GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction

Yuxuan Liang[1,2], **Songyu Ke**[3,2], **Junbo Zhang**[2,4], **Xiuwen Yi**[4,2], **Yu Zheng**[2,1,3,4]
[1] School of Computer Science and Technology, Xidian University, Xi'an, China
[2] Urban Computing Business Unit, JD Finance, Beijing, China
[3] Zhiyuan College, Shanghai Jiao Tong University, Shanghai, China
[4] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China
{yuxliang, songyu-ke, msjunbozhang, xiuwyi, msyuzheng}@outlook.com

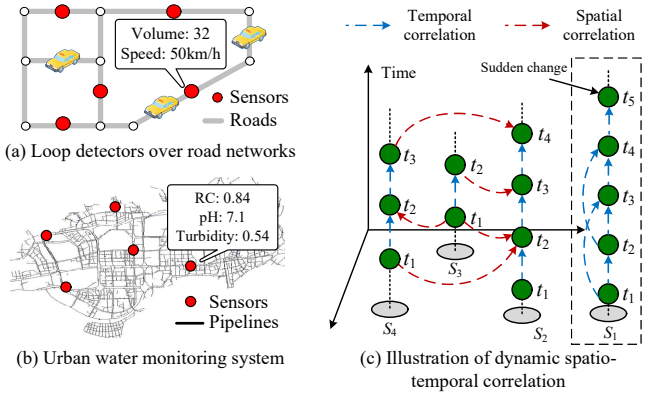
Abstract

Numerous sensors have been deployed in different geospatial locations to continuously and cooperatively monitor the surrounding environment, such as the air quality. These sensors generate multiple geo-sensory time series, with spatial correlations between their readings. Forecasting geo-sensory time series is of great importance yet very challenging as it is affected by many complex factors, *i.e.*, dynamic spatio-temporal correlations and external factors. In this paper, we predict the readings of a geo-sensor over several future hours by using a multi-level attention-based recurrent neural network that considers multiple sensors' readings, meteorological data, and spatial data. More specifically, our model consists of two major parts: 1) a multi-level attention mechanism to model the dynamic spatio-temporal dependencies. 2) a general fusion module to incorporate the external factors from different domains. Experiments on two types of real-world datasets, *viz.*, air quality data and water quality data, demonstrate that our method outperforms nine baseline methods.

1 Introduction

There are massive sensors, such as meteorological sites, that have been deployed in the physical world. Each of them has a unique geospatial location, constantly generating time series readings. A group of sensors collectively monitor the environment of a spatial region, with the spatial correlation between their readings. We call such sensors' readings *geo-sensory time series*. Additionally, it is common that one sensor generates multiple kinds of geo-sensory time series as it monitors different target conditions simultaneously. For example, as shown in Figure 1(a), the loop detectors in roads report timely readings about the vehicles passing by as well as their travel speed. Figure 1(b) presents that the sensors generate three different chemical indexes about water quality every 5 minutes. Besides monitoring, there is a rising demand for geo-sensory time series prediction, *e.g.*, traffic prediction.

However, forecasting geo-sensory time series is very challenging, affected by the two following complex factors:



(a) Loop detectors over road networks

(b) Urban water monitoring system

(c) Illustration of dynamic spatio-temporal correlation

Figure 1: (a)-(b) Examples of geo-sensory time series. (c) The future readings of a sensor depends on its past readings and that of nearby sensors, where the weights are changing over the location and time.

1) *Dynamic spatio-temporal correlations*.

- Complex inter-sensor correlations. Figure 1(c) shows the spatial correlation between different sensors' time series is highly dynamic, changing over time. Moreover, geo-sensory time series varies by locations non-linearly. When modeling dynamic pairwise correlation, classical methods (*e.g.*, probabilistic graphical models [Koller and Friedman, 2009]) have extremely heavy computational cost due to their massive parameters.
- Dynamic intra-sensor correlations. First, a geo-sensory time series usually follows a periodic pattern (*e.g.*, S_1 in Figure 1(c)), which changes over time and varies geographically [Zhang *et al.*, 2017]. Second, sensors' readings sometimes fluctuate tremendously and suddenly change, quickly decreasing the impact of their previous values. Thus, how to select the relevant previous time intervals to make predictions remains a challenge.

2) *External factors*. Sensors' readings are also affected by the surrounding environment such as meteorology (*e.g.*, a strong wind), time of day (*e.g.*, rush hours) and land use.

To tackle these aforementioned challenges, we propose a Multi-level Attention Network (GeoMAN) to predict the readings of a geo-sensor over a couple of future hours. The contributions of our study are three-fold:

- *Multi-level attention mechanism.* We develop a multi-level attention mechanism to model the dynamic spatio-temporal correlations. Specifically, in the first level, we propose a novel attention mechanism, consisting of local spatial attention and global spatial attention, to capture the complex spatial correlations between different sensors' time series (*i.e.*, inter-sensor correlation). In the second level, a temporal attention is applied to model the dynamic temporal correlations (*i.e.*, intra-sensor correlation) between different time intervals in a time series.
- *External factor fusion module.* We design a general fusion module to incorporate the external factors from different domains. The learned latent representations are fed into the multi-level attention networks to enhance the importance of these external factors.
- *Real evaluation.* We evaluate our approach based on two real-world datasets. Extensive experiments show the advantages of our method against all baselines.

