# Predicting Fine-Grained Air Quality Based on Deep Neural Networks

Xiuwen Yi, Zhewen Duan, Ruiyuan Li, Junbo Zhang, Tianrui Li, Yu Zheng

**Abstract**—Nowadays, many cities are suffering from air pollution problems, which endangered the health of the young and elderly for breathing problems. For supporting the government's policy-making and people's decision making, it is important to predict future fine-grained air quality. In this paper, we predict the air quality of the next 48 hours for each monitoring station and the daily average air quality of the next 7 days for a city, considering air quality data, meteorology data, and weather forecast data. Based on the domain knowledge about air pollution, we propose a deep neural network based approach, entitled DeepAir. Our approach consists of a deep distributed fusion network for station-level short-term prediction and a deep cascaded fusion network for the city-level long-term forecast. With the data transformation preprocessing, the former network adopts a neural distributed architecture to fuse heterogeneous urban data for simultaneously capturing the direct and indirect factors affecting air quality. The latter network takes a neural cascaded architecture to learn the dynamic influences from previously existing data and future predicted data on future air quality. We have deployed a real-time system on the cloud, providing fine-grained air quality forecasts for 300+ Chinese cities every hour. Our system mainly consists of three components: data crawler, task scheduler, and prediction model, which are implemented with a multi-task architecture to improve the system's efficiency and stability. Based on the datasets from three-year nine Chinese cities, experimental results demonstrate the advantages of our proposed method.

**Index Terms**—Air Quality Prediction; Deep Learning; Data Fusion; Urban Computing

## 1 Introduction

With the rapid development of urbanization, air pollution is becoming a severe issue for many cities [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 on breathing problems [2]. For monitoring real-time air pollution, Chinese governments have built many air quality monitoring stations and published air quality information to the public every hour [3]. Besides monitoring, there is a rising demand for predicting future fine-grained air quality. Such predictions can inform the government's policy-making (e.g., performing traffic control) and people's decision making (e.g., whether to exercise outdoors tomorrow).

However, predicting future air quality is very challenging because of the following reasons:

First, air quality has multiple influential factors. As shown in Figure 1, air pollutant sources mainly come from vehicle exhaust, industrial emission, coal burning, and dust [4], where each source has different spatio-temporal patterns and pollutant particles. Moreover, the air quality is affected by local emission, regional transport, meteorological conditions [5]. Depending on the impact, these factors fall into two groups. Local emission and regional transport are direct factors as they mainly determine the formation of pollutants; meteorological conditions, secondary productions, terrain, and time are indirect factors as they primarily decide the development environment of pollutants. However, we do not have sufficient and accurate data to model these factors precisely [6]. For example, it is almost impossible to obtain city-wide pollutant emissions. Likewise, weather forecasts are not accurate enough as "The longer the forecast horizon is, the less accurate the forecast will be."

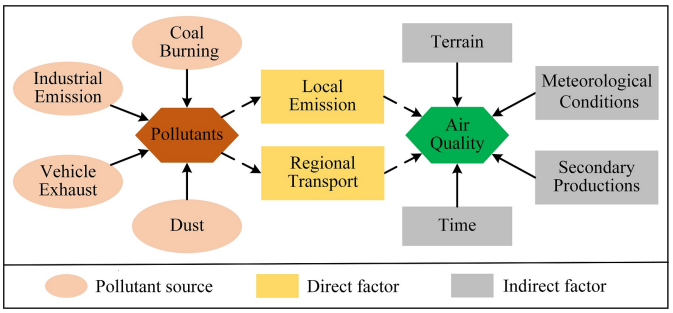
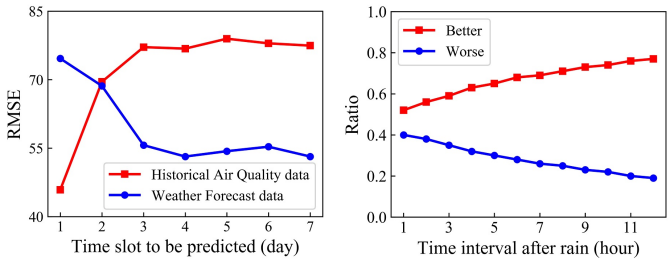



Fig. 1. Multiple influential factors on air pollution

Second, the interactions between these factors are complex. When predicting air quality with only one kind of data using multi-layer perceptron, the results of Beijing for $PM_{2.5}$ are shown in Figure 2(a). We can see the RMSE of air quality and weather forecast is opposite along time, where the former increases while the latter decreases. The reason behind it is that the historical air quality data is constant for the day 1 to day 7 predictions, while the weather forecast data captures the future dynamics until the time slot to be predicted. As a result, the importance


- *Xiuwen Yi is with JD Intelligent Cities Research and Tsinghua University. E-mail: xiuwenyi@foxmail.com*
- *Zhewen Duan and Ruiyuan Li are with Xidian University and JD Intelligent Cities Research. E-mail: {duanzhewen, ruiyuan.li}@jd.com*
- *Junbo Zhang and Yu Zheng are with JD Intelligent Cities Research and JD Intelligent Cities Business Unit. E-mail: {msjunbozhang, msyuzheng}@outlook.com*
- *Tianrui Li is with Institute of Artificial Intelligence, Southwest Jiaotong University, China. E-mail: trli@swjtu.edu.cn*

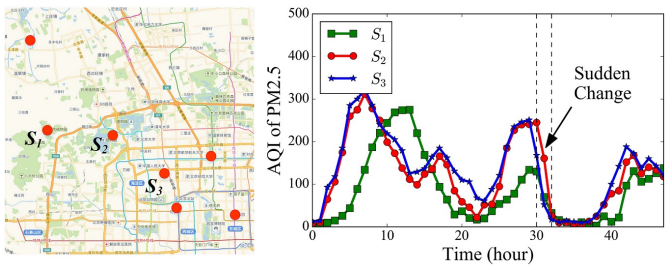



a) Effects from different data b) Rain affect air quality

Fig. 2. Complex interaction among influential factors

of different data sources is different over time. Thus, it is important to design an appropriate mechanism to fuse these data. Moreover, many people have a sense of air quality will be better after the rain. However, air quality will be worse in some cases. Figure 2(b) shows the effect of rain on air quality based on the statistical results of three-year data in Beijing from our dataset. We calculate ratios by counting the proportion of $\Delta_k = AQI_{t+k} - AQI_t$, where $AQI_t \geq 100$, $Weather_t = rain$ and $k$ is the time interval after rain. Here, the sum of the raising and dropping ratio is less than 1 as it exists the situation of unchanged after rain. We can find that it still has more than 20% ratio that air quality will be worse after rain with 12 hours later. This is because air quality is affected by multiple factors simultaneously, where the effect of a single influential factor is not absolute.

Third, air quality changes over location and time significantly and sometimes coming with a sudden change. As shown in Figure 3, the air quality always fluctuates along time without apparent daily and weekly periodic patterns and change differently over locations. Moreover, we can find some sudden changes where the air quality index (AQI) drops very sharply in a very short period [7]. As illustrated in Figure 3(b), AQI of monitoring station $S_2$ at the $30^{th}$ timestamp drops over 200 in the coming two hours due to a strong wind blowing from the southeast. Such a sudden change is important, where people always pay more attention to sudden changes than general cases in daily life. 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 the whole dataset. Among the three-year air quality data, the presence of sudden changes is less than 2.3%. Such a data imbalance phenomenon brings many difficulties for air quality prediction.




a) Monitoring stations b) Air quality data

Fig. 3. Air quality change over location and time

To address these challenges, we propose a DNN based approach to predict the air quality of the next 48 hours for a monitoring station and the daily average air quality of the next 7 days for a city, considering air quality data, meteorology data, and weather forecasts. Our approach is inspired by the domain knowledge about air pollution, which can help design model structure with more interpretations. For short-term predictions, as direct and indirect factors have different influences on air quality and all indirect factors will affect direct factors, we capture these individual and holistic influences simultaneously by distributed fusion architecture. For long-term predictions, considering the opposite effect of historical air quality and weather forecast along time, we combine the advantages of these two features for capturing the dynamic interactions by cascaded fusion architecture. Our contributions are listed as below:

- We develop a real-time air quality prediction system, providing short-term and long-term prediction services for 300+ cities. To improve the system's efficiency and stability, we implement three core system components, data crawler, task scheduler, and prediction model, with a multi-task architecture.
- For station-level short-term prediction, we propose a deep distributed fusion network, which adopts a novel distributed architecture to fuse heterogeneous urban data for simultaneously modeling the individual and holistic influences.
- For city-level long-term prediction, we propose a deep cascaded fusion network, which adopts a novel cascaded architecture to fuse the previously existing data and future predicted data for learning the contextual influences.
- Based on three-year data from nine Chinese cities, the results demonstrate the advantages of our proposed approach for both station-level short-term and city-level long-term air quality prediction.

