MSSTN: Multi-Scale Spatial Temporal Network for Air Pollution Prediction

Zhiyuan Wu
Department of Electronic Engineering
Tsinghua University
Beijing, China
wu-zy18@mails.tsinghua.edu.cn

Yue Wang
Department of Electronic Engineering
Tsinghua University
Beijing, China
wangyue@tsinghua.edu.cn

Lin Zhang
Tsinghua-Berkeley Shenzhen Institute
Tsinghua University
Shenzhen, China
linzhang@tsinghua.edu.cn

***Abstract*—Air pollution has become an important factor constraining city development and threatening public health in recent years. Air pollution prediction has been considered as the key part for the early warning of pollution event. Considering the multi-scale nature of geo-sensory data such as air pollution signal, in this paper we adopt a multi-level graph data structure for better utilization of multi-scale spatio-temporal information. We further present a novel deep convolutional neural network model, named Multi-Scale Spatial Temporal Network (MSSTN), for the learning task on this data structure. The MSSTN is specially designed to better discover multi-scale spatial temporal patterns and their high-level interactions, by explicitly using multi-scale neural network structure in both spatial and temporal component. We conduct extensive experiments and ablation studies on Urban Air Pollution Datasets in North China, where the MSSTN can make hourly PM2.5 concentration predictions jointly for a number of cities. And our results shows an outstanding prediction accuracy as well as high computational efficiency compared to existing works.**

***Index Terms*—Air Pollution Prediction, Multi-scale Model, Spatial-Temporal Neural Network**

I. Introduction

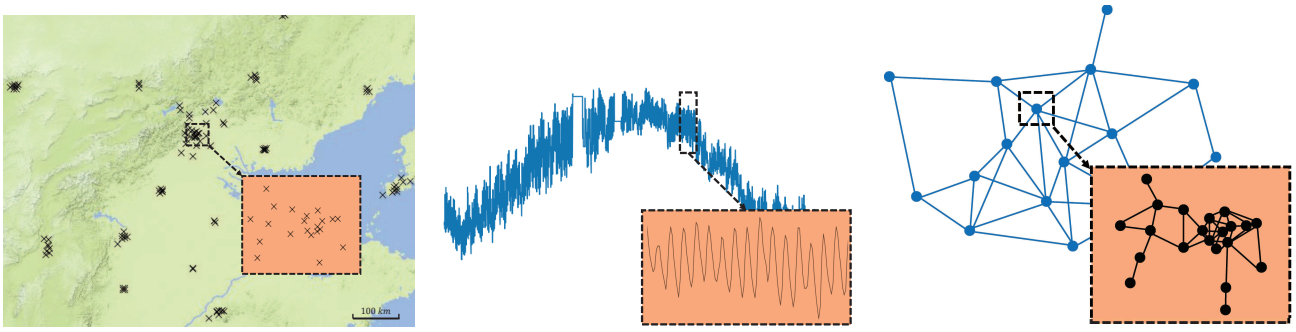
In recent years, the public and the governments have become increasingly concerned about air quality issues, because air pollution can have negative impact on public health as well as city development. Under this circumstance, there is a great need to make accurate air pollution predictions, with which the government can do further pollution analysis, release early warnings, and adopt emergency actions. And citizens can also plan their outdoor activities in advance. In particular, predictions at different spatio-temporal scales can be helpful in various scenario, i.e. take appropriate measures to deal with pollution events of different scales. And therefore the damage caused by air pollution can be significantly reduced [18].

The *multi-scale* nature is an ubiquitous feature of geo-sensory systems like air pollution monitoring systems, from both spatial and temporal aspects. For instance, there exist short-term fluctuation, mid-term periodicity and long-term trend in temporal aspect, city-scale pattern and region-scale transportation in spatial aspect. There also exist more complicated spatio-temporal interactions, which can be especially complicated. It is well acknowledged that this multi-scale phenomena is the complex result of factors at various scales, temporally from seasonal changes to daily periodicity, and spatially from atmospheric circulation to street-level diffusion. An example from real world air pollution monitoring datasets is illustrated in Figure 1 (a)(b), where the multi-scale nature is clearly shown in both spatial and temporal viewpoint. And therefore the appropriate explicit multi-scale modeling is of great necessity and is the key to better prediction accuracy.

There have been extensive related researches explored by scholars of various disciplines. On the one hand, after the great achievement in atmospheric physics and its successful application in weather forecast, a number of numerical physical models have been proposed [4, 13] to predict air pollution. These models usually refer to an interpretable physical simulations. However, physical models are suffered from prediction accuracy, computational cost, and transferability, as air pollution is heavily influenced by human activities and local conditions which are hard to be modeled physically. On the other hand, with the rapid development of big data and machine learning, data-driven models are attracting more and more attention [12, 23]. Though numerical physical models have great interpretability, data-driven models are becoming popular because of high computational efficiency as well as high accuracy. Recently a lot of deep learning models for air pollution prediction have been proposed, focusing on from time series modeling to utilization of spatial information. It is well acknowledged that explicit modeling that fits data characteristic can significantly improve the performance of deep learning models, for instance, CNNs explicitly model the local correlations in images and so dominate many tasks in computer vision. However, most of existing deep neural network solutions neglect the multi-scale nature of air pollution signals, and designed deep models lack the capability of capturing multi-scale patterns in the data, which leads to degradation to the performance.

In order to address challenges above, in this paper we adopt a multi level graph data structure for air pollution monitoring systems. And we further propose a novel deep convolutional neural network, called Multi-Scale Spatial Temporal Network (MSSTN), for the air pollution prediction task in one end-to-end framework. The architecture of MSSTN is specially

This work was supported by the National Key Research and Development Program of China (No.2017YFC0212100).



(a) (b) (c)

Fig. 1. (a)The distribution of sensors in a geo-sensory systems. Each cross stands for a air pollution monitoring stations in the Urban Air Pollution Datasets in North China. The distance between sensors in a city is at scale of 1km while the distance between cities can be at scale of 100km. This is a example of spatial multi-scale property because sensors are usually clustered in cities while cites are sparsely distributed. (b)The illustration of sensor reading as a time series. Here we use temperature reading to show the temporal multi-scale property where signal shows long term trend at scale of 1 year and periodicity at scale of 1 day as well as fluctuation at scale of 1 hour. (c)The illustration of data structure and notations. The proposed data structure can be abstracted from the distribution of sensors. Air pollution sensors in a region can be clustered as a multi-level graph. In a *city-scale* graph (G^a), each node stands for an air pollution sensor. And in a *region-scale* graph ($G^\diamond$), each node stands for a city.

designed to better deal with the multi-scale nature of air pollution data, also considering the dynamic, non-linear and spatio-temporal characteristic of the data. The proposed model can learn short-term/long-term temporal dependencies, city-scale/region-scale spatial patterns, and their interactions in a joint manner, providing better air pollution prediction accuracy.