2 Preliminary

2.1 Notations

Suppose there are N_g sensors, each of which generates N_l kinds of time series. Among them, we specify one time series as *target series* for making predictions, while other kinds of series are used as features. Given a time window of length T, we use $\mathbf{Y} = \left(\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{N_g}\right) \in \mathbb{R}^{N_g \times T}$ to denote the readings of all target series during past T hours, where $\mathbf{y}^i \in \mathbb{R}^T$ belongs to sensor i. We use $\mathbf{X}^i = \left(\mathbf{x}^{i,1}, \mathbf{x}^{i,2}, \ldots, \mathbf{x}^{i,N_l}\right)^\top = \left(\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_T^i\right) \in \mathbb{R}^{N_l \times T}$ as the local features of a sensor i, where $\mathbf{x}^{i,k} \in \mathbb{R}^T$ is the k-th time series reported by this sensor, and $\mathbf{x}_t^i = \left(x_t^{i,1}, x_t^{i,2}, \ldots, x_t^{i,N_l}\right)^\top \in \mathbb{R}^{N_l}$ denotes the readings of all time series from sensor i at time t. Besides the local features of sensor i, other sensors also share plenty of information that is useful to our predictions due to the geospatial correlations between different sensors. To this end, we combine the local features of each sensor into a set $\mathcal{X}^i = \left\{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^{N_g}\right\}$ as the global features of sensor i.

2.2 Problem Statement

Given the previous readings of each sensor and the external factors, predict the readings of the sensor i over next τ hours, denoted as $\hat{\mathbf{y}}^i = \left(\hat{y}_{T+1}^i, \hat{y}_{T+2}^i, \ldots, \hat{y}_{T+\tau}^i\right)^\top \in \mathbb{R}^\tau$.

3 Multi-level Attention Networks

Figure 2 presents the framework of our approach. Following the encoder-decoder architecture [Cho *et al.*, 2014b], we employ two separate LSTMs [Lin *et al.*, 1996], one to encode the sequence of input, *i.e.*, historical geo-sensory time series, and another one to predict the output sequence $\hat{\mathbf{y}}^i$. More specifically, our model GeoMAN is composed of two major parts as follows: 1) *Multi-level attention mechanisms*. It consists of an encoder with two kinds of spatial attention mechanisms and a decoder with temporal attention. In the encoder, we develop two different attention mechanisms, *i.e.*, local spatial attention and global spatial attention as depicted in Figure 2,

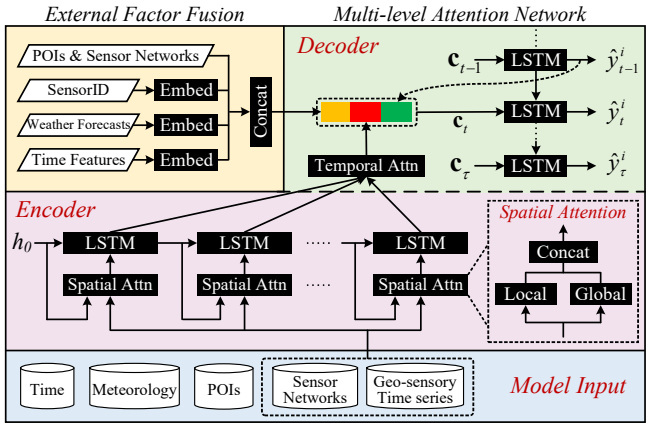


Figure 2: The framework of our approach. Attn: attention. Local: local spatial attention. Global: global spatial attention. Concat: concatenation layer. $\hat{y}_t^i$: the predicting value at time t. $\mathbf{c}_t$: the context vectors at time t. h_0: the initial state of encoder.

which can capture complex inter-sensor correlations at each time slot by referring to the previous hidden state of encoder, previous values of sensors as well as the spatial information (*i.e.*, sensor networks). In the decoder, we use a temporal attention to adaptively select the relevant previous time intervals for making predictions. 2) *External factor fusion*. This module is used to handle the effects of external factors, and its output is fed to the decoder as a part of its inputs. Here, we use $\mathbf{h}_t \in \mathbb{R}^m$ and $\mathbf{s}_t \in \mathbb{R}^m$ to denote the hidden state and cell state of the encoder at time t, respectively. Likewise, $\mathbf{d}_t \in \mathbb{R}^n$ and $\mathbf{s}' \in \mathbb{R}^n$ represent those of the decoder.

