Modeling fine-grained spatio-temporal pollution maps with low-cost sensors

SHIVA R. IYER, New York University, USA
ANANTH BALASHANKAR, New York University, USA
WILLIAM H. AEBERHARD, Swiss Data Science Center, Switzerland
SUJOY BHATTACHARYYA, Columbia University, USA
GIUDITTA RUSCONI, State Secretariat for Education, Research and Innovation (SERI), Switzerland
ANANT SUDARSHAN, University of Chicago, USA
ROHINI PANDE, Yale University, USA
LAKSHMINARAYANAN SUBRAMANIAN, New York University, USA

The urban populations of lower income countries are increasing at unprecedented rates. Unfortunately, so are urban air pollution rates. The centrality of air quality monitoring networks to the design of effective policy responses to this urban health emergency is well-recognized. But, weak state capacity and limited city finances constrain the ability of lower-income city governments to install reference grade air quality monitoring networks. Our study site of Delhi, India for instance, is home to over 15 million residents but has a network of only 30 reference grade air quality monitors. Can low-cost sensor based networks provide a robust monitoring alternative? In this paper, we evaluate this possibility using data from a low-cost monitoring network of 28 custom-designed low-cost portable air quality sensors installed in a dense network in Delhi, over a period of two years. Using data from reference grade monitors for validation, we show that low-cost sensors can be used to derive a real-time spatio-temporal high-precision pollution sensing map. Using these, we build effective forecasting models for both spatial and temporal pollution sensing. The main challenge is high spatio-temporal variability exhibited by low-cost sensors, owing to factors such as sensor faults, network and power issues. We present a novel methodology grounded in domain knowledge of pollution monitoring, that combines geostatistical approaches, spatio-temporal modeling and message-passing recurrent neural networks. We are able to model spatio-temporal variations effectively and make forecasts within 15 minute time-windows at 9.8%, 10.9% and 10.3% Mean Absolute Percentage Error (MAPE) over our low-cost monitors, reference grade monitors and the combined monitoring network respectively. With these accurate fine-grained pollution sensing maps, we provide a way forward to build citizen-driven low-cost monitoring systems that detect hazardous urban air quality.

1 INTRODUCTION

Pollution forecasting in cities with dense populations can be critical for generating fine-grained policy recommendations and public health warnings [11, 29]. The scale of accurate sensor based monitoring required to achieve this can come at a huge cost and thus inhibit building a dense fine-grained pollution sensing map. In this paper, we describe a methodology to model and forecast urban air quality at a fine-grained level using dense and noisy *low-cost sensors*. There are two main questions we seek to answer in this paper – *i)* how can we use a network of low-cost and portable air quality monitors in order to build a fine-grained pollution heatmap in a city that provides accurate forecasting?, *ii)* does it help to augment existing monitoring networks by the local governments with low-cost air quality sensors? We develop a hybrid model that combines many state-of-the-art approaches

Authors' addresses: Shiva R. Iyer, New York University, Department of Computer Science, New York, New York, USA, shiva.iyer@cs.nyu.edu; Ananth Balashankar, New York University, Department of Computer Science, New York, New York, USA, ananth@nyu.edu; William H. Aeberhard, Swiss Data Science Center, Bern, Switzerland, william.aeberhard@sdsc.ethz.ch; Sujoy Bhattacharyya, Columbia University, New York, New York, USA; Giuditta Rusconi, State Secretariat for Education, Research and Innovation (SERI), Bern, Switzerland; Anant Sudarshan, Department of Economics, University of Chicago, Chicago, Illinois, USA, anants@uchicago.edu; Rohini Pande, Department of Economics, Yale University, New Haven, Connecticut, USA, rohini.pande@yale.edu; Lakshminarayanan Subramanian, New York University, Department of Computer Science, New York, New York, USA, lakshmi@cs.nyu.edu.

for spatio-temporal modeling: (a) a polynomial spline model that models daily trends; (b) a spatio-temporal hierarchical model (STHM) for imputing missing data; and (c) a message-passing recurrent neural network (MPRNN) that uses neighboring sensors' information.

The deployment of low-cost particulate matter sensors to replace or augment reference grade pollution air quality monitoring systems has been studied extensively recently, and have addressed issues of calibration [13, 23, 24], design [27, 35], data selection [3] and personal exposure quantification [25, 39]. However, building a highly accurate large scale fine-grained pollution sensing and monitoring map that leverages the size of a pollution network has been largely unexplored. Specifically, modeling the behavior of noisy low-cost sensors in cities with high pollution and population density has not been studied previously, with recent state-of-the-art mapping approaches providing errors only in the range of 30-40% [6, 32]. This high error lends the pollution sensing map unusable for policy making and air quality hazard detection. In this paper, we build on prior work and model the pollution network in its entirety, with prediction models at each sensor location based on a recurrent neural network model dependent on sensor-reading messages sent from near-by sensor locations.

Our adaptive statistical approach can incorporate data from several noisy and low-cost sensors and provides an attractive and more viable alternative. We employ a data-driven approach in which first we fit a cubic spline function that captures the daily and hourly mean trends, then we fit a spatio-temporal hierarchical model (STHM) to impute missing values of sensors (due to power and network outages) and create a "baseline" spatio-temporal field, and then finally model the residuals using a message-passing graph neural network that incorporate spatial priors and temporal trends from nearby sensors' data. We aim to forecast a given sensor's readings of the concentration of fine particulate matter ($PM_{2.5}$) measured in $\mu g/m^3$ using historical data of up to 8 hours from all the sensors in the network. We make this choice because the primary advantage of low-cost sensors lies in their ability to provide a large number of noisy measurements. By learning the variability of each of these noisy measurements through message passing neural networks which have the ability to model each sensor separately, we learn to not only separate the signal from the noise, but build an accurate sensing network of low-cost sensors that achieves 10% Root Mean Squared Error (RMSE) in forecasting up to one hour in advance over a fine-grained spatio-temporal grid as compared to baseline modeling approaches that provide 30% RMSE. By using a sparse network of sensors, whose signals are shared through neural network embeddings, we learn to capture the information from nearby sources that might affect the readings of nearby sources (e.g. factory) and ignore the ones which are heavily localized (e.g. food cart). Such an accurate fine-grained pollution sensing map ($\leq$ 10% MAPE) is usable by policy makers in deciding which neighborhoods of the city need interventions to improve the air quality and population health (Figure 1). Estimating such models provides a way to efficiently use information from several monitors to make predictions over a fine-grained grid, with the ability to seamlessly and flexibly incorporate low-cost sensors in developing countries.

2 MODEL

We model our problem as a graph prediction problem, where we attempt to predict a value at every node at a certain time from neighboring and historical values. In our setting, each sensor location $v \in \mathcal{V}$ is a node in an undirected graph. Assuming that air pollutants in one region can impact pollution in another space, we make the graph complete, where an edge exists between every pair of nodes. The end goal for us is to train a model that predicts at any node/location, the pollution level, measured in terms of the concentration of fine particulate matter ($PM_{2.5}$), at time $t + 1$ given one or more readings from neighboring locations prior to $t + 1$. We adopt a three-step approach to achieve this. We first fit a cubic spline based on daily trends at each sensor location, then we fit a spatio-temporal hierarchical model to impute missing data and then finally train a Message-Passing Recurrent Neural Network (MPRNN) (§A.3) to predict raw PM values. In order to account for the amount of influence based on the pairwise distances, we include the Euclidean distance between sensors as part of our