The MSSTN is a full convolutional neural network consisting of three subnets, named T-Net that extracting multi-scale temporal features, S-Net that extracting multi-scale spatial features, and F-Net that make fusion of information and give final predictions. We use a collection of dilated convolutional networks (DCNs) as T-Net, whose receptive field grows exponentially as the network go deeper, so that features at different temporal scale can be extracted at different layers. S-Net is constructed by a set of graph convolutional neural networks (GCNs) at different spatial scale, which can adapt to sparse and irregular spatial data structure and extract features efficiently. F-Net collects all extracted features and fuse them with a dense connection. Comprehensive experiments are conducted on a real-world air pollution datasets named Urban Air Pollution Datasets in North China, which involve some of largest cities with urgent air pollution issues in China. The result shows that the MSSTN is effective in air pollution prediction task and outperforms several state-of-the-art models. And the ablation result also shows effectiveness of the proposed network structure. Besides, the high computational efficiency of the MSSTN, because of its full convolutional structure, make it more convenient in real time applications.

Our contributions are as follows.

- We suggest to use a multi level graph data structure to better represent the geo-sensory systems and better discover high level spatial temporal patterns. Further we propose a novel deep convolutional neural network named MSSTN for the air pollution prediction task on the proposed data structure.
- Three subnets are specially designed for MSSTN in order to process multi-scale spatial temporal data explicitly, that are a set of dilated casual convolutional network named T-Net, a set of graph convolutional neural networks named S-Net, and a fusion network with dense connections named F-Net.
- We deploy our model on a real world air pollution datasets, Urban Air Pollution Datasets in North China, and result shows an outstanding performance compared to many state-of-the-art methods.

The rest of this paper are organized as follows. In Section II we briefly discussed published works related to this topic. In Section III we introduce the proposed neural network architecture in detail. In Section IV we present the detailed experiments settings and conduct a comprehensive ablation study. And finally in Section V we conclude this paper with a brief summary.

II. Related Works

In this section we provide a detailed overview of published works related to this paper. The application of data mining and deep learning in the domain of environmental big data have attracted a lots of interest in recent years. The related topics can involve geo-sensory system analysis tasks like concentration prediction, field reconstruction, signal decomposition, anomaly detection and hybrid models etc. Specifically for prediction tasks, efforts have been devoted into two categories, that are temporal modeling and utilization of spatial information.

A. Temporal Modeling

Time series modeling have a long history under the topic of signal processing and statistical signal analysis. Many well-known models have been successfully applied to predict air pollutants like PM2.5 (particulate matters with an aerodynamic diameter less than 2.5 micron meters). As some examples, Zhang et al. [28] applied AutoRegressive Integrated Moving Average (ARIMA) model, a popular time-series forecasting

model, to predict PM2.5 in Fuzhou, China, also providing analysis about seasonal patterns and correlations with other pollutants. Ming et al. [14] use a variant of Hidden Markov Model (HMM) named hidden semi-Markov models (HSMM) by introducing the temporal structures into the HMM and use them to predict the concentration levels of PM2.5. All of these models are usually based on some statistical assumptions like stationarity or linear dependencies, which can have nonnegligible gap with reality and hence cause accuracy loss.

To address this issue, extensive machine learning methods have been explored. Long et al. [12] use a Least Square Support Vector machine (LSSVM) to jointly consider the non-linear relationship between pollutants concentration and meteorological data. Yu et al. [26] use a set of linear regression (LR) models to make fusion of multi-mode information like pollutant concentration and spatial feature, and different LR components are responsible for different factors. These classical machine learning models remove the mentioned statistical assumptions by parametric settings. However, it remains a problem to design a representive feature and it is still hard for these model to catch complicated spatial temporal patterns in the data.

Considering problems above, deep learning approaches take the position in the recent years. Fawaz et al. [6] gives a nice review on the deep learning method for time series classification. As a nature extension of sequence to sequence learning (S2S) [19], a lot of recurrent neural network (RNN) based approaches were proposed by deep learning communities. Du et al. [5] proposed a hybrid model stacked by a one dimensional convolutional neural network (CNN) and a bi-directional Long Short Term Memory network (Bi-LSTM) for representation learning of multivariate air quality related time series data, and report an outstanding prediction accuracy. Liang et al. [10] adopt a encoder-decoder framework, together with sophisticated attention mechanism, to apply a S2S learning. Though RNN based networks like LSTM can intuitively learn long term dependencies in the data, there exist researches [7, 20] reporting that they are suffering from vanishing gradient and parallelization issues and often failed to learn multi-scale features. To overcome these shortcomings, numerous full convolutional models (e.g. ConvS2S [7]) and full attention models (e.g. Transformer [20]) for S2S learning have been proposed.

In addition to the flaws mentioned above, most existing deep learning approaches for air pollution prediction are lack of effective multi-scale temporal modeling. Inspired by WaveNet presented by Oord et al. [15], where dilated convolution operation was introduced [25], we design our temporal subnets in order to extract multi-scale features from data by a full convolutional way.

B. Spatial Modeling

The unique spatial structure of geo-sensory data like air pollution signals make it different from other spatio-temporal data analysis tasks, e.g. video processing and multi-variants time series processing. The sparseness of sensors location and dynamics in sensor numbers make it challenging to adapt deep learning modules to extract spatial features. As some examples, Yi et al. [23] design a handcrafted spatial feature by setting a polar grid at point of interests and aggregating surrounding information, then use this feature for further temporal learning. Wang et al. [21] transfer sensor networks into dense images by simple cubic interpolation method, and similarly to video processing, a CNN based module is adopted to extract spatial features. Liang et al. [10] use sophisticated attention operation with a distance based prior to aggregate spatial information, while Zhang et al. [27] design a residual network that apply convolution on space and time simultaneously. In addition to these works, Graph Convolutional Networks (GCNs [16, 29]) based networks have attracted great interests because the geo-sensory networks can be naturally described by a topology of graph. Similar to Convolutional LSTM (ConvLSTM [22]), Yu et al. [24] as well as Lin et al. [11] combined GCNs with LSTM to get a novel hybrid spatio-temporal model for the task.