3.1 Spatial Attention

Local Spatial Attention

We first introduce the local spatial attention mechanism. For a certain sensor, there is a complex correlation among its local time series. For instance, an air quality monitoring station reports different time series such as PM2.5 (Particular Matter), NO and SO2. In fact, the concentration of PM2.5 is usually affected by other time series, including other air pollutants and local weather conditions [Wang *et al.*, 2005]. To address this issue, given the k-th local feature vector of the i-th sensor (*i.e.*, $\mathbf{x}^{i,k}$), we employ an attention mechanism to adaptively capture the dynamic correlation between the target series and each local feature with:

$$e_t^k = \mathbf{v}_l^\top \tanh\left(\mathbf{W}_l\left[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}\right] + \mathbf{U}_l \mathbf{x}^{i,k} + \mathbf{b}_l\right), \quad (1)$$

$$\alpha_t^k = \frac{\exp\left(e_t^k\right)}{\sum_{j=1}^{N_l} \exp\left(e_t^j\right)}, \quad (2)$$

where $[\cdot;\cdot]$ is a concentration operation. Here, the learnable parameter are $\mathbf{v}_l, \mathbf{b}_l \in \mathbb{R}^T$, $\mathbf{W}_l \in \mathbb{R}^{T \times 2m}$ and $\mathbf{U}_l \in \mathbb{R}^{T \times T}$. The attention weights of local features are jointly determined by the input local features and the historical states (*i.e.*, $\mathbf{h}_{t-1}$ and $\mathbf{s}_{t-1}$) in the encoder. This score semantically represents the importance of each local contributing feature. Once we obtain the attention weights, the output vector of local spatial attention at time step t is computed with:

$$\tilde{\mathbf{x}}_t^{local} = \left(\alpha_t^1 x_t^{i,1}, \alpha_t^2 x_t^{i,2}, \ldots, \alpha_t^{N_l} x_t^{i,N_l}\right)^\top. \quad (3)$$

Global Spatial Attention

To a target series reported by a sensor, that of other sensors have direct impacts on it. However, the impacting weights are highly dynamic, changing over time. Since there might be many irrelevant series, directly using all kinds of time series as the encoder inputs to capture the correlations between different sensors results in very high computational cost and degrades the performance. Note that such impacting weights are affected by the local condition of other sensors. For example, when the wind blows strongly from the remote places, the air quality in a certain region is more affected by these places than it used to be. Inspired by this fact, we develop a new attention mechanism to capture the dynamic correlations between different sensors. Given the i-th sensor as our predictive target and another sensor l, we calculate the attention weight (*i.e.*, impacting weight) between them as follows:

$$g_t^l = \mathbf{v}_g^\top \tanh\left(\mathbf{W}_g\left[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}\right] + \mathbf{U}_g \mathbf{y}^l + \mathbf{W}'_g \mathbf{X}^l \mathbf{u}_g + \mathbf{b}_g\right),$$

where $\mathbf{v}_g$, $\mathbf{u}_g$, $\mathbf{b}_g \in \mathbb{R}^T$, $\mathbf{W}_g \in \mathbb{R}^{T\times 2m}$, $\mathbf{U}_g \in \mathbb{R}^{T\times T}$ and $\mathbf{W}'_g \in \mathbb{R}^{T\times N_l}$ are the parameters to be learned. By referring to the target series and local features of other sensors, this attention mechanism can adaptively select the relevant sensors to make predictions. Meanwhile, the historical information is spreading across the time steps by considering the previous hidden state $\mathbf{h}_{t-1}$ and cell state $\mathbf{s}_{t-1}$ in the encoder.

Note that the spatial factors also contribute to the correlations between different sensors. Generally, geo-sensors are interconnected with each other through an explicit or underlying network. Here, we use a matrix $\mathbf{P} \in \mathbb{R}^{N_g\times N_g}$ to measure the geospatial similarity (such as the inverse of geospatial distance), where $P_{i,j}$ denotes the similarity between sensor i and j. Different from the attention weight, the geospatial similarity can be considered as a prior knowledge. In particular, if N_g is too large, an alternative is by using the nearest or closest ones instead of all sensors. After that, we employ a softmax function to ensure all the attention weights sum to one, jointly considering the geospatial similarity as follows:

$$\beta_t^l = \frac{\exp\left((1-\lambda)g_t^l + \lambda P_{i,l}\right)}{\sum_{j=1}^{N_g}\exp\left((1-\lambda)g_t^j + \lambda P_{i,j}\right)}, \tag{4}$$

where λ is a tunable hyperparameter as a trade-off. if λ is large, the term will force the attention weight to be as similar as the geospatial similarity. With these attention weights, the output vector of the global spatial attention is computed as:

$$\tilde{\mathbf{x}}_t^{global} = \left(\beta_t^1 y_t^1, \beta_t^2 y_t^2, \ldots, \beta_t^{N_g} y_t^{N_g}\right)^\top. \tag{5}$$

