The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

A Neural Attention Model for Urban Air Quality Inference: Learning the Weights of Monitoring Stations

Weiyu Cheng, Yanyan Shen,[*] Yanmin Zhu, Linpeng Huang
Department of Computer Science and Engineering, Shanghai Jiao Tong University
Email: {weiyu_cheng, shenyy, yzhu, lphuang}@sjtu.edu.cn

Abstract

Urban air pollution has attracted much attention these years for its adverse impacts on human health. While monitoring stations have been established to collect pollutant statistics, the number of stations is very limited due to the high cost. Thus, inferring fine-grained urban air quality information is becoming an essential issue for both government and people. In this paper, we propose a generic neural approach, named ADAIN, for urban air quality inference. We leverage both the information from monitoring stations and urban data that are closely related to air quality, including POIs, road networks and meteorology. ADAIN combines feedforward and recurrent neural networks for modeling static and sequential features as well as capturing deep feature interactions effectively. A novel attempt of ADAIN is an attention-based pooling layer that automatically learns the weights of features from different monitoring stations, to boost the performance. We conduct experiments on a real-world air quality dataset and our approach achieves the highest performance compared with various state-of-the-art solutions.

Introduction

Urban air pollution is undoubtedly a severe problem in the world, responsible for a growing number of health effects. The acquisition of spatially fine-grained urban air quality information is of great importance for both government and urban people to understand the problem and take necessary actions in time. Recent effort has been devoted to establishing monitoring stations to collect air quality statistics. However, due to the high monetary cost (about $200,000 per station), the number of available monitoring stations is very limited, e.g., Beijing has only 36 monitoring stations in a total area of $16,410\ km^2$ (Center 2017). As a result, it is becoming crucial to infer a large amount of air quality information in areas without monitoring stations.

Existing approaches to inferring spatially fine-grained air quality information mainly fall into two categories: *physical methods* and *data-driven approaches*. Physical methods estimate air quality in unmonitored locations by simulating the complex physical dispersion process of air pollutants based on observed data and several empirical assumptions (Arystanbekova 2004; Kim, Park, and Kim 2012). However, the necessary data such as the distribution of all kinds of pollution sources, accurate weather conditions (Godish, Davis, and Fu 2014) and specific street configurations (Kim, Park, and Kim 2012), are always difficult to obtain in practice. Furthermore, some empirical assumptions may not reflect real scenarios accurately, which degrades the model performance. For example, the concentration of air pollutants may not follow the Gaussian distribution as assumed in Gaussian Plume models (Arystanbekova 2004).

Data-driven approaches exploit the effects of the available spatio-temporal urban data on air quality inference (Hasenfratz et al. 2014; Chen et al. 2016a; Zheng, Liu, and Hsieh 2013). Intuitively, various factors from external data sources such as POIs, land-use, traffic and meteorology in a particular location, can be partially or fully acknowledged to its air quality. By augmenting the limited statistics from monitoring stations with plentiful spatio-temporal data, data-driven approaches are generally more effective at capturing local information that relates closely to a location's air quality, thus achieving better inference results than those physical methods.

Typically, data-driven approaches have to address two challenging issues. The first challenge is: *how to incorporate auxiliary multi-source data with monitoring data*? The recent work (Zheng, Liu, and Hsieh 2013) propose to train multiple prediction models with different feature sets and then conduct co-training to retrain each model iteratively. However, developing models separately can hardly capture complex interactions between different features, and hence fails to make accurate inference. The second challenge to be addressed is: *how to differentiate the importance degrees of air quality data from different monitoring stations*? Since not all the monitoring data contribute equally to predicting the air quality in a particular location, existing methods adopt a random scheme (Zheng, Liu, and Hsieh 2013) or k-nearest neighbor strategy (Chen et al. 2016a) to select a subset of monitoring stations and only model the effects of the selected monitoring station data for inference. Unfortunately, the random selection scheme may cause the inconsistency problem (Chen et al. 2016a), while the features from k nearest stations are unnecessarily the most effective and the significance of the same stations may rather vary with time (see details in the Experiments Section).

In this paper, we address the aforementioned problems

[*] Corresponding author

by introducing a generic neural attention model, named ADAIN (Attentional Deep Air quality Inference Network), for spatially fine-grained urban air quality inference. We explore the use of deep neural networks (DNNs) for: 1) modeling heterogeneous data (e.g., air quality data, POIs, road networks, meteorological data) in a unified way, and 2) learning complex feature interactions without costly handcrafted feature engineering. In general, ADAIN combines two kinds of neural networks: i.e., feedforward neural networks to model static data and recurrent neural networks to model sequential data, followed by hidden layers to capture feature interactions. A novel attempt of ADAIN is to utilize the attention mechanism (Bahdanau, Cho, and Bengio 2014), to learn the importance degrees of monitoring stations for inferring air quality in a particular location automatically. The importance degree of each monitoring station is incorporated in ADAIN to dynamically re-weight the features from each station during prediction. We conduct experiments on real-world air quality data and the results demonstrate the superiority of our approach in inference performance. Furthermore, the learned weights of monitoring stations shed light on the potential behaviors of air pollutant emissions and variations, which are valuable for practitioners in addressing air quality issues.