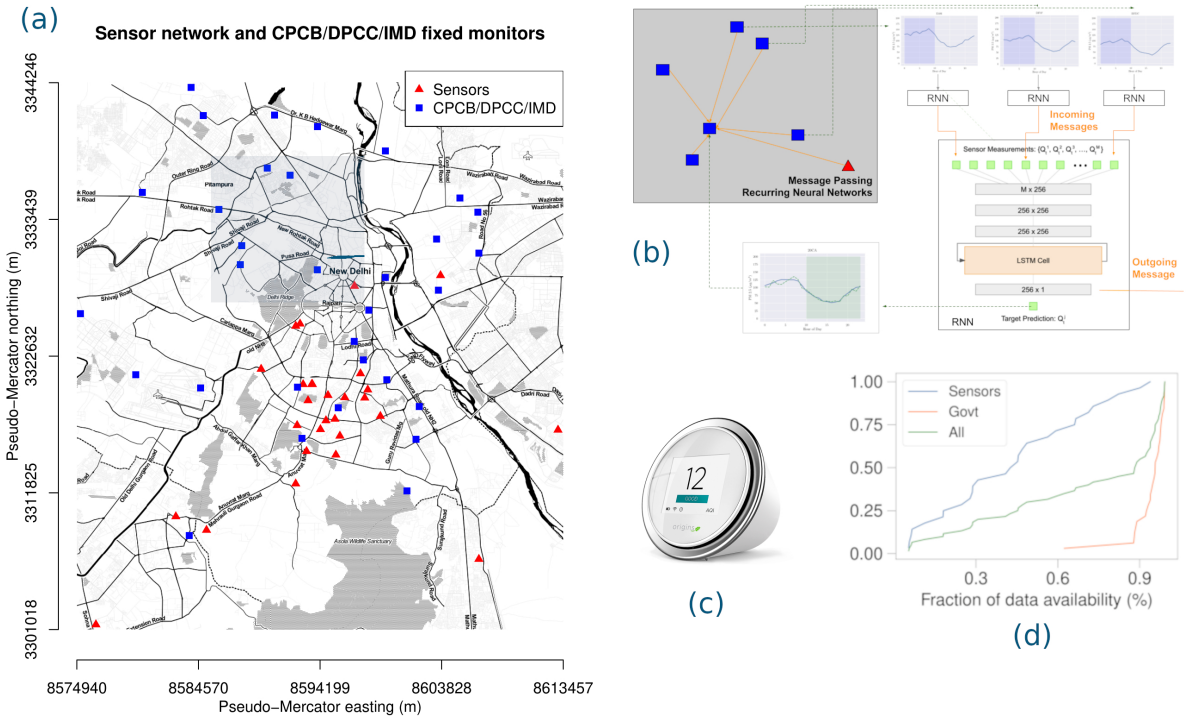


Fig. 1. **Message Passing Recurrent Neural Network for Pollution Monitoring in Delhi (a)** Network of air quality monitors in the entire greater Delhi region **(b)** Model architecture, showing M sensor inputs feeding into the layers and producing a single real output, illustrated by zooming on the selected region in (a). The computation goes from top to bottom. The green boxes represent input PM concentrations from a set of locations, the grey boxes the hidden linear transformation layers, with the numbers in the boxes representing the number of internal parameters to be learned, and the orange box shows the RNN with the LSTM cells. Here 256 is the embedding size of the hidden layer messages passed, that was chosen empirically based on performance. The final output is the single real value of PM concentration. The input to the RNN is the vector output of length 256 from the hidden layer. More details are in the supplementary text. **(c)** Sample model of a low-cost sensor **(d)** Our experimental testbed of monitors, and the quality of the $PM_{2.5}$ data obtained. We had to contend with frequent outages and communication issues that plagued our sensor network and affected data availability.

feature embedding in our message-passing formulation. We test this model by predicting values at locations where sensors, and therefore ground truth information, are present, but the model is generalized enough to be used to predict at locations where there is no ground truth data available. If $y_{v,t}$ is the reading of the sensor at location v, at timestamp t, and $\hat{y}_{v,t}$ is our corresponding prediction, we aim to minimize the mean absolute percentage loss:

$$MAPE = \sum_{v} \sum_{t} \frac{|\hat{y}_{v,t} - y_{v,t}|}{y_{v,t}} \tag{1}$$

For pre-processing prior to the above message passing model being fit, we model daily temporal patterns per sensor and per location. For example, if our prediction error follows a temporal pattern of say, higher prediction error in the morning, while lower in the afternoon, we leverage this pattern by fitting piecewise polynomial functions, called a *spline*. The residual sensor readings after the spline model is then used to fit the MPRNN

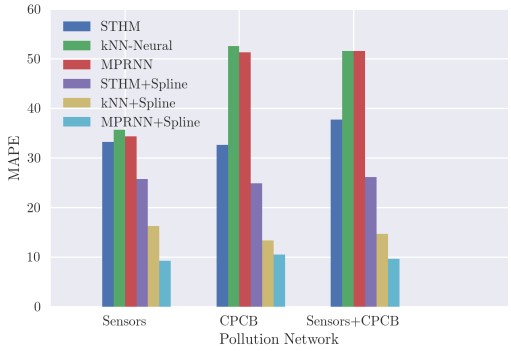
model after an imputation step. As a default imputation procedure, we use the value at the same time of day from the closest other day when data was available. Using this method to fill up the gaps in the data resulted in reasonably good performance. We were further able to improve the performance slightly though by using a more principled imputation approach that using Spatio-Temporal Hierarchical Model (STHM) (§A.7) to build a smooth spatio-temporal field over the entire spatial and temporal domain, and then use that to fill up the gaps when data was unavailable.

We contrast our combined model with two alternative state-of-the-art modeling approaches in order to set a baseline to benchmark the MPRNN model performance. The first one is a model from geostatistics, similar to the STHM model. When STHM is used solely for prediction, it performs poorly as it does not take in neighboring sensors inputs when they are missing, whereas in the best performing model, we use the STHM imputations as part of the messages in the MPRNN model. The second baseline is an alternative neural network formulation that employs information from a specified number (K) of nearest neighbors to predict the value at a location, called the k-Nearest Neighbor (k-NN) Spatial Neural Network (§A.7).

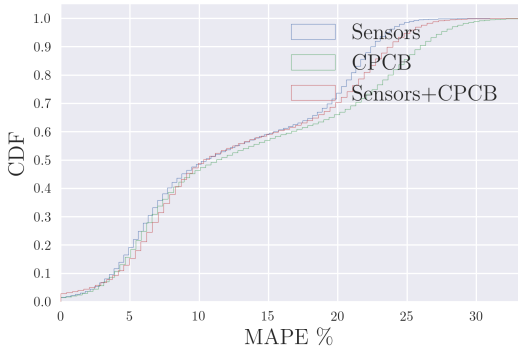
3 RESULTS

Model	Our sensors		Govt monitors		Combined	
	RMSE	*MAPE*	*RMSE*	*MAPE*	*RMSE*	*MAPE*
STHM	29.5	33.2%	38.3	32.7%	31.4	37.8%
k-NN v1	73.2	59.3%	108.3	64.7%	–	–
k-NN v2	38.8	35.7%	69.7	52.6%	54.2	51.6%
MPRNN	37.1	34.4%	65.2	51.3%	56.3	51.6%
Per-Sensor Spline	25.1	32.8%	60.4	49.1%	47.3	36.5%
STHM + Spline	21.8	25.8%	27.2	24.9%	24.2	26.2%
k-NN v2 + Per-Sensor Residual Spline	11.6	16.3%	18.1	13.4%	12.8	14.7%
MPRNN + Per-Sensor Residual Spline	9.8	10.2%	13.2	11.7%	10.4	12.6%
Per-Sensor Spline + Residual MPRNN	10.1	10.5%	14.7	12.2%	10.7	13.5%
MPRNN with STHM imputation + Per-Sensor Residual Spline	9.5	9.3%	12.7	10.5%	10.1	9.7%
Per-Sensor Spline + MPRNN with STHM imputation	9.5	9.4%	12.6	10.5%	10.1	9.6%
MPRNN with STHM imputation + Average Residual Spline	10.1	9.8%	13.2	10.9%	11.2	10.3%

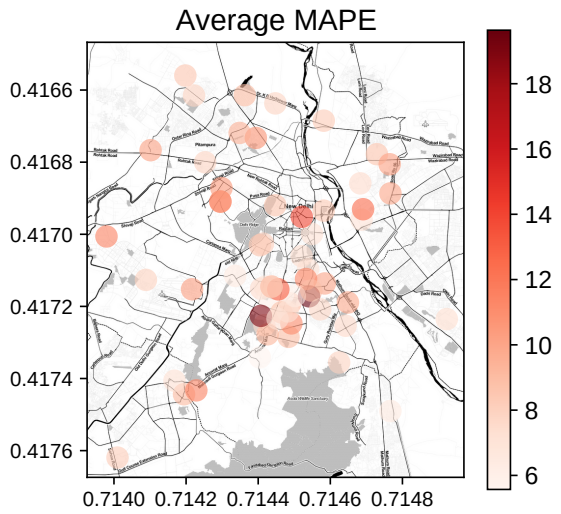
Table 1. RMSE and MAPE of prediction of PM concentrations, averaged across all the sensor locations. The RMSE is in units of $\mu g/m^3$. The best performing model is shown in boldface. The MPRNN works well to predict a "baseline" PM concentration based on historical trends and neighboring influences, while a well-trained spline helps to finely tune the prediction after subtracting the baseline prediction from the ground truth. In comparison with the k-Nearest Neighbor neural network model, the MPRNN is definitely an improvement, and this improvement is more noticeable after the spline correction. This is because it explicitly models influences among the nodes in the graph in the form of message-passing. The v1 is a simplistic model not too different from a time series modeling method, and assumes an inverse square law distribution between the influence of neighboring readings and the respective distances. Clearly, a time series-based approach alone, discounting neighboring influences, performs rather poorly. On the other hand, k-NN v2 does not assume any such prior relationship between the distance and influence. The v2 embeds the distance and bearing into the feature vector and lets the training process determine the right parameters and relationship among them, thus providing better results than v1.



