AIRDELHI: Fine-Grained Spatio-Temporal Particulate Matter Dataset From Delhi For ML based Modeling

Sachin Kumar Chauhan, Sayan Ranu, Rijurekha Sen
Department of Computer Science
IIT Delhi
{csz188012, sayanranu, riju}@cse.iitd.ac.in

Zeel B Patel, Nipun Batra
Department of Computer Science
IIT Gandhinagar
{patel_zeel, nipun.batra}@iitgn.ac.in

Abstract

Air pollution poses serious health concerns in developing countries, such as India, necessitating large-scale measurement for correlation analysis, policy recommendations, and informed decision-making. However, fine-grained data collection is costly. Specifically, static sensors for pollution measurement cost several thousand dollars per unit, leading to inadequate deployment and coverage. To complement the existing sparse static sensor network, we propose a mobile sensor network utilizing lower-cost $PM_{2.5}$ sensors mounted on public buses in the Delhi-NCR region of India. Through this exercise, we introduce a novel dataset AIRDELHI comprising $PM_{2.5}$ and PM_{10} measurements. This dataset is made publicly available at *https://www.cse.iitd.ac.in/pollutiondata*, serving as a valuable resource for machine learning (ML) researchers and environmentalists. We present three key contributions with the release of this dataset. Firstly, through in-depth statistical analysis, we demonstrate that the released dataset significantly differs from existing pollution datasets, highlighting its uniqueness and potential for new insights. Secondly, the dataset quality been validated against existing expensive sensors. Thirdly, we conduct a benchmarking exercise (*https://github.com/sachin-iitd/DelhiPMDatasetBenchmark*), evaluating state-of-the-art methods for interpolation, feature imputation, and forecasting on this dataset, which is the largest publicly available PM dataset to date. The results of the benchmarking exercise underscore the substantial disparities in accuracy between the proposed dataset and other publicly available datasets. This finding highlights the complexity and richness of our dataset, emphasizing its value for advancing research in the field of air pollution.

1 Introduction

United Nations Sustainable Development Goals [UN, 2015], especially the Goal-11 Sustainable Cities and Communities, is a primary research focus in institutions of developing countries like India [IIITD, 2023], with related research works in sustainable transport [Chauhan *et al.*, 2020] and pollution [Shukla *et al.*, 2020; Bikkina *et al.*, 2019; Iyer *et al.*, 2022; Abidi *et al.*, 2022].

Air pollution has now reached life-threatening levels in Delhi-National Capital Region (NCR), India [Tripathi *et al.*, 2019; Mannucci and Franchini, 2017], which is one of the most densely populated urban centers. The population of Delhi-NCR exceeds 46 million people [Nagar *et al.*, 2017] and it has been reported that 50% of all children staying in this region suffer from irreversible lung damage [Chatterji, 2021; ORF, 2021]. *Particulate Matter (PM)* is especially dangerous, since

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

our breathing cannot filter out the ultra-fine particles. To mitigate the effects of air pollution, there is an urgent need to identify causes of pollution and strategies to curb its spread. [Sahu *et al.*, 2020; Sutaria, 2022] suggests to use one sensor per km^2 for better pollution analysis. The *Central Pollution Control Board (CPCB)* and *Delhi Pollution Control Committee (DPCC)* have only 81 realtime air pollution measurement centers in Delhi-NCR Sutaria [2022] with 65 manually monitored centers, which are thoroughly inadequate [Guttikunda *et al.*, 2023; ET, 2022] to cover the vast geography of $55,000$ km^2 [NCRPB, 2018].

In the literature, several models have been proposed for predicting pollution levels at same/future time points [Patel *et al.*, 2022; Gao and Li, 2021; Kurt *et al.*, 2008; Tsai *et al.*, 2018; Le *et al.*, 2020], and identifying factors affecting pollution [Apte *et al.*, 2011; Google, 2014; Messier *et al.*, 2018; Apte *et al.*, 2017; Alexeeff *et al.*, 2018]. There exists *interpolation models* [Qiao *et al.*, 2019; Ras and Williams, 2005; Hamilton *et al.*, 2017; Patel *et al.*, 2022] to reliably predict pollution levels at unseen locations based on a sufficient number of pre-installed sensors. These models can improve with fine-grained pollution data. The interpolation and forecasting models are *supervised* in nature and hence can do better with more training data. Unfortunately, collecting pollution data using realtime centers is highly expensive as each instrument costs thousands of US Dollars.

In this work, we aim to mitigate the problem of lack of sufficient data in a cost-effective manner. We design a low-cost sensing mechanism (thoroughly compared in quality against high cost sensors) that allows us to collect PM data over a subset of the Delhi-NCR region at a fine spatio-temporal granularity. The key highlights and contributions of our work are:

1. Quality dataset: As it is not cost-effective to repeat even the low cost sensors per km^2, we establish a low-cost vehicle-mounted PM sensing network and release the largest $PM_{2.5}$ dataset from one of the most polluted regions in the world. This dataset is shown to be as good as the data collected from the few high-cost static-sensor deployed in the same region. As it is very challenging to collect such dataset in a developing country due to constraints in infrastructure and government permissions, we document our data collection experience briefly in the paper. (§ 3.2).

2. Unique dataset: This dataset complements the static sensor data available from the government deployed instruments in important ways. The static sensors are located at the top of high towers to get precise recordings of ambient pollution values, not affected by local sources. Our mobile sensors, on the other hand, are installed in the bus driver's cabin to measures the ground level pollution that daily commuters breathe in. We also perform a thorough comparison with PM datasets available from other parts of the world and establish that the released dataset is unique in terms of scale and statistical characteristics. Hence, it can be of immense value to environmental think tanks. (§ 3.3).

3. Utility for ML modeling: Through extensive benchmarking using state-of-the-art Machine Learning (ML) algorithms, we demonstrate the utility of this new dataset for modeling problems using ML, like spatio-temporal interpolation, missing data imputation and forecasting. The dataset is shown to be more challenging to model with ML algorithms, compared to previously available datasets, as Delhi has much higher variance in PM across space and time. This dataset, therefore opens opportunities for ML researchers for designing and benchmarking new ML algorithms, to reduce the interpolation, missing data imputation or forecasting errors. (§ 4).

2 Related Work

A primer for Air Pollution Monitoring is available at Urbanemissions [2023]. Spatio-temporal (ST) interpolation involves predicting air quality at unmonitored locations in the past and/or present time using training data observed from the sensors during the past and present time. Zheng *et al.* [2013] developed a co-training-based approach for ST interpolation using $PM_{2.5}$ values captured every hour from ground stations of 4 cities in China which are converted to AQI (Air Quality Index), along with meteorological and traffic data. Cheng *et al.* [2018] proposed an attention-based hybrid model involving LSTM and dense layers and Patel *et al.* [2022] proposed a domain-inspired non-stationary Gaussian process model for ST interpolation which can also be used for ST forecasting. The two used 36 monitoring stations in Beijing with the collection time interval of 1 hour (with the latter additionally using London data), alongside meteorological data.

Missing data imputation problem can be considered a variation of spatio-temporal interpolation where observations on the spatio-temporal cube are missing at random and we want to impute the missing data. Models that work for ST interpolation can mostly be adapted readily for this problem.

Spatio-temporal forecasting aims to predict air quality at a particular location in future using the past and current data available at all the installed sensors. Kurt *et al.* [2008] developed an online neural network based approach to predict air quality maximum 3 days ahead in time using 1 year PM_{10} data

for 1 region in Turkey. Zheng *et al.* [2015] develop and deploy a machine learning based air quality forecasting system with the Chinese Ministry of Environmental Protection. Yi *et al.* [2018] develop a deep learning based approach to provide short-term and long-term air quality forecasts. The two used meteorological data along with pollution data generated every hour from 2,296 stations in 302 Chinese cities, and converted these concentrations into corresponding (individual) AQIs according to Chinese AQI standards. Air quality forecasting was posed as a challenge in KDD2018, where Luo *et al.* [2019] presented a winning solution based on a combination of classical machine learning and deep learning models using the provided data from stations in Beijing and London. Gao and Li [2021] propose a graph-based LSTM model for air quality forecasting and evaluate on Northwest China hourly data from 32 china stations. All these prior arts utilize the static ground stations Air Quality data for the analysis, which enforces a restricted spatial coverage. They also use meteorological data from the respective regions. Bhattacharyya *et al.* [2022] provides a similar USA AQI dataset collected from Air Quality Open Data Platform.

There also have been studies on low cost sensors available in market for developed (EU) regions Karagulian *et al.* [2019]. Also, projects about installing low cost sensors at different roadside locations, like Schneider *et al.* [2023] and Iyer *et al.* [2022], to complement the existing expensive static sensor network are done recently, but they kept the sensors at fixed locations. Vehicle mounted air pollution sensing has also been conducted [Apte *et al.*, 2011; Google, 2014; Apte *et al.*, 2017; Alexeeff *et al.*, 2018; Guo *et al.*, 2016; Adams and Corr, 2019; Li *et al.*, 2012], with certain limitations. Abidi *et al.* [2022] used similar low-cost sensors to analyze the relation of PM with static (green cover, buildup, commercial, residential) and dynamic (meteorological, traffic) factors, particularly at traffic intersections with odd-even policy in Delhi. As per Yi *et al.* [2015]; Pavani and Rao [2017], static sensors are better in data quality, endurance and temporal resolution, a low-cost static sensor is better in temporal resolution which can provide (static) pollution maps with better temporal resolution for the installed locations. But the effective static pollution maps require careful placement of sensor nodes in large quantity leading to resource wastage. Mobile sensors are better in mobility, geographic coverage, maintenance, cost-efficiency which can provide (dynamic) pollution maps with greater geographic coverage, with limitation of communication overheads and redundant sampling. In contrast, we are working with the PM data, collected with mobile sensors deployed on frequent bus routes, which is fine-grained and provides better spatio-temporal coverage, and our benchmarked models do not rely on meteorological factors.

3 Dataset Description

3.1 Dataset Collection Challenges

Creating the mobile PM dataset (as a replacement for low cost static PM dataset and high-cost ground station PM dataset) required us to design and implement our own embedded platform, choosing and calibrating appropriate sensors for maximum accuracy at low cost. The complete design is presented in Goyal *et al.* [2018]. We opted to install our device in public buses, to utilize their pre-defined/fixed and frequent routes of travel. Packaging was challenging to securely mount the instruments in the public buses, avoiding theft and ensuring enough ambient air to measure PM. Cellular connectivity was intermittent as the buses traversed the city, requiring us to augment real time data transfer when signal was present, with local storage to save data when signal strength dropped. Finally, getting permissions from different government entities to instrument the public bus fleet needed strict safety certifications that our devices do not interfere with the electrical and mechanical functioning of bus.

