Deep Distributed Fusion Network for Air Quality Prediction

Xiuwen Yi[1,2], Junbo Zhang[2,1,+], Zhaoyuan Wang[1,2], Tianrui Li[1], Yu Zheng[2,1,3]
[1]School of Information Science and Technology, Southwest Jiaotong University, China
[2]Urban Computing Business Unit, JD Finance, China
[3]School of Computer Science and Technology, Xidian University, China
xiuwenyi@foxmail.com, {msjunbozhang, wang_zhaoyuan, msyuzheng}@outlook.com, trli@swjtu.edu.cn

ABSTRACT

Accompanying the rapid urbanization, many developing countries are suffering from serious air pollution problem. The demand for predicting future air quality is becoming increasingly more important to government's policy-making and people's decision making. In this paper, we predict the air quality of next 48 hours for each monitoring station, considering air quality data, meteorology data, and weather forecast data. Based on the domain knowledge about air pollution, we propose a deep neural network (DNN)-based approach (entitled DeepAir), which consists of a spatial transformation component and a deep distributed fusion network. Considering air pollutants' spatial correlations, the former component converts the spatial sparse air quality data into a consistent input to simulate the pollutant sources. The latter network adopts a neural distributed architecture to fuse heterogeneous urban data for simultaneously capturing the factors affecting air quality, *e.g.* meteorological conditions. We deployed DeepAir in our AirPollutionPrediction system, providing fine-grained air quality forecasts for 300+ Chinese cities every hour. The experimental results on the data from three-year nine Chinese-city demonstrate the advantages of DeepAir beyond 10 baseline methods. Comparing with the previous online approach in AirPollutionPrediction system, we have 2.4%, 12.2%, 63.2% relative accuracy improvements on short-term, long-term and sudden changes prediction, respectively.

CCS CONCEPTS

• **Applied computing** → Environmental sciences;

KEYWORDS

Air Quality Prediction; Deep Learning; Urban Computing

ACM Reference format:

Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, Yu Zheng. 2018. Deep Distributed Fusion Network for Air Quality Prediction. In *Proceedings of KDD'18, London, United Kingdom, August 19-23, 2019*, 9 pages. DOI: https://doi.org/10.1145/3219819.3219822

1 INTRODUCTION

With the rapid development of urbanization, air pollution is becoming a severe environmental and societal issue for all developing countries around the world [1]. Air pollution consists of a mixture of particulate matter (*i.e.* $PM_{2.5}$ and PM_{10}) and gaseous species (*i.e.* NO_2, CO, O_3 and SO_2), which have both acute and chronic effects on human health, especially for young and elderly [2]. From statistical results [3], Beijing recorded 46 days of heavy pollution during 2015, accounting for 12.6 percent of the year. For monitoring real-time air pollution, Chinese governments have built many air quality monitoring stations and published air quality data every hour in recent years [4]. Besides monitoring, there is a rising demand for predicting future air quality, which can inform governments' policy-making (such as performing traffic control when the air is polluted seriously) and people's decision making (like whether to exercise outdoors).

Predicting future air quality for a monitoring station, however, is very challenging because of the following reasons:

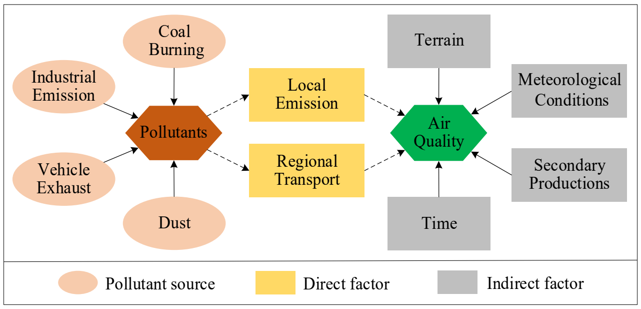


Figure 1: Multiple influential factors

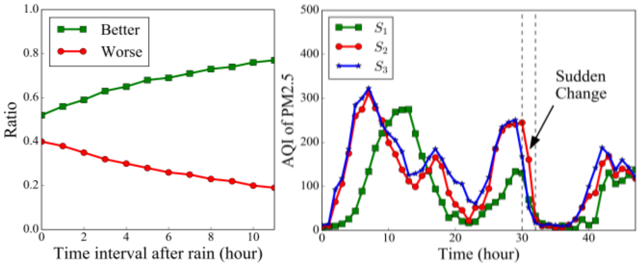
First, air quality has multiple influential factors. As illustrated in Figure 1, air pollutant sources mainly come from vehicle exhaust, industrial emission, coal burning, and dust [5], which each source has a different spatial distribution and temporal pattern. From atmospheric science research [6], the accumulation and dissipation of air pollutants are mainly confounded by local emission, regional transport, meteorological conditions and so on. Depending on the impact on air quality, these factors fall into two groups. Local emission and regional transport are direct factors as they are derived from pollutant sources and determine the formation of pollutants. Meteorological conditions, secondary productions, terrain and time are indirect factors as they determine the development environment of pollutants. However, we do not have sufficient and accurate data to model all these

+ Junbo Zhang is the corresponding author of this paper.

KDD'18, August 19–23, 2018, London, United Kingdom.
© 2018 ACM. 978-1-4503-5552-0/18/08...$15.00.
DOI: https://doi.org/10.1145/3219819.3219822

factors precisely. For example, it is almost impossible to obtain city-wide emission data from pollutant sources. Likewise, weather forecast data are not accurate enough as “The longer the forecast horizon is, the less accurate the forecast will be.”

Second, the interactions between these factors are complex. Most people have the sense of air quality will be better after rain. However, based on the statistical results of three-year data in Beijing which depicted in Figure 2(a), it still has above 20% ratio that air quality will be worse in the next few hours after rain. Though rain has a higher ratio of air quality will be better, it is still difficult to model the interaction as the effect is not absolute. The reason behind it is that air quality is affected by multiple factors simultaneously shown in Figure 1, which the interactions are complex. As a result, it is hard to determine the weight of each factor due to the dynamic environment. Likewise, it is very difficult to capture the spatio-temporal characterizations of air pollution dispersion as air quality changes over location and fluctuate along time without an obvious periodic pattern.