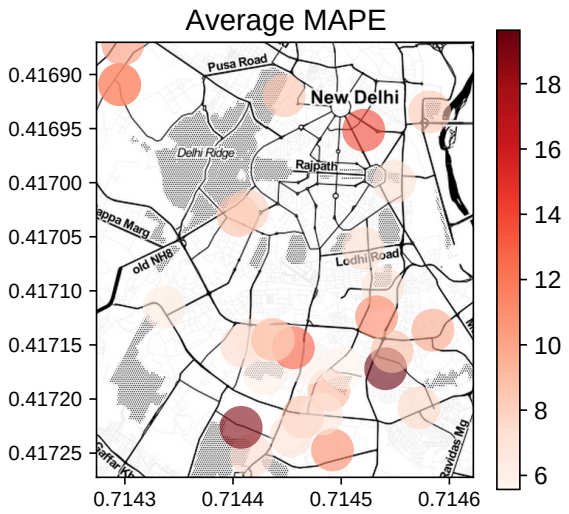
(a) Bar plot comparing our methodology with other competing approaches



(b) Distribution of MAPE across all the locations shown as a cumulative density function (CDF)



(c) Prediction errors of the best performing model (MPRNN+Spline) at every monitoring location on the map



(d) Errors of the final prediction zoomed into the regions with highest concentration of sensors (New Delhi and South Delhi)

Fig. 2. Prediction errors of $PM_{2.5}$ during the test period (Nov 1, 2019 - May 1, 2020) shown as the Mean Absolute Percentage Error (MAPE) of the ground truth and predicted $PM_{2.5}$ concentration. In this period, the $PM_{2.5}$ concentration values ranges between 0 and 1000 $\mu g/m^3$, and average value being ~130 $\mu g/m^3$

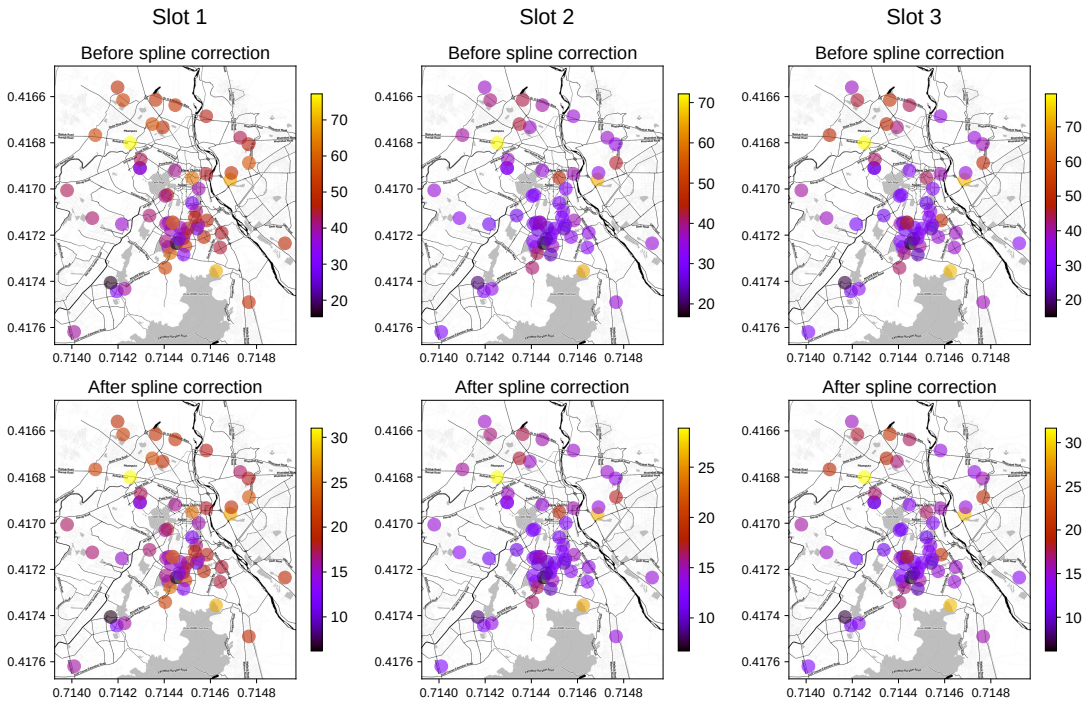
In our data available from May 1, 2018 till May 1, 2020, we use the data until Oct 30, 2019 for training (75%) and hold out the remaining (25%) for testing. We report two criteria – the root mean squared error (RMSE) and the mean absolute percentage error (MAPE). We evaluate our models on the data from the combined set of our 28 low-cost sensors and the 32 government monitors, as well as separately on each set. For each of these locations, we compare our model based predictions with the ground truth of the measurement of the pollution sensor.

Overall, the MPRNN model with imputed data (see Methods), along with the spline correction either done during pre-processing or post-processing, work well to predict a "baseline" PM concentration level, over which a spline correction provides impressively good performance on predicting the PM concentration across all locations. By estimating a spline per location, we are able to improve our predictive performance significantly. But we can perform nearly equally well by estimating an "average" spline over all the locations. Across all locations, the median RMSE and MAPE are 9.15 $\mu g/m^3$ and 8.64% respectively. The best case values are 4.28 $\mu g/m^3$ and 5.57% respectively, and the worst case values are 24.1 $\mu g/m^3$ and 19.64% respectively. The location where we have minimum MAPE is at a location in Green Park, a very busy area of south Delhi, further validating the need for fine-grained pollution sensing in a large city like Delhi.

It is interesting that even though our approach provided better results overall, the STHM does a better job of predicting a spatio-temporal field when using the data from the public monitors, which is spread over a larger geographical area, is more superior in quality and also has far fewer gaps in the data in this time period, all in total contrast to our network. We infer that the hierarchical model is better suited for building coarse spatio-temporal fields and imputing data, whereas the neural network models are better suited for fine-tuning our prediction numbers due to the amount of control they provide. This is also evident from the fact that the hierarchical model provides better performance when using the public monitoring network alone, in comparison to the other methods, prior to the spline correction. For the same reason, our final fine-tuned predictive performance using *Spline+MPRNN+STHM* on the public monitoring network does not match that on our low-cost sensor network.

4 DISCUSSION

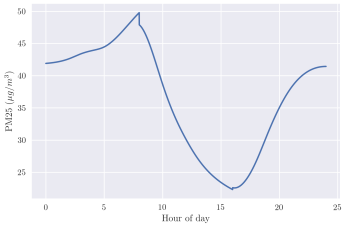
Our contributions are significant when compared to the recent and fast-growing literature that explores the use of distributed sensor networks to gather information on air pollution and other meteorological variables in urban contexts. Clements et al.[7] provide a comprehensive review of many such works. In the last few years, researchers have sought to learn more about how pollution sensing systems of low-cost sensors may be deployed in urban contexts [10, 19, 21, 26, 31, 33, 36]. With the exception of Gao et al. [10], who examine the performance of fine particulate sensors in Xi'an in China, most of these deployments have occurred in areas with significantly lower air pollution than the city of Delhi in India. In this paper, we provide evidence of modeling a fine-grained low-cost pollution sensing map from a highly polluted city like Delhi. Gao et al. [10] also point out that low-cost $PM_{2.5}$ sensors may perform worse in very low pollution environments, suggesting that they may be relatively more useful when particulate concentrations are high. While their study focused on Xi'an, a large city (area: 3,898 mi^2) with only 8 low-cost sensors, we dramatically increase the density of the deployment by 28× in Delhi (area: 573 mi^2) with 28 sensors. Further, the large longitudinal dataset we have been able to capture over 2 years as compared to prior work which captured at most a few weeks of data, allows us to model long-term seasonal changes and train more complex neural network models that can adapt to seasonal and daily patterns and produce significantly low RMSE. Related approaches in this space can be broadly classified into three groups – spatial interpolation approaches, land-use regression and dispersion models Xie et al. [37] Jerrett et al. [18]. In the case of dispersion models, they assume that an appropriate chemical transport model is identified along with their parameter values, and a high-quality emissions inventory. In the case of land-use regression models, having access to environmental characteristics that significantly influence pollution is critical. This additional data is often suited for longer range predictions, as the geographical and meteorological data vary over a longer temporal and coarser spatial grids [38]. For instance, in the US EPA dispersion model, parameters are estimated on grid cell squares with a length in the order of a few kilometers [9], while the parameters are used for inference of meteorological outputs at spatial resolutions of up to 500m. Our approach, in contrast relies on fine-grained positioning of low-cost sensors and makes the case for crowdsourcing pollution sensing. This



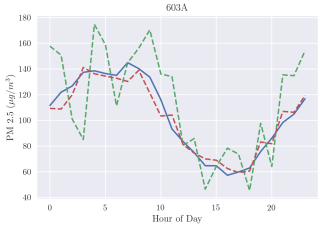
(a) Distribution of residuals: Slot 1 (12 AM - 8 AM)

(b) Distribution of residuals: Slot 2 (8 AM - 4 PM)

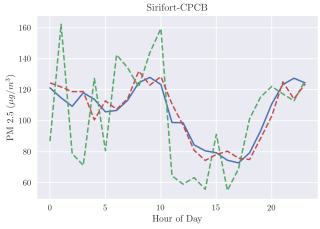
(c) Distribution of residuals: Slot 3 (4 PM - 12 AM)



(d) Cubic spline correction



(e) Ground truth $PM_{2.5}$ (blue), along with MPRNN prediction (green) and final prediction after spline correction (red) at one of our sensor locations in Chanakyapuri in New Delhi.