We mounted pollution tracking sensors on the permissible 13 public buses in Delhi for 3 months (Nov 1, 2020 to Jan 31, 2021), in collaboration with Delhi Integrated Multimodal Transport System, after rigorous tests for automotive safety certification and appropriate permissions and letters of support from the Delhi Ministry of Transport and Delhi Pollution Control Committee (provided in Appendix **??**). The Covid'19 restrictions were relaxed by this time, limiting to containment zones at local level only [MHA, 2020]. So, the impact of the COVID'19 on the collected dataset is expected to be minimal, if not none.

As discussed in Goyal *et al.* [2018], the inside of our custom-made instrument comprising *(a)* PM sensor measuring $PM_{2.5}$, PM_{10} and PM_1, *(b)* GPS sensor to locate the bus, *(c)* 4G radio to communicate data from bus to server, *(d)* SD card for locally storing data when 4G signal is unavailable, *(e)* BME sensor [BME, 2023], a sensor especially developed for mobile applications and wearables, to record temperature and relative humidity and *(f)* micro-controller to orchestrate the sense-store-communicate software (See Fig. 1a). The mounting location in the bus driver's cabin, next to two open windows to allow enough air-flow (Fig. 1b-1c). Each bus commutes for

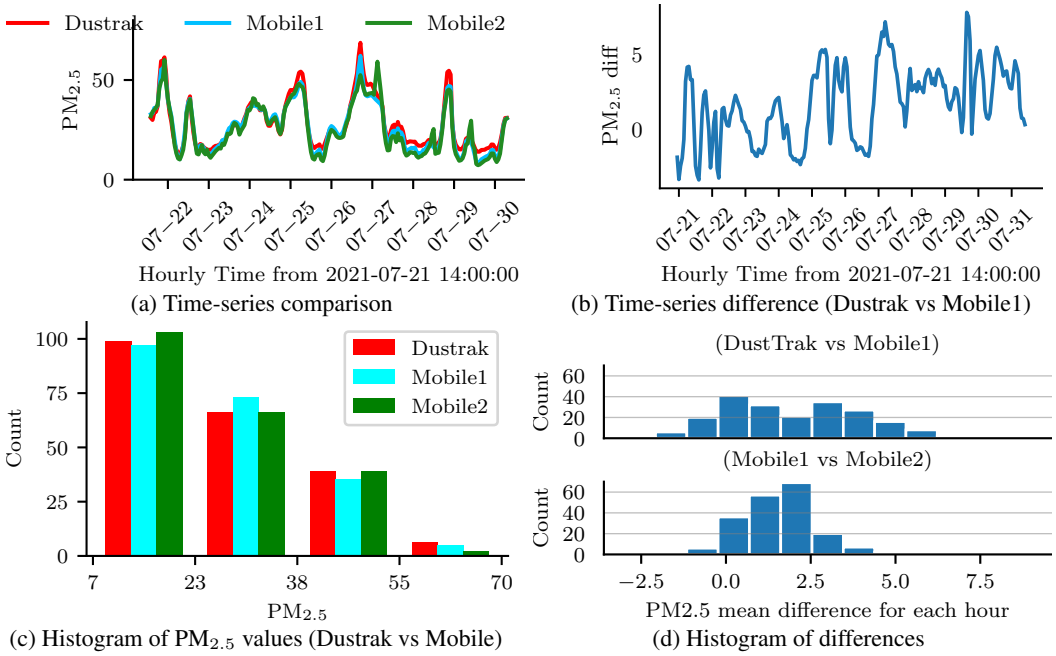
(a) Measuring device (b) Mounting location (c) Mounted device (d) Bus trajectories

Figure 1: (a) Inside of our PM measuring IoT unit. (b) Mounting location in bus driver's cabin in non air-conditioned public bus (below the existing white box). (c) Mounted IoT unit in the bus (below the existing white box). (d) Government deployed static sensors installed in and around our bus trajectories, as location icons.

16-20 hours per day, and our instruments collect data at a fine granularity of 20 samples per minute. Overall, the bus trajectories cover 559 km^2, along the main arterial roads in North-West, North, North-East and South-East Delhi (Fig. 1d). The dataset, having 3 pollution parameters: PM_1, $PM_{2.5}$ and PM_{10} with 7 non-pollution parameters: latitude, longitude, time, deviceId, pressure, temperature, relative humidity, has been made available at `https://www.cse.iitd.ac.in/pollutiondata/` and `https://huggingface.co/sachin-iitd/DelhiPollDataset` with proper documentation, under a Creative Commons Attribution 4.0 International License [CC-by4, 2013].

3.2 Data Quality Analysis

Fig. 2a plots $PM_{2.5}$ values measured by two low cost PM sensors built by us (cost USD 30), and the same measured by an industry grade reference instrument TSI DustTrak (cost USD 9500), while all three instruments are placed close to each other. The plot shows hours of the day along x-axis and sensed $PM_{2.5}$ values along y-axis, for 10 sample days Jul 21-31, 2021. This is after the deployment of the low cost sensors in the buses is over, and the sensors have been brought back to the lab.



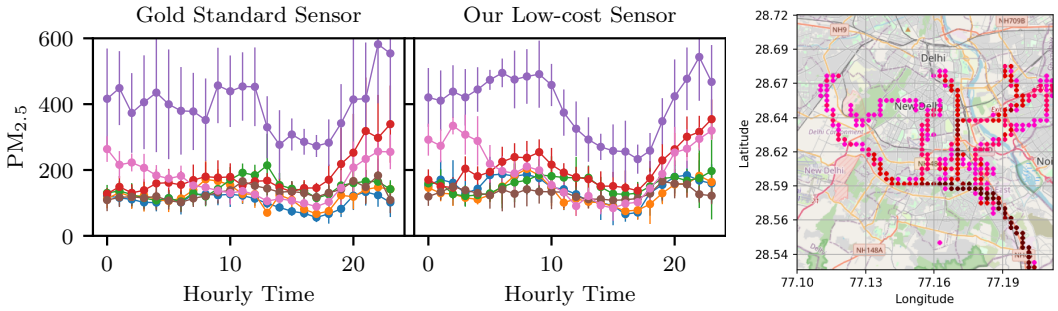
(a) Time-series comparison (b) Time-series difference (Dustrak vs Mobile1)

(c) Histogram of $PM_{2.5}$ values (Dustrak vs Mobile) (d) Histogram of differences

Figure 2: (a) $PM_{2.5}$ values measured by our low-cost mobile PM sensors (USD 30) vs. TSI DustTrak (USD 9500) between Jul 21-31, 2021. (b) Mean Difference and (d) Histogram of pointwise differences of $PM_{2.5}$ values measured by DustTrak and low cost mobile PM sensors. (c) Histogram of $PM_{2.5}$ in the intervals on x-axis. The values are almost identical.

Fig. 2b shows the difference of hourly mean $PM_{2.5}$ between DustTrak and one mobile sensor, the mean difference is 6.16%. Also, Fig. 2c shows the histograms of hourly mean $PM_{2.5}$ for the shown $PM_{2.5}$ intervals for the three devices. Fig. 2d shows the histogram of difference of hourly mean $PM_{2.5}$ between DustTrak and one mobile sensor, and two mobile sensors, for the same 10 days. Also, Fig. 9 in Appendix C shows the similar mean and standard deviation between the 3 devices. While the cost gap between the instruments is huge, the gap between their sensed $PM_{2.5}$ values, as seen in the plots, is negligible. This pattern has been observed consistently by us and other researchers [Zheng *et al.*, 2018; Cheng *et al.*, 2014; Gao *et al.*, 2015; Rai *et al.*, 2017; Jiao *et al.*, 2016; Zheng *et al.*, 2019].

We additionally compare the distribution of $PM_{2.5}$ values recorded by our mobile sensors vs. those by the high-cost static sensors, deployed at sparse locations by CPCB and DPCC in Delhi-NCR. Fig. 3a(Left) shows hours of day along x-axis and average $PM_{2.5}$ for that hour, as measured by reference grade static monitors, with standard-deviation bars along y-axis. Fig. 3a(Right) shows the same averaged over all bus mounted sensors. We select the static sensors that are within 1km of mobile sensor trajectory for each hour, and plot for 7 sample days. Fig. 3a reveal that both static and bus mounted sensors show similar PM distributions for each day, in spite of the difference in heights they have been installed at, and the difference in PM measurement technique. We see this agreement for the entire 3 months deployment period. The agreement between low cost mobile sensors, and a co-located high cost TSI Dusttrak, as well as reference grade static monitors, give us confidence to release the dataset to the research community.



(a) Static v/s Mobile Sensors Data (b) Heat Map

Figure 3: (a) Distribution of $PM_{2.5}$ collected by our low-cost sensor and gold standard sensor over 7 random days. The distributions are similar across the two sets of instruments. (b) Heat Map for all bus routes for Dec 15, 2020 (darker locations contain more samples).

Heat Map: During our analysis, we discovered variations in data availability across different timestamps and spatial locations. It was evident that certain timestamps were not available at all spatial locations. Furthermore, some spatial locations, which were situated along routes with fewer bus visits throughout the day, exhibited limited temporal samples. As illustrated in Fig. 3b, a typical day (Dec 15, 2020) demonstrated this pattern, where the outermost locations (depicted in light/pink color) contained samples from 4 hours duration within the 16.5-hour effective temporal window. Conversely, the darker/brown locations near the bottom right of the figure displayed a higher number of samples, ranging from 14 to 16.5 hours. These locations are associated with common bus routes that connect with the depot.

3.3 Dataset Novelty

Tables 1 and 2 summarize the statistics of the dataset. While vehicle mounted air pollution sensing has been conducted [Apte *et al.*, 2011; Google, 2014; Apte *et al.*, 2017; Alexeeff *et al.*, 2018; Guo *et al.*, 2016; Adams and Corr, 2019; Li *et al.*, 2012], our dataset is unique in characteristics and scale. Specifically, only two studies from Ontario, Canada [Adams and Corr, 2019] and Zurich, Switzerland [Li *et al.*, 2012] have made their datasets publicly available. The Zurich dataset does not include PM values. Compared to the Canada dataset, our dataset is 1000 times larger and has a significantly different distribution of PM values (See Tables 1 and 2).