(a) Rain affect air quality (b) AQI change over time

Figure 2: Air quality change over multiple factors. Ratios in a) is calculated by $\Delta = AQI_{t+k} - AQI_t$, where $AQI_t > 100$, $Weather_t = rain$ and k is the time interval after rain.

Third, in addition to normal fluctuation, it exists some sudden changes which are caused by some specific kinds of factor. Here, sudden changes mean that air quality index (AQI) drops very sharply in a very short time span [7]. As illustrated in Figure 2(b), AQI of air quality monitoring station S_2 at the 30th timestamp drops over 200 in the coming two hours due to a strong wind blowing from the southeast. Such a sudden change is very important to real-time monitoring and further data analysis. In daily life, most people always pay more attention to sudden changes than to general cases, as they only care about future air quality once the air is polluted seriously and want to know how long it will be good. However, the presence of sudden changes is very infrequent in whole datasets. Among entire observations of three-year air quality data in Beijing, the presence of sudden changes is less than 2.3%. Such data imbalance phenomenon brings much difficulty for air quality prediction.

To address these challenges, we propose a DNN-based air quality prediction approach, entitled DeepAir. Our approach is inspired by the domain knowledge about air pollution, which can help design model structure with more interpretations. We deployed DeepAir in real-time AirPollutionPrediction system [8], providing 48-hour fine-grained air quality forecasts for 300+ Chinese cities. Our contributions are listed as below:

- Considering the dispersion of air pollutants, we design a spatial transformation component to convert the spatial sparse air quality data into a consistent input for simulating second-hand pollutant sources. With the signals from spatial neighbors, DeepAir results in a better performance on general cases and sudden changes.
- Considering direct and indirect factors have different effects on air quality, we propose a deep distributed fusion network to fuse heterogeneous urban data for capturing all influential factors. The network adapts a novel distributed architecture to simultaneously model the interactions between these factors for learning the individual and holistic influences.
- Based on three-year data from 9 Chinese cities, the results demonstrate the advantages of DeepAir compared with 10 baselines. We deployed DeepAir in AirPollutionPrediction system, providing fine-grained air quality forecasts for 300+ cities. Comparing with previous online approach [7], we have 2.4%, 12.2%, 63.2% relative accuracy improvements on short-term, long-term and sudden changes prediction.

2 OVERVIEW

2.1 Problem Formulation

Given air quality data $\{AQI_S^t\}_{t=1}^T$, meteorological data $\{M_S^t\}_{t=1}^T$, and weather forecast data $\{W_S^t\}_{t=1}^{T+K}$, where S is the set of air quality monitoring stations $\{S_1, S_2, \cdots, S_n\}$, and T is current timestamp, we aim to predict the AQI over the next K hours for each monitoring station $\{AQI_S^{T+k}\}_{k=1}^K$.

As PM2.5 (Particulate matter with a diameter smaller than 2.5 micrometers) is most reported and most difficult-to-predict, in the following, we take AQI of PM2.5 for example.

2.2 AirPollutionPrediction System

AirPollutionPrediction system [8] is deployed through a “cloud + client” framework, where the cloud continuously collects real-time data and make predictions [9], and the web client public air quality information available. Figure 3 presents the website of AirPollutionPrediction , where the chart on the map showing AQI forecasts. For visualization, we show the min-max range of AQI for time intervals 7-12, 12-24, and 24-48 hours.

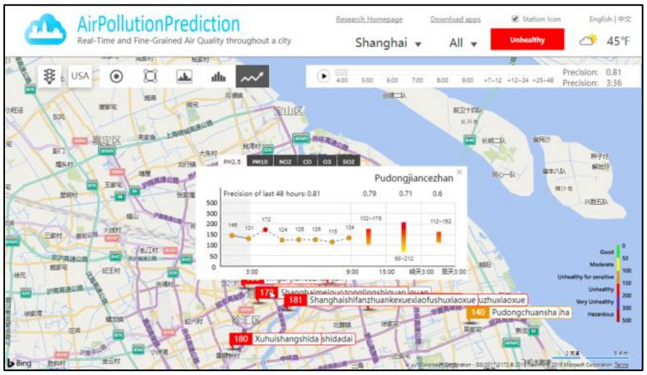


Figure 3. Web client of AirPollutionPrediction

2.3 Framework of the Predictive Model

As shown in Figure 4, the framework of DeepAir consists of two parts: spatial transformation component and deep distributed fusion network. As air pollutants are dispersed in geographical space, the former component regards the readings recorded by air quality monitoring stations as second-hand pollutant sources. Considering air pollutants' spatial correlations, spatial transformation component uses the spatial partition, spatial aggregation, and spatial interpolation to convert the spatial sparse air quality data into a consistent input, named AQIs. Then, AQIs and other datasets, *i.e.* meteorology, weather forecast, other pollutants, time, and station ID are fed into deep distributed fusion network, which adapts DNN to fuse heterogeneous urban data. We first use embedding method to transform the raw features of each domain data into a low-dimensional space for capturing temporal correlation and learning the intra-dynamics. Here, we use the embedding of AQIs to simulate the direct factors from local emission and regional transport and use the embedding of rest datasets as indirect factors respectively. Then, we propose a distributed fusion architecture to simultaneously model the interactions between these factors for learning the individual and holistic influences. As each indirect factor has own effort on direct factors affecting future air quality, we build four subnets (HW, WF, SP, and MP) to capture the individual influences from the historical weather, weather forecast, secondary productions, and meta properties from time and terrain, respectively. Besides individual influences, we build a subnet (HI) to learn the holistic influence by fusing all direct and indirect factors together. After that, the outputs of five subnets are aggregated by weighted merge to capture the high-level effects of these factors. Finally, the aggregation is mapped into [0, 1] by a Sigmoid function to generate final prediction results.