(f) Ground truth $PM_{2.5}$ (blue), along with MPRNN prediction (green) and final prediction after spline correction (red) at the CPCB monitor at Sirifort in South Delhi.

Fig. 3. This figure aims to show the interpretation of the spline correction, and its effect on the residual. The top two rows show the distribution of the residuals (in PM units of $\mu g/m^3$) over space, before and after the spline correction. Three different splines were fitted over the residuals in three different time slots in the day. We observe that for the most part, locations that exhibited high residual errors after MPRNN fit (in the upper quantiles of the residual error distribution) continued to show high error (relative to other locations) even after spline correction, even though the magnitude of the residual does decrease. This phenomenon is partially explained by the high baseline values of the sensors with high residual errors, that is often coupled with high variance in measurement - which we are yet to better capture in our modeling.

way, we demonstrate that a collective effort from citizens using low cost sensors can actually help in building a high quality pollution sensing map.

The low MAPE and RMSE across all monitors in Delhi provided by our Per-Sensor Spline+MPRNN with STHM imputation model is significant as it can detect hazardous air quality with high precision. The WHO air quality standards prescribe that $PM_{2.5}$ levels should not exceed 10 $\mu g/m^3$ and 35 $\mu g/m^3$ at an annual and daily average levels, while the Indian Government air quality standards prescribe 40 $\mu g/m^3$ and 60 $\mu g/m^3$ respectively. We note that for the 60 sensors, Delhi has exceeded these prescribed levels 371 out of the 641 days on a daily level, across 2 years of our measurement. The 9.7 % MAPE error that we are able to achieve, corresponds to the ability to detect hazardous air quality as per Indian government standards with 93.5% precision and 90.8% recall. This further indicates that the low error rate we have obtained leads to an almost exact forecasting of hazardous air quality. This enables citizen-driven sensing where pollution sensor readings can be crowdsourced and effective policy interventions like clean energy policies that penalize construction sites that have PM 2.5 levels more than 25% higher than the nearest monitoring center can be operationalized [1]. Specifically, the improvement in forecasting power is achieved in specific pollution hotspots like bus stations, markets, etc (Figures 2c, 2d). In addition, we can provide transparency of the overall average pollution of the city [2] and contribute towards increasing the co-benefits of clean energy policies [28, 34]. The development of fine-grained pollution sensing maps at low-costs can further catalyze the deployment of such monitoring networks in other polluted cities, where the pollution networks are sparse. With citizens procuring, deploying and modeling pollution of cities accurately, this paper provides a way forward for developing high-quality fine-grained pollution sensing maps.

ACKNOWLEDGEMENTS

The work done by authors Shiva Iyer, Ananth Balashankar, and Lakshminarayanan Subramanian in this paper was supported by NYUWIRELESS Group (https://www.nyuwireless.com). We acknowledge our collaboration with Kaiterra (https://kaiterra.com/), a company that manufactures smart indoor air quality monitors and air filters, in this endeavor. We are very grateful to them for their efforts in enabling the sensor installations for this project. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NYUWIRELESS or Kaiterra. We also acknowledge the contributions of Ulzee An, in building specific baseline models.

COMPETING INTERESTS AND CONFLICT OF INTERESTS DISCLOSURES

Dr. Subramanian is a co-founder of Entrupy Inc, Velai Inc, and Gaius Networks Inc and has served as a consultant for the World Bank and the Governance Lab. Mr. Balashankar is a Ph.D student at New York University, and is also funded in part, by the Google Student Research Advising Program. Dr. Subramanian reports that Velai Inc broadly works in the area of socio-economic predictive models. No other disclosures were reported.

[1] https://indianexpress.com/article/cities/delhi/dust-management-committee-recommends-air-quality-monitors-at-large-delhi-construction-sites-7437599/

[2] https://www.downtoearth.org.in/blog/air/delhi-s-air-quality-and-number-games-76214

REFERENCES

[1] Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. https://pytorch.org/.

[2] A. Beloconi, N. Chrysoulakis, A. Lyapustin, J. Utzinger, and P. Vounatsou. Bayesian geostatistical modelling of PM10 and PM2.5 surface level concentrations in Europe using high-resolution satellite-derived products. *Environment International*, 121:57–70, 2018.

[3] J. Bi, N. Carmona, M. N. Blanco, A. J. Gassett, E. Seto, A. A. Szpiro, T. V. Larson, P. D. Sampson, J. D. Kaufman, and L. Sheppard. Publicly available low-cost sensor measurements for pm2.5 exposure modeling: Guidance for monitor deployment and data selection. *Environment International*, 158:106897, 2022. ISSN 0160-4120. doi: https://doi.org/10.1016/j.envint.2021.106897. URL https://www.sciencedirect.com/science/article/pii/S0160412021005225.

[4] M. Cameletti, R. Ignaccolo, and S. Bande. Comparing spatio-temporal models for particulate matter in Piemonte. *Environmetrics*, 22(8): 985–996, 2011.

[5] M. Cameletti, F. Lindgren, D. Simpson, and H. Rue. Spatio-temporal modeling of particulate matter concentration through the SPDE approach. *AStA Advances in Statistical Analysis*, 97(2):109–131, 2013.

[6] H.-J. Chu, M. Z. Ali, and Y.-C. He. Spatial calibration and pm 2.5 mapping of low-cost air quality sensors. *Scientific reports*, 10(1):1–11, 2020.

[7] A. L. Clements, W. G. Griswold, J. E. Johnston, M. M. Herting, J. Thorson, A. Collier-Oxandale, and M. Hannigan. Low-cost air quality monitoring tools: From research to practice (a workshop summary). *Sensors*, 17(11):2478, 2017.

[8] N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data.* John Wiley & Sons, Hoboken, NJ, 2011.

[9] EPA. Aermod implementation guide. 2021.

[10] M. Gao, J. Cao, and E. Seto. A distributed network of low-cost continuous reading sensors to measure spatiotemporal variations of pm2. 5 in xi'an, china. *Environmental pollution*, 199:56–65, 2015.

[11] G. Geng, Y. Zheng, Q. Zhang, T. Xue, H. Zhao, D. Tong, B. Zheng, M. Li, F. Liu, C. Hong, et al. Drivers of pm2. 5 air pollution deaths in china 2002–2017. *Nature Geoscience*, 14(9):645–650, 2021.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1263–1272, 2017.

[13] M. R. Giordano, C. Malings, S. N. Pandis, A. A. Presto, V. McNeill, D. M. Westervelt, M. Beekmann, and R. Subramanian. From low-cost sensors to high-quality data: A summary of challenges and best practices for effectively calibrating low-cost particulate matter mass sensors. *Journal of Aerosol Science*, 158:105833, 2021. ISSN 0021-8502. doi: https://doi.org/10.1016/j.jaerosci.2021.105833. URL https://www.sciencedirect.com/science/article/pii/S0021850221005644.

[14] A. C. Harvey. *Forecasting, structural time series models, and the Kalman filter.* Cambridge University Press, Cambridge, UK, 1989.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, nov 1997. doi: 10.1162/neco.1997.9.8.1735.

[16] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, jan 1989. doi: 10.1016/0893-6080(89)90020-8.

[17] S. R. Iyer, U. An, and L. Subramanian. Forecasting sparse traffic congestion patterns using message-passing rnns. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3772–3776, 2020. doi: 10.1109/ICASSP40776.2020.9052963.

[18] M. Jerrett, A. Arain, P. Kanaroglou, B. Beckerman, D. Potoglou, T. Sahsuvaroglu, J. Morrison, and C. Giovis. A review and evaluation of intraurban air pollution exposure models. *Journal of Exposure Science and Environmental Epidemiology*, 15(2):185, 2005.

[19] W. Jiao, G. Hagler, R. Williams, R. Sharpe, R. Brown, D. Garver, R. Judge, M. Caudill, J. Rickard, M. Davis, et al. Community air sensor network (cairsense) project: evaluation of low-cost sensor performance in a suburban environment in the southeastern united states. *Atmospheric Measurement Techniques*, 9(11):5281, 2016.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[21] C. Lin, J. Gillespie, M. Schuder, W. Duberstein, I. Beverland, and M. Heal. Evaluation and calibration of aeroqual series 500 portable gas sensors for accurate measurement of ambient ozone and nitrogen dioxide. *Atmospheric Environment*, 100:111–116, 2015.

[22] F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society, Series B*, 73(4):423–498, 2011.

[23] H.-Y. Liu, P. Schneider, R. Haugen, and M. Vogt. Performance assessment of a low-cost pm2. 5 sensor for a near four-month period in oslo, norway. *Atmosphere*, 10(2):41, 2019.