This is understandable as Delhi-NCR is an air pollution hotspot, whereas Zurich and Ontario have negligible PM levels. We also compare our dataset with a recent USA AQI dataset Bhattacharyya *et al.* [2022] collected from Air Quality Open Data Platform. We also purchased Wind Speed (WS) data

Table 1: Details of Delhi, India and Hamilton, Ontario, Canada and USA datasets.

Metric	Delhi-NCR	Canada	USA
Total area	559 km^2	1138 km^2	54 cities
Total samples	12,542,183	46,080	35,596
Samples with PM2.5	12,542,183	12,154	35,134
Pollutants covered	PM_1, $PM_{2.5}$ and PM_{10}	CO, NO, NO_2, SO_2, O_3, PM_1, $PM_{2.5}$ and PM_{10}	CO, NO_2, SO_2, O_3, $PM_{2.5}$ and PM_{10}
Meteorological	Temp, RH, Pressure, Wind Speed *	-	Temp, RH, Pressure, Dew, Wind Speed, Wind Gust
Sensor source	Public bus	Commercial van	OpenDataPlatform
Monitoring days	91	114	668

Table 2: Statistical comparison of PM values in Delhi, Canada and USA datasets.

Metric	Delhi-NCR			Canada			USA	
	PM_1	$PM_{2.5}$	PM_{10}	PM_1	$PM_{2.5}$	PM_{10}	$PM_{2.5}$ AQI	PM_{10} AQI
Mean	120.35	207.92	226.11	12.15	15.08	46.45	31.15	17.67
Std-dev	57.27	114.36	123.86	9.02	12.87	97.36	17.11	11.00
Missing %	0	0	0	71.71	73.62	72.24	1.30	52.34

for Nov 2020 - Jan 2021 from *www. windfinder. com* to complement our Delhi-NCR dataset for the meteorological analysis performed in Appendix E and Fig. 15, showing pollutants correlation with different meteorological factors and peculiar situations due to external factors (like impact of stubble burning episodes). Due to this behaviour, interpolation and forecasting are hard problems. Still, in Appendix D and Fig. 14, we observed covariance among some spatial locations, and in Fig. 11, we observed some autocorrelation over the entire dataset. Similar trends were observed with the bus route analysis shown in Appendix F. Hence we decided to formulate an interpolation problem and a 24 hour forecasting problem to analyze the model performance, next.

4 ML Modeling Benchmarks

In this section, we benchmark the machine learning problems of **(1)** spatio-temporal interpolation, **(2)** spatio-temporal data imputation and **(3)** spatio-temporal forecasting on the proposed, Canada and USA datasets. This benchmarking study serves two roles. First, it allows us to compare the complexities of the three datasets beyond just statistical characterization. Secondly, spatio-temporal interpolations, data imputations, and forecasting methods are crucial for environmental research, policy-making, and individual decision-making. They empower various stakeholders to gain a comprehensive understanding of air pollution, proactively address potential increases in pollution levels, and make informed choices to reduce personal exposure. In order to harness the full potential of spatio-temporal forecasting, interpolations, and data imputations, it is crucial to benchmark and evaluate the performance of algorithms designed to tackle these problems.

4.1 Formulation of different ML Prediction Problems

(a) Spatio-temporal Interpolation: Given set of visible/available locations with input features (latitude, longitude and time) and $PM_{2.5}$ available for T+1 days, we wish to estimate $PM_{2.5}$ for a set of held-out locations for the T+1th day using the input features (latitude, longitude and time). This approach is compatible to the scenario where we have data for some locations and we use interpolation algorithms to know the $PM_{2.5}$ values at new locations.

(b) Spatio-temporal Missing Data Imputation: Given set of locations with input features (latitude, longitude and time) and $PM_{2.5}$ available for T days and a set of visible locations with input features (latitude, longitude and time) and $PM_{2.5}$ available for the T+1th day, we wish to estimate $PM_{2.5}$ for a set of held-out locations for the T+1th day using the input features (latitude, longitude and time). This setting is compatible to the scenario where we have intermittent data missing throughout the day and we use interpolation algorithms to predict the missing points taking past & present data as input.

(c) Spatio-temporal Forecasting: Given a set of all locations with input features (latitude, longitude and time) and $PM_{2.5}$ available for T days, we wish to estimate $PM_{2.5}$ for a set of all locations for the T+1th day using the input features (latitude, longitude and time).

4.2 ML Algorithms Benchmarked in this Paper

(a) Mean Predictor is the simple mean value of all visible samples which is used as the value of all the held-out locations. The mean value of all visible $PM_{2.5}$ locations C is used as the value of the held-out $PM_{2.5}$ locations P.

$PM^p_{2.5} \leftarrow mean \ \ \forall p \in P$, where $mean \leftarrow \frac{1}{|C|} \sum PM^c_{2.5} \ \ \forall c \in C$

(b) Inverse Distance Weighting (IDW) is the weighted average value of all visible C samples in terms of distance, which is used as the value of the held-out P locations.

$PM^p_{2.5} \leftarrow \sum \frac{PM^c_{2.5}}{F(d_{cp})} \ \ \forall c \in C \ \ \forall p \in P$, where F is a linear function on distance d.

(c) Random Forest (RF) is a non-linear model capable of modeling complex spaces. It is known to perform efficiently on non-linear regression tasks, using an ensemble of multiple decision trees, taking the final output as the mean of the output from all trees.

(d) XGBoost (XGB) iteratively combines the results from weak estimators. It uses gradient descent while adding new trees during training.

(e) ARIMA or Auto-Regressive Integrated Moving Average is a statistical time-series forecasting model that uses linear regression. It is configured using parameters (p, d, q) as: p is the number of lag observations included in the model, d is the number of times raw observations are differenced, and q is the size of the moving average window. We use ARIMA with parameters (3, 1, 1).

(f) N-BEATS is Neural Basis Expansion Analysis for Time Series, a deep learning model for zero-shot time-series forecasting [Oreshkin *et al.*, 2020]. We use the code from Python library "Darts".

(g) Non-Stationary Gaussian Process (NSGP) is a recent gaussian processes baseline [Patel *et al.*, 2022]. It learns a non-stationary covariance [Plagemann *et al.*, 2008] for latitude and longitude and locally periodic covariance for time. In general, Gaussian process Ras and Williams [2005] a.k.a. Kriging is a Bayesian non-parametric model known as the best unbiased predictor in spatial interpolation domain. With only three tunable parameters, it is considered a strong baseline.

(h) Graphsage is a graph neural network model to learn and predict values at unknown spatio-temporal locations [Hamilton *et al.*, 2017]. We transform the PM data to a graph, and use Graphsage for interpolation and missing data imputation. Our graph formulation is available in Appendix A.

4.3 Dataset Pre-processing and Evaluation Metrics for the Analysis

To benchmark interpolation and missing data interpolation ML modeling algorithms, we split the data into two parts for *visible* and *held-out/hidden* to use the held-out part for testing purpose. For forecasting based ML modeling, the visible/held-out split is not required, as predictions are made for future timestamps for all locations. For the Delhi dataset, we focus on the data collected from Nov 12, 2020, to Jan 30, 2021, excluding the initial days when there were fewer instruments on the buses and limited sample data. Additionally, we exclude the nightly data between 10 PM IST and 5:30 AM IST when buses remain stationary at confined bus-depots. To facilitate analysis, we divide the geographical area into square spatial grids with a side length of 1 km. These grids are further converted into spatio-temporal cells with a time interval of 30 minutes. To obtain representative PM values, we compute the average of all samples within each spatio-temporal cell. Subsequently, we employ K-fold cross-validation to partition the data into K $PM_{2.5}$ sets for each day, out of which 1 set is reserved for test and others used in training. The results obtained from the Delhi dataset are denoted as *Delhi (Day)* in the generated plots.

Additionally, we utilize two open-sources $PM_{2.5}$ datasets, from Hamilton in Ontario, Canada [Adams and Corr, 2019] and from USA [Bhattacharyya *et al.*, 2022]. For the Canada dataset, we process the data from 18 distinct days in the year 2015 using the same methodology. These results are presented as *Canada (Day)* in the respective experiments. As the data for Canada exhibits temporal sparsity, we project the data for each year onto a single day and treat it as equivalent to 11 days (from 2006 to 2016). The outcomes of this processing approach are depicted as *Canada (Year)* in the experiments. For the USA data, we use the available $PM_{2.5}$ data across 54 cities from Jan 1, 2019 to Dec 11, 2020, and the results are presented as *USA (Day)*. We benchmark the datasets on Nvidia DGX Workstation (with 4X Tesla V100 GPUs) and the benchmarking code is available at `https://github.com/sachin-iitd/DelhiPMDatasetBenchmark`.

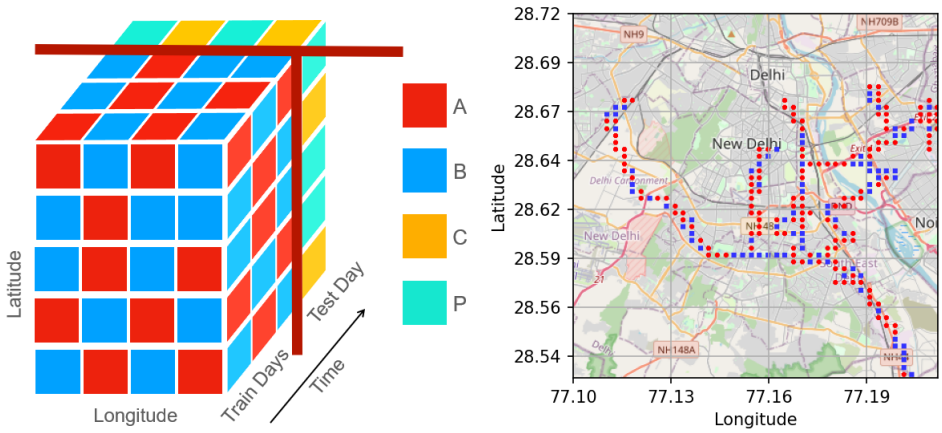
Evaluation Metrics: The Loss is computed as:

$$RMSE(L'_p, L_p) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y'_i - y_i)^2} \tag{1}$$

where y'_i is the predicted and y_i is the true $PM_{2.5}$ value, and N is the total number of samples.

For each of the K folds, we separately compute RMSE for that fold, and then plot average with standard deviation bars over the K folds. The lower RMSE being the better.