Overview

Definition and Problem

Definition 1 (AQI and IAQI) *The air quality index (AQI) is widely used to measure air quality. For a specific air pollutant, its individual air quality index (IAQI) in an area is measured by a monitoring station, reflecting the real-time concentration of the pollutant. AQI is the highest IAQI values among all kinds of air pollutants. We denote by* D_a^t *the set of IAQI values for a certain pollutant in a city during time period* t*.*

Definition 2 (POI) *A point of interest (POI) represents a specific location, with name, category, coordinates and several auxiliary attributes. We denote by* D_p *the set of all POIs in a city.*

Definition 3 (Road Network) *A road network* D_r *consists of a set of linked road segments in a city. Each road segment includes coordinates of the start and end points, and is associated with a road type (e.g., motorway).*

Definition 4 (Meteorological Data) *A meteorology dataset* D_m *includes district-level meteorological records of a city. Let* D_m^t *denote the real-time meteorological information like weather, temperature, pressure, humidity, wind speed and wind direction during time period* t*.*

In this paper, we aim to infer spatially fine-grained urban air quality based on the above heterogeneous data.
Problem Statement. Consider a particular air pollutant. Given its IAQI data $D_a = \{D_a^t\}_{t=1}^T$ from monitoring stations, POI data D_p, road network D_r and meteorological data $D_m = \{D_m^t\}_{t=1}^T$ of a city, we aim to predict IAQI value for any location l without monitoring stations during time period T.

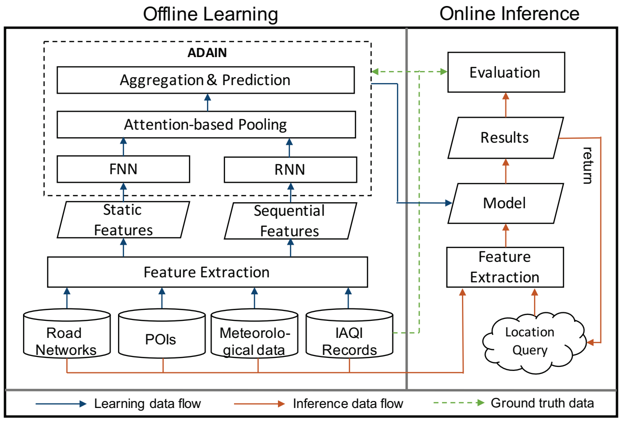


Figure 1: Framework of our approach

Since different pollutants are typically influenced by the observed data differently, we develop an individual model for each pollutant. Note that AQI values can be easily derived from IAQIs by choosing the maximum.

Framework

Figure 1 provides the framework of our proposed solution, which consists of two major components: offline learning and online inference.

- **Offline learning.** We first extract features from heterogeneous data. The features are generally divided into two groups: static ones that are mostly time-invariant (e.g., features from POIs and road networks), and sequential ones that vary with time (e.g., features from meteorology and monitoring data).

To obtain training data, we deliberately remove a monitoring station, and associate the features extracted from data in its affecting region (i.e., within certain distance) and data collected by the remaining monitoring stations with the ground-truth IAQI value as one training example.

Our prediction model ADAIN employs feedforward neural networks (FNN) to handle static features, and uses recurrent neural networks (RNN) to absorb sequential ones. The transformed features are further combined to learn a unified representation that models feature interactions well. ADAIN incorporates the attention mechanism to discriminate the importance of features from different monitoring stations automatically. Note that ADAIN serves two benefits: 1) it is generic to deal with new features that are either static or sequential; 2) it also provides possible explanation on which monitoring data contributes more to the prediction. While the training data is limited by the number of monitoring stations, ADAIN still outperforms the advanced semi-supervised methods experimentally.

- **Online inference.** The online inference process tries to predict IAQI value for a target location in a given time period. To do this, we first extract features from the heterogeneous data observed in the affecting region of the target location. We then combine these features with the ones from real-time monitoring data, and feed them to the trained

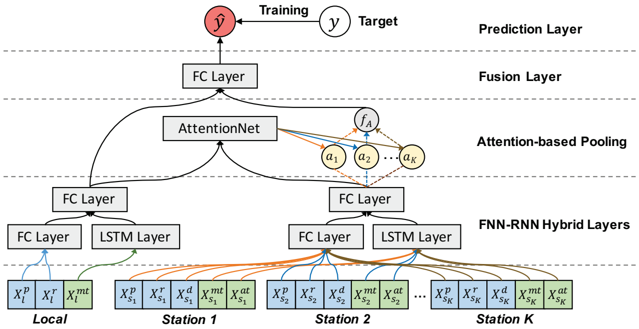


Figure 2: Structure of ADAIN model

model, producing the inferred air quality result.

Methodology

Feature Extraction

We first introduce the features used in this paper, which have been proved to be useful in previous works (Xu and Zhu 2016; Zheng, Liu, and Hsieh 2013). Without loss of generality, we focus on estimating air quality for location l and extract features from the data within l's affecting region, i.e., within certain distance d. By default, d is set to 2 kilometers.

Meteorological features $\mathbf{X}^m$. The concentrations of air pollutants are easily influenced by meteorological factors. In this paper, we consider six meteorological features: *weather, temperature, pressure, humidity, wind speed and wind direction*. Among these features, weather and wind direction are categorical with 12 and 10 categories each, while the others are numerical. We adopt one-hot encoding to represent weather and wind direction features. For numerical ones, we normalize their values to be in the range of $[0, 1]$. As meteorological data varies with time and locations, we extract features for each region periodically (e.g., every 1 hour). We denote by $\mathbf{X}^{mt}$ the set of meteorological features during time period t and omit the region label with the context is clear.