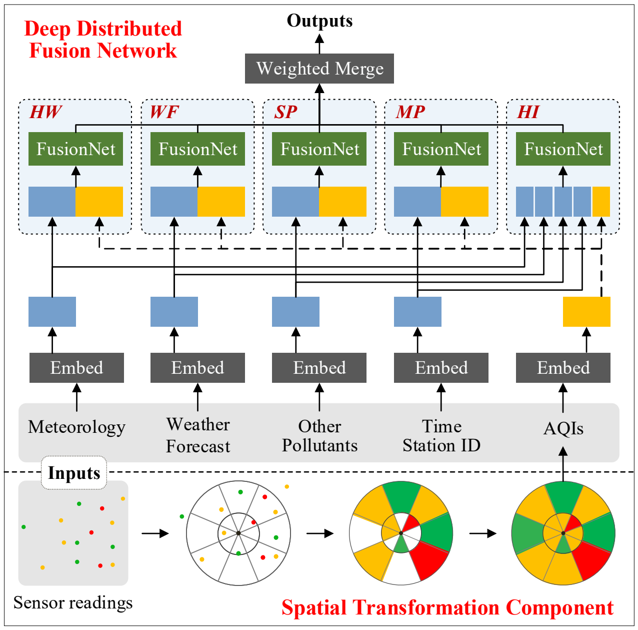


Figure 4: Framework of our approach

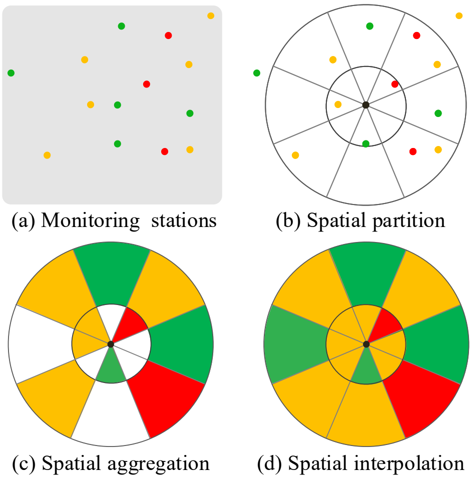
Specifically, for temporal granularity, we collectively predict the air quality in a couple hours, *e.g.* 1-3 hours, as weather forecasts are usually segmented into 3-hour time intervals. For spatial granularity, we build one predictive model for all monitoring stations in the same city as spatial transformation component will generate a consistent input for each monitoring station and data augmentation for training DNN.

3 SPATIAL TRANSFORMATION

As pollutants are dispersed in geographical space, the air quality of a geo-location not only depends on its previous air quality but also depends on the air quality of its neighbors. For converting spatial sparse air quality data into a consistent input for the further predictive model, we devise the spatial transformation component, which can be applied to other spatial sparse datasets.

As shown in Figure 5(a), air quality monitoring stations (marked as dot) are randomly scattered in geographical space, where color on the dot means the level of air quality. Firstly, we partition the geographical space into 16 regions by four lines and two circles, *e.g.* 20 km and 100 km semidiameter. As depicted in Figure 5(b), all regions share the target monitoring station (denoted by the black point) as common center and regions in the inner circle have a small area, while regions in the outer circle have a big area. Also, regions with different angles fit eight wind directions, which may be further captured by meteorological conditions. Furthermore, we aggregate the readings of air quality recorded by monitoring stations within the regions, illustrated in Figure 5(c). As a result, regions with at least one station will have one average AQI. However, from the partition results of Beijing, we find that different target stations have different missing patterns and about 33% regions do not have monitoring stations. Thus, we fill the missing values in these regions shown in Figure 5(d). More specifically, we first random generate some fake monitoring stations in these regions. Then, we use a classic spatial interpolation method, inverse distance weighting (IDW) [11], to interpolate the AQI of fake monitoring stations. Considering the readings of geospatially adjacent stations located in both inside and outside the outer circle, IDW assigns a weight to each available reading of geospatially adjacent stations by the distance to target sensor, and then aggregates these weights and readings by weighted average. After that, we aggregate the interpolated values of fake stations to calculate average AQI for the region. Finally, we get 17 AQI in one timestamp which 1 AQI come from target station and 16 AQI come from neighbor regions. We conduct the same process for each monitoring stations along time.

We design the spatial transformation component considering the following three aspects. 1) Air pollution dispersion. Although we do not have first-hand city-wide pollutant emission data, the readings of air quality recorded by monitoring stations can be regarded as second-hand pollutant sources as air pollutants are dispersed among different locations. With the signals from spatial neighbors, the further predictive model can incorporate more information. 2) Spatial correlations. Spatial partition merge the scattered air quality data into regions, which closer regions have a finer granularity and farther regions have a coarser granularity. Moreover, regions with different distance show different impacts varying by distance, which follows the First Law of Geography [12], *i.e.* "*Everything is related to everything else, but near things are more related than distant things.*" 3) Scalability. Spatial aggregation reduces model complexity as it sets an upper bound (the number of regions) for the input. Moreover, spatial interpolation overcomes spatial sparsity by filling the missing values and generating a consistent input for all monitoring stations, which enable us to use different stations' data together to train a single model with more training data.


(a) Monitoring stations (b) Spatial partition

(c) Spatial aggregation (d) Spatial interpolation

Figure 5: Spatial transformation

4 DEEP DISTRIBUTED FUSION

For simultaneously capturing the factors affecting future air quality, *e.g.* meteorological conditions, we design a DNN-based method to fuse cross-domain data. As depicted in Figure 4, we build five subnets (HW, WF, SP, MP, and HI) to capture these factors. Though air quality is affected by multiple factors, the degree of influences from these factors may be different. Inspired by such observation, the outputs of five subnets are weighted merged using a parametric-matrix-based fusion [13] to model the dynamic influences and generate the final results:

$$\hat{\boldsymbol{y}} = Sigmoid(\boldsymbol{y}_{hw} \circ \boldsymbol{w}_{hw} + \boldsymbol{y}_{wf} \circ \boldsymbol{w}_{wf} + \boldsymbol{y}_{sp} \circ \boldsymbol{w}_{sp} + \boldsymbol{y}_{mp} \circ \boldsymbol{w}_{mp} + \boldsymbol{y}_{hi} \circ \boldsymbol{w}_{hi}) \qquad (1)$$

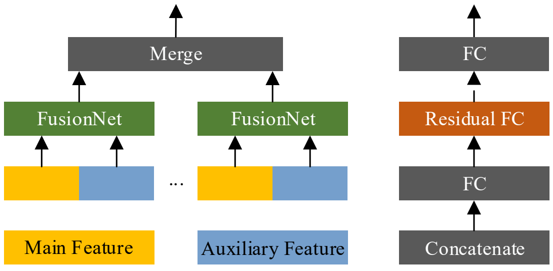
where $\hat{\boldsymbol{y}} \in R^h$ are the predicted results, $\boldsymbol{y}_{hw}$, $\boldsymbol{y}_{wf}$, $\boldsymbol{y}_{sp}$, $\boldsymbol{y}_{mp}$, $\boldsymbol{y}_{hi}$ are the outputs of five subnets, $\circ$ is Hadamard product, and $\boldsymbol{w}_{hw}$, $\boldsymbol{w}_{wf}$, $\boldsymbol{w}_{sp}$, $\boldsymbol{w}_{mp}$, $\boldsymbol{w}_{hi}$ are the learnable parameters that adjust the degrees affected by these subnets. Here, the prediction results are mapped into [0, 1] by Sigmoid function. And later, we denormalize the predictions to get the actual air quality.