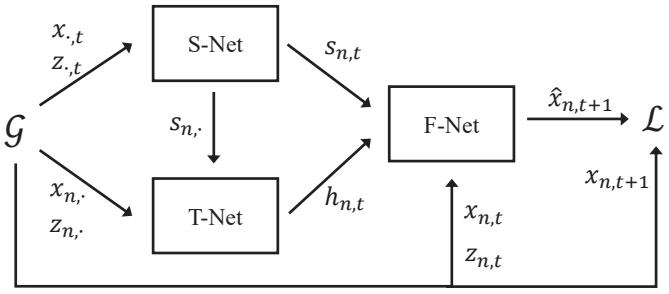


Fig. 2. The illustration of dataflow in MSSTN. The MSSTN is composed of three subnets, named S-Net, T-Net, and F-Net. The input data at time stamp t is first processed by S-Net, generating spatial features $s_{n,t}$ for each node at two different spatial scales. And then $s_{n,t}$, as well as original data are fed into T-Net, generating multi-scale temporal features $h_{n,t}$. Finally all extracting features are fed into F-Net to generate prediction $\hat{x}_{n,t+1}$. The loss function is then evaluated by comparing the prediction and ground truth.

However, most spatial modeling solutions from existing works mentioned above neglect the inherent multi-scale spatial correlations in the data. And these models may encounter the problem of over-fitting because of limited amount of available data compared to complicated model size. In this paper we propose to use a set of GCNs to extract spatial features at different scales, and together with other components to form a full convolutional framework, which can have less parameters while capable of modeling multi-scale spatio-temporal patterns.

III. The MSSTN Model

A. Formulation

We first introduce some notations that will be used through this paper. On one hand, the air pollution sensors in a **city** (e.g. Beijing) can be denoted as a graph

$$G = \{N, E, X, Z\} \tag{1}$$

where N is the set of nodes (a node for a sensor), E is the set of (undirected) edges, $X = \{x_{n,t}\}$ is the set of sensor

readings and $Z = \{z_{n,t}\}$ is the set of auxiliary data. A sensor reading $x_{n,t}$ is a vector composed of readings of pollutants concentration, for the sensor n at time stamp t. A auxiliary data $z_{n,t}$ is similarly defined for meteorological records as well as other used features (categorical features indicating weekdays and hours in this paper). In addition, there are some useful attributes of G: $|N|$ denotes the number of nodes in G. $A \in \mathbb{R}^{|N|\times|N|}$ is adjacency matrix and $D \in \mathbb{N}^{|N|\times|N|}$ is degree matrix of the graph G.

On the other hand, The air pollution data in a **region** (e.g. a group of cities around Beijing) will be similarly denoted as a graph $G = \{N, E, X, Z\}$, however, in region-scale case, a node will stand for a city. To avoid confusion, we add a superscript to all notations related to G. For example, we use G^a, $x^a_{n,t}$, and A^a for city a, as a city-scale graph. And Specifically, superscript $\Diamond$ to identify the region-scale graph: $G^\Diamond$, $x^\Diamond_{n,t}$, and $A^\Diamond$. The data of a city $x^\Diamond_{n,t}$ and $z^\Diamond_{n,t}$ are defined as a function of all sensors belong to that city n. And in this paper, this function is simply a picking-out operation, i.e. pick out a pre-selected sensor to be the representation of the city.

Whats more, we will use $\mathcal{G} = \{G^a, \ldots, G^\diamond\}$ to denote the set of all graphs, including both city-scale and region-scale graph. We also abbreviate the set $\{x^G_{n,t}\}_{n\in N^G}$ to $x^G_{\cdot,t}$ and set $\{x^G_{n,t}\}_{-L\le t<0}$ to $x^G_{n,\cdot}$ for convenience, where superscript can be further eliminated without causing confusion. The illustration of notations is shown in Figure 1.

The task of air pollution prediction is to predict future pollutants concentration for all sensors given all the data in the past. And this can be formulated as, to find a function f, such that

$$\{x^G_{\cdot,t}\}_{t=0,1,\ldots,T-1;G\in\mathcal{G}} = f(\{\mathcal{G}\}_{t=-1,-2,\ldots,-L}) \tag{2}$$

where T is number of time stamps need to predict, and L is the temporal receptive field of the predicting model.

Specifically, equation (2) is the formulation for *Multi Step Prediction* task. However, in this paper we will focusing on *One Step Prediction* task, that is only to predict one time stamp forward with $T = 1$. The inference algorithm for *Multi Step Prediction* under this setting will be discussed later. This simplification, however, reduce the prediction space significantly and boost the convergence as well as performance of the model, under the reality that the amount of training data is very limited.

B. Overview

Our proposed MSSTN is composed of three subnets, named S-Net for multi-scale spatial feature extraction, T-Net for multi-scale temporal feature extraction, and F-Net for feature fusion and final prediction. The overview of our model is illustrated in Figure 2. The input data is first processed by S-Net and T-Net, and then extracted features are fed into F-Net to generate prediction. The loss function is evaluated by comparing the prediction and ground truth.

1) S-Net: S-Net is a group of GCNs, which process data on a time slice. Specifically for each graph $G \in \mathcal{G}$, there will be a corresponding GCN function $GCN_G(\cdot)$. At any time stamp t $(-L \le t < 0)$, the formulation of computation is written as

$$s^G_{\cdot,t} = GCN_G(x^G_{\cdot,t}, z^G_{\cdot,t}) \tag{3}$$

where $s^G_{\cdot,t}$ is a set of spatial feature vectors defined on G. Note that $GCN_G(\cdot)$ will be different for different graph, but remains identical at all time stamps for certain graph.

2) T-Net: T-Net is a group of one-dimensional dilated convolutional networks (DCN), which process data along the time axis. Specifically for each graph $G \in \mathcal{G}$, there will be a corresponding DCN function $DCN_G(\cdot)$, At any node n $(n \in N)$, the formulation of computation is written as

$$h^G_{n,\cdot} = DCN_G(x^G_{n,\cdot}, z^G_{n,\cdot}, s^G_{n,\cdot}, s^\diamond_{G,\cdot}) \tag{4}$$

where $h^G_{n,\cdot}$ is a set of temporal feature vectors defined on G. Note that $DCN_G(\cdot)$ also take the output of S-Net as input, in order to take high level spatio-temporal features into consideration. Besides, $DCN_G(\cdot)$ is a *casual function*, which only make use of information in the past at any time stamps. And, again, $DCN_G(\cdot)$ will be different for different graph, but remains identical for any node for certain graph.