POI features $\mathbf{X}^p$. Intuitively, areas having many factories tend to have poor air quality due to the emission of air pollutants, while those surrounded by public parks are more likely to have fresh air. As POIs well capture the characteristics of locations, we leverage POI data for air quality inference. We consider a set C^p of 12 POI categories specified in (Zheng, Liu, and Hsieh 2013) and compute the *number of each POI category* within a region as one feature. Let $\mathbf{X}^p = \{x^p_c\}_{c \in C^p}$ denote the POI features extracted for a location l. We have:

$$x^p_c = |\{l' \in D_p \mid dist(l, l') \leq d \wedge l'.category = c\}| \quad (1)$$

Road Network features $\mathbf{X}^r$. The structure of road networks also affects local air quality as vehicles are known to be an important source of urban air pollutants (Faiz et al. 1997). We divide all road segments in D_r into three categories: $C^r =$\{highway, trunk, others\}. To capture the intensiveness of road segments in different types, we measure the *total length of road segments per category* within a region as a feature $x^r_c \in \mathbf{X}^r$, $c \in C^r$:

$$x^r_c = \sum_{seg \in S^r_c} seg.length \quad (2)$$

where $S^r_c = \{seg \in D_r \mid seg.category = c \wedge seg$ is overlapped with l's affecting region$\}$.

Monitoring features $\mathbf{X}^d$ and $\mathbf{X}^a$. For each monitoring station s, we extract meteorological, POI and road network features $\mathbf{X}^m_s, \mathbf{X}^p_s, \mathbf{X}^r_s$ from data within s's affecting region. In addition, we provision each station s with relative position features $\mathbf{X}^d_s$ that records the *distance and direction* of s to the target location l, and IAQI features $\mathbf{X}^a_s$ that contain a sequence of *observed IAQI values* in s over time.

Proposed Model

Figure 2 provides the neural network structure of our ADAIN model. The input layer of ADAIN consists of two groups of input features: *local* features $\mathbf{X}^m_l \cup \mathbf{X}^p_l \cup \mathbf{X}^r_l$ for location l, and *station-oriented* features $\mathbf{X}^m_s \cup \mathbf{X}^p_s \cup \mathbf{X}^r_s \cup \mathbf{X}^d_s \cup \mathbf{X}^a_s$ for each station s. Recall that the output of ADAIN is the estimated IAQI value for location l. In what follows, we introduce each layer of ADAIN in detail.

FNN-RNN Hybrid Layers. This layer tries to identify latent features based on raw input features and model feature interactions. Since some input features such as $\mathbf{X}^m, \mathbf{X}^a$ are temporally related, we propose to use recurrent neural networks (RNN) to encode these sequential features. We observe the fact that the values of sequential features often exhibit long periodicity, e.g., temperature. However, traditional RNN can hardly capture long-term dependencies because of gradient vanishing and exploding problems (Hochreiter and Schmidhuber 1997). Hence, ADAIN employs the Long Short-Term Memory (LSTM) (Graves 2013) to encode each of the sequential features (i.e., $\mathbf{X}^m$ and $\mathbf{X}^a$), which leverages the gate mechanism to address the long-term dependency problem.

The regular LSTM contains memory cells $\mathbf{c}$ with self-connections to store temporal states. Each memory cell is associated with input gate $\mathbf{i}$, forget gate $\mathbf{f}$ and output gate $\mathbf{o}$ to control the flow of sequential information. Consider a sequence $\{\mathbf{X}^{mt}\}^T_{t=1}$ of meteorological features for example. The LSTM maps the input sequence to an output sequence by calculating various unit activations using the following equations.

$$\begin{aligned}
\mathbf{i}^t &= \sigma(\mathbf{W}_{ix}\mathbf{X}^{mt} + \mathbf{W}_{ih}\mathbf{h}^{t-1} + \mathbf{W}_{ic} \odot \mathbf{c}^{t-1} + \mathbf{b}_i) \\
\mathbf{f}^t &= \sigma(\mathbf{W}_{fx}\mathbf{X}^{mt} + \mathbf{W}_{fh}\mathbf{h}^{t-1} + \mathbf{W}_{fc} \odot \mathbf{c}^{t-1} + \mathbf{b}_f) \\
\mathbf{c}^t &= \mathbf{f}^t \odot \mathbf{c}^{t-1} + \mathbf{i}^t \odot tanh(\mathbf{W}_{cx}\mathbf{X}^{mt} + \mathbf{W}_{ch}\mathbf{h}^{t-1} + \mathbf{b}_c) \quad (3) \\
\mathbf{o}^t &= \sigma(\mathbf{W}_{ox}\mathbf{X}^{mt} + \mathbf{W}_{oh}\mathbf{h}^{t-1} + \mathbf{W}_{oc} \odot \mathbf{c}^t + \mathbf{b}_o) \\
\mathbf{h}^t &= \mathbf{o}^t \odot tanh(\mathbf{c}^t)
\end{aligned}$$

where $\mathbf{X}^{mt}$ and $\mathbf{h}^t$ are one input element and the corresponding memory cell output activation vector at time t, respectively. The $\mathbf{W}$ terms denote weight matrices (e.g., $\mathbf{W}_{ix}$ is the weight matrix from input gate to the input) and $\mathbf{b}$ terms are bias vectors. $\odot$ denotes the Hadamard product. σ represents the standard sigmoid function. $\mathbf{i}, \mathbf{f}, \mathbf{o}, \mathbf{c}$ are the activation vectors in the same size for input gate, forget gate, output gate and memory cell, respectively. We conduct similar operations over sequential station-oriented features $\mathbf{X}_s^m, \mathbf{X}_s^a$. Specifically, we concatenate $\mathbf{X}_s^{mt}, \mathbf{X}_s^{at}$ across all stations to obtain a bigger sequential feature and feed it to LSTM as one input element for time period t. As local and station-oriented sequential features are typically extracted from different locations, we treat them separately using different LSTMs.

