Modelling Air Pollution Transfers for Prediction, Analysis and Simulation

PATRICK DEUTSCHMANN
GRAZ UNIVERSITY OF TECHNOLOGY

This project introduces an air pollution prediction model based on a graph neural network (GNN). It distinguishes itself from similar approaches in that its core component is an interpretable pollution transfer matrix that can be used to analyse the influence of specific locations on different ones. The model achieves performances comparable to the state-of-the-art for short-term predictions while also explaining its results. Moreover, it can simulate how reducing emissions at one location would impact air pollution in the area.

1 INTRODUCTION

In recent years, air pollution has become an increasingly influential factor in human health. With the industrialisation of more and more parts of the world and ever-growing economic output, it has increased steadily and become an important research area. In its Air Quality Guidelines [15], the WHO cites robust scientific evidence that air pollutants can severely affect human health in a variety of ways. One way of measuring air pollution is $PM_{2.5}$ (particles with diameters of 2.5 micrometres and less). As shown in [4], it is absorbed by the lungs, and extensive exposure can lead to several neurodevelopmental disorders, including autism, attention deficit disorder, and cognitive delays.

Air pollutants can be grouped into two categories: Primary pollutants that are directly emitted into the air and secondary pollutants that are formed by chemical reactions with the atmosphere itself. Depending on the type of pollutant, they can also be transferred over long distances with the wind. The emission of primary pollutants can either have natural causes or can be caused by human behaviour, i.e. road transport or stationary combustion, for example, in factories. Secondary pollutants include ozone and oxides of nitrogen and are heavily influenced by weather conditions. The characteristics of the air pollutant $PM_{2.5}$ are depicted in Figure 1.

The variety of influence factors renders modelling $PM_{2.5}$ non-trivial. This project aims at accurately predicting its concentration at different stations considering weather information, date and transfer events while also remaining interpretable with respect to the local influences. It does so using a graph neural network (GNN) that takes as input historical $PM_{2.5}$ values and weather information to predict the $PM_{2.5}$ concentration in the future.

In short, our contributions are as follows:

- We extend existing approaches of $PM_{2.5}$ prediction by an interpretable transfer matrix and obtain results comparable to the state-of-the-art for our predictions.
- We show how to analyse influences of surrounding cities on local $PM_{2.5}$ concentration.
- We demonstrate an approach that allows our transfer matrix to be used for simulations about the impact of local pollution reduction. For example, we can predict how much the air quality at a particular measurement station would be improved if the emissions of a nearby factory were reduced by a certain factor.

2 RELATED WORK

Traditional approaches for predicting air pollution such as [8, 19] rely on physical and chemical models that build on expert knowledge. While they utilise much scientific understanding of the

Author's address: Patrick Deutschmann
Graz University of Technology.

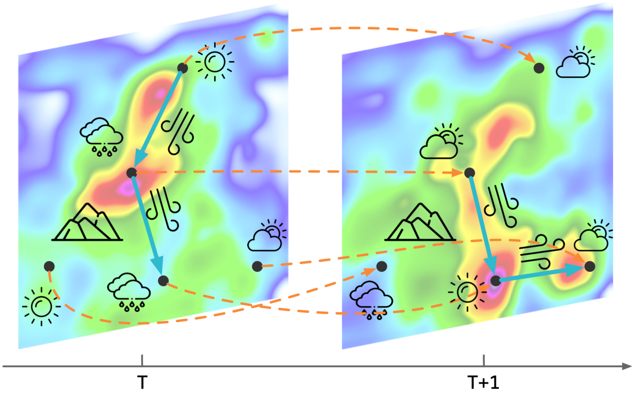


Fig. 1. This figure from [23] demonstrates the characteristics of $PM_{2.5}$ that are highly influenced by weather conditions. The background colour indicates the concentration, blue arrows show spatial transfers due to wind, and the orange dotted arrows indicate dispersion effects that occur over time.

underlying phenomena, they are often computationally expensive and display inaccuracies if circumstances occur that are not considered in the models [25].