Notation: We use T (consecutive) days data for the training and take the next day for test/evaluation. For interpolation benchmarking, Fig. 4a denotes the various subsets of this T+1 days data as A, B, C and P, for one fold of $K = 5$ fold validation. The P is the set over which *Prediction* is to be done. the C set can be understood as the *Context* for prediction. For a given fold, A is the visible set with 80% of all T train days data, B is the held-out set with the remaining 20% of the T train days data. $A \cup B$ forms the whole dataset for the T train days. Similarly, C is the visible set with 80% of the test day data, P is the held-out set with the remaining 20% of the test day and $C \cup P$ forms the whole dataset for the test day. The held-out locations B and P will be different for each of the K folds. The



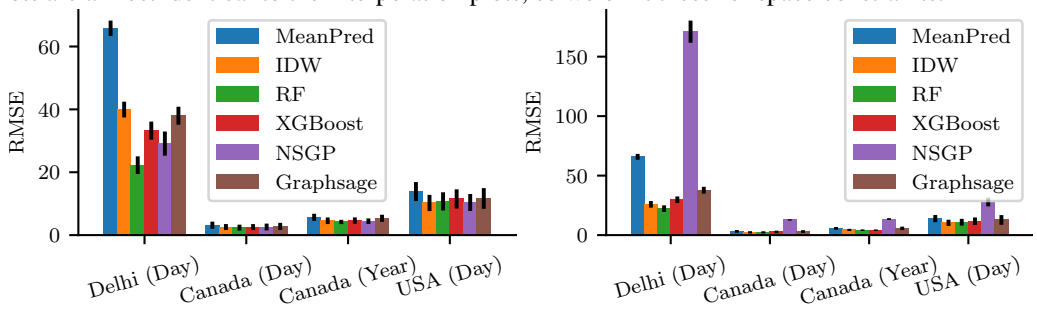
(a) An arbitrary split of A, B, C and P sets. (b) Visible (A) and Held-out (B) sets over map.

Figure 4: PM Data Splits. (a) A ∪ B forms the whole dataset for the T train days, C ∪ P forms the whole dataset for the test day. Spatial locations of C or P are independent of spatial locations of A or B. (b) Set of A and B spatial locations for 3 PM to 4:30 PM on Dec 15, 2020 from all bus-routes.

A and C though both being visible sets, are independent of each other. The exact number of locations in A, B, C and P change across the K folds. In Fig. 4b, we show set of A and B spatial locations in Delhi dataset for 3 PM to 4:30 PM on Dec 15, 2020.

4.4 Observations and Inferences

Fig. 5, shows the RMSE for interpolation, using 5-fold cross validation for the two training configurations ACT in Fig. 5a and C in Fig. 5b, for 3 training days. ACT uses the visible set from both training and test days, while C uses only the test day's $PM_{2.5}$ visible set. The missing data imputation plots are almost identical to the interpolation plots, so we omit these for space constraints.



(a) Train:ACT, Input:ACT, Predict:P (b) Train:C, Input:C, Predict:P

Figure 5: Interpolation RMSE. Training days' data is used by ML model in (a) and not used in (b).

Observation 1: Delhi dataset is harder to model. All experiments over Delhi data show higher RMSE and all experiments over Canada and USA data show low RMSE, for both interpolation and forecasting, in Figures 5, and 7. This shows that Delhi data is more challenging for ML modeling, than the currently available PM datasets. Also, the majority of samples present in the data is below 800 $PM_{2.5}$. Omitting the 29 test samples with higher $PM_{2.5}$ for the purview of low-cost sensor accuracy, in Fig. 6 we can see the MAE and RMSE for different $PM_{2.5}$ levels for Random Forest (RF) algorithm. We observe that the modeling errors increase with increasing $PM_{2.5}$ levels. For existing dataset like Canada with a very low $PM_{2.5}$ of mean 15 ± 13 (refer the Table 2), the expected modeling error would automatically be low, making the data easier to model. For the Delhi dataset with high $PM_{2.5}$ of mean 208 ± 114, makes the dataset hard to model.

Observation 2: Learning from data helps in modeling the Delhi dataset. All ML based algorithms show significant improvement over Mean Predictor for Delhi data in Figures 5a, whereas improvement for Canada and USA data over Mean Predictor is not significant. In Fig. 5a, all ML algorithms exhibit less than 40 RMSE while Mean Predictor RMSE is 65.80 for Delhi data (best case improvement is 66.2% for RF and worst case 39.3% for IDW). For Canada data, best case improvement is $\sim$ 27% and worst case sees no improvement, whereas for USA AQI data, improvement is within 16% - 26%.

Observation 3: Traditional ML algorithms do as well as the recent models for the Delhi dataset. Learning from data matters, as the ML based models do better than the mean predictor. But the recent

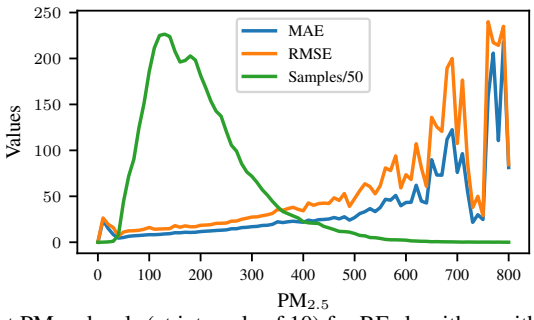


Figure 6: Errors for different $PM_{2.5}$ levels (at intervals of 10) for RF algorithm, with the number of test samples. Modeling shows high errors at high $PM_{2.5}$, making Delhi dataset hard to model due to high $PM_{2.5}$ levels.

complex Bayesian models like NSGP, and the neural network based models like GraphSage (for interpolation) and N-BEATS (for forecasting), do not outperform powerful traditional ML models like Random Forest. For instance, RF performs best for interpolation (RMSE 22.24 in Fig. 5), and XGBoost performs best for forecasting (RMSE 84.15 in Fig. 7).

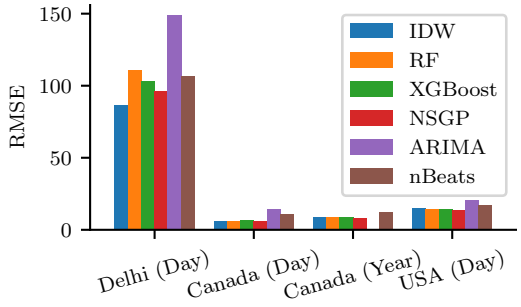
Observation 4: Historical training data adds no value for interpolation. For the spatio-temporal interpolation problem, just using data from the visible set C from test day is enough to predict the held-out P data with low RMSE. For example, the RMSE for RF is similar (22.24) for test day only data C in Fig. 5b and with including train day data ACT in Fig. 5a. And XGBoost is better for C with RMSE 29.73 than for ACT with RMSE 33.24. NSGP is the only algorithm, which sees a huge jump in RMSE when not using training data from past days. Thus PM for a given day is mostly unrelated to PM on past days, and using historical training data has no significant impact on interpolation RMSE.

Observation 5: A location's air quality is related to nearby location. In Fig. 5, for Delhi data, we can observe that IDW performs significantly better than the Mean Predictor. Mean Predictor does not take the distance into account, whereas the IDW gives more weightage to nearby locations data. So, the impact of an adjacent location is significant for IDW w.r.t Mean Predictor, pointing that nearby locations air quality impacts the air quality of a location.

Fig. 7 shows RMSE of forecasting. Graphsage does not work in this setting as it requires a subset of test day's data for edge formation to the data being predicted. So we drop Graphsage, and add two forecasting specific baselines: ARIMA and N-BEATS, that are not suitable for interpolation.

Observation 6: Forecasting is a harder problem than interpolation. Forecasting RMSEs are significantly higher than interpolation RMSEs. The best model in forecasting is XGBoost in Fig. 7 with RMSE 84.15, whereas the best model for interpolation in Fig. 5 is RF with RMSE 22.24. Higher forecasting RMSE compared to interpolation also supports that previous day's data has less impact on test day's PM data. Hence forecasting using only past days' data for an unseen future test day is hard.

Observation 7: How time is normalized affects forecasting accuracy. In Fig. 7a, time normalization is done across days, i.e. time starts at 0 on first train day and increases to 1 till last train day. ARIMA / N-BEATS don't normalize the time directly, they take all PM values in a sequence corresponding to time from start to end. RF/XGBoost takes input in random sequence and hence takes the time as a state parameter, which can be normalized from start to end, or for each day. Table 7b compares this time normalization across days (T), to normalizing separately for each day. RF, XGBoost and NSGP show lower RMSE for separate normalization for each day, while IDW does better with normalization across days. This pre-processing step of time normalization therefore should be carefully decided based on the ML algorithm.



(a)

	Config		Delhi (Day)	
Algo	Train	Input	Train	Test
IDW	ABT	ABT	-	86.52
	AB	AB	-	270.73
RF	ABT	ABT	8.09	110.49
	AB	AB	45.85	89.54
XGBoost	ABT	ABT	33.19	102.68
	AB	AB	57.12	84.15
NSGP	ABT	ABT	24.34	95.83
	AB	AB	51.90	86.34
ARIMA	ABT	ABT	204.87	148.86
N-BEATS	ABT	ABT	-	106.41

(b)

Figure 7: Forecasting RMSE. (a) Normalization is done across all days. (b) Comparison of normalization across all days (T) vs normalization over each day.

5 Limitations and Future Work

We acknowledge this dataset is limited to 3 months, and we are working to scale in another metropolitan city in India for a comprehensive 6+ months data collection and analysis. The bus routes selected by us covered the different types of areas, including green cover, residential, commercial etc, termed as static factors. Hence we have tried our best to reduce the bias towards any particular factor. Still, there is limited spatial distribution due to the bus routes over urban arterial roads. An alternative could have been to put sensors in private vehicles, but it is not reasonable to drive cars just to gather the data. Also, commercial cabs can be used for data collection, which we can check in our future work. We are anyhow moving from a high limitation of few data points in the same vicinity to million+ points.

In Appendix G, we also discuss anomaly detection used to fix any faults in the low cost sensors. We check for samples recorded per minute, number of minutes each device is active in an hour, number of active hours in a day, samples recorded per region, inter-sensor PM values variation, and intra-sensor PM values variation, during dataset collection to effectively keep validating the data being collected.

In our future work, we aim to address the problem of recommending suitable locations for installing new expensive sensors effectively within budget constraints, a challenging task in a developing country like India. By leveraging the insights gained from this research, we strive to optimize the allocation of resources and enhance the efficiency of the monitoring network, further strengthening pollution mitigation efforts.