For non-sequential local and station-oriented features (i.e., $\mathbf{X}_l^p \cup \mathbf{X}_l^r$ and $\mathbf{X}_s^p \cup \mathbf{X}_s^r$), we simply apply an individual stack of the fully connected (FC) layers to learn high-order interactions for each feature group. The definition of the FC layers over non-sequential features is as follows.

$$\mathbf{z}_*^{(n)} = \begin{cases} \phi(\mathbf{W}_*^{(n)}(\mathbf{X}_*^p \oplus \mathbf{X}_*^r \oplus \mathbf{X}_*^d) + \mathbf{b}_*^{(n)}), & n = 1 \\ \phi(\mathbf{W}_*^{(n)}\mathbf{z}_*^{n-1} + \mathbf{b}_*^{(n)}), & 1 < n \leq L \end{cases} \tag{4}$$

where $*$ can be l for local features or s for station-oriented features, and L is the number of basic FC layer. ϕ is the activation function and we use the rectifier ReLU in this paper if not otherwise specified, which yields good performance. Note that $\mathbf{X}_*^d = \emptyset$ when modeling local features. In our design, we share the same set of network parameters among all stations to control model complexity and increase model flexibility when the number of stations changes.

To capture interactions among sequential and non-sequential features, we further develop FC layers on top of the basic FC and LSTM layers. Formally, the high-level FC layers transform the last hidden state $\mathbf{h}^T$ from LSTM and the output vector $\mathbf{z}^{(L)}$ of basic FC layers via the following operations.

$$\mathbf{z}_*^{(m)} = \begin{cases} \phi(\mathbf{W}_*^{(m)}(\mathbf{z}_*^{(L)} \oplus \mathbf{h}_*^T) + \mathbf{b}_*^{(m)}), & m = L + 1 \\ \phi(\mathbf{W}_*^{(m)}\mathbf{z}_*^{(m-1)} + \mathbf{b}_*^{(m)}), & m \in [L+2, L+L'] \end{cases} \tag{5}$$

where $*$ can be l or s, and L' denotes the number of high-level FC layers. To summarize, the output of hybrid layers contains latent local features $\mathbf{z}_l^{(L+L')}$ and latent station-oriented features $\mathbf{z}_s^{(L+L')}$ for each station.

Attention Layer. Since not all monitoring data contributes equally to predicting air quality in the target location, we propose to leverage the attention mechanism (Bahdanau, Cho, and Bengio 2014) to our ADAIN model that learns the importance of different station data automatically. The attention mechanism has been incorporated into neural network modeling in various domains such as computer vision (Chen et al. 2016b), information retrieval (Xiong, Callan, and Liu 2017) and recommendation (Xiao et al. 2017). The key idea is to assign weights to different feature parts during prediction. In our context, we compute a weighted sum over latent station-oriented features from different stations as follows.

$$f_A(\{z_s^{(L+L')}\}_{s \in S}) = \sum_{s \in S} a_s \mathbf{z}_s^{(L+L')} \tag{6}$$

where S denotes all monitoring stations and a_s is the *attention score* for latent features $\mathbf{z}_s^{(L+L')}$ of station s learned from the above hybrid layers. Intuitively, a_s discriminates the importance of different station features to benefit model prediction. Existing station selection schemes (e.g., random or k nearest) set values of a_s to 0 or 1 based on certain rules. The resultant weight can hardly distinguish the significance among the selected station features where a_s equals to 1. To address the problem, we parameterize the attention scores based on a multi-layer perceptron (MLP), called AttentionNet, as shown in Figure 2. The input to AttentionNet is the concatenation of both $\mathbf{z}_l^{(L+L')}$ and $\{\mathbf{z_s}^{(L+L')}\}_{s \in S}$. It then encodes the interactions between local features and the ones for station s to decide the attention score a_s:

$$\begin{aligned} a'_s &= \mathbf{w}_s^\intercal \phi(\mathbf{W}_a(\mathbf{z}_l^{(L+L')} \oplus \mathbf{z}_s^{(L+L')}) + \mathbf{b}_a) + b_s \\ a_s &= \frac{\exp(a'_s)}{\sum_{s \in S} \exp(a'_s)} \end{aligned} \tag{7}$$

where the matrix $\mathbf{W}_a$ and the bias vector $\mathbf{b}_a$ are model parameters in the first layer of MLP; vector $\mathbf{w}_s$ and bias b_s are second-layer parameters. The length of $\mathbf{w}_s$ equals the size of hidden layer in MLP. The attention scores are normalized via softmax such that they can be interpreted as the importance of different feature groups for prediction.

Fusion Layer. Hybrid layers and attention layer model the latent local features $\mathbf{z}_l^{(L+L')}$ for location l and high-level station features $f_A(\{z_s^{(L+L')}\}_{s \in S})$, respectively. It is intuitive to combine these features via concatenation and use a hidden layer to learn high-order interactions. Hence, we develop a fusion layer above the hybrid and attention layers, which is defined as follows.

$$\mathbf{z}_f = \phi(\mathbf{W}_f(\mathbf{z}_l^{(L+L')} \oplus f_A(\{z_s^{(L+L')}\}_{s \in S})) + \mathbf{b}_f) \tag{8}$$

where $\mathbf{z}_f$ is the output vector of the fusion layer, matrix $\mathbf{W}_f$ and bias vector $\mathbf{b}_f$ are model parameters. While multiple fusion layers can be stacked together, we observe good performance based on a single fusion layer and omit further layers to reduce model parameters.