Hence, data-driven approaches that suppose much less domain knowledge and rely on advances in machine learning to make their predictions are commonly used today. They reach from technologically simpler, but interpretable models such as GAMs [5] to complicated deep learning models. [6] groups the latter into

- models that perform fine-grained air pollution estimation, i.e. models that make spatial predictions for locations for which there are no measurements available,
- temporal models that produce air pollution forecasts and
- models that do both, such as [6].

There exist spatial models that build on attention mechanisms [3] and physically-inspired dispersion models that learn their parameters using deep autoencoders with convolutional LSTMs [14]. Forecast models, such es the one implemented in this paper can be based on RNNs [1, 9], ensembles [24] or perform hybrid spatio-temporal modelling, i.e. use both temporal and spatial aspects for their forecasts. These approaches typically build on graphs for the different measurement stations [11] and either use convolutions [17], attention [22] or recurrent units [23].

While these approaches yield good predictive performance, they are not interpretable by design. They consider transfers of air pollution but cannot explain the influences of different cities on the predictions. This project aims at resolving the issue while achieving comparable performance. There exist model-agnostic approaches that add interpretability to deep learning models, such as SHAP [13], that could also be applied to air pollution prediction. However, they would not directly give insights about the transfers and could not support simulations due to their post hoc nature.

3 METHODOLOGY

This section describes the precise problem this project is addressing, the data used, and the model proposed.

3.1 Problem formulation

The problem addressed in this paper is predicting the $PM_{2.5}$ concentration at time step t, denoted as $\hat{c}^t \in \mathbb{R}^N$ for N measurement stations. The inputs for these predictions are the current $PM_{2.5}$

concentration c^0 and weather information for the prediction time step $x^t \in \mathbb{R}^{K \times N}$ with K being the number of weather features, as described in detail in subsection 3.2. Predictions are made for a short-term time horizon where $t \in [1, \tau]$, with $\tau \in \{6, 12, 24, 48, 72\}$ for the purposes of our experiments.

3.2 Data

The geographical region on which we evaluate our approach is Beijing and the Hebei province in China, as they are highly polluted areas, in which also previous work has been done to serve as a means of comparison. We used two different types of data with different sources for this project.

The first is the $PM_{2.5}$ data used as historical values in the model inputs and as ground truth for the prediction targets. The data we used[1] comes from a network of highly accurate sensors by the Chinese government that is, however, rather sparsely deployed. It provides air quality information for various attributes such as $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3 and CO. We restrict ourselves to the prediction of $PM_{2.5}$ for the purposes of this project.

The second type of data we used is weather information from ECMWF[2], which serves as input to our model. We consider the following features:

- **Temperature** influences $PM_{2.5}$ through chemical effects and ventilation of cold fronts [21].
- **Planetary Boundary Layer (PBL) height**[3] relates to the vertical diffusion of $PM_{2.5}$ and is inversely correlated with its concentration [12, 20].
- **Humidity** promotes the formation of ammonium nitrate which is a component of $PM_{2.5}$ [21].
- **Air pressure** has a significant positive correlation with $PM_{2.5}$ [10].
- **Precipitation** reduces the $PM_{2.5}$ concentration due to drag effects [16, 18].
- **Wind** transfers $PM_{2.5}$ across distances of up to 250km [7].

The data is provided as a grid, and we retrieve the values closest to the $PM_{2.5}$ sensors of the first data set. Thereby we then achieve a network of measurement stations with both $PM_{2.5}$ as well as weather values.

3.3 Model

The model we are proposing is based on [23] and improves it in several ways. It operates on a graph $G = (V, E)$ where V are the nodes (measurement stations) with $|V| = N$ and E is the set of edges between them. Inputs to our model are grouped into two categories:

- **Node attributes** $x^t \in \mathbb{R}^{K \times N}$: These K attributes are assumed to locally influence the $PM_{2.5}$ concentration at a measurement station. They include temperature, PBL, humidity, air pressure and precipitation.
- **Edge attributes** $y^t \in \mathbb{R}^{L \times |E|}$: In contrast to the node attributes, these L attributes are assumed to influence the transfers between different measurement stations. Following [23], they incorporate domain knowledge and are derived from the wind speeds and directions at the different nodes and static information, i.e. the distance between cities and the altitude of the mountains between them. In particular, we compute the directional wind alignment and relative speed to make it easier for the network to learn spatial relations.