6 Conclusion

Delhi-NCR, with its notorious air pollution problem, poses a significant health risk to its population of approximately 46 million individuals. In this paper, we present a novel PM dataset AIRDELHI collected from this region using low-cost IoT devices deployed on public buses. This dataset serves as a valuable resource for environmental researchers and medical practitioners, offering insights into ground-level PM exposure for daily commuters and temporal variations in PM levels over days and weeks. It provides a comprehensive view of spatial variations across different locations within the region.

Through thorough statistical analysis and benchmarking studies, we have established that the released dataset is distinct from any other existing pollution dataset. By comparing the performance of machine learning algorithms on the released dataset against the Canada and USA datasets, we have demonstrated the significant differences in characteristics and challenges associated with the Delhi-NCR dataset. This highlights the need for specialized approaches and tailored solutions to address the unique complexities of air pollution in this region.

The low-cost mobile data collection has the potential to complement the expensive static sensor network in the city, empowering citizens to make informed decisions regarding local PM levels. This includes determining the safety of engaging in outdoor activities, choosing appropriate protective measures such as face-masks or air purifiers, and selecting optimal commuting routes and transportation modes to minimize PM exposure. Such considerations are vital for safeguarding public health and promoting environmental sustainability.

To foster further advancements in the field of environmental sustainability, we release both the code and data associated with this study. This allows researchers to build upon our work, explore new avenues of inquiry, and contribute to the collective understanding and management of air pollution-related challenges.

Acknowledgments and Disclosure of Funding

This work was supported by DST-SERB for funding through IMPRINT-II research grant. We also thank the Delhi Integrated Multimodal Transit System (DIMTS) for allowing us to instrument their fleet.

References

Ismi Abidi, Sagar Ravi Gaddam, Saswat Kumar Pujari, Chinmay Shirish Degwekar, and Rijurekha Sen. Complexity of factor analysis for particulate matter (pm) data: A measurement based case study in delhi-ncr. In *ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS)*, COMPASS '22, page 45–56, New York, NY, USA, 2022. Association for Computing Machinery.

Matthew D. Adams and Denis Corr. A mobile air pollution monitoring data set. *Data*, 4(1), 2019.

Stacey E Alexeeff, Ananya Roy, Jun Shan, Xi Liu, Kyle Messier, Joshua S Apte, Christopher Portier, Stephen Sidney, and Stephen K Van Den Eeden. High-resolution mapping of traffic related air pollution with google street view cars and incidence of cardiovascular events within neighborhoods in oakland, ca. *Environmental Health*, 17:1–13, 2018.

Joshua S Apte, Thomas W Kirchstetter, Alexander H Reich, Shyam J Deshpande, Geetanjali Kaushik, Arvind Chel, Julian D Marshall, and William W Nazaroff. Concentrations of fine, ultrafine, and black carbon particles in auto-rickshaws in new delhi, india. *Atmospheric Environment*, 45(26):4470–4480, 2011.

Joshua S Apte, Kyle P Messier, Shahzad Gani, Michael Brauer, Thomas W Kirchstetter, Melissa M Lunden, Julian D Marshall, Christopher J Portier, Roel CH Vermeulen, and Steven P Hamburg. High-resolution air pollution mapping with google street view cars: exploiting big data. *Environmental science & technology*, 51(12):6999–7008, 2017.

Samir Avdakovic, Maja Muftic Dedovic, Nedis Dautbasic, and Jasenka Dizdarevic. The influence of wind speed, humidity, temperature and air pressure on pollutants concentrations of pm10—sarajevo case study using wavelet coherence approach. In *2016 XI International Symposium on Telecommunications (BIHTEL)*, pages 1–6. IEEE, 2016.

Mayukh Bhattacharyya, Sayan Nag, and Udita Ghosh. Deciphering environmental air pollution with large scale city data. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5031–5037. International Joint Conferences on Artificial Intelligence Organization, 7 2022. AI for Good.

Srinivas Bikkina, August Andersson, Elena N Kirillova, Henry Holmstrand, Suresh Tiwari, Atul K Srivastava, Deewan S Bisht, and Örjan Gustafsson. Air quality in megacity delhi affected by countryside biomass burning. *Nature Sustainability*, 2(3):200–205, 2019.

BME. Humidity sensor bme280, 2023.

CAQM. Revised graded response action plan (grap) for ncr, 2022.

CC-by4. Attribution 4.0 international (cc by 4.0), 2013.

Arpan Chatterji. Air pollution in delhi: filling the policy gaps. *Massach Undergr J Econ*, 17(1), 2021.

Sachin Chauhan, Kashish Bansal, and Rijurekha Sen. Ecolight: Intersection control in developing regions under extreme budget and network constraints. 2020.

Yun Cheng, Xiucheng Li, Zhijun Li, Shouxu Jiang, Yilong Li, Ji Jia, and Xiaofan Jiang. Aircloud: A cloud-based air-quality monitoring system for everyone. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, 2014.

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.

DIMTS. Delhi integrated multi-modal transit system ltd.

DST-SERB. Science and engineering research board.

ET. Caqm asks delhi ncr states to install sensors to check pollution at construction sites and hotspots, 2022.

Xi Gao and Weide Li. A graph-based lstm model for pm2. 5 forecasting. *Atmospheric Pollution Research*, 2021.

Meiling Gao, Junji Cao, and Edmund 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.

Goal-11. Make cities and human settlements inclusive, safe, resilient and sustainable.

Google. Mapping the invisible: Street view cars add air pollution sensors, 2014.

Tanishka Goyal, Ankita Singh, Smriti Chhaya, Aditi Vikas, Poorva Garg, Ritika Malik, and Rijurekha Sen. Low cost platform design for pollution measurement in delhi-ncr using vehicle-mounted sensors. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, page 759–761, New York, NY, USA, 2018. Association for Computing Machinery.

Hongjie Guo, Guojun Dai, Jin Fan, Yifan Wu, Fangyao Shen, and Yidan Hu. A mobile sensing system for urban monitoring with adaptive resolution. *Journal of Sensors*, 2016, 2016.

Sarath K. Guttikunda, Sai Krishna Dammalapati, Gautam Pradhan, Bhargav Krishna, Hiren T. Jethva, and Puja Jawahar. What is polluting delhis air? a review from 1990 to 2022. *Sustainability*, 15(5), 2023.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *31st NeurIPS Conference*, 2017.

HT. Pollution in november 2020 up from last year, shows cpcb data, 2020.

IIITD. Sdg research, 2023.

IMPRINT-II. Impacting research innovation and technology (imprint-ii).

Shiva R Iyer, Ananth Balashankar, William H Aeberhard, Sujoy Bhattacharyya, Giuditta Rusconi, Lejo Jose, Nita Soans, Anant Sudarshan, Rohini Pande, and Lakshminarayanan Subramanian. Modeling fine-grained spatio-temporal pollution maps with low-cost sensors. *npj Climate and Atmospheric Science*, 5(1):76, 2022.

Wan Jiao, Gayle Hagler, Ronald Williams, Robert Sharpe, Ryan Brown, Daniel Garver, Robert Judge, Motria Caudill, Joshua Rickard, Michael 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), 2016.

Federico Karagulian, Maurizio Barbiere, Alexander Kotsev, Laurent Spinelle, Michel Gerboles, Friedrich Lagler, Nathalie Redon, Sabine Crunaire, and Annette Borowiak. Review of the performance of low-cost sensors for air quality monitoring. *Atmosphere*, 10(9), 2019.

Atakan Kurt, Betul Gulbagci, Ferhat Karaca, and Omar Alagha. An online air pollution forecasting system using neural networks. *Environment international*, 2008.

Van-Duc Le, Tien-Cuong Bui, and Sang-Kyun Cha. Spatiotemporal deep learning model for citywide air pollution interpolation and prediction. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 55–62. IEEE, 2020.

Jason Jingshi Li, Boi Faltings, Olga Saukh, David Hasenfratz, and Jan Beutel. Sensing the air we breathe: The opensense zurich dataset. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 323–325. AAAI Press, 2012.

Yansui Liu, Yang Zhou, and Jiaxin Lu. Exploring the relationship between air pollution and meteorological conditions in china under environmental governance. *Scientific reports*, 10(1):14518, 2020.

Zhipeng Luo, Jianqiang Huang, Ke Hu, Xue Li, and Peng Zhang. Accuair: Winning solution to air quality prediction for kdd cup 2018. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1842–1850, 2019.

Pier Mannuccio Mannucci and Massimo Franchini. Health effects of ambient air pollution in developing countries. *International journal of environmental research and public health*, 14(9):1048, 2017.

Kyle P Messier, Sarah E Chambliss, Shahzad Gani, Ramon Alvarez, Michael Brauer, Jonathan J Choi, Steven P Hamburg, Jules Kerckhoffs, Brian LaFranchi, Melissa M Lunden, et al. Mapping air pollution with google street view cars: Efficient approaches with mobile monitoring and land use regression. *Environmental science & technology*, 52(21):12563–12572, 2018.

MHA. Guidelines on surveilance- containment and caution, 2020.

Pavan K Nagar, Dhirendra Singh, Mukesh Sharma, Anil Kumar, Viney P Aneja, Mohan P George, Nigam Agarwal, and Sheo P Shukla. Characterization of pm 2.5 in delhi: role and impact of secondary aerosol, burning of biomass, and municipal solid waste and crustal matter. *Environmental Science and Pollution Research*, 24:25179–25189, 2017.

William Navidi. *Statistics for Engineers and Scientists*. McGraw-Hill, 2009.

NCRPB. Ncr constituent areas, 2018.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

Tung Nguyen, Jason Jewik, Hritik Bansal, Prakhar Sharma, and Aditya Grover. Climatelearn: Benchmarking machine learning for weather and climate modeling. *arXiv preprint arXiv:2307.01909*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

ORF. Delhi is failing its children, air pollution is choking their future, 2021.

Zeel B Patel, Palak Purohit, Harsh M Patel, Shivam Sahni, and Nipun Batra. Accurate and scalable gaussian processes for fine-grained air quality inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12080–12088, Jun. 2022.

Movva Pavani and P Trinatha Rao. Urban air pollution monitoring using wireless sensor networks: A comprehensive review. *International Journal of Communication Networks and Information Security*, 9(3):439–449, 2017.

Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II 19*, pages 204–219. Springer, 2008.