## 2 System Overview

Figure 4 shows the system architecture, which mainly consists of three parts: Data Crawler, Task Scheduler, and Prediction Model. Here, we consider air quality data, meteorology data, and weather forecast data as real-time data sources. Data crawler, deployed on the cloud, continuously collect these real-time data from web pages or API interfaces and then feed these data into Database (e.g., MySQL) and cache (e.g., Redis). If the collected data meets the requirements of prediction, the task scheduler will invoke the prediction model. Note that, data crawler and task scheduler are implemented with a multi-thread and multi-queue based multi-task architecture for improving the system's efficiency. As for the prediction model, we respectively predict station-level short-term air quality and city-level long-term air quality after data preprocessing. Here, we train the neural networks in the local GPU servers and run predictions on the cloud, where online prediction is implemented with multi-task architecture. Then, prediction results are stored in another cache for fast data changing to end-user. To back up the data, prediction results will backup to the Database periodically. Finally, we visualize the real-time prediction results on the web through web services.

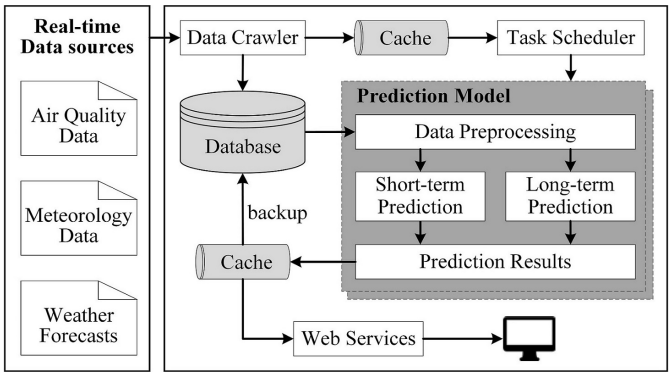



Fig. 4. Architecture of air quality prediction system

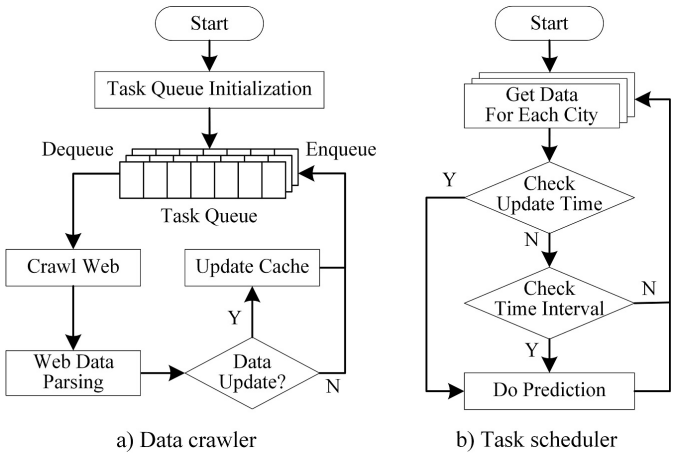



Fig. 5. Procedure of data crawler and task scheduler

### 2.1 DATA CRAWLER

Data crawler is the most basic and fragile component of a real-time system, which provides essential data to keep the system stable. However, it is difficult to crawl these dynamic data effectively. The reasons are two-fold: First, numerous different data sources need to be collected. The meteorology/weather forecast/air quality data are collected from 302 Chinese cities, 2,812 districts, and 2,296 air quality monitoring stations. Second, the updating frequency of different data sources is completely different due to many complex factors. Typically, the updating time of air quality data and meteorology data is one-hour, while for the weather forecast data, the updating time is twelve-hour. However, the updating time of different stations/cities are different, some are fast, and some are slow. Even for a station, the updating time is unstable.

To improve our system's efficiency and robustness, we designed the data crawler with a multi-thread and multi-queue based multi-task architecture. Here, we consider crawling one kind of data of a city as a task, where each task will be run with a thread, respectively. As the number of tasks is huge, we aggregate all tasks from the same web domain as a task queue, e.g., crawl meteorology data of all Chinese cities from http://www.weather.com.cn/. Thus, we can crawl these data in a parallel manner for fast data collecting in a near-real-time manner. Moreover, as the characteristics of all web pages in the same web domain are similar, it is also simplified configuration and fast management with this grouping strategy.

Figure 5(a) depicts the procedure of data crawler. When the program starts, we first initialize each task queue configuration, including the target URLs, maximum parallel number, maximum crawl time interval, time sleeping interval. Then, we check each crawl task's time interval after the same task's last crawl time. If it exceeds a threshold, we pop this task from the task queue and crawl the web page. Here, we can not continually crawl one web page for two reasons: resource-wasting and the web may block IP. After web data parsing, we can get the target data. If the data is updated comparing with the last updated time, we update it into our database and cache. After that, this task will be inserted into the tail of the queue and waiting for the next task call. For all tasks in the same task queue, they are run in a sequential manner; while for the tasks in the different task queues, they are run in a parallel manner.

### 2.2 TASK SCHEDULER

As the updating time of air quality data for 300+ cities is different, it is important to make predictions for each city asynchronously. Thus, we design a task scheduler module, which schedules the time for prediction after checking the collected data. As is shown in Figure 5(b), the schedule module is a loop execution service with multi-thread based multi-task architecture. Here, we consider scheduling the prediction for a city as a task, where each task will be run with a thread. For each task, it reads the updating time for each station firstly. Then, we check the updating time of all stations in a city. If the number of updated stations on a city's total stations exceeds a threshold, we conduct the predictions. If not, we check the time interval from the last prediction. If we find the time interval does not exceed a threshold, the thread will be hung up and wait for the next wake-up. Thus, the task scheduler can decide the time and order. With such a parallel architecture, our task scheduler can improve the system's efficiency.

### 2.3 Prediction Model

The prediction model is a core part of the system. Here, we propose a deep distributed fusion network for predicting station-level air quality of the next 48 hours and a deep cascaded fusion network for predicting city-level daily average air quality of the next 7 days. The details about the prediction model are described in Section 3. As the updating time of different cities is different, we scale a few instances for parallel running predictions following the command of the task scheduler. With such multi-task architecture, we can improve the systems efficiency.

### 2.4 WEB INTERFACE

Figure 6 shows the web interface of our air prediction system. All stations of a city are marked on the map and attached with their real-time AQI, where the darker color represents the worse air condition. In our system, we can get station-level air quality prediction for the following 48 hours, as is depicted in the center of the figure, and the city-level daily air quality prediction for the next 7 days, as is shown in the right chart of the figure. Here, we provide different cities and pollutants for users to select.


Fig. 6. Web interface of our system

## 3 Prediction Model

As different users have different prediction requirements, some care about station-level short-term, and some care about city-level long-term. Thus, we predict the fine-grained air quality in both settings with a deep distributed fusion network and a deep cascaded fusion network.

### 3.1 Station-level Short-term Prediction

As shown in Figure 7, for station-level short-term air quality prediction, we propose an approach, entitled DA-Short, which consists of a spatial transformation component and deep distributed fusion network. Considering the spatial correlations, the spatial transformation component uses the partition, aggregation, and 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, are fed into a deep distributed fusion network to model the individual and holistic influences simultaneously. Here, we use the embedding of AQIs to simulate the direct factors and use the embedding of rest datasets as indirect factors. As we knew, each indirect factor has its effort on direct factors affecting future air quality. Thus, 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. After that, the outputs of five subnets are aggregated to generate the final results.