3) F-Net: F-Net is a group of dense connection networks (DNN), which process data locally on a node. Specifically for each graph $G \in \mathcal{G}$, there will be a corresponding DNN function $DNN_G(\cdot)$, At any time stamp t $(-L \le t < 0)$ and any node n $(n \in N)$, the formulation of computation is written as

$$\hat{x}_{n,t+1} = DNN_G(x_{n,t}, z_{n,t}, s_{n,t}, h_{n,t}) \tag{5}$$

where $\hat{x}_{n,t+1}$ is the prediction for pollutants concentration at node n and at a time step forward.

The illustration of the data flow is shown in Figure 2. And in the rest of this section, we will present the detailed structure and formulation for three subnets.

C. S-Net

In the past decade, CNNs have been developed as a powerful tool for spatial representation learning, especially under the topic of computer vision and video processing. However, geo-sensory systems like air pollution sensors network can be considerable irregular and sparse, as sensors are usually settled randomly in the city and cities are sparsely distributed in the region. And therefore, sensor readings at a time slice do not have a regular grid structure, making it hard to use classical CNN modules to handle the task. However, originated from graph spectral theory, GCNs make it possible to apply convolution operations on graph defined in this paper. For a graph $G \in \mathcal{G}$, Let $L = D - A$ be the *Laplacian Matrix* of G, whose eigen-decomposition is denoted as $L = Q\Lambda Q^T$ where $\Lambda \in \mathbb{R}^{|N|\times|N|}$ is the matrix of eigenvalues and $Q \in \mathbb{R}^{|N|\times|N|}$ are corresponding eigenvectors. A GCN defined in spectral domain then can be written as

$$u^{(l+1)} = \sigma(Q\Theta^{(l)}(\Lambda)Q^T u^{(l)}) \tag{6}$$

where u is a signal on G and l indicates the different layers of GCN. $\Theta^{(l)}(\Lambda)$ is the spectral of the filters to be designed

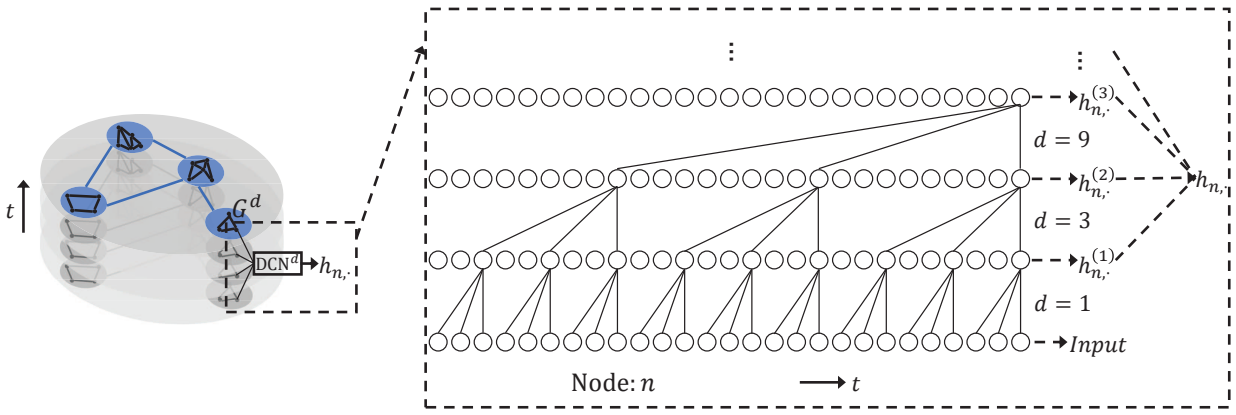


Fig. 3. The illustration of T-Net (left) and DCN (right) structure. T-Net is a collection of DCNs, where all nodes in a graph G share a DCN. The DCN operates on a time series, extracting multi-scale temporal features $h_{n,\cdot}$ for air pollution prediction. Similar to [15], the dilated convolution as well as gated non-linearity is the key to DCN. The dilated convolution generalize discrete convolution by skip several inputs (dilations). And then by stacking dilated convolution and gated non-linearity layer by layer with exponentially growing dilated ratio (3^l in this paper), informative features at different temporal scales are generated, which can be concatenated into final output of DCN.

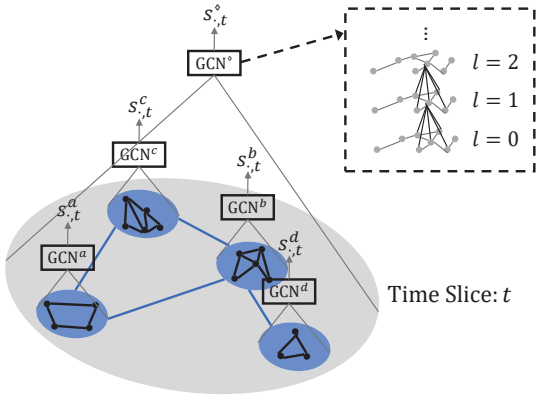


Fig. 4. The illustration of S-Net and GCN (in dashed box) structure. The S-Net is a collection of GCNs, where one GCN for one graph including both city-scale and region-scale graph. The ChebNet, a special version of GCN, is adopted for multi-scale spatial feature extraction. Since the ChebNet is strictly localized in space domain, the computation can be viewed as a spatial information spreading followed by a non-linearity activation, which is similar to the diffusion process of air pollutants.

or parameterized, usually being a function of Λ for the computational convenience in the space domain. $\sigma(\cdot)$ is the non-linear activation.