Prediction Layer. At last, the output vector of the fusion layer $\mathbf{z}_f$ is transformed to the final prediction score, i.e., the estimated IAQI value for target location l during time period T:

$$\hat{y} = \mathbf{w}_p^\intercal \mathbf{z}_f + b_p \tag{9}$$

where $\mathbf{w}_p$ is the neural weight vector for the prediction layer, and b_p is a bias scalar.

Summary. It is worth mentioning that the structure of our ADAIN model is generic to incorporate more features that are either static or sequential. When the number of monitoring station increases, we may keep a subset of them and still leverage the attention layer to determine the importance of the feature group for each selected station.

Learning and Optimization

The air quality inference problem can be considered as a regression task or a classification task (e.g., IAQI values are organized into categories). As numerical IAQI values are more accurate and valuable, we treat the inference problem as a regression task and adopt the following squared loss as the objective function:

$$L_{loss} = \sum_{\mathbf{X} \in \mathcal{T}} (\hat{y}(\mathbf{X}) - y(\mathbf{X}))^2 \tag{10}$$

where $\mathcal{T}$ denotes the set of all training instances with ground-truth IAQI values $y(\mathbf{X})$.

Instead of using vanilla stochastic gradient descent (SGD) to optimize the objective function, we adopt Adam (Kingma and Ba 2014) as the optimizer. Based on adaptive estimates of lower-order moments, the Adam optimizer dynamically tunes the learning rate during training process and leads to faster convergence. It is also known to be computationally efficient with little memory requirement.

To prevent our model from overfitting, we consider two widely used regularization techniques: dropout and L_2 regularization. The main idea of dropout is to randomly drop some neurons along with their connections during training (Srivastava et al. 2014), which prevents units from too much co-adapting. In ADAIN , we employ dropout on each hidden layer. Besides, we apply L_2 regularization on model weights to prevent possible overfitting. Formally, the actual objective function we optimize is:

$$L_{loss} = \sum_{\mathbf{X}\in\mathcal{T}} (\hat{y}(\mathbf{X}) - y(\mathbf{X}))^2 + \lambda \left\|\mathbf{W}\right\|^2 \tag{11}$$

where λ is a hyperparameter to control the regularization strength and $\mathbf{W}$ denotes all weights in ADAIN .

Experiments

Experimental Settings

Datasets. To evaluate the performance of our proposed approach, we use the following available heterogeneous data collected in Beijing, China.

(1) *Air quality data:* The air quality data (Zheng et al. 2015) was collected by 36 monitoring stations in Beijing, from 2014/05/01 to 2015/04/30, with the collection time interval of 1 hour. Each record contains IAQI values for different air pollutants observed by a station in an hour. We focus on predicting three important pollutants PM2.5, PM10, NO_2. All IAQI values follow the Chinese AQI standard.

(2) *Meteorological data:* The meteorological data (Zheng et al. 2015) consists of real-time district-level meteorological records. Each record contains weather, temperature, pressure, humidity, wind speed and direction in an area.

(3) *POI data:* We query Map World APIs (world 2017) and obtain about 151,000 POIs in Beijing.

(4) *Road network data:* We download the road network for Beijing from OpenStreetMap (Openstreetmap 2017). The number of road segments is 65,991, with a total length of 27,889km.

Settings and Compared Methods. We divide all the monitoring station data into the training and test sets with the proportion of 2:1. The separation is based on stations and is repeated randomly for 10 times, in order to avoid using historical air quality data to infer current air quality information for the same location. This is reasonable as we prefer to predict air quality of locations without monitoring statistics. We also select 10% of training data as the validation set and allow training to be early stopped according to the validation score. In our experiment, we construct a single basic FC layer (L=1) with 100 neurons and two LSTM layers with 300 memory cells per layer. We then build two layers of high-level FC network (L'=2) with 200 neurons per layer. We initialize all the model parameters by sampling from the uniform distribution between -0.1 and 0.1. We compare ADAIN with the following approaches.

(1) *k nearest neighbors (KNN)*: This method selects the k monitoring stations closest to the inferred location, and compute the average IAQI value from these stations as result. We set k to be 3 in our experiments.

(2) *Linear Interpolation (LI)*: This method calculates the weighted average IAQI value based on data from all stations. The weight of a station s is inversely proportional to its distance d_s to the inferred location:

$$\hat{y} = \sum_{s\in S} \frac{s.IAQI \times \frac{1}{d_s}}{\sum_i \frac{1}{d_s}}$$

(3) *Gaussian Interpolation (GI)*: This is another interpolation method based on a Gaussian distribution $N(0, \sigma)$:

$$\hat{y} = \frac{1}{Z}\sum_{s\in S} s.IAQI \times f(s), \;\; f(s) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d_s^2}{\sigma^2}}$$

where σ is the average distance between two monitoring stations, and Z is the normalizing factor.

(4) *Gaussian Process Regression (GPR)*: GPR is a nonparametric Bayesian regression model. We follow the formulation of GPR in (Cheng et al. 2014) and use the following kernel function:

$$K(x_i, x_j, \lambda) = e^{-\lambda\left\|x_i - x_j\right\|^2}$$

where λ is a hyperparameter and set to 0.01 by default.