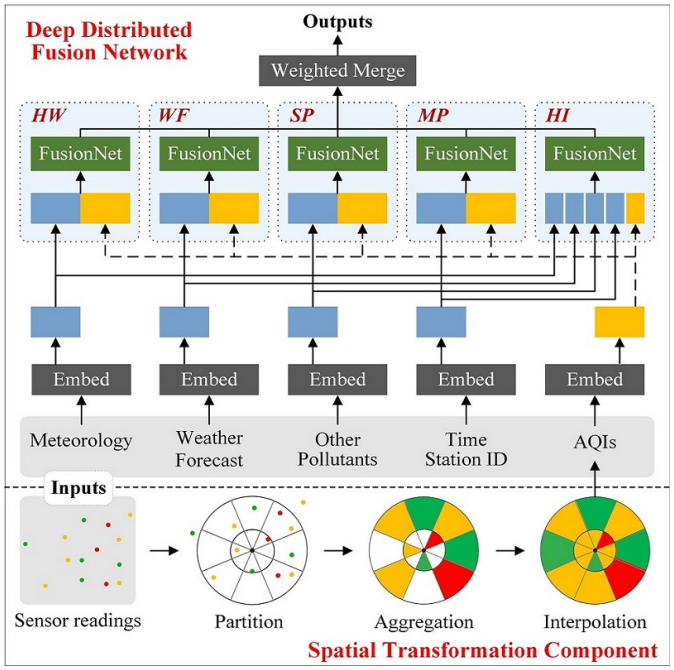



Fig. 7. Framework of station-level short-term prediction

Here, for temporal granularity, we collectively predict the air quality in a couple of hours, e.g., 1-3 hours, as weather forecasts are segmented into 3-hour time intervals. For spatial granularity, we build one predictive model for all monitoring stations in the same city as the spatial transformation component will generate a consistent input.

#### 3.1.1 Spatial Transformation

As pollutants are dispersed in geographical space [6], 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. However, as shown in the left bottom of Figure 7, air quality monitoring stations are randomly scattered in geographical space, where the color on the dot means the level of air quality. For converting spatial sparse air quality data into a consistent input for the further predictive model, we devise the spatial transformation component, which mainly consists of partition, aggregation, and interpolation.

Firstly, we partition the geographical space into 16 regions by four lines and two circles, e.g., 20 km and 100 km semidiameter. Thus, all regions share the target monitoring station as a 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. 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. Specifically, we first randomly generate some fake monitoring stations in these regions. Then, we use a classic spatial interpolation method, inverse distance weighting (IDW) [8], to interpolate the AQI of fake monitoring stations. Considering the geo-spatially adjacent stations located both inside and outside the outer circle, IDW assigns a weight to each available AQI reading of adjacent stations by the distance to the target sensor, and then aggregates these weights and readings by weighted average. After that, we aggregate the interpolated values of fake stations to calculate the average AQI for the region. Finally, we get 17 AQI in one timestamp where 1 AQI comes from the target station, and 16 AQI come from neighbor regions. We conduct the same process for each monitoring station over time.

We design the spatial transformation, 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 merges the scattered air quality data into regions. Closer regions have finer granularity, and farther regions have a coarser granularity. Moreover, regions with different distances show different impacts varying by distance, which follows the First Law of Geography [9], 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 will further facilitate data augmentation for training deep neural networks.

### 3.1.2 DEEP DISTRIBUTED FUSION

As we know, air pollution has multiple influential factors, where local emission and regional transport are direct factors, while meteorological conditions, terrain and time are indirect factors. Moreover, direct and indirect factors have different influences on future air quality. At most times, all indirect factors will simultaneously affect direct factors. Also, each indirect factor has its effect on direct factors. For simultaneously capturing these influential factors, we propose a distributed fusion architecture based neural network, as shown in the top area of Figure 7.

Firstly, we use the embedding [10] of influential factors to simulate the direct factors and indirect factors. For categorical features, embedding can transform the features represented by one-hot encoding to a real-valued vector for capturing the categories' similarity. While for numerical features, embedding can transform the raw features into a low-dimensional space for learning the hidden representation. Then, we design a distributed fusion architecture with five subnets (HW, WF, SP, MP, and HI) to model the holistic influence from all influential factors and the individual influences from the historical weather, weather forecast, secondary productions, and meta properties. Our model is end-to-end and we can optimize the embedding parameters together with other parameters in the neural network during the model training phase.

As illustrated in 8(a), distributed fusion architecture specifies the direct factor as main feature and other indirect factors as auxiliary features. Then, main feature respectively interacting with each auxiliary feature in a parallel manner and merges the outputs to learn the joint effects. With such architecture, we can highlight the main feature and capture the influences from auxiliary features. The reason for the partition is main feature and prediction target come from the same domain, while auxiliary features and prediction targets come from different domains. In our task, we specify the embedding of AQIs as main feature and the embedding of other features (i.e. meteorology) as auxiliary features, where main feature can simulate the direct factors from local emission and regional transport, while auxiliary features can represent the indirect factors. Here, the main feature is shared across all subnets and all subnets have the same network structure, FusionNet.

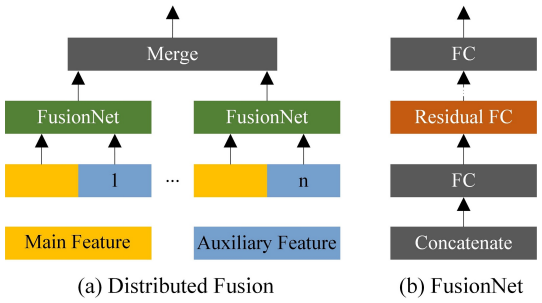



Fig. 8. Architecture of distributed fusion

As shown in Figure 8(b), FusionNet comprises a concatenate layer, some fully-connected (FC) layers, and a residual FC layer. Firstly, we merge all features together by using a concatenate layer, then use some FC layers to learn higher-order feature interactions. For training the neural network more robust, we add one residual FC layer [10] between FC layers, the previous information of which can be directly passed to the following layers through the shortcut connections. Residual FC layer is derived from ResNet [11], where the residual mapping is learned by $H(x) = x + \mathcal{F}(x)$ and broadcast to the following layers.

We build historical weather subnet (HW) and weather forecast subnet (WF) for capturing historical and future weather conditions, respectively. The reason for building such two subnets is about data realism and time interval, where historical weather provides hourly real weather conditions while the weather forecast provides 3-hour segmented forecasted weather. Here, we consider the weather, wind speed, wind direction, humidity, and pressure as features for historical weather data. While for weather forecast data, we consider the weather, wind strength, and wind direction as features. After feeding AQIs and historical weather into HW subnet, we get $\mathbf{y}_{hw}$ as the output. Besides, we will get $\mathbf{y}_{wf}$ as output when we feeding AQIs and weather forecasts into WF subnet.

Besides the direct emission of pollutants, it exists some secondary chemical reaction among pollutants in the atmosphere. Thus, we design a secondary production subnet (SP) to simulate the chemical interaction. 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 $\mathbf{y}_{sp}$.

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 the temporal dimension, e.g. winter always has a bad air quality than summer. 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 $\mathbf{y}_{mp}$.

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) for learning the holistic influence by fusing all direct and indirect factors. Then, we get $\mathbf{y}_{hi}$.

Though air quality is affected by multiple factors, the degree of influences may be different. Inspired by such ob-

servation, the outputs of five subnets are weighted merged using a parametric-matrix-based fusion [12] to model the dynamic influences and generate the final results:

$$\hat{\mathbf{y}} = Sigmoid(\mathbf{y}_{hw} \cdot \mathbf{w}_{hw} + \mathbf{y}_{wf} \cdot \mathbf{w}_{wf} + \mathbf{y}_{sp} \cdot \mathbf{w}_{sp} + \mathbf{y}_{mp} \cdot \mathbf{w}_{mp} + \mathbf{y}_{hi} \cdot \mathbf{w}_{hi}) \quad (1)$$

where $\hat{\mathbf{y}} \in R^h$ are the predicted results, $\mathbf{y}_{hw}, \mathbf{y}_{wf}, \mathbf{y}_{sp}, \mathbf{y}_{mp}, \mathbf{y}_{hi}$ are the outputs of five subnets. $\cdot$ is Hadamard product, and $\mathbf{w}_{hw}, \mathbf{w}_{wf}, \mathbf{w}_{sp}, \mathbf{w}_{mp}, \mathbf{w}_{hi}$ 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.