In this paper we use a special version of GCNs named ChebNet [3]. ChebNet use a Kth-order linear combination of Chebyshev polynomials as filters

$$\Theta^{(l)}(\Lambda) = \sum_{k=0}^{K-1} \theta_k^{(l)} \tau_k(\tilde{\Lambda}) \tag{7}$$

where τ_k is the the Chebyshev polynomials of order-k, $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I$, λ_{max} is the maximum eigenvalue in Λ, I is the identical matrix, and $\theta_k^{(l)}$ are parameters to be determined. Note that the convolution kernel given by Equation (7) is strictly K-localized. In a single convolution operation, the information of a node will only be spread within its K-neighborhood, which share a similar property with the diffusion procedure of air pollutants. Under this settings, the equation (6) can be easily evaluated without eigenvalue decomposition

$$u^{(l+1)} = \sigma\Big(\sum_{k=0}^{K-1} \theta_k^{(l)} c_k^{(l)} \Big) \tag{8}$$

where $c_k^{(l)}$ can be calculated by Chebyshev recursive relation

$$\begin{aligned} c_k^{(l)} &= 2\tilde{L} c_{k-1}^{(l)} - c_{k-2}^{(l)} \\ c_0^{(l)} &= u^{(l)} \\ c_1^{(l)} &= \tilde{L} u^{(l)} \end{aligned} \tag{9}$$

with $\tilde{L} = 2L/\lambda_{max} - I$.

We denote Equation (8) and Equation (9) as our GCN function $GCN_G(\cdot)$. Then the S-Net is a collection of GCNs applied on $G \in \mathcal{G}$ defined by Equation (3). Specifically, for any $G \in \mathcal{G}$ and any time stamp t, concatenate $x_{\cdot,t}$ and $z_{\cdot,t}$ as a signal on the graph G, i.e. as $u^{(0)}$ in Equation (9). After processing by $GCN_G(\cdot)$, $s_{\cdot,t}$, a spatial feature on G is generated. We will have a multi-scale spatial feature on $\mathcal{G}$ after applying it on graphs of different scales. This procedure is illustrated in Figure 4.

D. T-Net

Most of previous work use RNN based networks for temporal modeling. However, there exist researches [7, 20] reporting that RNNs are suffering from vanishing gradient and parallelization issues and often failed to learn multi-scale features. On the other hand, convolution has been a key concept under the topic of signal processing, and one dimensional CNN is considerable popular in time series deep learning. In this paper, we use a elaborate CNN named dilated convolutional networks (DCN) with dilated convolution operation and gated non-linearity [15]. These special designs enable DCN to have

exponentially growing receptive field as well as high non-linearity even with few layers, in order to extract features at different temporal scales explicitly and efficiently.

The d-dilated convolution is different from common discrete convolution only by using a dilated kernel $k(s)$, i.e. $k(s) \neq 0$ iff. $d|s$. An equivalent expression is to use a ordinary kernel $k(s)$ but a d-dilated convolution operation $*_d$ defined as [25]

$$(k *_d u)(t) = \sum_{\substack{ds+s'=t \\ s\geq 0}} k(s)u(s') \tag{10}$$

where u is a time series like signal and k is a convolution kernel with parameters. The bias of convolution is omitted here. It is easy to find that the dilated convolution defined in Equation (10) satisfies the causality requirement of $DCN_G(\cdot)$, i.e. the dependencies of $(u *_d k)(t)$ is strictly restricted to $\{u(s)\}_{s\leq t}$, and thus is a valid building block.

The gated non-linearity is a generalization of element-wise non-linear activation function by adding an parametric gate to control the recalibration of activations. And it is widely adopted in neural networks design such as LSTM and SE-block [8], which is believed to be able to boost the representive capability of neural networks with little additional computation. The gated non-linearity, along with residual connections, in this paper is defined as

$$\begin{aligned} h^{(l)} =& \tanh(W_0^{(l)} *_{d^{(l)}} u^{(l)}) \odot \sigma(W_1^{(l)} *_{d^{(l)}} u^{(l)}) \\ u^{(l+1)} =& W_2^{(l)} h^{(l)} + u^{(l)} \end{aligned} \tag{11}$$

where $u^{(l)}$ is a time series like signal in layer l, and $h^{(l)}$ is a feature map extracted at layer l. $W_0^{(l)}$, $W_1^{(l)}$, are convolution parameters for gate and signal respectively, $W_2^{(l)}$ is parameters for residual connection, all with bias term but omitted here. $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid non-linearity. $\odot$ is the element-wise production.

In order to extract informative features on multi temporal scales, we stack the dilated convolution and gated non-linearity for several layers with exponentially growing dilated factors $d = 3^l$. As illustrated in Figure 3, intuitively features on different temporal scales will be extracted across layers, where short term patterns are described in first layers and long term dependencies are encoded in last layers. Then we concatenate all $\{h^{(l)}\}$ to get

$$h_{n,\cdot} = Concat(\{h^{(l)}\}) \tag{12}$$

We denote Equation (10) (11) and (12) as our DCN function $DCN_G(\cdot)$. Then the T-Net is a collection of DCNs applied on $G \in \mathcal{G}$ defined by Equation (4). Specifically, for node n in any $G \in \mathcal{G}$, concatenate $x_{n,\cdot}$ and $z_{n,\cdot}$, together with city-scale spatial feature $s_{n,\cdot}$ as well as region-scale spatial feature $s^{\diamond}_{G,\cdot}$ as a time series, i.e. as $u^{(0)}$ in Equation (11). After processing by $DCN_G(\cdot)$, $h_{n,\cdot}$, a multi-scale temporal feature on G is generated. This procedure is illustrated in Figure 3.

E. F-Net

We use multi layer perceptrons (MLPs) as building blocks of F-Net, in order to make fusion of all extracted multi-scale spatial and temporal features and strengthen high level spatio-temporal interactions modeling for final predictions. In detail, the MLP for $G \in \mathcal{G}$ is of single hidden layer, mapping all extracted features as well as original data, $Concat(x_{n,t}, z_{n,t}, s_{n,t}, h_{n,t})$ to a vector $v_{n,t}$ of length $2^{n_{bits}}$, with a Softmax activation $Softmax(x)(k) = e^{x(k)}/\sum_i e^{x(i)}$ on the last layer.