[24] X. Liu, R. Jayaratne, P. Thai, T. Kuhn, I. Zing, B. Christensen, R. Lamont, M. Dunbabin, S. Zhu, J. Gao, D. Wainwright, D. Neale, R. Kan, J. Kirkwood, and L. Morawska. Low-cost sensors as an alternative for long-term air quality monitoring. *Environmental Research*, 185: 109438, 2020. ISSN 0013-9351. doi: https://doi.org/10.1016/j.envres.2020.109438. URL https://www.sciencedirect.com/science/article/pii/S0013935120303315.

[25] S. Mahajan and P. Kumar. Evaluation of low-cost sensors for quantitative personal exposure monitoring. *Sustainable Cities and Society*, 57:102076, 2020. ISSN 2210-6707. doi: https://doi.org/10.1016/j.scs.2020.102076. URL https://www.sciencedirect.com/science/article/pii/

S2210670720300639.

[26] S. Moltchanov, I. Levy, Y. Etzion, U. Lerner, D. M. Broday, and B. Fishbain. On the feasibility of measuring urban air pollution by wireless distributed sensor networks. *Science of The Total Environment*, 502:537–547, 2015.

[27] J. Prakash, S. Choudhary, R. Raliya, T. S. Chadha, J. Fang, and P. Biswas. Real-time source apportionment of fine particle inorganic and organic constituents at an urban site in delhi city: An iot-based approach. *Atmospheric Pollution Research*, 12(11):101206, 2021. ISSN 1309-1042. doi: https://doi.org/10.1016/j.apr.2021.101206. URL https://www.sciencedirect.com/science/article/pii/S1309104221002701.

[28] H. Qian, S. Xu, J. Cao, F. Ren, W. Wei, J. Meng, and L. Wu. Air pollution reduction and climate co-benefits in china's industries. *Nature Sustainability*, 4(5):417–425, 2021.

[29] N. D. Rao, G. Kiesewetter, J. Min, S. Pachauri, and F. Wagner. Household contributions to and impacts from air pollution in india. *Nature Sustainability*, pages 1–9, 2021.

[30] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society, Series B*, 71(2):319–392, 2009.

[31] A. A. Shusterman, V. E. Teige, A. J. Turner, C. Newman, J. Kim, and R. C. Cohen. The berkeley atmospheric co 2 observation network: initial evaluation. *Atmospheric Chemistry and Physics*, 16(21):13449–13463, 2016.

[32] G. C. Spyropoulos, P. T. Nastos, and K. P. Moustris. Performance of aether low-cost sensor device for air pollution measurements in urban environments. accuracy evaluation applying the air quality index (aqi). *Atmosphere*, 12(10), 2021. ISSN 2073-4433. doi: 10.3390/atmos12101246. URL https://www.mdpi.com/2073-4433/12/10/1246.

[33] L. Sun, K. C. Wong, P. Wei, S. Ye, H. Huang, F. Yang, D. Westerdahl, P. K. Louie, C. W. Luk, and Z. Ning. Development and application of a next generation air sensor network for the hong kong marathon 2015 air quality monitoring. *Sensors*, 16(2):211, 2016.

[34] K. Tibrewal and C. Venkataraman. Climate co-benefits of air quality and clean energy policy in india. *Nature Sustainability*, 4(4):305–313, 2021.

[35] J. Tryner, M. Phillips, C. Quinn, G. Neymark, A. Wilson, S. H. Jathar, E. Carter, and J. Volckens. Design and testing of a low-cost sensor and sampling platform for indoor air quality. *Building and Environment*, 206:108398, 2021. ISSN 0360-1323. doi: https://doi.org/10.1016/j.buildenv.2021.108398. URL https://www.sciencedirect.com/science/article/pii/S0360132321007952.

[36] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi. Gas sensor network for air-pollution monitoring. *Sensors and Actuators B: Chemical*, 110(2):304–311, 2005.

[37] X. Xie, I. Semanjski, S. Gautama, E. Tsiligianni, N. Deligiannis, R. Rajan, F. Pasveer, and W. Philips. A review of urban air pollution monitoring and exposure assessment methods. *ISPRS International Journal of Geo-Information*, 6(12):389, Dec 2017. ISSN 2220-9964. doi: 10.3390/ijgi6120389. URL http://dx.doi.org/10.3390/ijgi6120389.

[38] C. Yeh, A. Perez, A. Driscoll, G. Azzari, Z. Tang, D. Lobell, S. Ermon, and M. Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature communications*, 11(1):1–11, 2020.

[39] M. Zusman, C. S. Schumacher, A. J. Gassett, E. W. Spalt, E. Austin, T. V. Larson, G. Carvlin, E. Seto, J. D. Kaufman, and L. Sheppard. Calibration of low-cost particulate matter sensors: Model development for a multi-city epidemiological study. *Environment International*, 134:105329, 2020. ISSN 0160-4120. doi: https://doi.org/10.1016/j.envint.2019.105329. URL https://www.sciencedirect.com/science/article/pii/S0160412019321920.

A MATERIALS AND METHODS

A.1 Data

The data used for the modeling the air pollution levels in Delhi was sourced from a combination of 32 local Delhi government monitors and a network of 28 low-cost sensors deployed by us in various locations of Delhi from May 2018 to May 2020. The average availability of each of these sensors are about 90% and 30% over the measured period respectively. Correspondingly, we calibrate our sensors, provided by Kaiterra[3], against the government sensors, by conducting a longitudinal comparison study by measuring in close proximity to the location of the government monitoring centers. The locations and their summary statistics of the sensors by location is given by the Tables 2 and 3.

A.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) belong to a broader family of deep neural networks which are general function approximators [16]. In this experiment, the purpose of using a deep model is to model the complex nonlinear dependencies between the input and output without the need to impose an explicit physical model. We specifically choose RNNs because they are well suited for modeling sequential or time series data. The working of an RNN can be described simply by the function Φ in the equation $y_t, h_t = \Phi(y_{t-1}, h_{t-1})$, where y_t refers to a label or value that is predicted by the network at time t, and h_t is an internal state that represents the "memory" of the network at time t. Given sequential data of the form $y_0, \ldots, y_t$, Φ is applied repeatedly to predict label y_i, state h_i and so on until time t. The initial internal state h_0 is assumed to be zero in most applications. The number of such recursive computations (equivalently, the number of cells in the unraveled RNN) defines the length of the *history* that is used in the learning process to predict the value at $t + 1$. However, while RNNs provide a semantic framework for prediction of sequential data, they provide no innate mechanism in deciding when the internal state h should be modified. This challenge is addressed by Long-Short Term Memory (LSTM) cells [15] which explicitly facilitates the persistence or re-initialization of the internal state vector h over real sequential data. The use of LSTM cells in RNN architectures have been empirically shown to improve predictive power in temporal data because of their ability to learn long-term dependencies, and hence we employ them in our model.

A.3 Message-Passing Recurrent Neural Network

Message-Passing Recurrent Neural Network (*MPRNN*), based on [12, 17], is a neural network architecture that is applied on a graph in order to predict values at each node in the graph. This approach enables to us incorporates spatial interactions between each pair of nodes as "messages" that are broadcast from every node to its neighbors. Each node has a modified version of a Long Short Term Memory (LSTM) network (§A.2) that iterates between message-passing and the recurrent computations.

We denote by $y_{v,t}$ the PM concentration, measured in $\mu g/m^3$, at a node v and time t. Mathematically, we would like to learn a function $\mathcal{F}$ such that $y_{v,t+1} = \mathcal{F}(v_1, y_{v_1,t}, v_2, y_{v_2,t}, \ldots; v_j \in \mathcal{V})$ where the set $\mathcal{V}$ denotes the set of all the nodes in the graph. A recurrent neural network unit (§A.2) is assigned to each node in the graph, with each node v maintaining a hidden state $h_{v,t}$ at time t. Through a message-passing phase and an time-recurrent phase, our model infers the next hidden state $h_{v,t+1}$ from which the PM value at v is decoded. A message-passing operation allows one segment to observe the hidden state of its neighboring segments.