In the course of the project, we have tried multiple architectures for the model, but the final one is described in Algorithm 1. It receives as input the observed $PM_{2.5}$ concentration right before the prediction starts ($t = 0$) and the node and edge attributes for the prediction window ($t \in [1, ..., \tau]$).

[1] https://quotsoft.net/air/

[2] https://www.ecmwf.int/en/forecasts/datasets

[3] https://www.britannica.com/science/planetary-boundary-layer

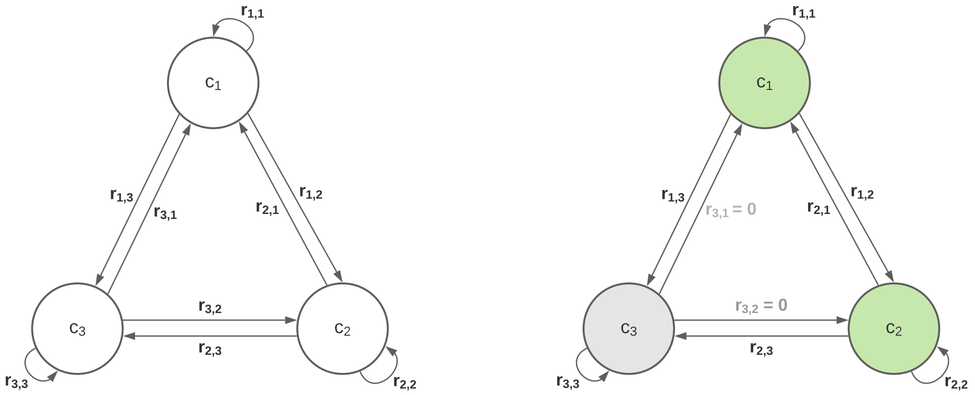
Algorithm 1: Prediction performed by our model

```
Input  : observed PM$_{2.5}$ concentration $c^0$
         node attributes $[x^1, ... x^\tau]$
         edge attributes $[y^1, ... y^\tau]$
Output : predicted PM$_{2.5}$ concentrations $[\hat{c}^1, ... \hat{c}^\tau]$
         transfer matrices $[R^1, ... R^\tau]$
$e_h = 0$;
$h_h = 0$;
Output_R = [] ;
Output_c = [] ;
for $t \in [1, ..., \tau]$ do
    if $t = 1$ then
        $\hat{c}^{t-1} = c^0$;
    end
    /* (1) compute local phenomena                                   */
    $h_h$ = NodeGRU($h_h$, $[x^t, \hat{c}^{t-1}]$) ;
    $h^t$ = NodeMLP($h_h$) ;
    /* (2) compute transfers                                         */
    $e_h$ = EdgeGRU($e_h$, generateEdgeAttributes($x^t$, $y^t$));
    $R^t$ = restructure(EdgeMLP($e_h$)) ;
    Output_R.append($R^t$) ;
    /* (3) execute transfers                                         */
    $\hat{c}^t = R^t h^t$ ;
    Output_c.append($\hat{c}^t$) ;
end
```

The output of the model are the predictions of the PM$_{2.5}$ concentrations $\hat{c}^t$ and a list of matrices R^t that contain the transfer influences, as explained in more detail in 3.3.1. After initialising the hidden states of the network's recurrent units and the output lists, the model iterates over all time steps $t \in [1, ..., \tau]$ of the prediction horizon and for each one goes through three stages:

(1) **Compute local phenomena**: A gated recurrent unit (GRU) *NodeGRU* gets as inputs the previous hidden state h_h, the weather information for the current time step x^t and the previous prediction of PM$_{2.5}$ concentration $\hat{c}^{t-1}$ and outputs an internal representation. At this stage, the network models interactions of weather phenomena with PM$_{2.5}$ individually for all measurement stations. Using an multi-layer perceptron (MLP) *NodeMLP*, the higher-dimensional hidden state h_h is reduced to the shape $\mathbb{R}^N$, so there is a value for every station.