## 3.2 City-level Long-term Prediction

With the DA-Short model, we can predict the next 48 hours' air quality for each monitoring station. Moreover, it is important to predict the daily average air quality of the next 7 days for a city. Most people only care about coarse-grained air quality, e.g., the average air quality in a city for the next few days, ignoring the hour-level air quality. Different from station-level short-term prediction, city-level long-term prediction faces some other challenges:

First, air quality shows strong continuity in a short period, while it does not exist daily or weekly periodicity. Thus, historical air quality plays a limited role in long-term air quality prediction, where "the longer the forecast horizon is, the less importance will be." Second, there is insufficient data to describe the future information except for weather forecasts. Though the accuracy of weather forecast decreases over time, it is still important to highlight the influence of weather forecast for long-term prediction. Third, it is hard to distinguish the main feature and auxiliary features as the effect of all features change a lot over time, let alone the next 7 days. Thus, we cannot directly reuse DA-Short for long-term air quality prediction.

As we know, the influences from historical air quality and weather forecasts are opposite along time, where the former decreases while the latter increases. For capturing these complex dynamic interactions, as shown in Figure 9, we propose a deep cascaded fusion network (DA-Long) to predict city-level long-term air quality. More specifically, we firstly embed air quality data, meteorology data, and weather forecast data to learn the intra-dynamics of each influential factor. Then, air quality data fuse weather condition data iteratively with FusionNet, which can simulate the dynamic interaction between these influential factors along time. Different from only considering the result of the last fusion, we treat all fusion results equally and aggregate all the fusion results using a weighted merge to generate the final prediction.

Here, we extract both region-level AQI of the past few hours by spatial transformation and city-level daily average AQI of the past few days for air quality data. Then, we fuse both features to simultaneously capture the fine-grained and coarse-grained spatial and temporal characteristics of air quality. As for the weather forecast data, we split the weather forecast data into daily granularity, where each day contains a sequence of weather forecast instances. Also, we regard the current and last few hours' meteorology data as the 0-day weather forecast for convenience.

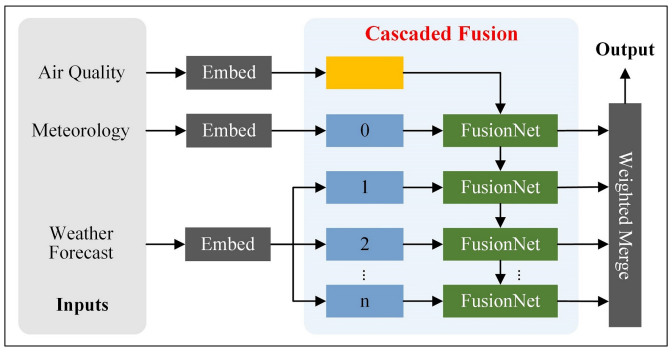



Fig. 9. Framework of city-level long-term prediction

### 3.2.1 Deep Cascaded Fusion

Considering the time information and data realism, we can specify the air quality and meteorology as previous existing data and specify weather forecast as future predicted data. The former data contains the ground information already happened, and the latter data contains the information happening in the future. As we know, the influences from previous existing data become weaken, and the influences from future predicted data become strengthen along time. For taking the advantages of both data, we design a cascaded fusion architecture to capture such dynamic interactions along time, as shown in Figure 10.

More specifically, we treat the two input parts of FusionNet are predecessor features and successor features. For example, the first predecessor feature is air quality data; the latter predecessor features are the results of FusionNet, and each slice of weather forecast is a successor feature. Cascaded fusion architecture captures the dynamic correlations as predecessor feature fuses with each successor feature sequentially. With the cascaded fusion architecture, we can weaken the original predecessor feature's effect and strengthen the last successor feature along with time. With this characteristic, comparing with distributed fusion architecture, cascaded fusion architecture is more suitable for long-term air quality predictions for simulating the dynamic interactions along time.

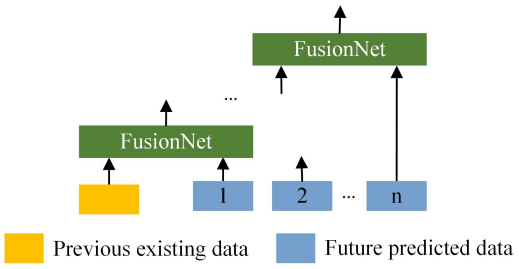



Fig. 10. Architecture of cascaded fusion

## 3.3 Algorithm

Algorithm 1 outlines the procedure of DA-Long. We first construct the training instances (line 1-8) and then train the model via backpropagation to minimize the loss (line 9-12). The pseudo-code of training DA-Short is similar. Here we ignore it, where can be viewed in our former paper [13].

**Algorithm 1:** DA-Long Training Algorithm

**Input:** Historical AQI observations $\{AQI_S^t\}_{t=1}^T$;
Historical weather coniditions $\{M_S^t\}_{t=1}^T$;
Weather forecasts $\{W_S^t\}_{t=1}^{T+k}$;Future time interval $k$;
Length of past sequence $h$;
**Output:** Learned DA-Long model

/* construct training instances */
**for** $\forall t \in [1, T]$ **do**
  **for** $\forall i \in S$ **do**
    $x_{ad}$=Daily_Aggregation($[AQI_S^{t-h}, \cdots, AQI_S^t]$)
    $x_{ah}$=Spatial_Transformations($[AQI_S^{t-h}, \cdots, AQI_S^t]$)
    $x_{aqi}$=$[x_{ad}, x_{ah}]$
    $x_m = [M_i^{t-h}, \cdots, M_i^t]$
    $x_{wf} = [W_i^t, \cdots, W_i^{t+k}]$
    $y$=Get_Prediction_Target($AQI_S^{t+k}$)
  Append($\{x_{aqi}, x_m, x_{wf}\}, y$ into $\mathcal{D}$)
/* train the model */
initialize all learnable parameters $\theta$ in DA-Long
**while** *stopping criteria is not met* **do**
  randomly select a batch of instances $\mathcal{D}_b$ from $\mathcal{D}$
  find $\theta$ by minimizing the loss function with $\mathcal{D}_b$

# 4 EXPERIMENTS

## 4.1 Settings

### 4.1.1 Datasets

Air quality data: Our system collects air quality data from 2,296 official air quality monitoring stations in 302 Chinese cities every hour. Each air quality record consists of the concentration of 6 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 seven days for each update, which is usually segmented into a 3-hour time interval. Each record consists of weather, temperature, wind strength, and wind direction.

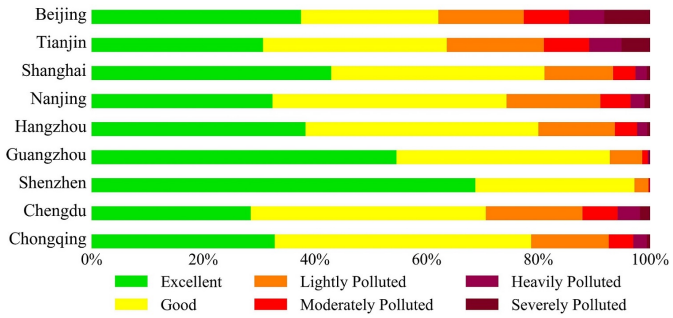



Fig. 11. Air pollution level among different cities

For evaluation, we use three-year data (from 2014/5/1 to 2017/4/30) in nine major Chinese cities (Beijing, Tianjin, Shanghai, Nanjing, Hangzhou, Guangzhou, Shenzhen, Chengdu, and Chongqing), where the first 24 months data for training and the last 12 months for testing. Figure 11 shows the distribution of AQI of $PM_{2.5}$ between 2014/5 to 2017/4 in nine cities, whose colors, defined by Chinese standards, represent the level of air pollution. As Beijing has the most complicated air quality, we focus on the data in Beijing when comparing with different baselines while showing overall results for the other eight cities. As $PM_{2.5}$ is the main concern for air quality, all results in experiments are based on $PM_{2.5}$. Table 1 details the statistical results in Beijing on $PM_{2.5}$. For predicting the air quality of 36 monitoring stations in Beijing, 74 neighbor air quality stations, 17 meteorology stations, and 17 weather forecast stations are retrieved within 100km (semidiameter). We collect 875,394 air quality records with 327,514 meteorology instances and 298,790 weather forecast instances. Among air quality records, 2.3% cases are sudden changes.