(5) *Support Vector Regression (SVR)*: This is a classical supervised regression model extended from support vector machine. For station-oriented features, we only consider the k (set to be 3) nearest stations as in (Chen et al. 2016a).

(6) *Feedforward Neural Networks (FNN)*: This method simply flattens all the features and feeds them into a multilayer feedforward neural network. For sequential features, we only use their latest values. The network contains three hidden layers, with 200 cells at each layer. We adopted dropout and L2 regularization to reduce overfitting.

(7) *Support Vector Machine (SVM)*: This is a classification model that absorbs the same inputs as the support vector regression model, while outputs categorical IAQI levels for the target location. We consider 6 IAQI values introduced in (Zheng, Liu, and Hsieh 2013).

(8) *U-Air* (Zheng, Liu, and Hsieh 2013): U-Air is a co-training based classification model. It trains two classifiers using data from different views, and improves the performance of the two classifiers iteratively. This model produces an inferred IAQI level by combining the outputs from both classifiers.

Metrics. We use the root mean squared error (RMSE) to measure the performance of various regression approaches that infer IAQI values:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}(\mathbf{X}_i) - y(\mathbf{X}_i))^2}{N}} \tag{12}$$

where N is the number of instances in the test set.

To compare with classification methods (i.e., SVM and U-air) that produce discrete IAQI levels, we convert the output

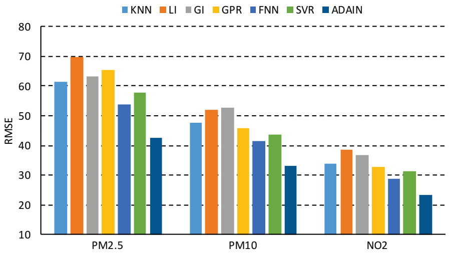


Figure 3: ADAIN v.s. competing regression methods

of our method into the corresponding IAQI levels and adopt accuracy as the measurement, which is defined as follows:

$$Accuracy = \frac{|\{\mathbf{X} \in TestSet \mid \hat{y}(\mathbf{X}) = y(\mathbf{X})\}|}{N} \quad (13)$$

where the numerator denotes the number of correct estimations for the test cases.

Results

We first compare our method with aforementioned baselines. We then evaluate the effectiveness of different features. Finally, we discuss the benefits of our attention model and provide qualitative visualization results to explain it.

Comparison Results. Figure 3 shows the performance of ADAIN and six regression methods, using RMSE metric. ADAIN produces the lowest RMSE values for predicting all three air pollutants. FNN provides the second best performance. The reason may be the ability of hidden layers in FNN that well model the feature interactions. However, on average, the relative improvement of ADAIN over FNN is still 20%. LI and GI perform worse than other methods on all three pollutants. This indicates that the affecting degree of air quality in areas with monitoring stations can hardly be quantified using a function of distance. Furthermore, we observe that the results on PM2.5 and PM10 are worse than those on NO_2 for all methods. This is reasonable because the concentration of NO_2 is more stable in different locations over time, compared wit PM2.5 and PM10.

Next, we compare our approach with competing classification methods, SVM and U-air. To do this, we convert the outputs of our model into IAQI levels accordingly. Figure 4 provides the comparison results. It can be seen that ADAIN achieves the highest accuracy than the other two models on all three pollutants. On average, the relative improvements against U-Air and SVM are 10% and 28%, respectively. The advantages of ADAIN over U-Air could be explained in two aspects. First, U-Air trains two separate classification models for spatial features and temporal features, respectively. Such separation may fail to capture feature interactions, thus degrading the prediction performance. Second, U-Air adopts random scheme when extracting features from monitoring data. It is very likely that informative features are eliminated due to the randomness. In contrast, ADAIN leverages the attention model to discriminate the importance of monitoring features from different stations effectively. The benefits of the attention will be described in the late part of this section.

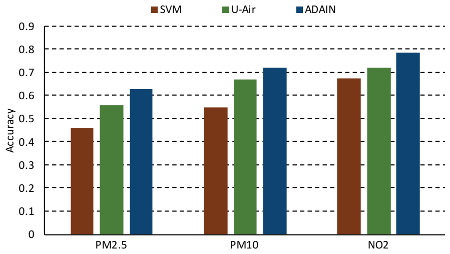


Figure 4: ADAIN v.s. competing classification methods

Table 1: Effects of various features in ADAIN

Features	PM2.5	PM10	NO_2
$\mathbf{X}^a + \mathbf{X}^d$	57.65	45.55	32.06
$\mathbf{X}^a + \mathbf{X}^d + \mathbf{X}^p + \mathbf{X}^r$	53.54	41.74	29.43
$\mathbf{X}^a + \mathbf{X}^d + \mathbf{X}^m$	49.45	37.37	27.28
$\mathbf{X}^a + \mathbf{X}^d + \mathbf{X}^m + \mathbf{X}^p + \mathbf{X}^r$	**42.60**	**33.01**	**23.19**