(2) **Compute transfers**: By applying a spatial attention mechanism, the model learns between which stations and to which degree transfers of PM$_{2.5}$ occur. To do so, edge attributes are generated from the weather information of the stations and static domain knowledge. They are then concatenated with the weather information of the source and target node for each edge and fed through a recurrent unit *EdgeGRU*. Reducing the dimensionality of the hidden state through *EdgeMLP* yields a contribution value $r_{i,j} \forall (i, j) \in E$ that can be interpreted as the influence of station i on station j. In order to allow the network to learn more easily that some stations might have no contribution at all, the final activation function of *EdgeMLP* is



(a) The degree to which the $PM_{2.5}$ concentrations c_i with $i \in \{1, 2, 3\}$ affect the neighbouring stations is determined by the factors $r_{i,j}$. For example, the estimate for $c_1^{t+1} = \sum_{i \in \{1,2,3\}} r_{i,1} f(c_i)$ where $f(\cdot)$ is a function using NodeGRU and NodeMLP as explained in more detail in Algorithm 1.

(b) The transfer matrix can also be used to perform simulations. For example, by setting $r_{3,1} = r_{3,2} = 0$, we can simulate how the $PM_{2.5}$ concentration at stations 1 and 2 would be affected, if emissions at station 3 were turned off completely.

Fig. 2. A schematic illustration of the transfer model operating on a graph with three nodes

a ReLU that encourages sparsity. For further processing, these edge-wise contributions are restructured to a transfer matrix $R^t \in \mathbb{R}^{N \times N}$ that is explained in more detail in 3.3.1

(3) **Execute transfers**: Finally, the transfers are executed by performing a matrix-multiplication of transfer matrix R^t with the local contributions h^t. This yields the final prediction vector $\hat{c}^t$.

3.3.1 Transfer matrix R and Pollution Reduction Impact Analysis. In order to make the predictions of our model more interpretable, our model produces as a by-product for every time step t a transfer matrix R^t as shown in equation 1. Note that for brevity purposes, the index t is omitted.

$$R = \begin{pmatrix} r_{1,1} & r_{2,1} & \cdots & r_{N,1} \\ r_{1,2} & r_{2,2} & \cdots & r_{N,2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,N} & r_{2,N} & \cdots & r_{N,N} \end{pmatrix} \tag{1}$$

The entries in this matrix can be used to interpret the influences of certain stations on others as depicted in Figure 2a. The explainability of the model depends on the sparsity of R. If there are small contributions of surrounding cities for every point in time for every target city, the results are difficult to understand and hardly reflect reality. Hence, we use a ReLU activation function in the end of the *EdgeMLP* to promote zero activations for the $r_{i,j}$. However, this might lead to the situation that all entries in a row vector of R become zero, and the model cannot predict properly anymore. Therefore, in the final version, we set the diagonal constantly to one, i.e. $r_{i,j} = 1$ where $i = j$. This makes sense from a domain knowledge point of view, as the local conditions at a measurement station always affect the air pollution concentration.

Additionally to making the model more interpretable, the transfer matrix R can be used to make simulations for hypothetical situations by altering the influences $r_{i,j}$. For example, by setting $r_{i,j} = 0$ for all $i \in V$ and one specific $j \in V$ at simulation time, we can predict a world without the influence

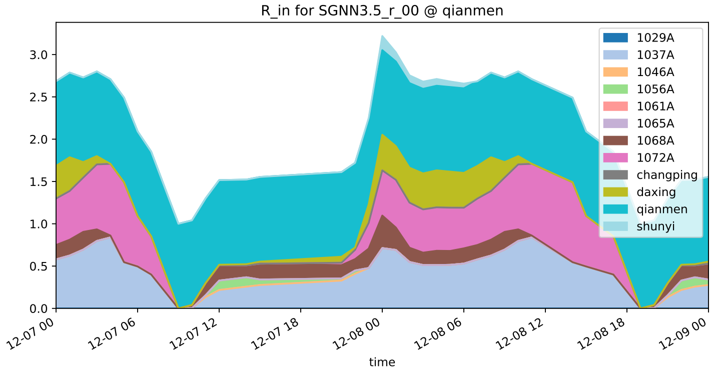


Fig. 3. This plot uses the transfer matrix R to explain how the model makes its predictions for the measurement station *Qianmen*. It shows how the column vector determines by what the station is influenced. We see that influences change over time.