4.1 Distributed Fusion Architecture

Based on domain knowledge, we know that direct and indirect factors have different effects on future air quality. At most time, all indirect factors will simultaneously determine the development environment of direct factors. Also, each indirect factor has an own individual effect on direct factors affecting future air quality. For capturing such individual and holistic influences, we propose a distributed fusion architecture as shown in Figure 6(a), which main feature fuses each auxiliary feature in a parallel manner, and then merge the outputs together. The key point in distributed fusion architecture is that we specify one feature as main feature and other features as auxiliary features. The reason for this partition is main feature and prediction target come from the same domain, while auxiliary features and prediction target come from different domains. Distributed fusion architecture highlights the main feature and captures the influences from auxiliary features as main feature respectively interacting with each auxiliary feature to learn the joint effects.

In our task, we specify the embedding of AQIs as main feature and the embedding of other features (*i.e.* meteorology, weather forecast, other pollutants, time and station ID) as auxiliary features, which main feature can simulate the direct factors from local emission and regional transport, while auxiliary features can represent the indirect factors. For modeling the interaction between these factors, we build five subnets to capture the holistic influence from all influential factors and the individual influences from the historical weather, weather forecast, secondary productions, and meta properties from time and terrain. Here, main feature is shared across all subnets and all subnets have the same network structure, FusionNet.

As shown in Figure 6(b), FusionNet treats all features equally by using a concatenate layer to merge all features together, then uses some fully-connected layers (FC) to learn higher-order feature interactions in a non-linear way. For training the neural network easier and more robust, we add some residual fully-connected layers [21] between fully-connected layers, which previous information can be directly passed to following layers through the shortcut connections.



(a) Distributed Fusion (b) FusionNet

Figure 6: Architectures of fusion

4.2 Subnets

We build historical weather subnet (HW) and weather forecast subnet (WF) for capturing historical and future meteorological conditions. The reason for building such two subnets is data realism and time interval, which historical weather provide hourly real weather conditions while weather forecast provide 3-hour segmented forecasted weather conditions. For historical weather data, we consider weather, wind speed, wind direction, humidity, and pressure as features; for weather forecast data, we consider weather, wind direction and wind strength as features. After feeding AQIs and features into subnets, we get $\boldsymbol{y}_{hw}$ and $\boldsymbol{y}_{wf}$.

Besides the direct emission of pollutants, it exists some secondary chemical reaction among pollutants in the atmosphere. Thus, we design a secondary production subnet (HI) to simulate the chemical interaction between pollutants. After fusing AQIs of $PM_{2.5}$ and other pollutants (PM_{10}, NO_2, CO, O_3, and SO_2) recorded by target station, we get $\boldsymbol{y}_{sp}$.

Meta property subnet (MP) models the time and terrain properties affecting air quality. Specifically, we use time (Month, DayOfWeek, TimeOfDay) to model the air quality pattern in temporal dimension, *e.g.* winter always has a higher AQI than summer due to heating. Also, we use station ID to simulate terrain affecting air quality, *e.g.* air quality is always worse in built-up areas than open areas. After fusing AQIs, time and station ID in FusionNet, we get $\boldsymbol{y}_{mp}$.

Except for the individual effects, all indirect factors will simultaneously determine the development environment of direct factors affecting future air quality. For capturing such information, we design the holistic influence subnet (HI) to learn the holistic influence by fusing all direct and indirect factors together. Then, we get $\boldsymbol{y}_{hi}$.

4.3 Embedding

Before distributed fusion, we use embedding [22] to capture temporal dependencies and learn intra-dynamics for each influential factor. For categorical features, embedding can transform the features represented by one-hot encoding to a real-valued vector and capture the similarity between different categories. For numerical features, embedding can transform the raw features to a low-dimensional space for reducing computational cost and learn the hidden representation.

Table 1. Embedding setting. Encoding is represented by timestamps * feature dimension in one timestamp.

Data	Feature	Encoding	Embedding
AQIs	PM2.5	6*17	36
Station ID	Beijing	36	3
Time	Month	12	3
	DayOfWeek	7	
	TimeOfDay	4	
Other Pollutants	PM_{10}	6*1	6
	NO_2	6*1	
	CO	6*1	
	O_3	6*1	
	SO_2	6*1	
Historical Weather	Weather	6*8	6
	Wind Speed	6*1	
	Wind Direction	6*4	
	Humidity	6*1	
	Temperature	6*1	
	Pressure	6*1	
Weather Forecast	Weather	(*k*/3)*8	6
	Wind Strength	(*k*/3)*4	
	Wind Direction	(*k*/3)*4	

As shown in Table 1, we detail the embedding settings for each influential factor. For AQIs, other pollutants, historical weather, we use the data in past and current 6 hours to incorporate the temporal information. For weather forecast, we use $k/3$ forecast instances to capture the dynamic changes of future weather conditions. Here, we combine the features from same domain together (*e.g.* Month, DayOfWeek, and TimeOfDay) to jointly learn the embedding for exploring intra-dynamics of each factor after feature interactions. Thus, we use the embedding of these domain data to simulate the direct factors and indirect factors.

4.4 Algorithm

Algorithm 1 outlines the DeepAir training process. We first construct the training instances from original heterogeneous urban data (lines 1-11). Then, DeepAir is trained via backpropagation to minimize the mean absolute error between predictions and ground values (lines 12-16).