Pengwei Qiao, Peizhong Li, Yanjun Cheng, Wenxia Wei, Sucai Yang, Mei Lei, and Tongbin Chen. Comparison of common spatial interpolation methods for analyzing pollutant spatial distributions at contaminated sites. *Environmental geochemistry and health*, 41(6):2709–2730, 2019.

Aakash C Rai, Prashant Kumar, Francesco Pilla, Andreas N Skouloudis, Silvana Di Sabatino, Carlo Ratti, Ansar Yasar, and David Rickerby. End-user perspective of low-cost sensors for outdoor air pollution monitoring. *Science of The Total Environment*, 607:691–705, 2017.

Carl Edward Ras and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Ravi Sahu, Kuldeep Kumar Dixit, Suneeti Mishra, Purushottam Kumar, Ashutosh Kumar Shukla, Ronak Sutaria, Shashi Tiwari, and Sachchida Nand Tripathi. Validation of low-cost sensors in measuring real-time pm10 concentrations at two sites in delhi national capital region. *Sensors*, 20(5), 2020.

Philipp Schneider, Matthias Vogt, Rolf Haugen, Amirhossein Hassani, Nuria Castell, Franck R. Dauge, and Alena Bartonova. Deployment and evaluation of a network of open low-cost air quality sensor systems. *Atmosphere*, 14(3), 2023.

Howard Seltman. *Experimental Design and Analysis*. Carnegie Mellon University, 2018.

Komal Shukla, Prashant Kumar, Gaurav S. Mann, and Mukesh Khare. Mapping spatial distribution of particulate matter using kriging and inverse distance weighting at supersites of megacity delhi. *Sustainable Cities and Society*, 54:101997, 2020.

Ronak Sutaria. Delhi plans mesh of sensors to monitor pollution air hot spots, 2022.

CB Tripathi, Prashant Baredar, and Lata Tripathi. Air pollution in delhi. *Current Science*, 117(7):1153–1160, 2019.

Yi-Ting Tsai, Yu-Ren Zeng, and Yue-Shan Chang. Air pollution forecasting using rnn with lstm. In *2018 IEEE 16th Intl DASC/PiCom/DataCom/CyberSciTech Conf*. IEEE, 2018.

UN. The 17 goals, 2015.

Urbanemissions. Air pollution monitoring 101, 2023.

Weather. Delhi pollution: Capital records 'emergency' levels of air pollution, 2020.

Hongmei Yang, Qin Peng, Jun Zhou, Guojun Song, and Xinqi Gong. The unidirectional causality influence of factors on pm2.5 in shenyang city of china. *Scientific Reports*, 10(1):8403, 2020.

Wei Ying Yi, Kin Ming Lo, Terrence Mak, Kwong Sak Leung, Yee Leung, and Mei Ling Meng. A survey of wireless sensor network based air pollution monitoring systems. *Sensors*, 15(12):31392–31427, 2015.

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, 2018.

Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444, 2013.

Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th SIGKDD conference on Knowledge Discovery and Data Mining*. KDD 2015, August 2015.

Tongshu Zheng, Michael H. Bergin, Karoline K. Johnson, Sachchida N. Tripathi, Shilpa Shirodkar, Matthew S. Landis, Ronak Sutaria, and David E. Carlson. Field evaluation of low-cost particulate matter sensors in high and low concentration environments. *Atmospheric Measurement Techniques*, 2018.

T. Zheng, M. H. Bergin, R. Sutaria, S. N. Tripathi, R. Caldow, and D. E. Carlson. Gaussian process regression model for dynamically calibrating and surveilling a wireless low-cost particulate matter sensor network in delhi. *Atmospheric Measurement Techniques*, 12(9):5161–5181, 2019.

Appendix

A Graphsage (with Graph formulation)

We aim at learning universal weights, similar to GraphSAGE Hamilton *et al.* [2017], which will signify the importance of a neighbour based on some known node values and edge weights. Here we define node values as the value of the pollutant PM2.5 while the edges are created using latitude, longitude and datetime features. Firstly, a graph is created from the train dataset, aggregating all inputs within 500m and 30 minutes of each other into a single node. An edge is created between two nodes if they lie within 2 hours of each other. The graph then goes through two graph-based layers to learn the required weights where embeddings are learnt using the max and mean aggregation layers, followed by 3 fully connected neural network layers to predict the final pollutant value.

Let $G = (V, E, \sigma, \mathcal{A})$ be a Directed Graph with V vertices/nodes, E edges, $\mathcal{A}$ attributes and σ as the label mapping, where

$\sigma : V \rightarrow \mathcal{L}$

$\mathcal{L}$ being the set of $PM_{2.5}$ values.

V corresponds to the spatiotemporal locations where $PM_{2.5}$ values are known (S: Red) or desired (U: Blue), i.e. V=S+U. E ($e \in E$) connects the V ($v \in V$) such that

e_{ij} = $(v_i, v_j) \mid v_i \in S \wedge v_j \in (S \vee U)$ and $t_{ij} \leq TimeLimit$, where t_{ij}=abs(v_i^t - v_j^t)

The Graph G comprises of separate connected components for different days.

e_{ij} = $(v_i, v_j) \mid v_i \in Day_p$ and $v_j \in Day_q \Rightarrow p = q$

Weight of each edge is inversely proportional to the spatial distance between the two nodes across the edge.

$w_{ij} = \frac{1}{1+d_{ij}}$, if e_{ij} exists, where d_{ij}=haversine(v_i, v_j)

Edges exist from all S nodes to each U node. No S to S edges exist.

$e_{ij} = (v_i, v_j) \mid v_i \in S$ and $v_j \in U \Rightarrow |e_{ij}\ \forall i| = |S|\ \forall j$

The graph G is of two types:

Train Graph G_{Train}: It is used for training Graphsage Neural Network.

$v \in Day_{Train} \Rightarrow v \in S \vee U \Rightarrow |v \in S| > 0$ and $|v \in U| > 0$

The RMSE loss on the nodes $v \in U$ is used for model training.

Test Graph G_{Test}: It is used for evaluating the trained Graphsage model on unseen test day data (Day_{Test}) along with full data from known days.

$v \in Day_{Test} \Rightarrow v \in S \vee U \Rightarrow |v \in S| > 0$ and $|v \in U| > 0$

The v is formed by taking the corresponding $PM_{2.5}$ label L and an indicator variable I.

$v_i = L_i | I_i$

$L_i \leftarrow PM_{2.5}, I_i \leftarrow= 1\ \forall\ v \in S$

$L_i \leftarrow 0, I_i \leftarrow= 0\ \forall\ v \in U$

The 2 layer mean-pool and max-pool model graphsage architecture is shown in Fig. 8.

The RMSE loss of the nodes $v \in U$ (or $v \in P$ in particular) is used as the reporting metric.

For Graphsage based evaluation, out the 80% training data in 5-fold cross validation, we use 40% as *visible* set, 40% as *held-out* set, to manage edges between these two sets.

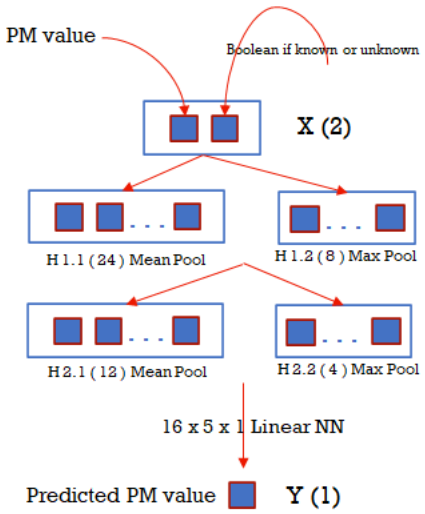


Figure 8: Graphsage model architecture.

B Complete ML Benchmarks

Table 3 shows the complete benchmark for Spatio-temporal Interpolation for different train and input configurations. An important subset of these benchmarks is presented in Fig. 5 and discussed in § 4.4 in the main paper.

Table 3: Spatiotemporal Interpolation RMSE for different configurations (* denotes partial experiments).

Algo	Config		Delhi (Day)		Canada (Day)		Canada (Year)		USA (Day)	
	Train	Input	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
MeanPred	-	C	65.80	2.44	3.13	1.14	5.66	1.13	13.85	3.02
IDW	ACT	ACT	39.94	2.51	2.56	0.95	4.56	1.05	**10.24**	2.57
	AC	AC	351.73	2.85	2.66	0.95	7.33	1.61	23.21	5.29
	C	C	25.83	2.77	**2.31**	0.98	4.35	0.91	10.32	2.60
RF	ACT	ACT	**22.24**	2.81	2.37	0.95	4.18	0.68	10.73	2.89
	AC	AC	77.30	2.67	2.69	0.98	6.05	0.93	13.93	3.20
	C	C	22.25	2.77	2.34	0.89	4.12	0.68	10.82	2.85
XGBoost	ACT	ACT	33.24	2.87	2.55	0.95	4.62	1.01	11.51	3.05
	AC	AC	65.04	2.55	2.90	0.98	6.03	0.84	14.19	3.32
	C	C	29.73	2.76	2.71	1.05	**4.09**	0.67	11.66	3.16
NSGP	ACT	ACT	29.11	3.84	2.57	1.09	4.41	0.89	10.39	2.69
	ACT	C	194.96	1.63	13.02	0.72	14.68	0.63	27.11	3.25
	AC	AC	69.75	3.65	2.89	0.90	5.99	0.95	13.42	3.09
	AC	C	37.46	4.63	3.17	1.12	5.25	1.22	22.14	3.46
	C	C	170.99	9.31	12.74	0.55	13.51	0.72	27.81	3.67
Graphsage	AC	C	38.63	3.89	2.96	1.25	5.37	1.13	11.66	3.29
	C	C	38.68	4.12	3.13	1.24	5.68	1.46	12.75	4.06

Table 4 shows the complete benchmark for Spatio-temporal Missing data Imputation for different train and input configurations. Missing data imputation is briefly discussed in § 4.4 in the main paper. The traditional and powerful RF (Random Forest) algorithm outperforms all other algorithms and methods.

Table 4: Missing Data Imputation RMSE for different configurations.