of station j, such as illustrated in Figure 2b. This might be useful to examine, for instance, how shutting down a factory would reduce the $PM_{2.5}$ concentration in surrounding cities. Naturally, the influences cannot only be set to zero but can also be multiplied with a certain factor in order to simulate the effects of any percentage reduction.

4 EXPERIMENTS

We ran all experiments on an NVIDIA Tesla K80 and split our data into a train (01/01/2018 – 30/09/2020), validation (01/10/2020 – 30/11/2020) and test set (01/12/2020 – 31/12/2020). All models were trained with RMSprop and an initial learning rate of 0.0005. Experimental results have shown that using a step-wise decaying learning rate that is halved every three epochs works best for our model. The data is provided in batches of size 32 and trained for 30 epochs with early stopping after five epochs. We run all experiments five times and report the average results. In order to improve generalisation performance, we used a weight decay of 0.0005. We used Weights & Biases [2] for tracking our experiments and for visualising the results.

An example of a prediction produced by our model is depicted in Figure 4. The overall performance on the test set is reported in Table 1. The longer the prediction horizon τ, the harder it is for the model to make accurate predictions, as can also be seen in Figure 5.

5 DISCUSSION

In this project, we introduced a deep learning model that can be used for predicting, analysing and simulating air pollution. Its prediction results are comparable to the state-of-the-art, while the model lends itself well to being explained out-of-the-box. Its structure allows simulations of how reducing emissions at certain stations would affect air pollution in the area. However, this project only lays the foundation for a more advanced model. The simple matrix multiplication for the transfers prevents learning more complicated relationships such as transfer lags. Introducing temporal attention might alleviate this issue. Further, the R matrix is sometimes unstable between different training runs, showing that it is hard for the model to learn the transfer events exactly. Improving these factors could allow our approach to make a relevant difference in the current landscape of air pollution prediction approaches.

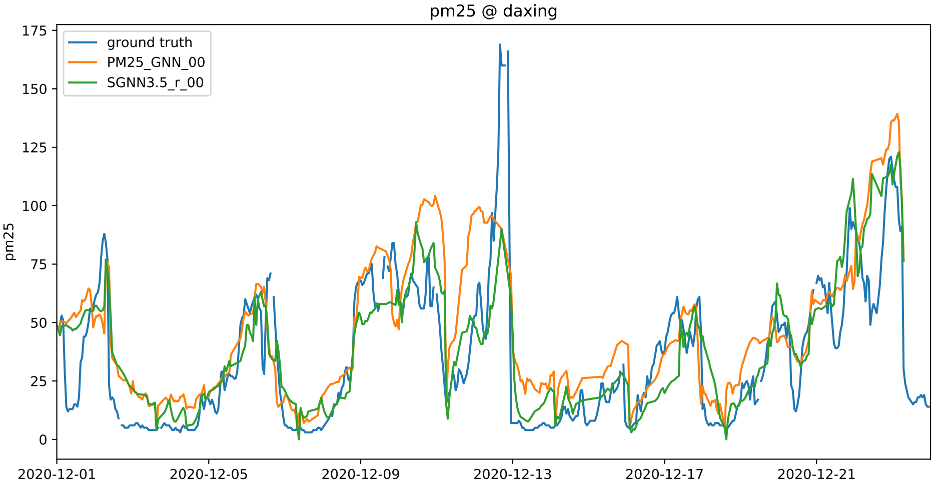


Fig. 4. This is an example prediction produced by our model at the station *Daxing* over the test time frame of December 2020. Overall, we see that the model can often capture the development very well, while some very sudden drops or jumps are difficult to predict.