TABLE 1
Data statistic of Beijing dataset

| | | |
|---|---|---|
| Air Quality | In-city stations | 36 |
| | Instances | 875,394 |
| | Sudden changes | 20,540 |
| | Average PM2.5 | 118.2 |
| | Neighbor stations | 74 |
| Meteorology | Sources | 17 |
| | Instances | 327,514 |
| Weather Forecast | Sources | 17 |
| | Instances | 298,790 |

### 4.1.2 Baselines

- LR: Linear Regression (LR) is a linear approach to model the relationship among features.
- ARIMA: Autoregressive integrated moving average (ARIMA) is a time series prediction model combining moving average and autoregression components. We set the orders as (6,1,1).
- LASSO: Lasso is a regression analysis method that performs both variable selection and regularization. Here, we set $\alpha$ as 0.00001, others as default value.
- GBRT: Gradient Boosting Regression Tree (GBRT) is a tree-based ensemble method. We set the number of trees as 100, learning rate as 0.1, max depth as 5.
- FFA [7]: A multi-view-based air quality prediction model containing spatial predictor, temporal predictor, inflection predictor and prediction aggregator. The parameters are the same as the original paper.
- FNN: Feedforward Neural Network (FNN), flattens all the features and then feeds them together into a multi-layer fully-connected network. The layer sizes are set as 64, 32, 16, and 8.
- LSTM: Long short term memory network (LSTM) is a recurrent model for modeling temporal correlations. Here, we use the recent 12 AQI as input. The number of units is set as 32. LSTM-STC and LSTM+fusion are two variants of LSTM, where the former considers spatial information of air quality stations, and the

latter fuses the output of LSTM with meteorological information. The basic settings of both methods remain consistent with LSTM.

- DeepST [14]: A CNN-based prediction approach for city traffic prediction. Here, we convert the spatial partition from circles to grids with image size (5 * 5). The channel size of CNN is [32,16,8].
- DMVST-Net [15]: Deep multi-view spatial-temporal network uses CNN and LSTM to jointly consider the spatial and temporal relations. The channel size of CNN is [32,16,8], and the length of LSTM is 16.
- DeepSD [10]: A fusion-based deep neural network for predicting car-hailing services. Here, we fuse air quality data with meteorology, weather forecast, and time information iteratively in a sequence. The parameters are set the same as our methods.
- DeepFM [16]: Factorization-machine based neural network integrates the architecture of DNN and FM to respectively model both high-order and low-order feature interactions. The parameters are set the same as our methods except for the fusion order.
- 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.

For data preprocessing, we use the min-max normalization to scale continuous features into [0,1] and use one-hot to encode discrete features. As for deep learning models, we apply Adam [17] to train the parameters with learning rate 0.001 and batch size 512. To prevent overfitting, we employ dropout [18] with rate 0.5 and apply L2 regularization with weight 0.1 on the final loss function. Besides, we use Sigmod function for the output layer and use exponential linear unit [19] for other layers. Here, all comparing methods are fed with all features except for ARIMA and LSTM, which only use air quality as input. The hyper-parameters of all baselines are generated by grid searching. All deep models are implemented on Keras with Tensorflow as backend. Here, we use a GPU server with Tesla K40m GPU.

#### 4.1.3 Model Details

The details of hyper-parameters and embedding for our models are defined as follows, where the settings on preprocessing, optimization, and activation functions are the same as baselines.

- Hyper-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, 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).
- Embedding. Table 2 detail the embedding settings for short-term prediction. For AQIs, other pollutants, historical weather, we use the data in the past and current 6 hours to incorporate the temporal information. For the weather forecast, we use k/3 forecast instances to capture the dynamic changes of future weather conditions. Here, we combine the features from the data to learn the embedding for exploring the intra-dynamics of each factor. For long-term prediction, we embed the AQI data in the past and current 6 hours into 36 and embed the weather forecast data in one day into 12. The embedding setting for long-term prediction is similar to short-term prediction.

TABLE 2
Embedding setting. Encoding is represented by timestamps * feature dimension in one timestamp.

| Data | Feature | Encoding | Embedding |
|---|---|---|---|
| AQIs | $PM_{2.5}$ | 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 | |

#### 4.1.4 Evaluation Metrics

We use prediction accuracy (acc) an d mean absolute error (mae) for evaluation, which are defined as follow:

$$acc = 1 - \frac{\sum_i |\hat{y}_i - y_i|}{\sum_i y_i}, mae = \frac{\sum_i |\hat{y}_i - y_i|}{n} \quad (2)$$

Where $y_i$ and $\hat{y_i}$ 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.

For the validation schema, each experiment is repeated 5 times and averaged, displaying the mean±standard deviation in Table 3 and Table 8.

### 4.2 Performance Comparison for short-term prediction

#### 4.2.1 Comparison with Different Baselines

Table 3 shows the performance of our proposed DA-Short approach comparing with other competing baselines. DA-Short achieves the highest accuracy in both general cases and sudden changes as it can automatically discover complicated air pollution patterns. By considering air quality data recorded by neighbor stations, LSTM-STC outperforms LSTM significantly, which shows the importance of air

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

| Methods | 1-6h | | 7-12h | | 13-24h | | 24-48h | | Sudden Change | | #Params |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 | / |
| GBRT | 0.792 | 21.8 | 0.629 | 38.8 | 0.540 | 48.0 | 0.458 | 56.5 | 0.321 | 77.8 | / |
| LSTM | 0.780 | $23.1 \pm 0.1$ | 0.606 | $41.2 \pm 0.1$ | 0.491 | $53.2 \pm 0.1$ | 0.380 | $64.8 \pm 0.1$ | 0.240 | $90.1 \pm 1.1$ | 4.3k |
| LSTM-STC | 0.794 | $21.6 \pm 0.2$ | 0.622 | $39.6 \pm 0.2$ | 0.508 | $51.4 \pm 0.1$ | 0.396 | $63.0 \pm 0.3$ | 0.314 | $82.5 \pm 1.6$ | 6.5k |
| DeepST | 0.806 | $20.4 \pm 0.1$ | 0.633 | $38.1 \pm 0.2$ | 0.545 | $47.5 \pm 0.2$ | 0.466 | $55.7 \pm 0.7$ | 0.380 | $74.5 \pm 2.9$ | 5.5k |
| DMVST-Net | 0.806 | $20.4 \pm 0.1$ | 0.638 | $37.8 \pm 0.3$ | 0.550 | $47.4 \pm 0.5$ | 0.481 | $53.9 \pm 0.7$ | 0.419 | $70.4 \pm 2.0$ | 6.4k |
| DeepFM | 0.808 | $20.1 \pm 0.1$ | 0.643 | $37.3 \pm 0.2$ | 0.549 | $47.2 \pm 0.6$ | 0.474 | $54.9 \pm 0.6$ | 0.396 | $72.3 \pm 1.9$ | 2.5k |
| DeepSD | 0.811 | $19.7 \pm 0.1$ | 0.645 | $37.1 \pm 0.2$ | 0.551 | $46.8 \pm 0.8$ | 0.479 | $54.3 \pm 0.7$ | 0.428 | $69.5 \pm 3.3$ | 4.7k |
| **DA-Short** | **0.812** | $\mathbf{19.5 \pm 0.2}$ | **0.656** | $\mathbf{36.1 \pm 0.2}$ | **0.569** | $\mathbf{45.1 \pm 0.1}$ | **0.500** | $\mathbf{52.1 \pm 0.3}$ | **0.471** | $\mathbf{63.8 \pm 2.8}$ | 6.8k |

quality' 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 pattern. Comparing with DeepST, the results show that CNN is not suited in air quality prediction tasks as air quality data is sparse, and the image size is small after preprocessing. As a result, DMVST-Net is not suited for air quality prediction task. Comparing with DeepFM, the results show the effectiveness of DA-Short as DeepFM is designed for high-dimensional and extremely sparse data in CTR tasks. Thus, a deep understanding of the problem and data is important, which we can not directly employ an existing model in other domains. By fusing the main feature with each auxiliary feature in a parallel manner, our proposed distributed architecture can model the underlying complex interactions of direct factors and indirect factors, which is more suited for short-term air quality prediction task. Besides, we illustrate the parameter size for 1-6h prediction in the last column. For other predicting settings, the parameters are slightly larger than that of 1-6h predicting model considering the weather forecasts.

#### 4.2.2 Comparison with Official Prediction

Table 4 shows the comparison between DA-Short and WFM during the period: 2014/10/1 to 2016/12/30. As WFM provides the predictions in district-level min-max range for the next 12 hours and DA-Short provides 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 DA-Short to district-level and get the min-max range for the next 12 hours. In both evaluation settings, DA-Short has higher accuracy than WFM with 22% accuracy improvements. Besides, DA-Short has a finer spatial and temporal granularity, a farther prediction period, and a faster updating frequency, which is robust and general enough on different prediction settings. The reason behind it is that WFM is a traditional dispersion model with high numerical computation complexity, which needs accurate input data. However, it is impossible to get all the factors completely and accurately. Thus, prediction accuracy is hard to be guaranteed.