Algo	Config		Delhi (Day)		Canada (Day)		Canada (Year)		USA (Day)	
	Train	Input	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
MeanPred	-	C	65.80	2.44	3.13	1.14	5.66	1.13	13.85	3.02
IDW	ABCT	ABCT	40.06	2.51	2.56	0.95	4.56	1.05	10.19	2.57
	ABC	ABC	399.44	1.14	2.69	0.93	7.92	1.47	68.63	8.00
RF	ABCT	ABCT	**22.26**	2.85	**2.34**	0.93	**4.22**	0.67	**9.42**	2.60
	ABC	ABC	78.90	2.71	2.70	0.96	6.21	0.96	14.09	3.13
XGBoost	ABCT	ABCT	33.46	2.87	2.53	0.91	4.63	1.02	10.23	2.74
	ABC	ABC	67.66	2.55	2.94	0.96	6.19	0.87	13.84	3.12
NSGP	ABCT	ABCT	29.06	3.64	2.52	0.95	4.40	0.85	9.62	2.46
	ABC	ABC	71.27	3.16	2.81	0.91	6.09	0.88	13.38	2.97
	ABC	C	171.94	8.08	12.71	0.53	13.29	0.94	21.76	3.18
	ABCT	C	194.98	1.55	12.90	0.60	14.58	0.68	26.80	3.08
	ABT	C	195.86	3.00	13.03	0.61	14.68	0.95	27.28	2.99
	AB	C	37.63	3.87	4.15	0.92	5.43	1.09	23.19	3.10
Graphsage	ABC	C	38.53	2.94	3.15	1.30	5.46	1.11	11.78	3.56
	AB	C	38.48	2.86	3.13	1.25	5.41	1.08	11.59	3.15

Table 5 shows the complete benchmark for Spatio-temporal Forecasting for different configurations. A subset of these benchmarks is presented in Fig. 7 and discussed in § 4.4 in the main paper.

Table 5: Forecasting RMSE for different configurations.

Algo	Config	Delhi (Day)	Canada (Day)	Canada (Year)	USA (Day)
IDW	ABT	86.52	5.65	8.31	14.61
	AB	270.73	**5.73**	11.23	69.20
RF	ABT	110.49	5.90	8.45	14.23
	AB	89.54	6.11	10.80	14.58
XGBoost	ABT	102.68	6.69	8.23	14.25
	AB	**84.15**	6.51	9.84	14.52
NSGP	ABT	95.83	5.76	**8.01**	**13.65**
	AB	86.34	6.08	10.22	14.34
ARIMA	ABT	148.86	13.87	12.85	20.12
nBeats	ABT	106.41	10.88	11.84	17.05

NSGP Variance

Non-stationary GP models provides us with uncertainty (variance) values around the expected mean PM2.5 value for each expected spatio-temporal location. We find that the average variance value for Delhi dataset is huge as compared to Canada (Day) experiments. It is more challenging for a model or algorithm to correctly understand and predict the PM values for Delhi dataset. Even the USA dataset with data over a big region does not exhibit such complexity for the algorithms.

Table 6: NSGP Variance.

	Delhi (Day)	Canada (Day)	Canada (Year)	USA (Day)
Spatio-temporal Interpolation	118.73	17.29	72.94	76.34
Missing Data Imputation	142.51	20.34	113.37	72.58
Forecasting	77.38	19.96	60.89	59.76

C Anova Tests Analysis for Low Cost Sensor

In continuation to the data quality analysis presented in § 3.2, we performed Anova Tests over the data collected by DustTrak and our Low Cost Mobile sensor devices at the same location. ANOVA Navidi [2009], Analysis of Variance, is a strong statistical factorial technique which involves one dependent variable known as response variable and one or more independent variables known as factors. The factors have different levels called treatments. The ANOVA tests compare two types of variation, the variation between the sample means and the variation within the samples.

Two-way ANOVA test between DustTrak reference sensor and our low-cost mobile sensor

In relation to our low cost sensor scenario, the observed $PM_{2.5}$ values are dependent on the sensor *Type* (DustTrak vs Low Cost) and the time(*Day*) of observation. As we have two factors, we need to perform two-way ANOVA test. For the *Day* factor, we take the hourly $PM_{2.5}$ mean samples grouped over each day (24 hours) of observations.

Two-way ANOVA tests three *null* hypotheses

(a) the means of observations grouped by factor *Type* are same

(b) the means of observations grouped by factor *Day* are same

(c) there is no interaction between the two factors *Type* and *Day*

Two-way ANOVA Assumptions

We make the standard assumptions of completeness, balanced design, normal distribution, similar variance, and sufficient replicates per treatment for validating ANOVA hypotheses. We take one device per sensor *Type* and same number (11) of *Day* as treatments under the two factors, with each *Type* and *Day* containing $PM_{2.5}$ samples. Fig. 9 shows the box-plot diagram with similar standard deviation for the DustTrak and our Low cost mobile sensors.

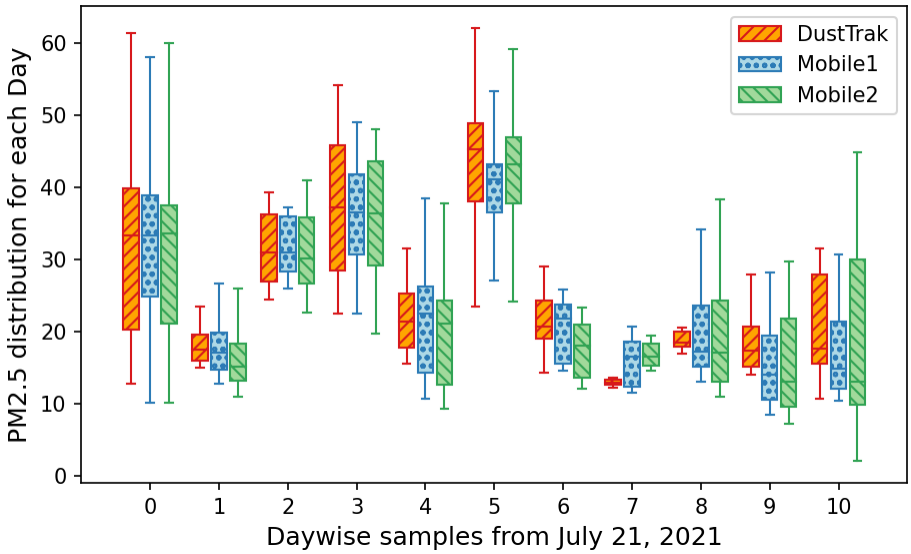


Figure 9: Mean and Standard Deviation for DustTrak and our Low Cost Mobile sensors.

Interpreting two-way ANOVA results

Table 7 shows the two-way ANOVA test results for DustTrak and our Low Cost Mobile sensor. As per Seltman [2018], the *SumSq* column represents the sum of squared deviations for each *Source* of variation. Each *Source* has a *df* (degrees of freedom) which is a measure of the number of independent pieces of information present in the deviations that are used to compute the corresponding *SumSq*. Each *MeanSq* is a variance estimate and the *SumSq* divided by the *df* for that *Source*.

Each *F*-statistic is the ratio of two *MeanSq* values. For the main effects, *Type* and *Day*, the denominators are all MSE which are pure estimates of group variance, unaffected by the validity of the null hypothesis. Each *F*-statistic is compared against it's null sampling distribution to compute a *p-value*. Interpretation of each of the *p-values* depends on the corresponding null hypothesis.

Table 7: Two-way ANOVA test for DustTrak Reference Sensor vs Our Low Cost Sensor Mobile Sensor 1

Effect	Source	df	SumSq	MeanSq	F	p-value	Significance
Main	*Type*	1	197.84	197.84	2.36	0.1248	Holds hypo (a)
	Day	10	30204.98	3020.50	36.10	< 0.0001	Reject hypo (b)
Interaction	*Type*Day*	10	261.76	26.18	0.31	0.9778	Holds hypo (c)
Error	Residual	444	37147.11	83.66			

In the presence of an interaction (*Type*Day*), the *p-value* for the interaction is most important and the main effects *Type* and *Day* p-values would be ignored if the interaction is significant. This is mainly because if the interaction is significant, then some changes in both explanatory variables (*Type* and *Day*) must have an effect on the outcome $PM_{2.5}$, regardless of the main effect *p-values*. The null hypothesis for the interaction *F*-statistic supports an additive relationship between the two explanatory variables, *Type* and *Day*, in their effects on the outcome $PM_{2.5}$. If the *p-value* for the interaction is less than α (usually 0.05), then we have a statistically significant interaction.

As we have a non-significant interaction $F_{1,10} = 0.31$ with *p-value* $= 0.9778$ which is greater than $\alpha = 0.05$, the null hypothesis (c) holds and the *p-values* for the main effects are valid for consideration. So, we can see that the *Day* has a significant *p-value* and thus it rejects the null hypothesis (b) meaning that there is impact of different *Day*'s observation on the observed $PM_{2.5}$ sample. This outcome aligns with a common understanding regarding the varying pollution across different days.

The analysis for the main effect sensor *Type* is more encouraging. It has a non-significant *p-value* $= 0.1248$ which holds the null hypothesis (a) that the means of the observations of the two device *Types*, DustTrak and our Low Cost Mobile sensor, are same. Hence, our Low Cost Mobile device can be effectively used to collect $PM_{2.5}$ observations in place of the expensive DustTrak sensors.

One-way ANOVA test between DustTrak reference sensor and our low-cost mobile sensor

Though the two-way ANOVA results hold for the main effects, we still perform one-way ANOVA test for the main effect *Type* (DustTrak vs Low Cost) for the observed $PM_{2.5}$ values. We ignore the *Day* factor in this analysis, so the $PM_{2.5}$ samples are only attributed with the *Type* factor. One-way ANOVA tests for the hypothesis (a) as of two-way ANOVA and with the standard assumptions of normal distribution and similar variance.

Table 8: One-way ANOVA test for DustTrak Reference Sensor vs Our Low Cost Sensor Mobile Sensor 1

Effect	Source	df	SumSq	MeanSq	F	p-value	Significance
Main	*Type*	1	197.84	197.84	1.36	0.2445	Holds hypothesis (a)
Error	Residual	464	67613.85	145.72			

Table 8 presents the results for one-way ANOVA, which too shows *Type* factor to have a non-significant *p-value* $= 0.2445$ which holds the null hypothesis (a). Hence with similar means of the observations, our Low Cost Mobile device can replace the expensive DustTrak sensors.

Two-way ANOVA test for our Low Cost device replaceability

We also show that our Low Cost Mobile devices are replaceable by each other. We perform two-way ANOVA tests between our Low Cost Mobile devices and the results are presented in Table 9.

As the *p-value* for the interaction is non-significant, main effects are valid. Likewise *Day* factor rejects hypothesis (b) and importantly *Type* factor holds hypothesis (a), allowing our Low Cost devices to replace each other as applicable.