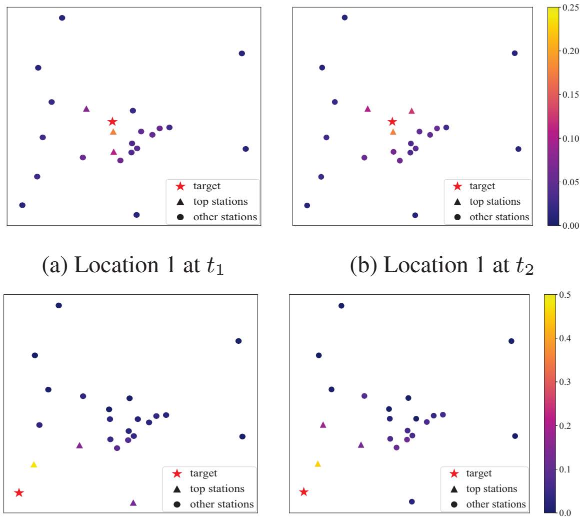
Effects of Different Features. To evaluate the effectiveness of different features in ADAIN , we manually remove some features and compute the prediction error based on the remaining features. Table 1 provides RMSE values of our model using different features. It is easy to see that incorporating more features improves prediction performance significantly and consistently over all three pollutants. The last row with all features achieves the lowest RMSE values. The first row provides the worst prediction results based on station-oriented features only. The second row incorporates non-sequential features from POI and road network data, while the third row leverages sequential features from meteorological data. Sequential meteorological features are more beneficial to air quality inference, which follows our intuition that meteorological factors are highly correlated with the concentration of air pollutants.

Effects of Attention Model. We now study the advantages of our attention model in selecting useful information from monitoring stations for prediction. To do this, we consider two variants of our model. Instead of using attention-based pooling, the two variants employ average pooling on the features of all stations and k nearest stations (we set k to be 3 by default), respectively. Table 2 shows the RMSE values using different pooling methods. Our attention-based pooling approach achieves the lowest RMSE values in predicting all three pollutants. The average pooling over all stations provides the worst performance, which ignores the relative importances of features from different stations. In particular, the averaged feature values may easily cancel out important information and introduce noise instead. Average-k-nearest pooling outperforms average-all pooling by producing lower RMSE values. This is because the features from areas in closer distances are intuitively more informative for inferring local air quality values. However, average-k-nearest pooling discriminates the importances of different stations based on distance factor only and hence results in inferior performance than our attention-based pooling method.

Attention Visualization. To better understand the effects of our attention model, we visualize the attention scores

Table 2: Effects of different pooling methods

Pooling methods	PM2.5	PM10	NO_2
Average-all pooling	65.83	53.18	39.89
Average-k-nearest pooling	48.37	38.72	26.89
Attention-based pooling	**42.60**	**33.01**	**23.19**



(a) Location 1 at t_1

(b) Location 1 at t_2

(c) Location 2 at t_1

(d) Location 2 at t_2

Figure 5: Attention visualization

of monitoring stations in Figure 5, where the shapes represent monitoring stations and the colors reflect their attention scores. We choose two target locations 1 and 2 and try to infer their air quality during two time periods t_1 and t_2. From the results, we can have the following observations. First, our attention model is able to dynamically identify important station data for prediction. The learned importance of features from a monitoring station can vary by location (location 1 vs 2) and time (t_1 vs t_2), mainly because of the chaotic air pollutants and unpredictable external incidents. Second, distance is a critical factor that determines the importance of monitoring stations, but not the only factor. We use diamonds to highlight the top-3 stations with highest attention scores. It can be seen from the colors that stations far away from the target location may have higher attention scores than those near the target location. For example, consider the top-3 stations for location 2 in Figure 5c. Third, the distribution of attention scores changes with the target location. For location 1 surrounded by many monitoring stations, major attention weights are more uniformly assigned to its nearby stations; for location 2 in remote area, high attention scores are concentrated on 1-2 nearby stations. This further verifies that the informative features from monitoring stations are typically dependent on the particular target location.

Related Work

There are two different ways predicting spatially fine-grained urban air quality. One way is based on classical emission models, including Gaussian Plume models (Arystanbekova 2004; Godish, Davis, and Fu 2014), Street Canyon models (Kim, Park, and Kim 2012) and Computational Fluid Dynamics (Scaar et al. 2012). These models simulate the dispersion of air pollutants based on a number of empirical assumptions and parameters. As some empirical assumptions may not correspond to the real situations and the required parameters such as emission density, street geometry and dispersion parameters are hard to get precisely, the prediction results are far from satisfactory (Zheng, Liu, and Hsieh 2013). The other is based on statistical models, such as linear regression, matrix factorization and neural networks for air quality inference (Shad et al. 2009; Hasenfratz et al. 2014; Xu and Zhu 2016). However, many of these models rely on local features from the target location for prediction, without taking care of the spatial-temporal correlations of air pollutants between adjacent areas. There are several air quality inference models that take such dependencies in account. For example, Zheng *et al.* (Zheng, Liu, and Hsieh 2013) and Chen *et al.* (Chen et al. 2016a) proposed semi-supervised based methods to estimate fine-grained air quality. They use random scheme or k-nearest neighbors to select nearby areas with monitoring stations to model spatial dependencies of air pollutants. However, the random scheme results in the inconsistency problem (Chen et al. 2016a) while k-nearest method uses the distance of station-oriented features to discriminate the importances of different stations. Different from these works, we employ the attention mechanism (Bahdanau, Cho, and Bengio 2014) to assign weights to each group of station-oriented features automatically, without human intervention. Moreover, our proposed framework is generic and flexible to incorporate more features to improve performance further.