TABLE 4
Compare with official prediction in Beijing

| Methods | Station Level | | District Level | | Update Hour | Grained Level |
|---|---|---|---|---|---|---|
| | acc | mae | acc | mae | | |
| WFM | 0.54 | 54.5 | 0.64 | 46.1 | 12 | District |
| **DA-Short** | **0.77** | **26.7** | **0.86** | **17.9** | **1** | **Station** |

#### 4.2.3 Comparison with Previous Online Model

Figure 12 shows the comparison between DA-Short and the previous online approach, FFA, on nine major Chinese cities. In general, comparing with FFA, DA-Short can achieve an average accuracy of (81.1%, 46%) with an average (1.8%, 17.3%) accuracy improvements on 1-6 hour and sudden changes predictions. The reason behind it is that FFA trains three separate prediction models for modeling spatial and temporal features and uses another model for the model ensemble, 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 used in FFA is not strong enough as it ignores the dynamic change of weather forecasts. Our 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.

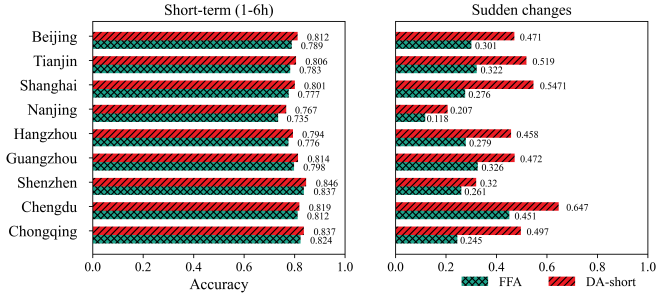



Fig. 12. Comparison with previous online model on nine Chinese cities

#### 4.2.4 Performance on Spatial Transformation

We show the effectiveness of spatial transformation component (STC) in Table 5. Comparing with only using the data from the target station, DA-Short has higher accuracy as it

learns the actual situation that air pollutants are dispersed in geographical space. With the signals from spatial neighbors, DA-Short can capture the dynamic changes of air quality from the spatial perspective. 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 different k-nearest stations, which may confuse the neural network for learning the spatial information. While the STC considers spatial correlations and generates a consistent input from eight directions, which is more suitable for real-world scenarios for simulating second-hand pollutant sources. In STC, we find that inner & outer circles have a better performance than a single inner circle. The reason behind it is that it considers the signals from both near and distant cities, where air pollution may disperse from a distant source by the wind. Especially for the sudden changes, the information from distant is more important than general cases.

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 | 0.370 | 75.2 |
| STC | Inner circle | 0.806 | 20.3 | 0.411 | 70.4 |
| | Inner & outer circle | **0.812** | **19.5** | **0.471** | **63.8** |

#### 4.2.5 Performance on Distributed Fusion

We show the effectiveness of distributed fusion architecture in Table 6. DA-Short outperforms all kinds of fusion combinations, bringing a significant improvement beyond 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 it has a worse result in 13-24h and 24-48h, which demonstrates 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 every individual influence, which also demonstrates that air quality is affected by multiple factors. With such distributed fusion architecture, we can simultaneously model the interactions, where highlight the importance of the main feature and capture the influences from auxiliary features.

TABLE 6
Results on different fusion architecture

| 1-6h | | 1-6h | 7-12h | 13-24h | 25-48h |
|---|---|---|---|---|---|
| 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 |
| **DA-Short** | | **0.812** | **0.656** | **0.569** | **0.500** |

#### 4.2.6 Performance on Embedding

We show the effectiveness of embedding in Table 7. After embedding, we can see a clear improvement in general cases and sudden changes as it can capture the intra-dynamics of each influential factor by learning the hidden representation. Besides, we combine the features from different time slots, where 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** |

### 4.3 Performance Comparison for Long-term Prediction

#### 4.3.1 Comparison with Different Baselines

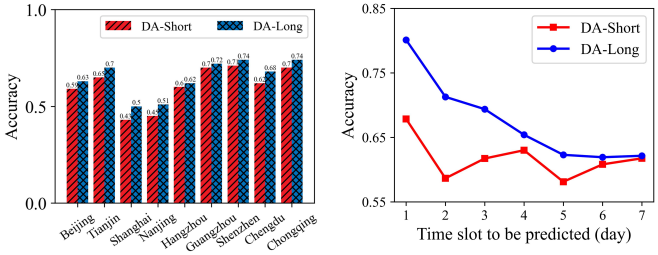
Table 8 shows the performance of different baselines for city-level long-term prediction. Here, we ignore the DeepST and DMVST-Net as the spatial sparse of air quality data. For FNN, it directly concatenates all features together, which ignores the temporal correlations for different days. LASSO performs much better than LR, as most features are discrete. LSTM+Fusion means using LSTM to learn the temporal dynamics of weather forecast data and then fuse with air quality data. However, the performance isn't so good as it ignores the dynamic interaction along time. As most input data is represented by one-hot encoding, DeepFM is good at dealing with sparse input. The results of DeepSD and DeepFM are fairly close. DA-Long performs best with an average accuracy of 63.4%, which indicates our proposed cascaded architecture is more suited for long-term air quality prediction task. By fusing the predecessor features with each slice of successor features iteratively, DA-Long can weaken the effect from the predecessor features and strengthen the importance of successor features over time. Thus, it can simulate the dynamic interaction between air pollutants and meteorological conditions along time, while the DeepSD model ignores it. Besides, we illustrate the size of parameters for day 1 prediction in the last column. Other prediction settings vary with different weather forecast size, whose parameter size is slightly larger than that of day-1.

#### 4.3.2 Comparing DA-Long with DA-Short Model

Figure 13(a) shows the comparison between DA-Long and DA-Short for future 7-day prediction. Overall, DA-Long has a better performance in all nine cities, with an average accuracy of 60.5% with 4.3% accuracy improvement comparing with DA-Short. In Figure 13(b), we show the comparison results on Chengdu, where DA-Long performs better than DA-Short for each day. This is because it is hard to distinguish main and auxiliary features as the effect of different features change a lot along time, where the effect of air quality decreases while that of weather forecast increases illustrated in Figure 2(b). With the ability to model dynamic correlations along time, DA-Long is more suitable for long-term prediction than DA-Short.

TABLE 8
Comparison with different baselines in Beijing. For neural network models, we run each of them 5 times and show mean±standard deviation"

| Methods | day 1 | | day 4 | | day 7 | | 7-day average | | #Params |
|---|---|---|---|---|---|---|---|---|---|
| | acc | mae | acc | mae | acc | mae | acc | mae | |
| LR | 0.685 | 31.3 | 0.431 | 56.1 | 0.413 | 57.8 | 0.464 | 52.9 | / |
| LASSO | 0.716 | 28.2 | 0.605 | 38.9 | 0.592 | 4.1 | 0.601 | 39.3 | / |
| GBRT | 0.742 | 25.6 | 0.582 | 41.2 | 0.581 | 41.2 | 0.606 | 38.9 | / |
| FNN | 0.714 | $28.4 \pm 1.3$ | 0.559 | $43.5 \pm 2.7$ | 0.559 | $43.4 \pm 0.7$ | 0.579 | $41.6 \pm 0.5$ | 19k |
| LSTM+Fusion | 0.739 | $25.9 \pm 0.4$ | 0.606 | $38.8 \pm 0.5$ | 0.600 | $39.4 \pm 0.1$ | 0.611 | $38.4 \pm 0.2$ | 5.3k |
| DeepFM | 0.741 | $25.7 \pm 1.1$ | 0.601 | $39.3 \pm 0.8$ | 0.601 | $39.3 \pm 0.6$ | 0.620 | $37.5 \pm 0.6$ | 1.8k |
| DeepSD | 0.744 | $25.4 \pm 0.2$ | 0.622 | $37.2 \pm 0.4$ | 0.600 | $39.4 \pm 0.4$ | 0.623 | $37.2 \pm 0.4$ | 2.1k |
| **DA-Long** | **0.749** | **24.9±0.5** | **0.629** | **36.6±0.6** | **0.613** | **38.0±0.2** | **0.634** | **36.1±0.3** | 9k |




a) Comparison with DA-Short among different cities

b) Comparison with DA-Short in different days

Fig. 13. Comparison with DA-Short on long-term prediction

#### 4.3.3 *Comparing with Different Feature Variants*

As shown in Table 9, we designed some feature variants for comparison. We can see that air quality data is useful in short-term prediction, but perform worse in long-term prediction. While for meteorology and weather forecast data, vise verse. With these characteristics, we design the cascaded fusion architecture to combine these different kinds of data. Though all features combination could not get the best result on day 7, we make a better trade-off for performing best in 7-day average prediction.