Algorithm 1: DeepAir Training Algorithm

```
   Input:  Historical AQI observations $\{AQI_S^t\}_{t=1}^T$;
           Historical weather conditions $\{M_S^t\}_{t=1}^T$;
           Weather forecasts $\{W_S^t\}_{t=1}^{T+k}$; Future time interval $k$;
           Length of past sequence $h$; Particular air pollutant $p$;
   Output: Learned DeepAir model
   // construct training instances
1  $\mathcal{D} \leftarrow \theta$
2  for all available time interval $t$ $(1 \le t \le T)$ do
3      $x_t$ = Extract_Feature_From_Time($t$)
4      for $\forall i \in S$ do
5          $x_{aqi}$ = Spatial_Transfrrmation($p$, $[AQI_S^{t-h}, \cdots, AQI_S^t]$)
6          $x_{hw} = [M_i^{t-h}, \cdots, M_i^t]$
7          $x_{wf} = [W_i^t, \cdots, W_i^{t+k}]$
8          $x_{sp}$ = Get_Other_Pollutants($p$, $[AQI_S^{t-h}, \cdots, AQI_S^t]$)
9          $x_{id}$ = One-Hot_Encoding($i$)
10         y = Get_Prediction_Target($p, AQI_S^{t+k}$)
11     Append ($\{x_{aqi}, x_{hw}, x_{wf}, x_{sp}, x_{id}, x_t\}, y$) into $\mathcal{D}$
   // train the model
12 initialize all learnable parameters $\theta$ in DeepAir
13 repeat
14     randomly select a batch of instances $\mathcal{D}_b$ from $\mathcal{D}$
15     find $\theta$ by minimizing the loss function with $\mathcal{D}_b$
16 until stopping criteria is met
```

4 EXPERIMENTS

4.1 Settings

Datasets

Air quality data: AirPollutionPrediction system [8] collects air pollutants data from 2,296 official air quality monitoring stations in 302 Chinese cities every hour. Each air quality record consists of the concentration of six pollutants: $PM_{2.5}$, PM_{10}, NO_2, CO, O_3, and SO_2. We convert these concentrations into corresponding AQI for each pollutant based on Chinese AQI standards.

Meteorological data: The system collects meteorological data from 3,514 cities/districts every hour. Most major cities have both district-level and city-level granularity for the data, while small cities only have a city-level report. Each record consists of weather (sunny, cloudy, overcast, foggy, snow, small rain, moderate rain, and heavy rain), humidity, temperature, pressure, wind speed, and wind direction.

Weather forecast data: The system collects weather forecast data for 2,612 cities/districts. The updating frequency of the forecasts is 12 hours, updating twice a day at 8 am and 8 pm. We collect the forecasts for the next three days for each update, which is usually segmented into 3-hour time interval. Each record consists of weather, temperature, wind strength and wind direction.

For evaluation, we use three-year (from 2014/5/1 to 2017/4/30) data in nine major Chinese cities (Beijing, Tianjin, Shanghai, Nanjing, Hangzhou, Guangzhou, Shenzhen, Chengdu, and Chongqing). Figure 7 shows the distribution about AQI of $PM_{2.5}$ between 2014/5 to 2017/4 in nine cities, which the colors, defined by Chinese standards, represent the level of air pollution. In general, Beijing and Tianjin have worse air quality and Shenzhen and Guangzhou are better. As Beijing has the most complicated air quality, we focus on Beijing's data when comparing with different baselines, while showing overall results for the other eight cities. Table 2 details the statistical results about Beijing dataset. To predict the air quality of 36 monitoring stations in Beijing, 74 neighbor stations within 100km (semidiameter) to Beijing are retrieved. Among all air quality records, 2.3% cases are sudden changes. In the experiments, the data in the first 24 months is used for training, and the data in last 12 months is used for testing.

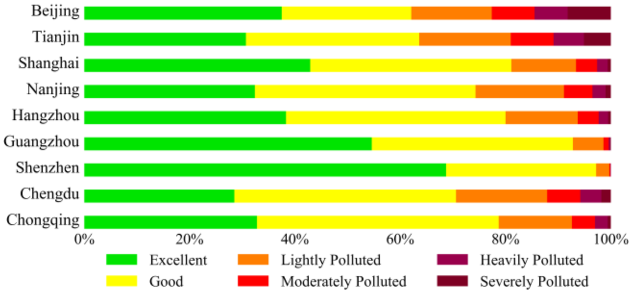


Figure 7: Distribution about AQI of PM2.5. Each color represents the level of air pollution.

Table 2. Data statistic of Beijing dataset

Air Quality	In-city stations	36
	Instances	875,394
	Sudden changes	20,540
	Average $PM_{2.5}$	118.2
	Neighbor stations	74
Meteorology	Sources	17
	Instances	327,514
Weather Forecast	Sources	17
	Instances	298,790

Baselines

We compare DeepAir with following ten baselines.

- **ARIMA**: Autoregressive integrated moving average (ARIMA) is a popular time series prediction model which combines moving average and autoregression components.
- **LASSO**: Lasso is a regression analysis method that performs both variable selection and regularization.
- **GBDT**: Gradient Boosting Decision Tree (GBDT) is a powerful and widely used ensemble method in data mining.
- **FFA** [7]: State-of-the-art air quality prediction model that is multi-view-based hybrid model considering spatial correlations, temporal dependencies, and sudden changes.
- **LSTM** [14]: Long-short-term-memory network (LSTM) is a special kind of recurrent neural network. Here, we use recent 12-hour AQI from target monitoring station as input.
- **DeepST** [15]: A CNN-based prediction approach for traffic prediction. Here, we convert the spatial partition from circles to grids with image size (5 * 5).
- **DMVST-Net** [26]: Deep multi-view spatial-temporal network uses CNN and LSTM to jointly consider the spatial, temporal, and semantic relations.
- **DeepSD** [19]: A sequential fusion architecture based deep neural network, fusing features iteratively in a sequence. DeepSD is designed for predicting car-hailing services.
- **DeepFM** [17]: Factorization-machine based neural network, modeling both high-order feature interactions and low-order feature interactions.
- **WFM**: A weather-forecast-based prediction method by Beijing municipal environmental monitoring center, providing a district-level min-max prediction for the next 12 hours, published at http://zx.bjmemc.com.cn/ at 8 am and 8 pm every day. We crawl the prediction results from 2014/10/1 to 2016/12/30.