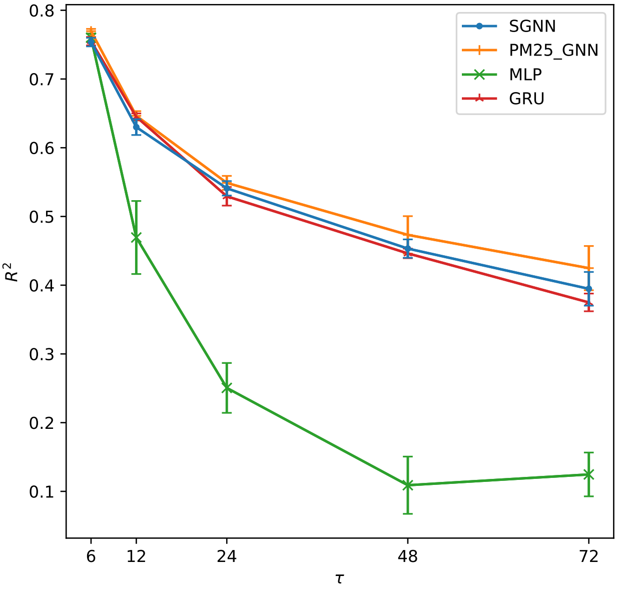


Fig. 5. The longer the prediction length τ, the harder it becomes for the models to accurately predict $PM_{2.5}$ concentration. This plot depicts our performance measured in R^2 to similar models and shows that it is comparable for most prediction lengths. The statistics have been computed over five individual training runs of all models.

τ		MLP	GRU	PM$_{2.5}$ GNN	SGNN
6	RMSE	12.8973	13.0061	**12.7615**	13.2216
	MAE	10.4380	10.5661	**10.4134**	10.7102
	R^2	0.7601	0.7551	**0.7697**	0.7537
12	RMSE	21.0701	16.6199	**16.5403**	17.1106
	MAE	17.8770	**13.6565**	13.6779	14.0696
	R^2	0.4694	0.6451	**0.6464**	0.6300
24	RMSE	26.2651	20.0557	19.9956	**19.6719**
	MAE	22.1770	16.3468	16.4179	**15.8836**
	R^2	0.2505	0.5293	**0.5489**	0.5410
48	RMSE	28.6216	21.9863	**21.5540**	21.9290
	MAE	23.3884	17.4836	**17.2278**	17.4317
	R^2	0.1090	0.4463	**0.4732**	0.4533
72	RMSE	28.4714	23.3448	**22.7929**	22.9120
	MAE	22.9170	18.2332	17.7749	**17.7527**
	R^2	0.1246	0.3749	**0.4248**	0.3947

Table 1. Performance comparison between our model (*SGNN*) and *MLP*, *GRU* and *PM$_{2.5}$ GNN* from [23]

REFERENCES

[1] V Athira, P Geetha, Rab Vinayakumar, and KP Soman. Deepairnet: Applying recurrent networks for air quality prediction. *Procedia computer science*, 132:1394–1403, 2018.

[2] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[3] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[4] Deborah A Cory-Slechta, Marissa Sobolewski, Elena Marvin, Katherine Conrad, Alyssa Merrill, Tim Anderson, Brian P Jackson, and Gunter Oberdorster. The impact of inhaled ambient ultrafine particulate matter on developing brain: Potential importance of elemental contaminants. *Toxicologic pathology*, 47(8):976–992, 2019.

[5] Johanna Einsiedler, Yun Cheng, Franz Papst, and Olga Saukh. Interpretable and transferable models to understand the impact of lockdown measures on local air quality. *arXiv preprint arXiv:2011.10144*, 2020.

[6] Yang Han, Qi Zhang, Victor OK Li, and Jacqueline CK Lam. Deep-air: A hybrid cnn-lstm framework for air quality modeling in metropolitan cities. *arXiv preprint arXiv:2103.14587*, 2021.

[7] Jianlin Hu, Yungang Wang, Qi Ying, and Hongliang Zhang. Spatial and temporal variability of pm2. 5 and pm10 over the north china plain and the yangtze river delta, china. *Atmospheric Environment*, 95:598–609, 2014.