The computation proceeds in five steps, as five layers of the neural network. In the first phase, the observation phase, the measurements $Y_t = \{y_{v,t} | v \in \mathcal{V}\}$ at time t are encoded into $h_{v,t}$ by the observation operation O_v. In the second and third phases, one or more iterations of messaging (M) and updating (U) operations are performed to propagate the observations in the graph. In the fourth phase, for each node, a time-recurrent operator T_v utilizing

[3]https://www.kaiterra.com/

Monitor ID	Count	Min	Max	Median	Max	Std. Dev
AnandVihar_DPCC	13562	0.0	985.0	97.0	141.4	130.4
AshokVihar_DPCC	15244	0.0	972.0	76.0	121.3	122.2
AyaNagar_IMD	14228	0.1	954.0	64.6	86.8	81.7
BurariCrossing_IMD	9593	0.1	989.7	85.7	125.8	121.2
CRRIMathuraRoad_IMD	14712	0.0	973.8	76.3	112.1	108.3
DwarkaSector8_DPCC	15208	0.0	958.3	71.8	108.6	102.4
IGIAirport_IMD	14168	0.1	867.2	60.3	90.8	88.2
IHBAS_CPCB	14518	0.0	989.6	83.8	111.3	93.0
ITO_CPCB	14410	0.0	989.3	82.0	115.9	103.9
Jahangirpuri_DPCC	15077	0.0	994.0	96.0	136.8	123.6
JNS_DPCC	14979	0.0	929.0	65.8	105.5	104.7
LodhiRoad_IMD	14322	0.3	980.8	64.1	87.7	79.8
MandirMarg_DPCC	14825	0.0	945.0	76.0	104.4	91.9
MDCNS_DPCC	15239	0.3	985.8	70.5	99.3	88.7
Mundaka_DPCC	13503	0.0	988.5	89.0	134.8	132.1
NehruNagar_DPCC	15262	0.0	997.5	74.8	128.0	134.5
NSIT_CPCB	15220	0.0	997.5	92.6	113.5	81.8
OkhlaPhase2_DPCC	15002	0.0	987.0	71.5	109.3	105.2
Patparganj_DPCC	14865	1.0	997.0	163.0	197.8	140.9
PunjabiBagh_DPCC	14899	0.0	997.0	178.3	217.7	154.0
Pusa_DPCC	13625	0.0	978.0	68.0	105.7	101.1
Pusa_IMD	14578	0.1	986.3	57.0	84.0	83.4
RKPuram_DPCC	13535	0.0	877.3	77.0	109.9	101.6
Rohini_DPCC	15045	0.0	967.0	84.3	132.6	125.4
Shadipur_CPCB	14627	0.0	997.2	90.3	117.6	94.6
Sirifort_CPCB	14706	0.0	994.3	70.8	104.8	99.2
SoniaVihar_DPCC	15021	0.0	984.0	75.3	112.6	106.3
SriAurobindoMarg_DPCC	13687	0.0	992.8	63.5	93.9	89.1
VivekVihar_DPCC	14922	0.0	957.3	70.5	113.6	113.7
Wazirpur_DPCC	15125	2.8	969.8	92.0	140.6	129.4

Table 2. Summary Statistics of Government Pollution Monitors

Monitor ID	Count	Min	Max	Median	Max	Std. Dev
113E	10173	2.17	959.00	66.33	113.71	113.95
1FD7	3804	1.49	444.67	59.96	85.89	74.46
20CA	8654	0.90	809.33	55.57	91.39	92.48
2E9C	1733	7.00	617.50	112.00	135.55	99.77
3ACF	813	0.00	319.00	32.25	44.94	39.78
498F	954	6.50	241.67	55.50	67.43	41.52
4BE7	9110	3.00	743.36	79.68	124.31	117.20
56C3	6546	2.58	664.08	61.25	101.83	101.36
5D7A	2776	6.17	427.07	41.55	52.22	40.37
603A	6274	2.75	1047.42	65.92	102.98	100.09
72CA	13862	2.75	909.75	80.54	118.64	104.55
8E2A	7480	0.00	1117.36	57.71	104.35	115.37
91B8	2753	9.62	1145.77	81.67	125.61	120.51
97D7	7001	0.00	507.83	54.67	89.30	83.99
A838	4429	4.08	827.67	99.08	143.51	128.52
A9BE	14436	1.50	1110.75	64.00	104.61	101.38
BB4A	4407	0.00	486.95	50.75	87.52	88.03
BC46	11223	4.17	1142.92	69.50	114.94	114.54
BFDC	859	5.33	371.08	56.33	75.10	61.09
C0A7	10246	1.42	696.83	62.17	97.26	90.50
CBC7	10952	1.42	915.73	62.67	95.30	87.14
D804	7220	2.08	563.00	54.69	88.01	86.25
DF07	6962	0.00	507.83	54.91	89.91	84.22
E1F8	4721	3.00	481.50	72.83	105.27	92.66
E47A	1009	13.00	274.67	63.08	72.65	41.49
E486	12058	1.00	954.82	77.58	111.79	100.88
E8E4	4280	7.83	1205.50	98.58	145.60	127.31
EAC8	12652	0.33	836.42	63.74	100.88	94.85

Table 3. Summary statistics of low-cost pollution sensor network

an LSTM unit takes as input the final hidden state $h_{v,t}$ and predicts the next hidden state $h_{v,t+1}$. The final phase is the readout operation R_v, which decodes the hidden state to produce the output value to be predicted $\hat{y}_{v,t+1}$. These five steps are shown below. The message function takes as input the hidden states of a pair of nodes v and n and the Euclidean distance between them, $d_{v,n}$ as the influence of the pollution at a given location on the pollution at another location would depend on the distance between them. Hence we include the distance in the embedding.

$$h_{v,t} = O_v(h_{v,t-1}, y_{v,t}) \tag{2}$$

$$m_{v,t} = \sum_{n \in V-v} M(h_{v,t}, h_{n,t}, d_{v,n}) \tag{3}$$

$$h_{v,t} = U(h_{v,t}, m_{v,t}) \tag{4}$$

$$h_{v,t+1} = T_v(h_{v,t}) \tag{5}$$

$$\hat{y}_{v,t+1} = R_v(h_{v,t+1}) \tag{6}$$

For a selection of nodes $\mathcal{W}$ in the graph, the components of the model $\{O_w, M, U, T_w, R_w, |w \in \mathcal{W}\}$ are defined. During inference, the states $H_t = \{h_{w,t}|w \in \mathcal{W}\}$ are maintained at each timestep. The hidden state for each segment is initialized at $t = 0$ randomly during training and evaluation $h_{v,0} \sim \mathcal{N}(0, 1)$.

A.4 Estimation of Residuals

The residual errors from the above MPRNN model is then fit based on the daily spatio-temporal patterns per sensor and per location. For example, if our prediction error follows a temporal pattern of say, higher prediction error in the morning, while lower in the afternoon, we can leverage this by fitting piecewise polynomial function called the spline on the residual errors. This spline can be of any order, but given our residual error patterns, we saw that a 3-way piecewise cubic spline works best. Given that at each timestamp t, after applying our MPRNN model's predictions, let the residual raw error be given by $\epsilon(v, t)$

$$\epsilon(v, t) = y_{v,t} - \hat{y}_{v,t} \tag{7}$$

This residual error can then be predicted by our residual spline model as follows using a piece-wise spline for a sensor v and time-period p:

$$\hat{\epsilon}_p(v, t) = \alpha_{v,p} * t^3 + \beta_{v,p} * t^2 + \kappa_{v,p} * t + \nu_{v,p} \tag{8}$$

Note that the chosen parameters per sensor $\alpha_{v,p}, \beta_{v,p}, \kappa_{v,p}, \nu_{v,p}$, where $p \in$ {"morning", "afternoon", "evening"}, depend on the patterns in our residual errors and are fit accordingly to minimize the root mean squared residual error:

$$RMSE(v) = \sum_t \sum_p \sqrt{(\epsilon(v, t) - \hat{\epsilon}_p(v, t))^2} \tag{9}$$

To check the resilience of these per-sensor splines, we also compute an average spline across all available sensor residual errors over the training data, by marginalizing over the sensors. Through our study, we show how per-sensor residual splines vary across geographies and how the average spline can sufficiently operate for bootstrapping or regions where we do not have enough sensor data to begin with. Not only are the sensor splines different across regions, we do see that regions with significantly high spline residual errors like the sensors A838, E8E4, 2E9C in Fig. 4a, are all located in central locations of Delhi with well established commercial activity like Connaught Place, Sardarjung Enclave and Lado Sarai respectively. Further, in Fig. 4b, the outliers with significantly high residual error splines among the government monitoring stations are Patparganj DPCC, PunjabiBagh DPCC and DKSSR DPCC. While Patparganj is situated next to an industrial area, Punjabi Bagh is a well known residential locality with established commercial activity centers and DKSSR is a shooting range located in the outskirts of Delhi next to an inter-state highway. The diversity of these splines across various

geographical regions further indicate the need to model fine-grained pollution profiles in seemingly remote as well as central locations of Delhi.

A.5 History length and Network Size

While a recurrent neural network is capable of predicting labels on a rolling basis, computing and backpropagating a loss function through an arbitrarily long history is not feasible. As a result, our recurrent neural network is trained on segments of fixed history length. That is, the state of the neural network is persisted through a number of points equal to the history length, and then reset. We experimented with several different reasonable history lengths during training and chose a history of 8 hours (32 measurements, at one reading per 15 minutes). Each training example consisted of a block of history length $H = 32$ collected at time step t for a chosen set of sensors except for the target sensor. Further, when we forecast by leaving out history of the past 24 hours and rely on previous day's data we see that the forecasting error increases by up to 18.3%, 19.5% and 17.2% MAPE error in our sensors, government monitors and the combined sensor network respectively.

Further, we see that the number of sensors we augment in addition to the government monitors reduces the overall MAPE for the size of the network chosen. We report the mean of 10 samples of networks at a given size along with the standard deviations in Figure 5.