As a trick to ensure a valid prediction ($0 \leq \hat{x}_{n,t+1} \leq x_{max}$ is a valid concentration reading if it is non-negative and within the measurement range), $v_{n,t} \in [0,1]^{2^{n_{bits}}}$ will be considered as a predictive distribution on a predefined alphabet of size $2^{n_{bits}}$. With the observation that small pollutant concentration take a larger part of the data, it is necessary to have a balanced distribution on the alphabet. For this consideration, we apply a n_{bits} μ-law logarithm quantization on range $[0, x_{max}]$

$$\mu_Q(x) = Q\left\lfloor \frac{\ln(1+Qx/x_{max})}{\ln(1+Q)} \right\rfloor \tag{13}$$

to get a n_{bits} alphabet, where $Q = 2^{n_{bits}}$, $\mu_Q(x)$ is the quantization index of input, and $\lfloor\cdot\rfloor$ is the Floor Function. Then the final prediction $\hat{x}_{n,t+1}$ is calculated as the expectation of distribution $v_{n,t}$ on this alphabet

$$\hat{x}_{n,t+1} = \sum_{i=0}^{2^{n_{bits}}-1} v_{n,t}(i)\mu_Q^{-1}(i) \tag{14}$$

where $\mu_Q^{-1}(i)$ is the decoder of the quantization.

We use Equation (13) and (14) as function $DNN_G(\cdot)$ in Equation (5). In F-Net, one MLP for one graph and all nodes in a graph share the same MLP. It is well worthy to note that MLP is equivalent to (ordinary) convolution operation of kernel size 1. And thus, together with graph convolution in S-Net and dilated convolution in T-Net, make the MSSTN a full convolutional neural network, which can have much smaller amount of parameters as well as great parallelization advantages for real time applications.

Finally a mean absolute error (MAE) is evaluated as loss function to form an end-to-end framework as MSSTN

$$\mathcal{L}_G = \sum_{n\in N^G} \sum_t \left|\hat{x}_{n,t}^G - x_{n,t}^G\right| \tag{15}$$

F. Multi-Step Inference

The MSSTN model trained for one step prediction can be easily extended to execute multi step prediction using a auto-regressive inference procedure. We can have an arbitrary length of predictions by using one step predictions as observations and feeding it back to the model. Note that in this inference procedure, the pollutant concentrations $\{x_{n,t}\}$ will be recursively updated, but the auxiliary information $\{z_{n,t}\}$ is considered to be provided. In the case of air pollution prediction task, we use meteorological data as auxiliary information, and therefore weather forecast data will be required for Multi-Step Inference.

IV. Experiments

In this section we present main experiments result as well as ablation study result of our MSSTN model. The code, data and trained models are available at https://github.com/Zhiyuan-Wu/MSSTN.

A. Settings

Datasets. In order to show the effectiveness of our proposed MSSTN model on the air pollution prediction task, we apply our model on a real world air pollution datasets named Urban Air Pollution Datasets in North China. There are in total 112 air monitoring stations located in 15 cites in the datasets. All of these cities are around Beijing, the capital of China, and all monitoring stations have hourly air pollutants concentration records. The detailed statistical information of the datasets can be found in Table I. We split data into training sets and independent testing sets with ratio 10:3. Specifically, records of first 10 months, i.e. before (UTC) 2017/11/1 14:00 (include) are used for training the model, and the records of last 3 months i.e. after (UTC) 2017/11/1 15:00 (include) are used for testing.

TABLE I
Details of Urban Air Pollution Datasets in North China

Number of Cities	15
Number of Sites	112
Latitude Span	36.53°N - 42.28°N
Longitude Span	111.55°E - 121.97°E
Central City	Beijing
Pollutants	PM2.5, PM10, O3
Meteorology	temperature, Humidity, Wind Speed, Wind Direction
Time Span	(UTC) 2017/1/1 14:00 - 2018/1/31 15:00
Records Interval	1 Hour, 3 Hours

Preprocessing. We apply several basic preprocessing procedure to make datasets ready for learning algorithm. (1)*Sampling rate alignment.* The recording rate for different items may be different. For example, the temperature is recorded hourly only in few large cites, but recorded every three hours in others. We simply adopt a linear up-sampling to unify all time series in the datasets into hourly records. (2)*The completion of missing values.* There exist approximately 10% missing values in raw sensor readings records. In most cases, we use a spatial interpolation value given by Inverse Distance Weighted (IDW) estimation from valid sensors in the same city at the same time slice. However, spatial interpolation will fail if most (even all) sensors are down simultaneously, and we have to use linear interpolation along the time instead. (3)*Normalization.* The numerical values of different quantities can be very different and it is important to normalize them into similar range for neural network based models. We apply a linear stretch where the upper and lower bound is decided by the corresponding 99% and 1% percentiles over the entire datasets. Note that the max-min normalization is not appropriate to avoid outliers. (4)*The adjacency matrix.* The terms in adjacency matrix are determined by the geographical distance between two nodes. Specifically, a gaussian kernel is applied on spherical distance computed by GPS coordinates pair, and values smaller than a threshold will be discarded, i.e. corresponding nodes pair are not considered to be connected.

Model structure. On one hand, some important hyper-parameters for model structures of MSSTN are as follows: For S-Net, all GCNs are of 3 layers and have maximum Chebyshev order of 3, and extracted spatial features are in 64 channels. For T-Net, all DCNs are of 4 dilated convolution layers with base dilated rate 3 and convolution kernel size 3. The extracted temporal features at each scales are all 16 channels, but are expended to 32 channels for next layer's input. For F-Net, the hidden number of MLPs are 128, and quantization is taken to be $n_{bits} = 5$. On the other hand, our implementation is based on the Tensorflow package [1], with default Adam optimizer and learning rate 0.0005. The early stopping trick is used for best validation performance.

Metrics Besides the mean absolute error (MAE), we also adopt rooted mean square error (RMSE) which is a popular metric for regression and prediction task.

$$\begin{aligned} E_{MAE} =& \frac{1}{M} \sum_{n \in N^G} \sum_{t} \left| \hat{x}_{n,t} - x_{n,t} \right| \\ E_{RMSE} =& \sqrt{\frac{1}{M} \sum_{n \in N^G} \sum_{t} \left(\hat{x}_{n,t} - x_{n,t} \right)^2} \end{aligned} \tag{16}$$

where M is the total number of terms in summation.