[8] Kyung-Hwan Kwak, Jong-Jin Baik, Young-Hee Ryu, and Sang-Hyun Lee. Urban air quality simulation in a high-rise building area using a cfd model coupled with mesoscale meteorological and chemistry-transport models. *Atmospheric Environment*, 100:167–177, 2015.

[9] Victor OK Li, Jacqueline CK Lam, Yun Chen, and Jiatao Gu. Deep learning model to estimate air pollution using m-bp to fill in missing proxy urban data. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[10] Xiaodong Li, Xuwu Chen, Xingzhong Yuan, Guangming Zeng, Tomás León, Jie Liang, Gaojie Chen, and Xinliang Yuan. Characteristics of particulate pollution (pm2. 5 and pm10) and their spacescale-dependent relationships with meteorological elements in china. *Sustainability*, 9(12):2330, 2017.

[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, 2018.

[12] Tian Luan, Xueliang Guo, Lijun Guo, and Tianhang Zhang. Quantifying the relationship between pm 2.5 concentration, visibility and planetary boundary layer height for long-lasting haze and fog–haze mixed events in beijing. *Atmospheric Chemistry and Physics*, 18(1):203–225, 2018.

[13] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing*

Systems 30, pages 4765–4774. Curran Associates, Inc., 2017.

[14] Rui Ma, Ning Liu, Xiangxiang Xu, Yue Wang, Hae Young Noh, Pei Zhang, and Lin Zhang. Fine-grained air pollution inference with mobile sensing systems: A weather-related deep autoencoder model. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–21, 2020.

[15] World Health Organization. *Air quality guidelines: global update 2005: particulate matter, ozone, nitrogen dioxide, and sulfur dioxide.* World Health Organization, 2006.

[16] HOT Pye, H Liao, S Wu, Loretta J Mickley, Daniel J Jacob, Daven K Henze, and JH Seinfeld. Effect of changes in climate and emissions on future sulfate-nitrate-ammonium aerosol levels in the united states. *Journal of Geophysical Research: Atmospheres*, 114(D1), 2009.

[17] Yanlin Qi, Qi Li, Hamed Karimian, and Di Liu. A hybrid model for spatiotemporal forecasting of pm2. 5 based on graph convolutional neural network and long short-term memory. *Science of the Total Environment*, 664:1–10, 2019.

[18] Zhang Renhe, Qiang Li, and RuoNan Zhang. Meteorological conditions for the persistent severe fog and haze event over eastern china in january 2013. *Science China Earth Sciences*, 57(1):26–35, 2014.

[19] Andrew Riddle, David Carruthers, Alan Sharpe, Christine McHugh, and Jennifer Stocker. Comparisons between fluent and adms for atmospheric dispersion modelling. *Atmospheric environment*, 38(7):1029–1038, 2004.

[20] Tianning Su, Zhanqing Li, and Ralph Kahn. Relationships between the planetary boundary layer height and surface pollutants derived from lidar observations over china: regional pattern and influencing factors. *Atmospheric Chemistry and Physics*, 18(21):15921–15935, 2018.

[21] Amos PK Tai, Loretta J Mickley, and Daniel J Jacob. Correlations between fine particulate matter (pm2. 5) and meteorological variables in the united states: Implications for the sensitivity of pm2. 5 to climate change. *Atmospheric environment*, 44(32):3976–3984, 2010.

[22] Chunyang Wang, Yanmin Zhu, Tianzi Zang, Haobing Liu, and Jiadi Yu. Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 616–634, 2021.

[23] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 163–166, 2020.

[24] Xiuwen Yi, Zhewen Duan, Ruiyuan Li, Junbo Zhang, Tianrui Li, and Yu Zheng. Predicting fine-grained air quality based on deep neural networks. *IEEE Transactions on Big Data*, 2020.

[25] Yang Zhang, Marc Bocquet, Vivien Mallet, Christian Seigneur, and Alexander Baklanov. Real-time air quality forecasting, part i: History, techniques, and current status. *Atmospheric Environment*, 60:632–655, 2012.