3.2 Temporal Attention

Since the performance of encoder-decoder architecture will degrade rapidly as the encoder length increases [Cho *et al.*, 2014a], an important extension is by adding a temporal attention mechanism, which can adaptively select the relevant hidden states of the encoder to produce output sequence, *i.e.*, model the dynamic temporal correlation between different time intervals in the target series. Specifically, to compute the attention vector at each output time t' over each hidden state of the encoder, we define:

$$u_{t'}^o = \mathbf{v}_d^\top \tanh\left(\mathbf{W}'_d\left[\mathbf{d}_{t'-1}; \mathbf{s}'_{t'-1}\right] + \mathbf{W}_d \mathbf{h}_o + \mathbf{b}_d\right), \tag{6}$$

$$\gamma_{t'}^o = \frac{\exp\left(u_{t'}^o\right)}{\sum_{j=1}^{T}\exp\left(u_{t'}^j\right)}, \tag{7}$$

$$\mathbf{c}_{t'} = \sum\nolimits_{o=1}^{T} \gamma_{t'}^o \mathbf{h}_o, \tag{8}$$

where $\mathbf{W}_d \in \mathbb{R}^{m\times m}$, $\mathbf{W}'_d \in \mathbb{R}^{m\times 2n}$, and $\mathbf{v}_d, \mathbf{b}_d \in \mathbb{R}^m$ are learnable. These scores are normalized by a softmax function to create the attention mask over the encoder hidden states.

3.3 External Factor Fusion

Geo-sensory time series has a strong correlation with the spatial factors, *e.g.*, POIs and sensor networks. Formally, these factors jointly feature the function of a region. Besides, there are many temporal factors affecting the readings of sensors, such as meteorology and time. Inspired by the previous works [Liang *et al.*, 2017; Wang *et al.*, 2018] focusing on the effects of external factors in spatio-temporal applications, we design a simple yet effective component to handle these factors.

As shown in Figure 2, we first incorporate the temporal factors including time features, meteorological features, and SensorID which specifies the target sensor. Since the weather condition at future time slot is unknown, we use weather forecasts to enhance our performance. Note that most of these factors are categorical which cannot be fed to neural networks directly, we transform each categorical attribute into a low-dimensional vector by feeding them into different embedding layers separately. In terms of the spatial factors, we use the POIs density of different categories as POIs features. Since the properties of sensor networks depend on the specific situation, we simply use the structural features of the networks, such as the number of neighbors and intersections. Finally, we concatenate the obtained embedded vectors together with the spatial features as the output of this module, denoted as $\mathbf{ex}_{t'} \in \mathbb{R}^{N_e}$, where t' is a future time step in the decoder.

3.4 Encoder-decoder & Model Training

In the encoder, we briefly aggregate the outputs from the local spatial attention and the global spatial attention with:

$$\tilde{\mathbf{x}}_t = \left[\tilde{\mathbf{x}}_t^{local}; \tilde{\mathbf{x}}_t^{global}\right], \tag{9}$$

where $\tilde{\mathbf{x}}_t \in \mathbb{R}^{N_l+N_g}$. We feed the concatenation $\tilde{\mathbf{x}}_t$ as the new input to the encoder and update the hidden state at time t by using $\mathbf{h}_t = f_e(\mathbf{h}_{t-1}, \tilde{\mathbf{x}}_t)$, where f_e is an LSTM unit.

In the decoder, once we get the weighted summed context vector $\mathbf{c}_{t'}$ at a future time step t', we combine it with the output of external factor fusion module $\mathbf{ex}_{t'}$ and the last output of decoder $\hat{y}_{t'-1}^i$ to update the decoder hidden state with $\mathbf{d}_{t'} = f_d\left(\mathbf{d}_{t'-1}, \left[\hat{y}_{t'-1}^i; \mathbf{ex}_{t'}; \mathbf{c}_{t'}\right]\right)$, where f_d is an LSTM unit used in the decoder. Then, we concatenate the context vector $\mathbf{c}_{t'}$ with the hidden state $\mathbf{d}_{t'}$, which becomes the new hidden state from which we make final predictions as follows:

$$\hat{y}_{t'}^i = \mathbf{v}_y^\top\left(\mathbf{W}_m\left[\mathbf{c}_{t'}; \mathbf{d}_{t'}\right] + \mathbf{b}_m\right) + b_y, \tag{10}$$

where the matrix $\mathbf{W}_m \in \mathbb{R}^{n\times(m+n)}$ and the vector $\mathbf{b}_m \in \mathbb{R}^n$ map the concentration $[\mathbf{c}_{t'}; \mathbf{d}_{t'}] \in \mathbb{R}^{m+n}$ to the size of the decoder hidden state. Finally, we use a linear transformation (*i.e.*, $\mathbf{v}_y \in \mathbb{R}^n$ and $b_y \in \mathbb{R}$) to generate the final output.