Model Details

- Preprocessing. We use min-max normalization to normalize the continuous features into [0, 1], and use one-hot encoding to transform discrete features. In the evaluation, we rescale the predicted values back to the normal values.
- Hyper-parameters. We set all FusionNet with same parameters. In a FusionNet, we set the sizes of fully-connected layers as {24, 3}, and use one residual fully-connected layer after the first fully-connected layer. We select 90% of the training data for training each model, and the remaining 10% is chosen as the validation set for parameter tuning and early stopping. Afterward, we continue to train the model on the full training data for some epochs (*e.g.* 25 epochs).
- Activation Function: We use Sigmoid function as the activation function for output layer and use exponential linear unit [18] as the activation function for other fully-connected layers.
- Optimization Method: We apply Adam [19] to train the parameters with learning rate is 0.001 and batch size is 512. To prevent overfitting, we employ dropout [20] with probability 0.5 on the last layer of each FusionNet. Also, we apply L_2 regularization with weight 0.1 on the final loss function.
- Experimental environment: We train the models on a GPU server with Tesla K40m GPU and programming environment is Keras with TensorFlow as backend.

Evaluation Metrics

We use prediction accuracy (*acc*) and mean absolute error (*mae*) to evaluate our algorithms, which are defined as follow:

$$acc = 1 - \frac{\sum_i |\hat{y}_i - y_i|}{\sum_i y_i} \quad (1)$$

$$mae = \frac{\sum_i |\hat{y}_i - y_i|}{n} \quad (2)$$

Where $\hat{y}_i$ and y_i mean the prediction value and real value of i timestamp, and n is the total number of cases.

For sudden changes [7], we select the cases whose AQI is bigger than 100 and decreases over a threshold in the next few hours, *e.g.* 50 in the coming one hour, or 100 in the coming two hours, or 150 in the coming three hours.

4.2 Performance Comparison

Comparison with Different Baselines

Table 3 shows the performance of the proposed approach with other competing baselines. DeepAir achieves the highest accuracy in both general cases and sudden changes as it can automatically discover complicated air pollution patterns by modeling the underlying complex interactions of direct factors and indirect factors. By considering air quality data recorded by neighbor stations, LSTM-STC outperforms LSTM significantly, which shows the importance of spatial signals. The results of LSTM methods are not good for two reasons. One is that air quality is affected by many complex factors and the other is air quality has temporal closeness without obvious daily/weekly/monthly patterns. Comparing with DeepST, the results show that CNN is not suited in air quality prediction task as air quality data is sparse and the image size is small after preprocessing. As a result, DMVST-Net is not suited as other influential factors are more important than spatio-temporal correlations in the complex environment of air pollution. Comparing with DeepFM, the results show the effectiveness of DeepAir as DeepFM is designed for high-dimensional and extremely sparse data. Thus, a deep understanding of problem and data is important. Comparing with DeepSD, the results show that distributed architecture is more suited for air quality prediction task than sequential architecture as each indirect factor has an individual effect on direct factors affecting future air quality.

Comparison with Official Prediction

Table 4 shows the comparison between DeepAir and WFM during the time span: 2014/10/1 to 2016/12/30. As WFM provides the predictions in district-level min-max range for the next 12 hours and DeepAir provide the predictions in station-level for each hour over the next 48 hours, we evaluate the prediction results in both hourly station level and 12-hour min-max district level. For hourly station-level, we split the predictions of WFM to hourly station-level by considering the average of min-max range; for district-level, we merge the predictions of DeepAir to district-level and get the min-max range for the next 12 hours. In both evaluation settings, DeepAir has a higher accuracy than WFM. In addition, DeepAir has a finer spatial and temporal granularity, a farther prediction period and a faster updating frequency. From the results, we can also find DeepAir has a good performance on 12-hour district-level min-max prediction, which means DeepAir is robust and general enough for other prediction settings.

Table 4. Compare with Official Prediction in Beijing

Methods	Station Level		District Level		Update Hours	Grained Level
	acc	*mae*	*acc*	*mae*		
WFM	0.54	54.5	0.64	46.1	12	District
DeepAir	0.77	26.7	0.86	17.9	1	Station

Comparison with Previous Online Model

Figure 8 shows the comparison between DeepAir and previous state-of-the-art online approach, FFA, in AirPollutionPrediction system on 9 major Chinese cities. In general, DeepAir can achieve an average accuracy of (81.1%, 63%, 46%) in (1-6h, 7-48h, sudden changes) for all cities. Comparing with FFA, our approach has a better performance in all nine cities, with 2.4%, 12.2%, 63.2% relative accuracy improvements on short-term, long-term and sudden changes prediction. The reason behind it is that FFA trains four separate prediction models for modeling influential features respectively, which may fail to capture the interactions among all factors. Also, FFA is a shallow method which cannot capture the underlying complex pattern of each factor. Moreover, the features in FFA is not strong enough as it ignores the dynamic change of weather forecasts.

Table 3. Comparison with different baselines in Beijing. For neural network models, we run each of them 5 times and show "mean ± standard deviation".

Method	1-6h		7-12h		13-24h		24-48h		Sudden Change	
	acc	*mae*	*acc*	*mae*	*acc*	*mae*	*acc*	*mae*	*acc*	*mae*
ARIMA	0.751	28.3	0.576	52.1	0.458	65.4	0.307	74.6	0.066	112.9
LASSO	0.790	21.9	0.620	39.7	0.534	48.9	0.452	57.1	0.273	87.2
GBDT	0.792	21.8	0.629	38.8	0.540	48.0	0.458	56.5	0.321	21.8
LSTM	0.780	23.1±0.1	0.606	41.2±0.1	0.491	53.2±0.1	0.380	64.8±0.1	0.240	90.1±1.1
LSTM-STC	0.794	21.6±0.2	0.622	39.6±0.2	0.508	51.4±0.1	0.396	63.0±0.3	0.314	82.5±1.6
DeepST	0.806	20.4±0.1	0.633	38.1±0.2	0.545	47.5±0.2	0.466	55.7±0.7	0.380	74.5±2.9
DMVST-Net	0.806	20.4±0.1	0.638	37.8±0.3	0.550	47.4±0.5	0.481	53.9±0.7	0.419	70.4±2.0
DeepFM	0.808	20.1±0.1	0.643	37.3±0.2	0.549	47.2±0.6	0.474	54.9±0.6	0.396	72.3±1.9
DeepSD	0.811	19.7±0.1	0.645	37.1±0.2	0.551	46.8±0.8	0.479	54.3±0.7	0.428	69.5±3.3
DeepAir	**0.812**	**19.5±0.2**	**0.656**	**36.1±0.2**	**0.569**	**45.1±0.1**	**0.500**	**52.1±0.3**	**0.471**	**63.8±2.8**