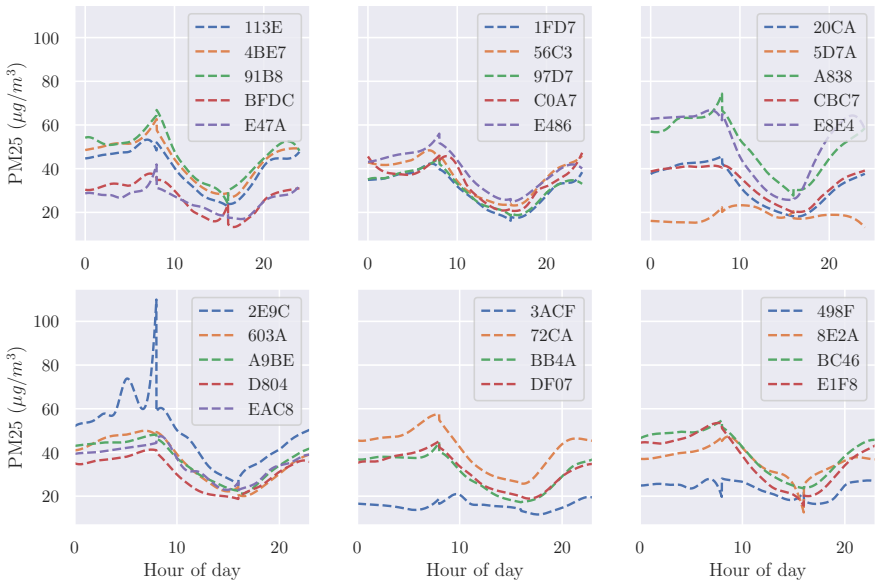
A.6 Training

Since neural networks take only fixed length input feature vectors, our formulation required the training of several models, one for each value of K. We trained a total of 10 models, for K from 1 to 10. By combining these 10 different models, we obtain a *master model*, that generalizes to predicting $y_{v,t}$ at any location v at a given time, regardless of the number of available neighboring input sensors at the time, using data from up to a maximum of 10 available input sensors. For each value of K, and for each sensor v in our set, we extracted blocks of available data of length $H = 32$ through the entire year from the K nearest neighbors to v. Then we merged all the blocks together for each version and each value of K, thus giving us a large dataset of training samples mapping K sensor readings to output sensor values over the history length H (i.e. a sample consisted of a block of dimensions $H \times K$ or $H \times 3K$ depending on the version). The list of samples was then shuffled prior to feeding into the neural network to reduce the chances of the optimization algorithm becoming stuck in local optima and and also increase the test prediction performance. We repeat this mechanism for every value of K, giving us totally 10 models for each version.

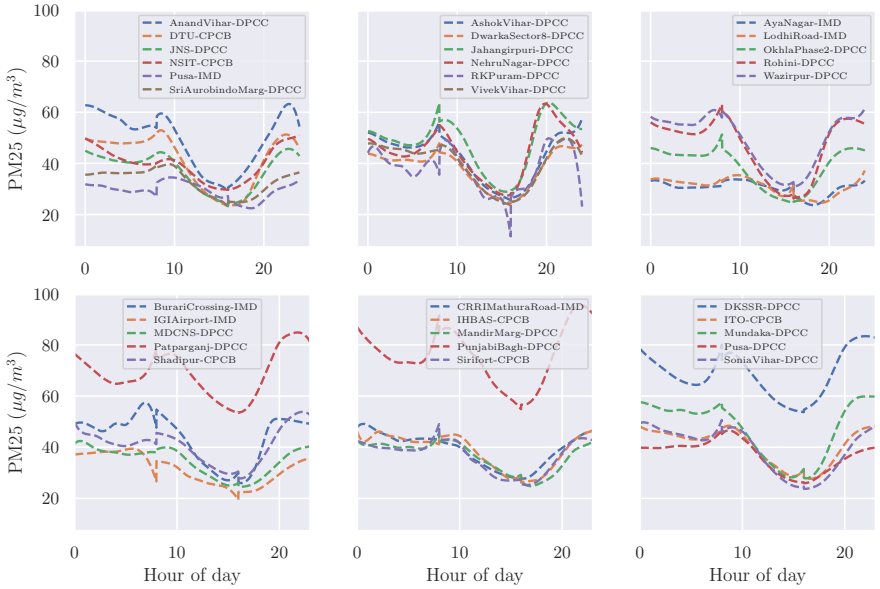
We needed at least 20 epochs for convergence. With a total of at least 2000 batches for every value of K, the training for each K and version took prohibitively large amount of time (several days). Hence we resorted to reduction of training time by selecting only a smaller number of samples from the entire corpus of samples for each K and version The shuffling of training samples thus allowed us to "effectively" reduce our training time and yet not lose generality since the shuffling ensured that data from all round the year was utilized for training. We used the `pytorch` [1] library in Python for implementing the neural network and the Adam optimizer [20] for training.

A.7 Baselines

Nearest Neighbor Spatial Neural Network: We contrast the MPRNN with a more simplified neural network model in which messages are not explicitly passed between pairs of nodes, but rather the sensor readings from a set of neighboring monitors to a location v at time t are directly used as input. Each node runs a neural network with an LSTM unit for predicting future values, similar to the MPRNN. The pairwise distances and relative positions are encoded in the feature vector along with the input sensor readings. At each node, only a certain number of closest neighbors are used as input to model the air quality at that location, in contrast to the MPRNN



(a) Splines for each of the 28 sensors



(b) Splines for each of the CPCB Government pollution monitors

Fig. 4. The daily variations in the splines learnt for each of the sensors show that there are temporal patterns which when incorporated into a prediction model can significantly improve prediction accuracy

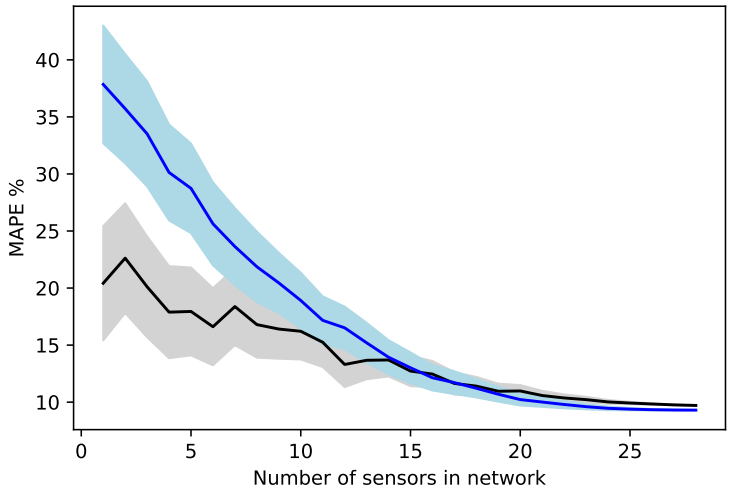


Fig. 5. Impact of sensor network size on forecasting error

where all the other sensors are used. We call this model the *K-Nearest Neighbor (K-NN) Spatial Neural Network* since the input for the prediction task at a location v is the set of sensory readings from K nearest monitors to v and their relative locations in terms of distances and positions, where K is a positive number.

The equation denoting the function to approximate is similar to that for the MPRNN, except that we restrict the number of input sensor locations to K, where K is a parameter: $y_{v,t} = \mathcal{F}(v_1, y_{v_1,t}, v_2, y_{v_2,t}, \ldots, v_K, y_{v_K,t}))$ In this equation, v_j denotes the j^{th} nearest neighbor to v. K is the maximum number of neighboring sensors that can provide input sensory data. Note that the set of nearest neighbors $v_1, v_2, \ldots v_K$ and K itself are functions of time, since at time t, these are the sensors at which data is available and the number of such sensors, respectively. For each input sensor at location v_j, we add as feature the triple of the sensor reading, the geodesic distance between v and v_j, the compass bearing of v_j with respect to v. The length of the feature vector is thus $3K$, where K is th number of available sensors at that time.

Spatio-temporal Hierarchical Model: The Spatio-Temporal Hierarchical Model (STHM) is a statistical modeling framework from geostatistics. It combines various sources of information, accommodates missing values and computes predictions in both space and time. This statistical model is hierarchical in that it distinguishes between observed variables, such as the actual PM measurements, and underlying processes that are not directly observed. In a state space terminology the latter are known as unobserved states, while in some statistics literature they are known as random effects or latent variables (see e.g. Harvey [14]). The hierarchy is explicit since the model is defined through multiple levels of equations, where a higher level typically involves variables conditioned on the variables defined at deeper levels. This allows the important identification of two sources of error: measurement error which applies to the observations, usually at the highest hierarchical level, and process error which enters the specification of the dynamics of the underlying processes at deeper levels. We refer to Cressie and Wikle [8] for details about the relevance of this hierarchical framework for spatio-temporal modeling, how it is currently considered state-of-the-art, and for links to the geostatistics literature. In particular, optimal spatial prediction (often referred to as Kriging) pertains to the prediction of underlying processes and not of the noisy measurements and is addressed below.