B. Results

One Step Prediction The Table II shows one step prediction result of the proposed MSSTN as well as many other popular models on the Urban Air Pollution Datasets in North China. These models include:

- Support Vector Regression (**SVR**). SVR is a popular kernel method in regression task. We split data of different cities into window slice and apply a SVR model with RBF kernel independently.
- Multi Layer Perceptrons (**MLP**). As a universal approximator, MLP plays a important role in neural network family. We use a MLP of 2 hidden layers with ReLu activation for comparison.
- LightGBM (**LGBM**) [9]. As an ensemble method, gradient boost decision tree (GBDT) is a powerful tool for machine learning task, of which LGBM is a fast and efficient implementation.
- Long Short Term Memory Network (**LSTM**). As a representation of RNN based time series predictors, we test a neural networks where we build a GRU network [2] for each city and map the output of GRU at each time step for a one step prediction.
- Diffusion Convolution Recurrent Neural Network (**DCRNN**) [11]. DCRNN is an outstanding spatial temporal network based on the combination of GCN

TABLE II
ONE STEP PREDICTION RESULT FOR 5 MAIN CITIES

Model	Beijing		Shijiazhuang		Taiyuan		Huhhot		Dalian		Average	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SVR	24.8063	29.7521	33.5855	43.1055	37.0533	46.3471	33.3685	40.1179	36.2745	41.0015	33.0176	40.4521
MLP	8.1176	13.5282	21.7218	30.5758	18.4101	27.1655	24.5930	32.0519	9.9266	14.5716	16.5538	24.8817
LGBM	6.5846	12.1617	13.0901	21.6894	10.5966	19.5544	11.5215	19.1660	4.7523	11.4777	9.3090	17.3192
LSTM	7.3519	12.7806	13.7945	22.6529	10.6122	19.4796	11.7648	19.5659	4.4704	9.4645	9.5987	17.4834
DCRNN	6.6594	11.7452	13.5136	22.2543	10.6928	**18.6544**	11.9366	19.3694	4.7834	10.5917	9.5171	17.1379
MSSTN	**6.1926**	**11.2896**	**12.2947**	**20.5379**	**10.0982**	18.6705	**10.8583**	**18.7679**	**4.1202**	**8.6335**	**8.7218**	**16.2764**

TABLE III
MULTI STEP PREDICTION RESULT IN BEIJING

Model	1h		3h		6h		12h		24h		48h	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SVR	24.8063	29.7521	25.6133	30.8975	29.0184	34.6545	36.2618	42.9871	47.7185	55.7933	55.0215	62.4976
MLP	8.1176	13.5282	10.6232	16.1388	13.8094	20.4534	19.7239	28.3191	25.2105	34.2607	28.2940	37.3437
LGBM	6.5846	12.1617	8.4465	14.5560	10.9659	19.5151	14.6347	25.3494	17.9910	**29.4341**	23.1302	37.0699
LSTM	7.3519	12.7806	9.6597	16.3933	11.9673	20.1492	15.3041	25.1131	19.3089	30.9065	22.2848	36.3147
DCRNN	6.6594	11.7452	10.0798	16.8269	13.8846	22.6989	17.9548	29.8268	24.6943	42.9330	27.4210	49.2022
MSSTN	**6.1926**	**11.2896**	**8.1644**	**13.4925**	**10.5302**	**17.3556**	**13.5465**	**22.8253**	**17.5055**	29.8782	**19.7373**	**33.5689**

and RNN and is reported to be successfully used in air pollution prediction.
- Multi Scale Spatial Temporal Networks (**MSSTN**). The proposed method.

Table II lists the prediction MAE and RMSE (both the lower the better) of all models above for 5 cities (Beijing, Shijiazhuang, Taiyuan, Huhhot and Dalian) in the dataset, and the average performance of these cities are computed in the last column. From Table II it is shown that our proposed MSSTN outperforms all of other baselines at all cites listed. The MSSTN improve the MAE error by 47.31%, 6.30%, 9.13%, 8.35%, and improve the RMSE error by 34.58%, 6.02%, 6.90%, 5.02% compared to MLP, LGBM, LSTM and DCRNN respectively. Therefore we can conclude that it is significantly beneficial to make use of multi-scale spatial temporal information in the data and to explicitly adopt a multi-scale network architecture. This make MSSTN be able to jointly and efficiently discover the related factors at various spatial temporal aspect and then make a more accurate prediction.

Multi Step Prediction In order to investigate the multi step prediction performance of the proposed MSSTN, we adopt a multi-step inference up to 48 hours for comparison. For convenience we only show the result in Beijing. Table III shows the average prediction error under different prediction window length $T \in \{$1h(identical to one step prediction), 3h, 6h, 12h, 24h, 48h$\}$. For example, under a 6h-prediction setting, the model is asked to give predictions in next 6 hours and the error is computed by averaging result for 1h-6h. The result shows that the MSSTN keep the advantages in prediction accuracy for multi step prediction task under various prediction length. Benefited from the utilization of multi-scale spatial temporal information, the MSSTN outperforms all baselines by a large margin especially in long term prediction task like $T = 48$h. These result can be intuitively interpreted as result of successful utilization of multi-scale spatial temporal information because empirically long term behavior of air pollution is heavily influenced by region transportation, which again prove the advantages of MSSTN.

C. Ablation Study

In order to further prove the effectiveness of the proposed architecture of MSSTN, we design a comprehensive ablation study.

Spatial Module Ablation. Most existing spatial feature extraction solutions ignore the multi-scale distribution of the sensor networks. In order to prove the advantages of multi-scale spatial modeling capability of S-Net, we verify the one step prediction performance in Beijing using different spatial feature extractor, however, with T-Net and F-Net as common temporal predictor. The baselines include
- No spatial module is used and the model will make the prediction locally only on temporal information.
- The handcrafted spatial feature proposed by [26], where the IDW interpolations are estimated within a polar grids centered at target sensors. And readings from those fake sensors are concatenated as spatial feature.
- A CNN extractor. The readings of sensor networks are first interpolated by IDW, and transformed into a heat

image with regular grids. Then a 3-layer CNN extractor are applied on the images, and the feature map of last layer are flattened as the spatial feature.

TABLE IV
SPATIAL MODULE ABLATION RESULT
(ONE STEP PREDICTION IN BEIJING)

Spatial Module	MAE	RMSE
None	6.4799	12.1641
Yu et al. [26]	6.4152	11.6030
IDW + CNN	6.3310	11.4886
S-Net (MSSTN)	**6.1926**	**11.2896**

The Table IV shows the spatial ablation result. The hand-crafted feature in [26] can be viewed as a special case of graph convolution operation but with fixed parameters and linear activation, which constraint the improvements. The CNN extractor can make adaptive representation automatically but still with very limited gains without multi-scale information. The proposed MSSTN shows the best performance in the test.