TABLE 9
Results on different feature variants

| Feature Combination | day 1 | day 4 | day 7 | 7-day average | |
|---|---|---|---|---|---|
| | acc | acc | acc | acc | mae |
| $X_{aqi}$ | 0.696 | 0.446 | 0.427 | 0.478 | 51.51 |
| $X_m + X_w$ | 0.579 | 0.623 | **0.618** | 0.608 | 38.69 |
| $X_{aqi} + X_m + X_w$ | **0.749** | **0.629** | 0.613 | **0.634** | **36.09** |

#### 4.3.4 *Performance on Weighted Merge*

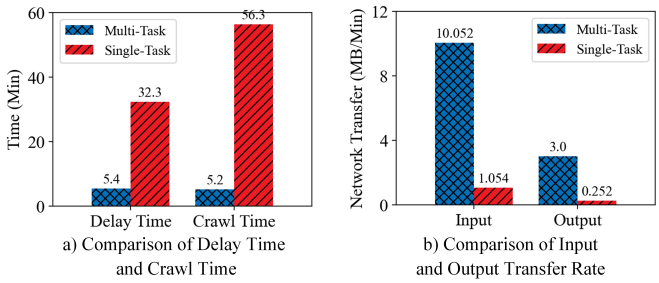
We compare two network variants with DA-Long in table 10. For the first variant, we remove the Weighted Merge layer and only use the last output of the FusionNet, leaving out the intermediate outputs. From the comparison result, we can see that DA-Long outperforms this model variant. The reason behind it is that Weighted Merge can learn the importance of each fusion output and fuse the results automatically, which can incorporate more information. As to the second variant, we replace the Weighted Merge layer with an Attention layer. The comparison result also demonstrates the Weighted Merge is more suitable than Attention for fusing all the output. Attention concatenates all results together and utilizes Softmax function to distribute weights, so the sum of the weight vector is restrained to 1, which may result in a relatively worse performance. As for Weighted Merge, it can learn the weights of each fusion output individually with fewer limitations.

TABLE 10
Comparison with different network variants

| Methods | day 1 | day 4 | day 7 | 7-day average | |
|---|---|---|---|---|---|
| | acc | acc | acc | acc | mae |
| w/o Weighted Merge | **0.749** | 0.614 | 0.608 | 0.629 | 36.60 |
| Attention based | 0.743 | 0.621 | 0.610 | 0.629 | 36.55 |
| DA-Long | **0.749** | **0.629** | **0.613** | **0.634** | **36.09** |

### 4.4 Performance on Data Crawler

To evaluate the performance of our data crawler (Multi-Task), we compare it with the traditional single-task based data crawler (Single-Task) on crawling the data. Figure 14(a) shows the comparison of delay time and crawl time, where the delay time denotes time intervals between data being published and data being stored into the database, and crawl time means how long it takes to execute all crawling tasks. We can find that Multi-Task architecture can crawl all data on average of 5.2 minutes with an average 5.4 minutes delay, which is 6 times faster on delay time and 10 times faster on crawl time than Singel-Task. Figure 14(b) presents the performance of input and output network transfer rate, where both rates of Multi-Task is 10 times of Single-Task, which demonstrates our data crawler can make better use of web resources. With the parallel mechanism, multi-thread and multi-queue based multi-task architecture outperform the single-thread based traditional crawler by an order of magnitude in terms of efficiency.




a) Comparison of Delay Time and Crawl Time

b) Comparison of Input and Output Transfer Rate

Fig. 14. Comparison of Different Data Crawler

### 4.5 Overall Discussion

The frameworks of our proposed approaches are based on a deep understanding of air pollution. We propose a distributed fusion architecture for short-term predictions, where the simple methods directly concatenate all features together for fusion. From the perspective of real-world scenarios, air pollution is affected by direct factors and indirect factors. At most times, all indirect factors will simultaneously affect direct factors. Also, each indirect factor has its effect on direct factors. Derived from such knowledge, we design our distributed fusion architecture, which can well capture these influential factors simultaneously. From the perspective of feature selection, different features have different impacts. By fusing the main feature with each auxiliary feature in a parallel manner, we can highlight the importance of main feature and capture the influences from auxiliary features. Besides, we use the Weighted Merge layer to aggregate the results from all subnets, which can model the dynamic effects of different higher-order influences. Thus, our distributed fusion architecture can automatically learn the importance of each feature and dynamically fuse them to achieve better prediction accuracy.

Besides, we propose a cascaded fusion architecture for long-term air quality prediction. From the perspective of real-world scenarios, the influences from historical air quality and weather forecasts for long-term predictions are opposite along time, where the former decreases while the latter increases. Thus, it is important to combine the advantages of these features and simulate the dynamic interactions along time. From the perspective of feature selection, we denote air quality and meteorology as predecessor features and weather forecast as successor features. By fusing the predecessor features with each slice of successor features iteratively, we can weaken the effect from the predecessor features and strengthen the importance of successor features over time. By using the Weighted Merge layer, the network can learn the importance of each fusion output and fuse the result automatically. That is why cascaded fusion architecture is more suitable for long-term prediction.

Here, we only use DNN without using CNN and LSTM in our proposed approach. Typically, CNN can learn spatial correlations. However, the air quality data is sparse in the spatial dimension. For example, there are more than 2500 grids in Beijing within six rings using 1 kilometer * 1 kilometer for partition, while only 36 air quality monitoring stations exist. The missing rate is bigger than 98%, which brings huge uncertainty even though using interpolation methods for filling missing values. If we directly convert the spatial partition in the spatial transformation component from circles to grids with image size (5 * 5), the experiment result is not good in section 4.2.1. That is why the CNN model is not suitable for handling such sparse data. LSTM is used for modeling temporal dependency. However, air quality is affected by multiple factors without a strong temporal dependence, where only has temporal closeness without daily/weekly/monthly periodic patterns. Considering the efficiency of the online prediction system, we chose DNN except for LSTM for fast running. Besides, we conduct the experiment about LSTM in section 4.2.1, where the performance is not good.

## 5 RELATED WORK

### 5.1 Air Quality Prediction

Air quality prediction methods mainly fall into two categories: numerical prediction models [20] and data-driven models [21]. Numerical prediction models, such as CMAQ, WRF/Chem, and CHIMERE, mainly identify the root cause of air pollution based on atmospheric dynamics and environmental chemistry [22]. Based on the data from emission sources and meteorological data, numerical prediction models construct equations to model the spatial-temporal distribution and transitions [23]. However, it is tough to get all these factors completely and accurately. Thus, prediction accuracy is hard to be guaranteed. Besides, the computation complexity is quite high, usually with a few hours. Data-driven models, such as artificial neural networks and gradient boosting decision tree, forecast air quality based on a variety of features for learning linear or nonlinear correlations [24], [13]. Recently, Zheng et al. proposed a multi-view-based hybrid model [7], which is our previous model in the online system. However, FFA is an ensemble method with a temporal predictor, a spatial predictor, a dynamic aggregator, and an inflection predictor. Our approach deeply learns the air pollution patterns, considering the interactions of these factors, which is more capable of air quality prediction than FFA.