Successful recent applications of such hierarchical models for the spatio-temporal modeling of air pollution include [4], [5] and [2]. Our model proposed below generally follows these references for the dynamics in space and time of an underlying specified random field modeled as a Gaussian process.

Notation: Let $\mathcal{D} \subseteq \mathbb{R}^2$ denote the spatial domain of interest and $Y(\boldsymbol{s}, t)$ denote the $PM_{2.5}$ concentration in $\mu g/m^3$ measured at location $\boldsymbol{s}$ and time t. The location vector $\boldsymbol{s} = (s_1, s_2)^\top \in \mathcal{D} \subseteq \mathbb{R}^2$ consists of geographical coordinates s_1 and s_2 in the plane following a map projection, such as easting or northing in km according to the Universal Transverse Mercator (UTM) coordinate system, so that a notion of distance $h(\boldsymbol{s}, \boldsymbol{s}') \in \mathbb{R}$ can be unequivocally defined. Following Cameletti et al. [4, 5], the top hierarchical level of our STHM is specified by the following measurement equation.

$$\log Y(\boldsymbol{s}, t) = \boldsymbol{z}(\boldsymbol{s}, t)^\top \boldsymbol{\beta} + \sum_{j=1}^{J} \alpha_j B_j(t) + X(\boldsymbol{s}, t) + \epsilon(\boldsymbol{s}, t), \tag{10}$$

Here $t = 1, 2, \ldots$ is a discrete representation of timestamps (regardless of the actual temporal resolution of the data), $\boldsymbol{z}(\boldsymbol{s}, t)$ is a p-vector of covariates which defines deterministic (fixed) effects along with the corresponding coefficient $\boldsymbol{\beta}$, $B_j(t)$ for $j = 1, \ldots, J$ is a set of specified (periodic) basis functions used to model seasonality effects along with the corresponding basis coefficients α_j, $X(\boldsymbol{s}, t)$ is a mean zero Gaussian process whose dependence structure in space and time is specified at the second hierarchical level, and the $\epsilon(\boldsymbol{s}, t)$'s are measurement error terms assumed independent and identically distributed as Gaussian with mean zero and constant variance σ_ϵ^2 (the latter known as the "nugget" effect in the geostatistics literature). As a result, the $\log Y(\boldsymbol{s}, t)$'s are independent conditionally on $X(\boldsymbol{s}, t)$, for all $\boldsymbol{s} \in \mathcal{D}$. The modeling on the natural logarithm scale ensures the positivity of Y. We note that although deterministic effects enter as a linear combination, any auxiliary information can be part of $\boldsymbol{z}(\boldsymbol{s}, t)$. For instance, outputs from a dispersion model predicting the propagation of fine particles from various environmental inputs would enter the STHM through $\boldsymbol{z}(\boldsymbol{s}, t)^\top \boldsymbol{\beta}$. In the application of this model to our Delhi sensor data, we however place ourselves in a data-poor situation with no extra information other than the air pollution measurements themselves, so that no auxiliary deterministic effects are estimated and $\boldsymbol{z}(\boldsymbol{s}, t)^\top \boldsymbol{\beta} = 0$ for all observations. We model daily seasonality with $J = 6$ quadratic B-spline bases over four disjoint time intervals: [00:00–06:00), [06:00-12:00), [12:00–18:00) and [18:00–00:00). This implies $J - 1 = 5$ fixed knots to facilitate interpretation of periodic patterns throughout the day. The corresponding α_j coefficients are estimated from the data but only $J - 2 = 4$ are free since we enforce two constraints for the continuity and differentiability of the resulting linear combination at the boundary at midnight. The intercept term (constant mean level) is included in the B-splines linear combination.

The second hierarchical levels describes the temporal dynamics and spatial dependence structure of the underlying stochastic process X. The process equation describes a stationary autoregressive (AR) process of first order through time:

$$X(\boldsymbol{s}, t) = \phi X(\boldsymbol{s}, t-1) + \delta(\boldsymbol{s}, t), \tag{11}$$

for $t = 1, 2, \ldots$ and $\boldsymbol{s} \in \mathcal{D}$, where the constraint on the AR coefficient $|\phi| < 1$ ensures stationarity, and the process error δ is distributed as Gaussian with expectation zero. The δ terms are temporally independent but spatially dependent:

$$\mathrm{Cov}[\delta(\boldsymbol{s}, t), \delta(\boldsymbol{s}', t')] = \begin{cases} 0 & t \neq t' \\ C(h(\boldsymbol{s}, \boldsymbol{s}'); \gamma, \sigma_\delta) & t = t', \end{cases}$$

where C is a (positive-definite) spatial covariance function; we set it to the stationary and isotropic exponential spatial covariance function for simplicity:

$$C(h(\boldsymbol{s}, \boldsymbol{s}'); \gamma, \sigma_\delta) = \sigma_\delta^2 \exp(-h(\boldsymbol{s}, \boldsymbol{s}')/\gamma),$$

where $h(\boldsymbol{s}, \boldsymbol{s}') = \sqrt{(s_1 - s_1')^2 + (s_2 - s_2')^2}$ is the Euclidean distance between locations $\boldsymbol{s}$ and $\boldsymbol{s}'$, σ_δ^2 is the process variance for $h(\boldsymbol{s}, \boldsymbol{s}') = 0$, and γ regulates the steepness of the exponential decay of the covariance with increasing distance. The initial states $X(\boldsymbol{s}, 0)$ follow the stationary distribution, i.e. a Gaussian distribution with mean zero and covariance matrix given by $C(h(\boldsymbol{s}, \boldsymbol{s}'); \gamma, \sigma_\delta)/(1 - \phi^2)$ for $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{D}$.

Overall, this STHM involves $(p + 8)$ fixed parameters,

$$\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \sigma_\epsilon^2, \phi, \gamma, \sigma_\delta^2)^\top,$$

while the dynamic spatial field X will be predicted given an estimate of $\boldsymbol{\theta}$. With the Delhi data, there is no $\boldsymbol{\beta}$ to estimate so that $p = 0$ and only 8 parameters are estimated.

Interpolation Given $\hat{\boldsymbol{\theta}}$, we predict the underlying dynamic spatial field by maximizing the joint log-likelihood

$$\hat{\boldsymbol{X}}(0), \ldots, \hat{\boldsymbol{X}}(T) = \arg\max_{\boldsymbol{x}} \log L(\hat{\boldsymbol{\theta}}; \boldsymbol{y}, \boldsymbol{x}). \tag{12}$$

This maximization can also be invoked for out-of-sample prediction, either for a new location, for forecasting in time, or for both simultaneously. That is, the $PM_{2.5}$ forecasting at time $t + 1$ for a new location $\boldsymbol{s}'$ is given by $\exp\left(\boldsymbol{z}(\boldsymbol{s}', t+1)^\top \hat{\boldsymbol{\beta}} + \sum_{j=1}^J \hat{\alpha}_j B_j(t) + \hat{X}(\boldsymbol{s}', t+1)\right)$, where $\hat{X}(\boldsymbol{s}', t+1)$ is indeed obtained by maximizing the joint log-likelihood for given parameter estimates. To be precise, such a forecast is not a prediction for the noisy measurement $Y(\boldsymbol{s}, t)$, but really a prediction of the true underlying $PM_{2.5}$ concentration represented by $\exp\left(\boldsymbol{z}(\boldsymbol{s}', t+1)^\top \boldsymbol{\beta} + \sum_{j=1}^J \alpha_j B_j(t) + X(\boldsymbol{s}', t+1)\right)$. Our predictions can thus be seen as Bayesian posterior modes, while spatial prediction by Kriging typically corresponds to posterior expectation, see Chapter 4 of Cressie and Wikle [8] for a discussion.

For a given time point t, plotting the predicted $PM_{2.5}$ concentrations as a smooth map by simple interpolation over the n measured locations likely gives rise to visual artifacts and distortions if either n is too small or if the measured locations are not spread evenly over $\mathcal{D}$. Such artifacts happen with clusters and empty spaces, as in the application to the Delhi data. The STHM provides a natural way to "fill-in" the spatial domain $\mathcal{D}$ with predictions at extra locations according to equation (12). Regarding the choice of extra locations, rather than constructing an inefficient regular grid of points, we follow here Lindgren et al. [22] by using a constrained Delaunay triangulation as implemented in the R package Integrated Nested Laplace Approximation [INLA; 30]. This triangulation method allows us to tessellate $\mathcal{D}$ with triangles such that their minimum interior angle is maximized under the constraint that measured locations correspond to vertices. This (constrained) maximin property ensures that the density of vertices somewhat follows the density of measured locations (i.e. more smaller triangles where sensors are clustered) while maintaining an even spread in empty areas, including beyond the convex hull of all measured locations. The predictions at these extra locations can be integrated within the fitting of the model since they are equivalent to missing values in $\boldsymbol{y}$ (locations where no observation is available).