Temporal Module Ablation. Table V shows the temporal prediction performance comparisons, where the temporal part of MSSTN, i.e. T-Net, is fairly compared with baselines those only make temporal prediction without using any spatial information. The baselines include SVR, MLP, LGBM and LSTM. Benefited from ensemble design, the LGBM is very competitive in this situation. However, the proposed T-Net and F-Net still have best performance in the test.

TABLE V
TEMPORAL MODULE ABLATION RESULT
(ONE STEP PREDICTION IN BEIJING)

Temporal Module	MAE	RMSE
SVR	24.8063	29.7521
MLP	8.1176	13.5282
LGBM	6.5846	**12.1617**
LSTM	7.3519	12.7806
T-Net + F-Net	**6.4799**	**12.1641**

D. Discussion

Multi-Task Learning Issue. The optimization of objective (15) is in fact a *multi-task learning* problem, where we want errors from all graphs to be small. The computation of these errors involved some shared parameters which usually can be, unfortunately, competitive [17]. In this work we simply make copy of shared parameters and optimize all errors independently, which can have slightly better performance than the naive sum-up solutions, i.e. $\mathcal{L} = \sum_{G \in \mathcal{G}} \mathcal{L}_G$ (See Table VI). There are inspiring works from community on the topic of multi-task learning which is left as future work.

Computational Efficiency. It is important for learning models to be computationally efficient in the applications of geo-sensory system or IoT system. The MSSTN is a network of full convolutional architecture, which can be efficiently parallelized on GPU devices by many popular deep learning packages like Tensorflow. In our experiment, the training time of MSSTN is around 1 hour for all cities on a single Titan V GPU, while RNN based method like DCRNN take more than one day. This make it convenient for applications such as real time prediction and online learning.

TABLE VI
MULTI-TASK OPTIMIZATION COMPARISON OF MSSTN
(MAE FOR ONE STEP PREDICTION IN BEIJING)

Optimization	Beijing	Shijiazhuang	Taiyuan	Huhhot	Dalian
Sum-up	**6.1609**	12.7942	10.1957	11.1541	4.1918
Independent	6.1926	**12.2947**	**10.0982**	**10.8583**	**4.1202**

S-Net and Pooling Operation. It is well worthy to note that the proposed S-Net, as a collection of GCNs on city-scale graph and region-scale graph, is very similar to the concept of *pooling* operation in the convolutional neural networks, both of which are extracting features from blocks of lower layers for the input of higher layers. However, in geo-sensory systems, intuitively it is better to use different convolution kernels for different blocks of lower layers, indicating more convenience of our approach.

V. CONCLUSION

In this paper we present the Multi Scale Spatial Temporal Network (MSSTN) for air pollution prediction task. We propose to use a multi-level graph data structure to fit the multi-scale nature of geo-sensory systems and the MSSTN performs well in this scenario. The MSSTN is specially designed to better use of multi-scale spatial temporal information and their high-level interactions, with explicit multi-scale neural network structure design. The MSSTN is composed of three subnets, named S-Net, T-Net and F-Net. S-Net is a collection of GCNs operating on graphs at different scale, extracting multi-scale spatial representations. T-Net is a collection of DCNs generating multi-scale temporal features. And F-Net is a dense network for feature fusion and making final predictions. We conduct comprehensive experiments on Urban Air Pollution Datasets in North China, whose results demonstrate the outstanding prediction accuracy compared to many other popular spatio-temporal models.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No.2017YFC0212100).

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. 2016.

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

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[4] Robin L. Dennis, Daewon W. Byun, Joan H. Novak, Kenneth J. Galluppi, Carlie J. Coats, and Mladen A. Vouk. The next generation of integrated air quality modeling: Epaš models-3. *Atmospheric Environment*, 30(12):1925–1938, 1996.

[5] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. Deep air quality forecasting using hybrid deep learning framework. *arXiv preprint arXiv:1812.04783*, 2018.

[6] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[7] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[10] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, pages 3428–3434, 2018.

[11] Yijun Lin, Nikhit Mago, Yu Gao, Yaguang Li, Yao-Yi Chiang, Cyrus Shahabi, and José Luis Ambite. Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 359–368. ACM, 2018.

[12] L. I. Long, M. A. Lei, H. E. Jianfeng, Dangguo Shao, Y. I. Sanli, Yan Xiang, and Lifang Liu. Pm2. 5 concentration prediction model of least squares support vector machine based on feature vector. *Journal of Computer Applications*, 34(8):2212–2216, 2014.

[13] Sylvain Mailler, Laurent Menut, Dmitry Khvorostyanov, Myrto Valari, Florian Couvidat, Guillaume Siour, Solène Turquety, Régis Briant, Paolo Tuccella, and Bertrand Bessagnet. Chimere-2017: from urban to hemispheric chemistry-transport modeling. *Geoscientific Model Development*, 10(6):2397–2423, 2017.

[14] Dong Ming, Dong Yang, Yan Kuang, David He, Serap Erdal, and Donna Kenski. Pm2.5 concentration prediction using hidden semi-markov model-based times series data mining. *Expert Systems with Applications*, 36(5):9046–9055, 2009.

[15] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[16] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[17] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[18] Science Technology Council. Air Quality Subcommittee. Air quality forecasting: A review of federal programs and research needs. *Environmental Policy Collection*, 2001.

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

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[21] Hao Wang, Bojin Zhuang, Yang Chen, Ni Li, and Dongxia Wei. Deep inferential spatial-temporal network for forecasting air pollution concentrations. *arXiv preprint arXiv:1809.03964*, 2018.

[22] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[23] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 965–973. ACM, 2018.

[24] Bing Yu, Haoteng Yin, and Zhanxing Zhu. St-unet: A spatio-temporal u-network for graph-structured time series modeling. *arXiv preprint arXiv:1903.05631*, 2019.

[25] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[26] Zheng Yu, Xiuwen Yi, Li Ming, Ruiyuan Li, and Zhangqing Shan. Forecasting fine-grained air quality based on big data. In *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 2015.

[27] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[28] Lanyi Zhang, Jane Lin, Rongzu Qiu, Xisheng Hu, Huihui Zhang, Qingyao Chen, Huamei Tan, Danting Lin, and Jiankai Wang. Trend analysis and forecast of pm2. 5 in fuzhou, china using the arima model. *Ecological indicators*, 95:702–710, 2018.

[29] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.