### 5.2 DNN for Spatio-Temporal Prediction

Recently, many works show the strength of DNN on solving spatio-temporal prediction tasks [25], [26]. Song et al. [27] proposed a recurrent neural network to simulate and predict human mobility. To predict citywide crowd flows, Zhang et al. proposed a spatio-temporal residual CNN-based network [12], [14], [28]. Yao et al. proposed a deep multi-view network to predict citywide taxi demand based on CNN and LSTM [15]. Athira et al. proposed an RNN based air quality prediction approach [29]. Qi et al. develop a general approach called DAL to unify the interpolation, prediction, feature selection, and analysis of the fine-grained air quality into one model [30]. Du et al. proposed a hybrid deep learning method to combine one-dimensional CNNs and Bi-directional LSTM for the single-step and multi-step predictions [31]. Different from that, our proposed methods are derived from the domain knowledge of air pollution. Our proposed deep distributed fusion network can simulate the individual and holistic effects of all influential factors for predicting station-level short-term air quality. Our proposed deep cascaded fusion network can capture the dynamic influences from previously existing data and future predicted data for predicting city-level long-term air quality.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we propose a DNN based approach to predict the air quality of the next 48 hours for each monitoring station and the daily average air quality of the next 7 days for a city. For short-term air quality prediction, considering complex interactions between direct and indirect factors, we propose a deep distributed fusion network to simulate the individual and holistic effects of influential factors. While for long-term air quality prediction, inspired by the

effects from different features varying differently along time, we design a deep cascaded fusion network to capture the dynamic influences from previously existing data and future predicted data. Experimental results on three-year data from nine Chinese cities, consistently demonstrate the effectiveness of our proposed approach. We have developed a real-time system, providing hourly station-level and daily city-level air quality predictions for 300+ Chinese cities. Moreover, we introduce a multi-task architecture based data crawler, task scheduler, and prediction model, which can improve the system's efficiency and stability.

In the future, we will explore the combination of numerical prediction models and data-driven models for improving the prediction accuracy, especially for sudden changes. Besides, we want to diagnose the root cause of air pollution from a data-driven perspective, such as study the correlation between vehicular emission and air pollution.

## ACKNOWLEDGMENT

This work was supported by China Postdoctoral Science Foundation, National Natural Science Foundation of China Grant (61773324), and National Key R&D Program of China (2019YFB2101800). Suggestions and comments from anonymous reviewers greatly improve this paper.

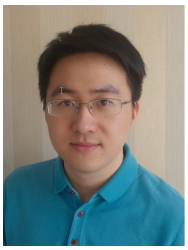
## REFERENCES

[1] H. Akimoto, "Global air quality and pollution," *Science*, vol. 302, no. 5651, pp. 1716–1719, 2003.
[2] M. Kampa and E. Castanas, "Human health effects of air pollution," *Environmental pollution*, vol. 151, no. 2, pp. 362–367, 2008.
[3] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1436–1444.
[4] J. Lu and X. Cao, "Pm2. 5 pollution in major cities in china: Pollution status, emission sources and control measures," *Fresenius Environ. Bull*, vol. 24, pp. 1338–1349, 2015.
[5] E. Baralis, T. Cerquitelli, P. Garza, and M. R. Kavoosifar, "Analyzing air pollution on the urban environment," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics*. IEEE, 2016, pp. 1464–1469.
[6] J. Y. Zhu, Y. Zheng, X. Yi, and V. O. Li, "A gaussian bayesian model to identify spatio-temporal causalities for air pollution based on urban big data," in *2016 IEEE Conference on Computer Communications Workshops*. IEEE, 2016, pp. 3–8.
[7] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. 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*. ACM, 2015, pp. 2267–2276.
[8] G. Y. Lu and D. W. Wong, "An adaptive inverse-distance weighting spatial interpolation technique," *Computers & geosciences*, vol. 34, no. 9, pp. 1044–1055, 2008.
[9] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic geography*, vol. 46, pp. 234–240, 1970.
[10] D. Wang, W. Cao, J. Li, and J. Ye, "Deepsd: supply-demand prediction for online car-hailing services using deep neural networks," in *2017 IEEE 33rd International Conference on Data Engineering*. IEEE, 2017, pp. 243–254.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[12] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[13] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 965–973.
[14] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2016, p. 92.
[15] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[16] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.
[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[19] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.
[20] Y. Zhang, M. Bocquet, C. Seigneur, and A. Baklanov, "Real-time air quality forecasting, part i: History, techniques, and current status," *Atmospheric Environment*, vol. 60, pp. 632–655, 2012.
[21] Y. Zhangg, M. Bocquet, V. Mallet, C. Seigneur, and A. Baklanov, "Real-time air quality forecasting, part ii: State of the science, current research needs, and future prospects," vol. 60. Elsevier, 2012, pp. 656–676.
[22] D. Byun and K. L. Schere, "Review of the governing equations, computational algorithms, and other components of the models-3 community multiscale air quality (cmaq) modeling system," 2006.
[23] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling air quality in street canyons: a review," *Atmospheric environment*, vol. 37, no. 2, pp. 155–182, 2003.
[24] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction." in *IJCAI*, 2018, pp. 3428–3434.
[25] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 305–313.
[26] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *The World Wide Web Conference*. ACM, 2019, pp. 2181–2191.
[27] X. Song, H. Kanasugi, and R. Shibasaki, "Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level." in *IJCAI*, vol. 16, 2016, pp. 2618–2624.
[28] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting city-wide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.
[29] V. Athira, P. Geetha, R. Vinayakumar, and K. Soman, "Deepairnet: Applying recurrent networks for air quality prediction," *Procedia computer science*, vol. 132, pp. 1394–1403, 2018.
[30] Z. Qi, T. Wang, G. Song, W. Hu, X. Li, and Z. Zhang, "Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2285–2297, 2018.
[31] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Deep air quality forecasting using hybrid deep learning framework," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

**Xiuwen Yi** is currently a Data Scientist of JD Intelligent Cities Research and Postdoctoral Researcher at Tsinghua University, focuses on using big data and AI technology to build real-world applications for tackling urban challenges. He got his Ph.D. degree in Computer Science and Technology from Southwest Jiaotong University in 2018. He was an intern in Urban Computing Group at MSR Asia from 2014 to 2017. His research interests include: Spatiotemporal Data Mining, Deep Learning, and Urban Computing. He has published over 15 research papers in refereed conferences (e.g., KDD, IJCAI) and journals (e.g., IEEE TKDE, AI). Also, he is experienced in building real-world applications of AI technology.

**Zhewen Duan** is a Undergraduate student in Xidian University, majoring in computer science and technology. His research interests mainly include spatiotemporal data mining with deep learning, urban computing. He is also an intern student in JD Intelligent Cities Business Unit.

**Ruiyuan Li** is a Ph.D. student at the School of Computer Science and Technology, Xidian University, China. He received his B.E. degree and M.S. degree from Wuhan University, in 2013 and 2016, respectively. His research focuses on Urban Computing, Spatiotemporal Data Management on the Cloud, and Distributed Computing. He is now an intern student in Urban Computing Lab, JD Group, China, under the supervision of Prof. Yu Zheng and Dr.Jie Bao.



**Junbo Zhang** is a Senior Researcher of JD Intelligent Cities Research and the head of AI Platform Division of Intelligent Cities Business Unit, JD Digits. Prior to that, he was a researcher at MSRA from 2015 - 2018. His research interests include urban computing, machine learning, and data mining. He currently serves as Associate Editor of ACM Transactions on Intelligent Systems and Technology. He has published over 30 research papers in refereed journals and conferences, among which one paper was selected as the ESI Hot Paper, three as the ESI Highly Cited Paper. He is a member of IEEE, ACM, CAAI and China Computer Federation.

**Tianrui Li** received the Ph.D. degree from Southwest Jiaotong University in 2002. He was a postdoctoral researcher with SCK·CEN from 2005 to 2006, and a visiting professor with Hasselt University in 2008, the University of Technology in 2009, and the University of Regina in 2014. He is currently a professor and the director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or coauthored more than 300 research papers in refereed journals and conferences. His research interests include big data, cloud computing, data mining, granular computing and rough sets. He is a fellow of IRSS and senior member of ACM and IEEE.

**Yu Zheng** is a Vice President of JD.COM and the Chief Data Scientist of JD Digits. He also leads the Intelligent Cities Business Unit as the president and serves as the managing director of JD Intelligent Cities Research. Before joining JD Digits, he was a senior research manager at Microsoft Research. Zheng currently serves as the Editor-in-Chief of ACM Transactions on Intelligent Systems and Technology. He has served as chair on over 10 prestigious international conferences, e.g. as the program co-chair of CIKM 2017 (Industrial Track). In 2013, he was named one of the Top Innovators under 35 by MIT Technology Review (TR35) and featured by Time Magazine for his research on urban computing. In 2014, he was named one of the Top 40 Business Elites under 40 in China by Fortune Magazine. In 2017, Zheng was named an ACM Distinguished Scientist.