Since our approach is smooth and differentiable, it can be trained via back-propagation algorithm [Rumelhart *et al.*, 1986]. During the training phase, we use a Adam optimizer [Kingma and Ba, 2014] to train our model by minimizing the mean squared error (MSE) between the predicted vector $\hat{\mathbf{y}}^i$ and the ground truth vector $\mathbf{y}^i \in \mathbb{R}^\tau$ at sensor i:

$$\mathcal{L}(\theta) = \left\| \hat{\mathbf{y}}^i - \mathbf{y}^i \right\|_2^2, \tag{11}$$

where θ are all learnable parameters in the proposed model.

4 Experiments

4.1 Settings

Datasets

We conduct our experiments over two different datasets as depicted in Table 1. Each dataset contains three sub-datasets: meteorological data, POIs data and sensor networks data.

- **Water quality**: The sensors throughout water distribution system in a city of southeast China provides the real-time water quality information every five minutes from a period of three years, *e.g.*, residual chlorine (RC), turbidity and PH. We consider the concentration of RC as target series since it is widely employed as the major water quality index in environmental science [Rossman *et al.*, 1994]. Totally, there are 14 sensors collectively monitoring 10 different indexes, which are interconnected through pipe networks. We use the metric proposed by [Liu *et al.*, 2016a] as the similarity matrix in this dataset.
- **Air quality**: Scratched from a public website[1], this dataset includes the concentration of many different pollutants (*e.g.*, PM2.5, SO2 and NO), together with some meteorological readings (*e.g.*, temperature and wind speed) collected by totally 35 sensors every hour in Beijing. Among them, the primary pollutant of air quality is PM2.5 in most cases, thus we employ its reading as the target series. We briefly use the inverse of geospatial distance to denote the similarity between two sensors.

In the experiment with respect to the water quality, we partition the data into non-overlapped training, validation and test data by a ratio of 4:1:1. *i.e.*, we use the first two-year data as the training set, the first half of the last year as the validation set, and the second half of the last year as the test set. Unfortunately, we cannot obtain such big data in the second dataset. Hence, we use a ratio of 8:1:1 to overcome it.

[1]http://zx.bjmemc.com.cn/

Dataset		Water Quality	Air Quality
Target series		RC	PM2.5
#Sensors		14	35
#Attributes		10	19
Time Spans		1/1/2012-2014/12/31	8/20/2014-2017/11/30
Time Intervals		5 minutes	1 hour
#Instances		4,415,040	920,640
Meteorology	#Sensors	8	16
	#Attributes	6	13
POIs	#POIs	185,841	651,016
	#Categories	20	20

Table 1: Detail of the datasets.

Evaluation Metrics

We use multiple criteria to evaluate our model, including the rooted mean squared error (RMSE) and the mean absolute error (MAE), both of which are widely used in regression tasks.

Hyperparameters

Following the previous works [Zheng *et al.*, 2015; Liu *et al.*, 2016b], we set $\tau = 6$ to make short-term predictions. During the training phase, the batch size is 256 and the learning rate is 0.001. In external factor fusion module, we embed SensorID to $\mathbb{R}^6$ and the time features to $\mathbb{R}^{10}$. Totally, there are 4 hyperparameters in our model, of which the trade-off parameter λ is empirically fixed from 0.1 to 0.5. For the length of window size T, we set $T \in \{6, 12, 24, 36, 48\}$. For simplicity, we use the same hidden dimensionality at the encoder and the decoder, and conduct a grid search over $\{32, 64, 128, 256\}$. Moreover, we use stacked LSTMs (the number of layers is denoted as q) as the unit of encoder and decoder to enhance our performance. The setting in which $q = 2$, $m = n = 64$ and $\lambda = 0.2$ outperforms the others in the validation set.

4.2 Baselines

We compare our model with nine baselines as follows:

- **ARIMA** [Box and Pierce, 1970]: It is a well-known model for forecasting future values in a time series.
- **VAR** [Ziv, 2006]: Vector Auto-Regressive can capture the pairwise relationships among all sensors with very high computational costs due to massive parameters.
- **GBRT** [Friedman, 2001]: Gradient Boosting Regression Tree (GBRT) is an ensemble method for the regression tasks and widely used in practice.
- **FFA** [Zheng *et al.*, 2015]: A multi-view based hybrid model considers spatio-temporal dependencies and sudden change simultaneously to forecast sensor's reading.
- **stMTMVL** [Liu *et al.*, 2016b]: It fuses the heterogeneous data from multiple domains, jointly capturing the local and global information of sensors to make predictions based on multi-task multi-view learning.
- **stDNN** [Zhang *et al.*, 2016]: A deep neural network (DNN)-based prediction model for spatio-temporal data.
- **LSTM**: We use 6 different LSTMs to forecast the readings of a sensor over the next 6 hours separately.
- **Seq2seq** [Sutskever *et al.*, 2014]: It uses a RNN to encode the input sequences into a feature representation and another RNN to make predictions iteratively.
- **DA-RNN** [Qin *et al.*, 2017]: A dual-staged attention model for time series prediction, which shows the state-of-the-art performance in time series prediction.