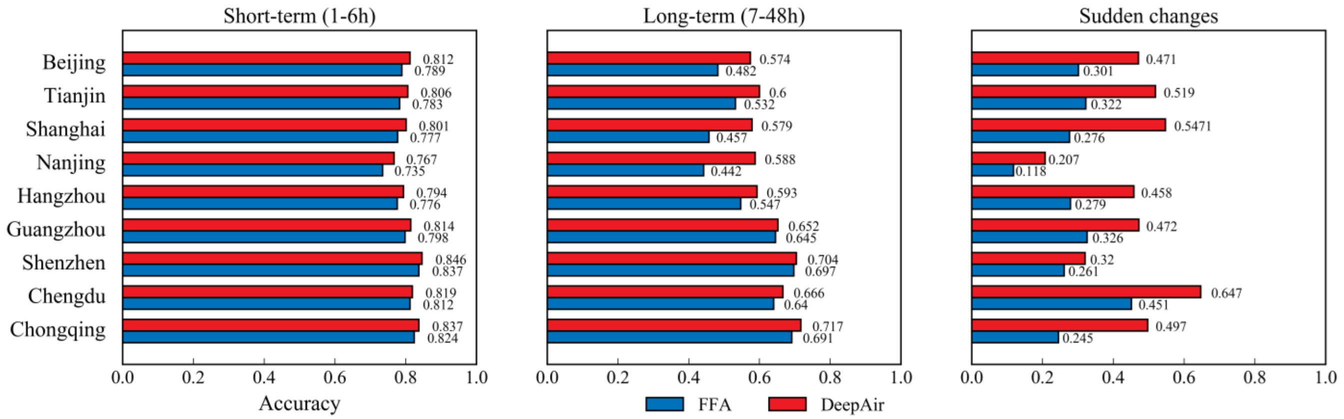


Figure 8: Comparison with the previous online model on nine major Chinese cities

Performance on Spatial Transformation

We show the effectiveness of our spatial transformation component (STC) in Table 5. Comparing with only using the air quality data from target station, DeepAir has a higher accuracy on general cases and sudden changes as air pollutants are dispersed in geographical space. With the signals from spatial neighbors, DeepAir can capture the dynamic changes of air quality. If we directly fed air quality readings from k-nearest stations (k=17, same size with STC) as inputs, the result is worse than STC. The reason behind it is each station has totally different k-nearest stations, while STC considers spatial correlations and generates a consistent input from eight directions. In STC, we find that inner & outer circles have a better performance than a single inner circle as it considers the signals from distant cities.

Table 5. Results on different preprocessing

Methods		1-6h		Sudden Change	
		acc	*mae*	*acc*	*mae*
Traditional	Target station	0.792	21.8	0.314	82.5
	17 nearest stations	0.802	20.1	0.37	75.2
STC	Inner circle	0.806	20.3	0.411	70.4
	Inner & outer circles	**0.812**	**19.5**	**0.471**	**63.8**

Performance on Distributed Fusion

We show the effectiveness of our distributed fusion architecture in Table 6. DeepAir outperforms all kinds of fusion combinations, bringing a significant improvement beyond direct influence and individual influences, a slightly better performance than holistic influence and distributed individual influences. Direct influence has a better result than individual influences in 1-6h and 7-12h, while has a worse result in 13-24h and 24-48h, which demonstrate that air quality changes a lot with the effects of other factors along time. Among all individual influences, WF has the best result for 13-48h, showing weather forecast is the most important factor for long-term prediction. Holistic influence and distributed individual influences have a better result than each individual influence, which also demonstrates that air quality is affected by multiple factors.

Table 6. Results on different fusion architectures

Methods (*acc*)		1-6h	7-12h	13-24h	24-48h
Direct Influence	AQIs	0.793	0.624	0.508	0.398
Individual Influence	HW	0.739	0.605	0.517	0.412
	WF	0.752	0.607	0.549	0.472
	SP	0.750	0.596	0.509	0.399
	MP	0.758	0.613	0.510	0.399
Holistic Influence	HI	0.772	0.630	0.564	0.496
Distributed (HW,WF,SP,MP)		0.808	0.653	0.565	0.495
DeepAir		**0.812**	**0.656**	**0.569**	**0.500**

Performance on Embedding

We show the effectiveness of embedding method in Table 7. After embedding, we can see a clear improvement on general cases and sudden changes after as it captures the intra-dynamics of each factor. Especially for direct factors, embedding can learn the spatio-temporal correlations of air pollution dispersion.

Table 7. Results on embedding setting

Methods	1-6h		Sudden Change	
	acc	*mae*	*acc*	*mae*
w/o embedding	0.807	20.2	0.429	68.1
with embedding	**0.812**	**19.5**	**0.471**	**63.8**

5 RELATED WORK

5.1 Air Quality Prediction

Air quality prediction methods mainly fall into two categories: classical dispersion models and data-driven models [23, 24]. Classical dispersion models identify the root cause of air pollution from chemical, emission, climatological and combinations of these factors. These models are most a numerical function of emissions from industry and vehicular, meteorology, and other factors. However, it is very difficult to get all these factors completely and accurately. Thus, the prediction accuracy is hard to be guaranteed. Also, the computation complexity is very high.

Data-driven models, *e.g.* artificial neural networks, forecast air pollutions based on a variety of features. Recently, Zheng et al. proposed a multi-view-based hybrid model [7], consisting of a temporal predictor, a spatial predictor, a dynamic aggregator, and an inflection predictor. However, FFA is a shallow ensemble method, which may fail to capture complex interactions between influential factors. Also, the features used in FFA are not strong enough. Our DeepAir approach learns the air pollution patterns in a deep manner, simultaneously considering the individual and holistic influences, which is more capable of predicting general cases and sudden changes than FFA.