Recently, many researches have developed deep learning based approaches to challenging tasks in urban computing (Zheng et al. 2014). For example, Zhang *et al.* proposed DNN-based prediction model to predict citywide crowd flows (Zhang, Zheng, and Qi 2016). Liang *et al.* utilized recurrent neural networks to predict metro density (Liang et al. 2016). Xing *et al.* and Grover *et al.* employed deep models for weather forecasting (Xingjian et al. 2015; Grover, Kapoor, and Horvitz 2015). Song *et al.* and Chen *et al.* applied deep neural networks to urban transportation systems (Song, Kanasugi, and Shibasaki 2016; Chen et al. 2016c). However, none of them concerns the problem of inferring spatially fine-grained urban air quality, which is the focus of this paper.

Conclusion

In this paper, we propose a generic neural attention model based on deep neural networks for urban air quality inference. We leverage both records from monitoring stations and various urban data (e.g., meteorology, road networks, POIs), and extract important features that are correlated with air quality. We model static and sequential features using different neural structures and incorporate the attention mechanism to discriminate the importance of features from different stations automatically, to boost the performance. The experimental results on a real dataset verify the superiority of our approach against compared methods.

Acknowledgment

We thank anonymous reviewers for their insightful and helpful comments, which improve the paper. This research is supported in part by 973 Program (no. 2014CB340303), NSFC (no. 61772341, 61472254, 61170238, 61602297 and 61472241), Singapore NRF (CREATE E2S2), and 863 Program (no. 2015AA015303). This work is also supported by the Program for Changjiang Young Scholars in University of China, and the Program for Shanghai Top Young Talents.

References

Arystanbekova, N. K. 2004. Application of gaussian plume models for air pollution simulation at instantaneous emissions. *Mathematics and Computers in Simulation* 67(4):451–458.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Center, B. M. E. M. 2017. http://zx.bjmemc.com.cn.

Chen, L.; Cai, Y.; Ding, Y.; Lv, M.; Yuan, C.; and Chen, G. 2016a. Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning. In *UbiComp*, 1076–1087. ACM.

Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; and Chua, T.-S. 2016b. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. *arXiv preprint arXiv:1611.05594*.

Chen, Q.; Song, X.; Yamada, H.; and Shibasaki, R. 2016c. Learning deep representation from big and heterogeneous data for traffic accident inference. In *AAAI*, 338–344.

Cheng, Y.; Li, X.; Li, Z.; Jiang, S.; and Jiang, X. 2014. Fine-grained air quality monitoring based on gaussian process regression. In *ICONIP*, 126–134. Springer.

Faiz, A.; Weaver, C. S.; Walsh, M.; Gautam, S.; and Chan, L. 1997. Air pollution from motor vehicles: Standards and technologies for controlling emissions. Technical report, World Bank Group, Washington, DC (United States).

Godish, T.; Davis, W. T.; and Fu, J. S. 2014. *Air quality*. CRC Press.

Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Grover, A.; Kapoor, A.; and Horvitz, E. 2015. A deep hybrid model for weather forecasting. In *KDD*, 379–386. ACM.

Hasenfratz, D.; Saukh, O.; Walser, C.; Hueglin, C.; Fierz, M.; and Thiele, L. 2014. Pushing the spatio-temporal resolution limit of urban air pollution maps. In *PerCom*, 69–77. IEEE.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Kim, M. J.; Park, R. J.; and Kim, J.-J. 2012. Urban air quality modeling with full o 3–nox–voc chemistry: Implications for o 3 and pm air quality in a street canyon. *Atmospheric Environment* 47:330–340.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liang, V. C.; Ma, R. T.; Ng, W. S.; Wang, L.; Winslett, M.; Wu, H.; Ying, S.; and Zhang, Z. 2016. Mercury: Metro density prediction with recurrent neural network on streaming cdr data. In *ICDE*, 1374–1377. IEEE.

Openstreetmap. 2017. http://www.openstreetmap.org/.

Scaar, H.; Teodorov, T.; Ziegler, T.; and Mellmann, J. 2012. Computational fluid dynamics analysis of air flow uniformity in a fixed-bed dryer for medicinal plants. In *International Symposium on CFD Applications in Agriculture*, 119–126.

Shad, R.; Mesgari, M. S.; Shad, A.; et al. 2009. Predicting air pollution using fuzzy genetic linear membership kriging in gis. *Computers, Environment and Urban Systems* 33(6):472–481.

Song, X.; Kanasugi, H.; and Shibasaki, R. 2016. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. IJCAI.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15(1):1929–1958.

world, M. 2017. http://map.tianditu.com/map/index.html.

Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; and Chua, T.-S. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 802–810.

Xiong, C.; Callan, J.; and Liu, T.-Y. 2017. Learning to attend and to rank with word-entity duets. SIGIR.

Xu, Y., and Zhu, Y. 2016. When remote sensing data meet ubiquitous urban data: Fine-grained air quality inference. In *Big Data*, 1252–1261. IEEE.

Zhang, J.; Zheng, Y.; and Qi, D. 2016. Deep spatio-temporal residual networks for citywide crowd flows prediction. *arXiv preprint arXiv:1610.00081*.

Zheng, Y.; Capra, L.; Wolfson, O.; and Yang, H. 2014. Urban computing: concepts, methodologies, and applications. *Transactions on Intelligent Systems and Technology* 5(3):38.

Zheng, Y.; Yi, X.; Li, M.; Li, R.; Shan, Z.; Chang, E.; and Li, T. 2015. Forecasting fine-grained air quality based on big data. In *KDD*, 2267–2276. ACM.

Zheng, Y.; Liu, F.; and Hsieh, H.-P. 2013. U-air: When urban air quality inference meets big data. In *KDD*, 1436–1444. ACM.