Our model, as well as the baselines, are implemented with TensorFlow [Abadi *et al.*, 2016] on the server with one Tesla K40m and Intel Xeon E5. We consider the previous 6-hour data as the input of ARIMA. In stMTMVL and FFA, we use the default settings by their authors. Similar to GeoMAN, we use the former $T = \{6, 12, 24, 36, 48\}$-hour data as the input of other baselines. Finally, we test different hyperparameters for them all, finding the best setting for each.

4.3 Model Comparison

In this section, we compare our model with the baselines on the two datasets. To be fair, we present the best performance of each method under different parameter settings in Table 2.

In terms of water quality prediction, our proposed method clearly outperforms all the baselines on both metrics. Specifically, GeoMAN shows 14.2% and 13.5% improvements beyond the state-of-the-art approach (DA-RNN) on MAE and RMSE respectively. On the other hand, since the concentration of RC follows a certain periodic pattern, stDNN and RNN-based methods (*i.e.*, Seq2seq, DA-RNN and GeoMAN) achieve better performance than stMTMVL and FFA by considering much longer temporal dependency. Compared to LSTM which makes predictions at each future time step separately, GeoMAN as well as Seq2seq bring significant improvements due to the positive effects of the decoder component. Remarkably, GBRT performs better against most baselines, which reveals the advantage of the ensemble methods.

Compared to relatively stable readings of water quality, the concentration of PM2.5 sometimes fluctuates tremendously, which makes it more difficult to forecast. Table 2 presents a comprehensive comparison on air quality data in Beijing. It is easy to be seen that our model achieves the best performance on MAE and RMSE simultaneously. Following the previous work [Zheng *et al.*, 2015] that focuses on MAE, we mainly discuss on such metric. Our approach has relatively from 7.2% up to 63.5% lower MAE than these baselines, demonstrating that it has better generalization performance on other applications. Another interesting observation is that stMTMVL works well in water quality prediction but shows inferiority here since the number of joint-learning tasks of air quality prediction is much larger than that of water quality.

4.4 Variant Comparison

To further investigate the effectiveness of each model component, we compare GeoMAN with its variants as follows:

- **GeoMAN-nl**: There is no local spatial attention in the first-level attention (*i.e.*, spaital attention).
- **GeoMAN-ng**: We simply remove the global spatial attention module from the first-level attention.
- **GeoMAN-nt**: To validate the temporal attention mechanism, we remove it from GeoMAN directly.
- **GeoMAN-ne**: This variant does not consider the effects of external factors. *i.e.*, no external factor fusion module.

Method	Water Quality		Air Quality	
	RMSE	MAE	RMSE	MAE
ARIMA	8.61E-02	7.97E-02	31.07	20.58
VAR	5.02E-02	4.42E-02	24.60	16.17
GBRT	5.17E-02	3.30E-02	24.00	15.03
FFA	6.04E-02	4.10E-02	23.83	15.75
stMTMVL	6.07E-02	4.16E-02	29.72	19.26
stDNN	5.77E-02	3.99E-02	25.64	16.49
LSTM	6.89E-02	5.04E-02	24.62	16.70
Seq2seq	5.80E-02	4.03E-02	24.55	15.09
DA-RNN	5.02E-02	3.52E-02	24.25	15.17
GeoMAN	**4.34E-02**	**3.02E-02**	**22.86**	**14.08**

Table 2: Performance comparison among different methods.

Evaluation on Spatial Attention

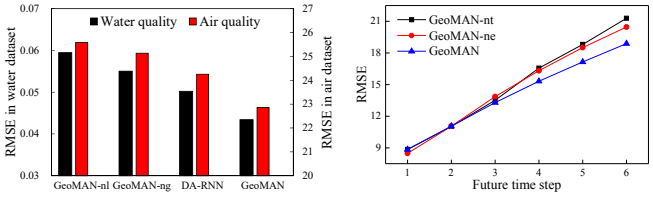
The experimental results are presented in Figure 3(a). From this figure, we observe that: 1) the combination of local and global spatial attention shows great superiority against each individual one, which demonstrates the importance of both the local and global information. 2) The fact that GeoMAN outperforms DA-RNN also verifies the advantage of our spatial attention against the input attention applied in the latter one. Despite that the local time series at a certain sensor and that from other sensors have different impacts on the target series, DA-RNN simply treats all these time series as equal and directly feeds them into the encoder to select relevant series by an input attention. It has the following two drawbacks. First, the input attention in DA-RNN cannot capture the spatial dependency between sensors. Second, the performance of DA-RNN decreases rapidly by the number of sensors.

Evaluation on External Factor Fusion

As a practical component of our model, this module provides additional information to boost the predictive performance. As illustrated in Figure 3(b), our model can significantly outperform GeoMAN-ne when predicting on more distant future over the second datset, since it allows our model to consider the temporally-related external factors in the future time step.