5.2 DNN for Spatio-Temporal Prediction

Currently, many works show the strength of DNN on solving spatio-temporal prediction problems. Song et al. proposed a recurrent neural network to simulate and predict human mobility [25]. To predict citywide crowd flows, Zhang et al. proposed a CNN-based network to extract features [13, 15, 16]. Yao et al. proposed a deep multi-view network to predict taxi demand based on CNN and LSTM [26]. Among these methods, CNN is wildly used for capturing spatial correlation and LSTM is used for modeling temporal dependency. In our task, we use DNN to learn the spatio-temporal correlations without CNN and LSTM due to the characteristics of air pollution. As air quality data is sparse in the spatial dimension, CNN is not suited for handling such sparse data. Another is air quality do not have strong temporal dependency as it is heavily affected by other factors.

For fusing cross-domain data by DNN, simple methods directly concatenate all features together. Recently, Wang et al. adapted a sequential fusion architecture, which fuses two features firstly, then fuses some new features in the same manner iteratively in a sequence [22]. However, sequential fusion need design the order of fusion sequence, which costs lots of time for tuning. Our DeepAir adapts a distributed fusion architecture to learn the feature interactions by enhancing main feature interacting with auxiliary features respectively, which is derived from domain knowledge as each indirect factor will have an individual effect on direct factors affecting future air quality.

6 CONCLUSION

In this paper, we propose a DNN-based approach to predict air quality. Based on the domain knowledge about air pollution, we adopt a novel distributed fusion architecture to fuse heterogeneous urban data, which can simultaneously capture the individual and holistic effects from all influential factors affecting air quality. Comparing with 10 baselines with three-year data from 9 Chinese cities, our approach achieves a higher accuracy in both general cases and sudden changes. We have deployed DeepAir in AirPollutionPrediction system, providing fine-grained air quality forecasts for 300+ Chinese cities every hour. Comparing with the previous online approach in the system, we have 2.4%, 12.2%, 63.2% relative accuracy improvements on short-term, long-term and sudden changes prediction, respectively.

In the future, we want to investigate the long-term sudden changes prediction as it is very difficult and important.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Grant (Nos. 61672399, U1609217, 61773324) and Cultivation Program for the Excellent Doctoral Dissertation of Southwest Jiaotong University.

References

[1] Akimoto Hajime. 2003. Global air quality and pollution. *Science* 302.5651 (2003), 1716-1719.

[2] Kampa Marilena, and Elias Castanas. 2008. Human health effects of air pollution. *Environmental pollution* 151, 2 (2008), 362-367.

[3] Julie Yixuan Zhu, Yu Zheng, Xiuwen Yi, Victor O.K. Li. 2016. A Gaussian Bayesian Model to Identify Spatio-temporal Causalities for Air Pollution based on Urban Big Data. In *INFOCOM WKSHPS.* IEEE, 3-8.

[4] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-Air: When Urban Air Quality Inference Meets Big Data. In *SIGKDD.* ACM, 1436-1444.

[5] Jianyi Lu, and Xin Cao. 2015. $PM_{2.5}$ Pollution in major cities in China: pollution status, emission sources and control measures. *Fresenius Environmental Bulletin* 24, 4A (2014): 1338-1349.

[6] Elena Baralis, Tania Cerquitelli, Silvia Chiusano, Paolo Garza, and Mohammad Reza Kavoosifar. 2016. Analyzing air pollution on the urban environment. In *MIPRO.* IEEE, 1464-1469.

[7] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting Fine-Grained Air Quality Based on Big Data. In *SIGKDD.* ACM, 2267-2276.

[8] AirPollutionPrediction Website: http://airprediction.urban-computing.com.

[9] Jie Bao, Ruiyuan Li, Xiuwen Yi, Yu Zheng. 2016. Managing massive trajectories on the cloud. In *ACM SIGSPATIAL.* ACM. 41.

[10] Xiuwen Yi, Yu Zheng, Junbo Zhang, Tianrui Li. 2015. ST-MVL: Filling Missing Values in Geo-sensory Time Series Data. In *IJCAI.* 2704-2710.

[11] George Y. Lu, David W. Wong. An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences,* 34,9 (2008), 1044-1055.

[12] Tobler, Waldo R.. 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography* 46, 2 (1970), 234-240.

[13] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI,* 1655-1661.

[14] Sepp Hochreiter, and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[15] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-Based Prediction Model for Spatio-Temporal Data. In *SIGSPATIAL GIS.* ACM, 92.

[16] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, Tianrui Li. 2018. Predicting Citywide Crowd Flows Using Deep Spatio-Temporal Residual Networks. *Artificial Intelligence*, 259, (2018), 147-166,

[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *arXiv preprint arXiv*:1703.04247 (2017).

[18] Djork-Arne Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint arXiv*, 1511.07289 (2015).

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

[20] Srivastava Nitish, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014): 1929-1958.

[21] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, Yu Zheng. 2018. When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks. In *AAAI.*

[22] Dong Wang, Wei Cao, Jian Li, Jieping Ye. 2017. DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks. In *ICDE.* IEEE, 243-254.

[23] Yang Zhang, Bocquet Marc, Mallet Vivien, Seigneur Christian and Baklanov Alexander. 2012 a. Real-time Air Quality Forecasting, Part I: History, techniques, and current status, *Atmospheric Environment*, 60, (2012), 632-655.

[24] Yang Zhang, Bocquet Marc, Mallet Vivien, Seigneur Christian and Baklanov Alexander. 2012 b. Real-time Air Quality Forecasting, Part II: State of the science, current research needs, and future prospects, *Atmospheric Environment*, 60 (2012), 656-676.

[25] Xuan Song, Kanasugi Hiroshi, and Shibasaki Ryosuke. 2016. DeepTransport: Prediction and simulation of human mobility and transport mode at a citywide level. In *AAAI.* 2618-2624.

[26] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *AAAI.*

[27] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, Yu Zheng. 2018. Multi-level Attention Networks for Geo-sensory Time Series Prediction. In *IJCAI.*