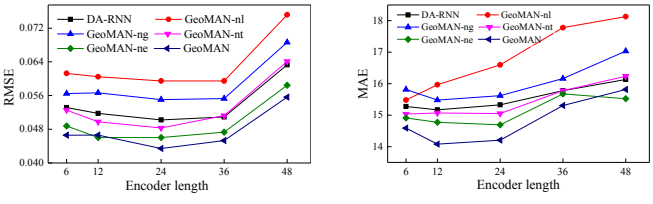
Evaluation on Temporal Attention

Temporal attention mechanism is employed to determine the discriminative encoder hidden state for making predictions. Hence, we attempt different encoder length T to verify its validity. As depicted in Figure 4(a), these methods follow a similar trend in water quality prediction. Most of them achieve the least errors when $T = 24$. Their performance will drop rapidly when T is large due to the difficulty with such long historical dependency. Different from water quality, Figure 4(b) reveals that the majority of models perform best when $T = 12$, because there is no such long temporal dependencies in the air quality dataset. Furthermore, Figure 3(b) shows our model outperforms GeoMAN-nt by a considerable margin since the temporal attention mechanism also enhances the long-term predictive performance.



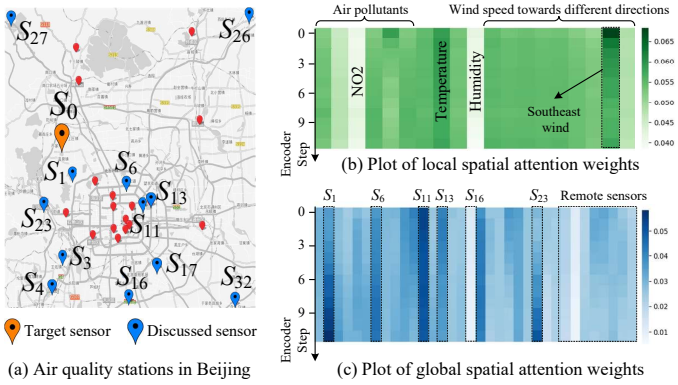
(a) Evaluation on spatial attention (b) Future time step vs. RMSE

Figure 3: Performance comparison among different vairants.



(a) Results on water quality. (b) Results on air quality.

Figure 4: Encoder length vs. metrics over the two datasets.



(a) Air quality stations in Beijing

(b) Plot of local spatial attention weights

(c) Plot of global spatial attention weights

Figure 5: (a) The distribution of air quality sensors throughout Beijing, where S_0 is the target sensor we discuss. (b)-(c) The attention matrix obtained from the local and global spatial attention mechanisms, where each row is the attention vector over the inputs.

4.5 Case study

To further investigate our approach, we perform a case study over air quality dataset from 4:00 to 16:00 on Feb. 28, 2017. Figure 5(a) presents the distribution of air quality monitoring sensors throughout Beijing. For succinctness, we omit some sensors in this figure. We take the sensor S_0 as an example to visualize the attention vectors for all encoder step t in Figure 5(b) and 5(c) respectively, where the attention weights are evolving across the encoder steps. Recall that the local weights semantically represent the relative importance of each local contributing series. According to the local weight matrix in Figure 5(b), the wind (from different directions) is an important factor affecting the concatenation of PM2.5, especially the local wind speed from the southeast. We also find that the temperature is closely related to the readings of PM2.5 at S_0 since humans in northern China consume fuel for heating in the winter. On the contrary, the humidity and NO2 have no obvious effect on the concentration of PM2.5 at this moment. All these facts demonstrate that the local spatial attention successfully captures the correlation between the local features and target series. Then, we discuss the complex correlations between different sensors. As shown in Figure 5(c), for remote sensors (*e.g.*, S_{16}, S_{26} and S_{27}), their attention weights are usually lower than that of nearby ones (*e.g.*, S_1 and S_6). During this time, there was a strong wind blowing from the southeast. That is the reason why S_{11} and S_{13} are far away from S_0 but have strong impacts on S_0. Another interesting observation is that the attention weight of S_{23} increases by the time step because of the increasing southwest wind at that sensor. According to this case study, our method is not only effective but can also be easily interpreted. Due to the page limitation, we do not further present the weights in the temporal attention mechanism.

5 Related Work

Geo-sensory Time Series Prediction. Autoregression-based models (*e.g.*, ARIMA and VAR) are widely used in time series prediction. Compared to traditional multivariate time series, geo-sensory time series has its own characteristics, *e.g.*, spatial correlation. Recently, cross-domain fusion-based methods showed superiority in many spatio-temporal applications. [Zheng *et al.*, 2015] forecast real-time air quality by fusing the past readings and the spatial factors. To predict the water quality, [Liu *et al.*, 2016b] proposed a multi-task multi-view learning model, which jointly captures the local information as well as global information of each sensor.

Deep Learning for Spatio-Temporal Data. Recurrent neural networks (RNNs) become popular due to their success in sequence learning [Sutskever *et al.*, 2014]. In particular, the incorporation of long short-term memory (LSTM) or gated recurrent unit (GRU) [Cho *et al.*, 2014b] enables RNNs to learn long-term temporal dependency. However, these works can only capture temporal dependency in time series, which ignore the unique characteristics of geo-sensory data, *e.g.*, spatial correlation. To overcome this problem, [Lv *et al.*, 2015] first proposed a deep learning approach to extract the latent traffic flow feature representation, such as the nonlinear spatial and temporal correlations from the traffic data. Currently, [Zhang *et al.*, 2017] developed a residual network considering both temporal and spatial dependencies to forecast the citywide crowd flow, a kind of geo-sensory time series.

Attention Mechanism. Recently, attention mechanisms became popular due to its success in general sequence-to-sequence problems. [Bahdanau *et al.*, 2014] first introduced a general attention model that did not assume a monotonic alignment. Later, researchers developed a number of multi-level attention-based models to select the relevant features and encoder hidden states in different applications [Wang *et al.*, 2016; Yu *et al.*, 2017]. To forecast the time series, [Qin *et al.*, 2017] proposed a dual-stage attention-based recurrent neural network (DA-RNN) to select the relevant driving series at each time interval. However, DA-RNN is not suitable for geo-sensory time series forecasting and the reason is detailed in Section 4.4. Thus, we propose two spatial different attention mechanisms to capture the dynamic inter-sensor correlations. To the best of our knowledge, no prior work studies our problem via an attention-based deep learning approach.

6 Conclusion and Future Work

In this paper, we propose a novel multi-level attention-based network for predicting the geo-sensory time series based on heterogeneous data from multiple domains. In the first level, local and global spatial attention mechanisms are applied to capture the dynamic inter-sensor correlations in geo-sensory data. In the second level, we employ a temporal attention to adaptively select the relevant time step to make predictions. Moreover, our model considers the effects of external factors by using a general fusion module. We evaluate our model on two types of geo-sensory datasets and the experiments show that our model achieves the best performance against 9 baselines in terms of the two metrics (RMSE and MAE) simultaneously. In addition, we visualize the attention weights to show the interpretation of our approach.

In the future, we will extend our method to solve the problem of long-term prediction. Moreover, we will explore the high-quality inference of geo-sensory time series through a limited number of sensors in real-world applications.

Acknowledgments

We thank Prof. Shuming Liu and Yipeng Wu from Tsinghua University for sourcing the water quality dataset in this study. This work was supported by the National Natural Science Foundation of China Grant No. 61672399, No. U1609217, and 973 Program, No. 2015CB352400.

References

[Abadi *et al.*, 2016] Martín Abadi, Ashish Agarwal, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[Box and Pierce, 1970] George EP Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[Cho *et al.*, 2014a] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

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

[Friedman, 2001] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[Liang *et al.*, 2017] Yuxuan Liang, Zhongyuan Jiang, and Yu Zheng. Inferring traffic cascading patterns. ACM SIGSPATIAL 2017, November 2017.

[Lin *et al.*, 1996] Tsungnan Lin, Bill G Horne, Peter Tino, and C Lee Giles. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1338, 1996.

[Liu *et al.*, 2016a] Ye Liu, Yuxuan Liang, Shuming Liu, David S Rosenblum, and Yu Zheng. Predicting urban water quality with ubiquitous data. *arXiv preprint arXiv:1610.09462*, 2016.

[Liu *et al.*, 2016b] Ye Liu, Yu Zheng, Yuxuan Liang, Shuming Liu, and David S. Rosenblum. Urban water quality prediction based on multi-task multi-view learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2576–2582, 2016.

[Lv *et al.*, 2015] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[Qin *et al.*, 2017] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

[Rossman *et al.*, 1994] Lewis A Rossman, Robert M Clark, and Walter M Grayman. Modeling chlorine residuals in drinking-water distribution systems. *Journal of environmental engineering*, 120(4):803–820, 1994.

[Rumelhart *et al.*, 1986] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[Wang *et al.*, 2005] Ying Wang, Guoshun Zhuang, Aohan Tang, Hui Yuan, Yele Sun, Shuang Chen, and Aihua Zheng. The ion chemistry and the source of pm2. 5 aerosol in beijing. *Atmospheric Environment*, 39(21):3771–3784, 2005.

[Wang *et al.*, 2016] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *ACL (1)*, 2016.

[Wang *et al.*, 2018] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. 2018.

[Yu *et al.*, 2017] Dongfei Yu, Jianlong Fu, Tao Mei, and Yong Rui. Multi-level attention networks for visual question answering. In *Conf. on Computer Vision and Pattern Recognition*, volume 1, page 8, 2017.

[Zhang *et al.*, 2016] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. Dnn-based prediction model for spatio-temporal data. ACM SIGSPATIAL 2016, October 2016.

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

[Zheng *et al.*, 2015] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2267–2276. ACM, 2015.

[Ziv, 2006] *Vector Autoregressive Models for Multivariate Time Series*, pages 385–429. Springer New York, New York, NY, 2006.