Table 9: Two-way ANOVA test for Our Low Cost Sensor Mobile Sensor 1 vs 2

Effect	Source	df	SumSq	MeanSq	F	p-value	Significance
Main	*Type*	1	145.65	145.65	1.65	0.1991	Holds hypothesis (a)
	Day	10	31204.66	3120.47	35.43	< 0.0001	Reject hypothesis (b)
Interaction	*Type*Day*	10	148.46	14.85	0.17	0.9982	Holds hypothesis (c)
Error	Residual	450	39632.11	88.07			

D Spatio-temporal Correlation and Covariance Analysis

To analyze the temporal correlation over the $PM_{2.5}$ values for different locations, we split the data in grids of 1km x 1km x 1hr and average the $PM_{2.5}$ values in each grid to get a representative value[1].

We observe high autocorrelation for different spatial grid locations, for 1 hour lag. In Fig. 10, we show the autocorrelation for 12 spatial grid locations, denoted as locations A-L, with the corresponding Latitude-Longitude marked alongside in the titles. The X and Y axis represent the $PM_{2.5}$ values for without lag (y(t)) and with 1 hour lag (y(t+1)) respectively.

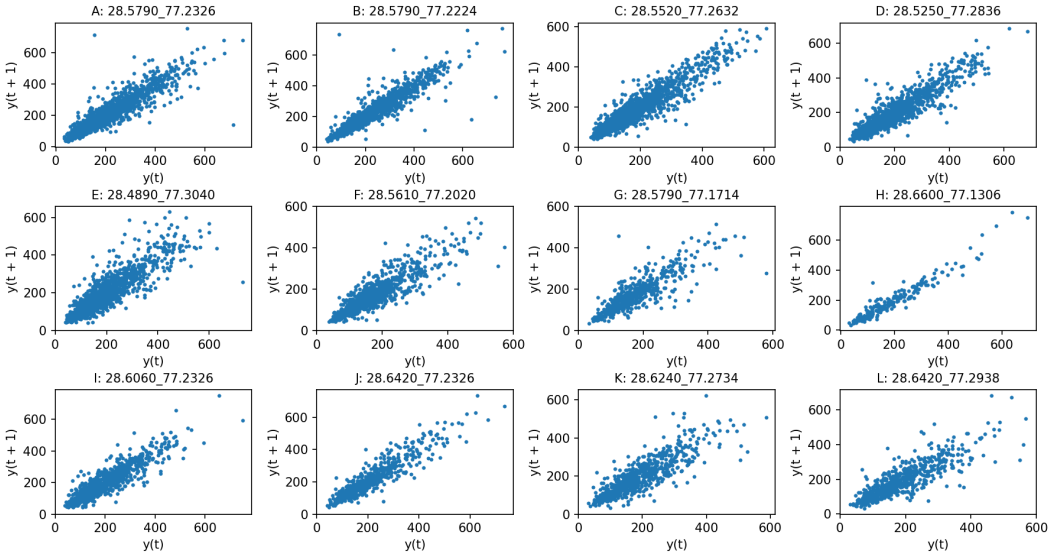


Figure 10: autocorrelation for 12 grid locations for 1 hour lag (the titles contain the latitude-longitude of the grid locations).

The autocorrelation decreases for lags of 2 hour or more at individual grid locations.

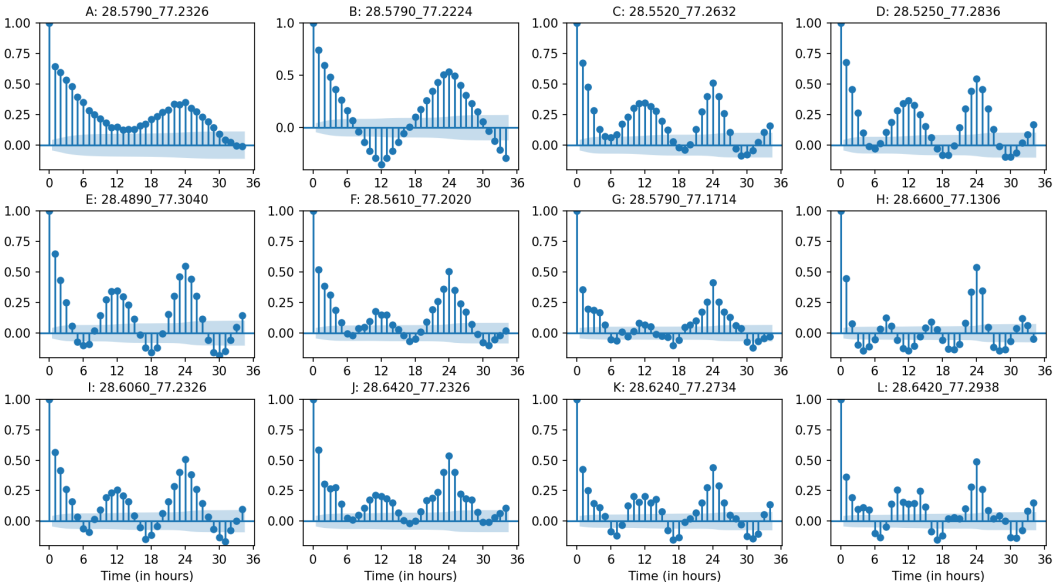


Figure 11: autocorrelation for 12 grid locations up to 1.5 days lags.

[1]This dataset version is available at `https://huggingface.co/datasets/sachin-iitd/DelhiPollDataset/tree/main/4.Grid(PM+Met)`

We further analyze the hourly data for 1.5 days for individual locations, and observe patterns of high and low autocorrelation. Fig. 11 shows autocorrelation for the same 12 grid locations for 36 hours. We observe a high autocorrelation at 24 hour period indicating similar pollution traits at the same time next day. We further observe that most locations (except A and B) exhibit a local high autocorrelation in the sub 12 or 8 hour periods as well, indicating repeated traffic patterns, like similar pollution characteristics around the morning and evening periods.

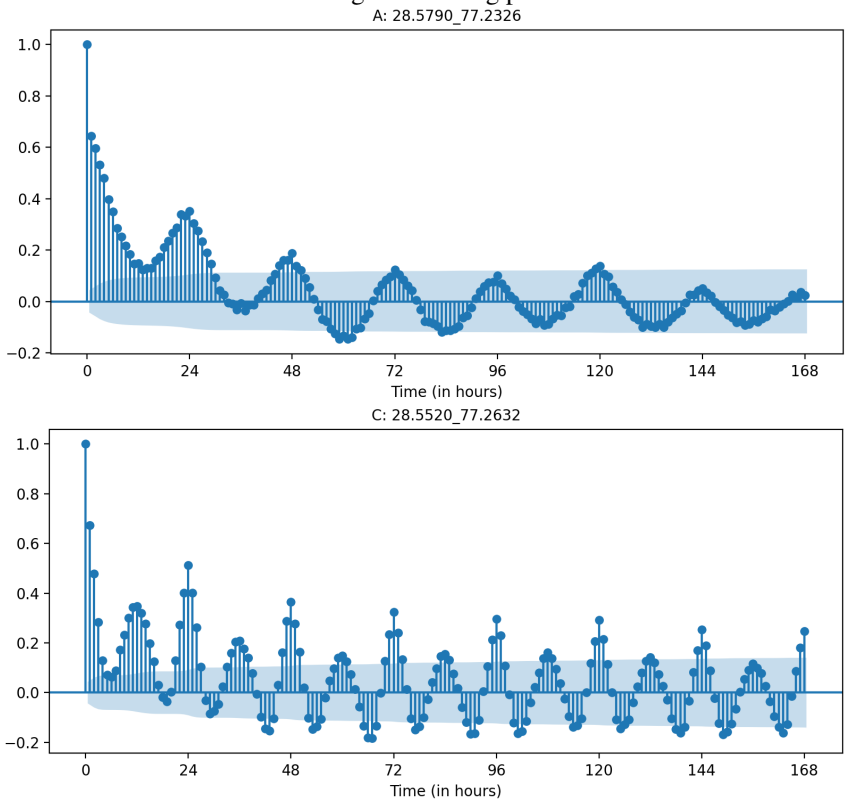


Figure 12: autocorrelation for A and C grid locations for up to 7 days lags. The autocorrelation decreases with increasing the lag.

The locations A and B seem to exhibit behaviour different from other locations, with low autocorrelation in the sub 12 hours period. Both these locations are adjacent to *Kushak Nalla Bus Depot* which hosts the buses for stopping between the runs and for overnight stopping while the bus services are down. Hence more data is collected here for longer periods which shows different pollution traits compared to all other grid locations. For grid locations A and C, Fig. 12 shows the autocorrelation for the two distinct behaviour for 7 days lags. The autocorrelation for 7 days for grid locations A and C is also presented as heat-map in Fig. 13.

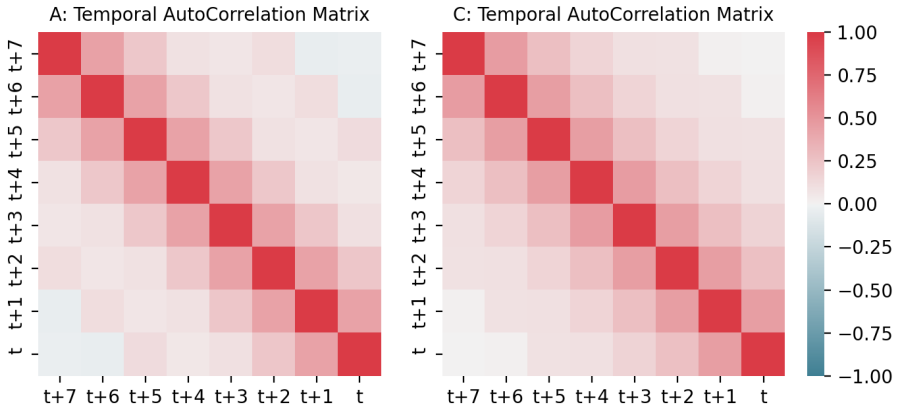


Figure 13: autocorrelation for A and C grid locations for up to 7 days lags as heat-map. The autocorrelation decreases with increasing the lag.

We also analyze the covariance among the spatial grid locations. In Fig. 14, we observe low or different covariances for locations at same distance from the base location. Hence there are different pollution traits at different locations, which sometimes match with neighboring locations due to common local/global factors, and sometimes differ due to some local factors.

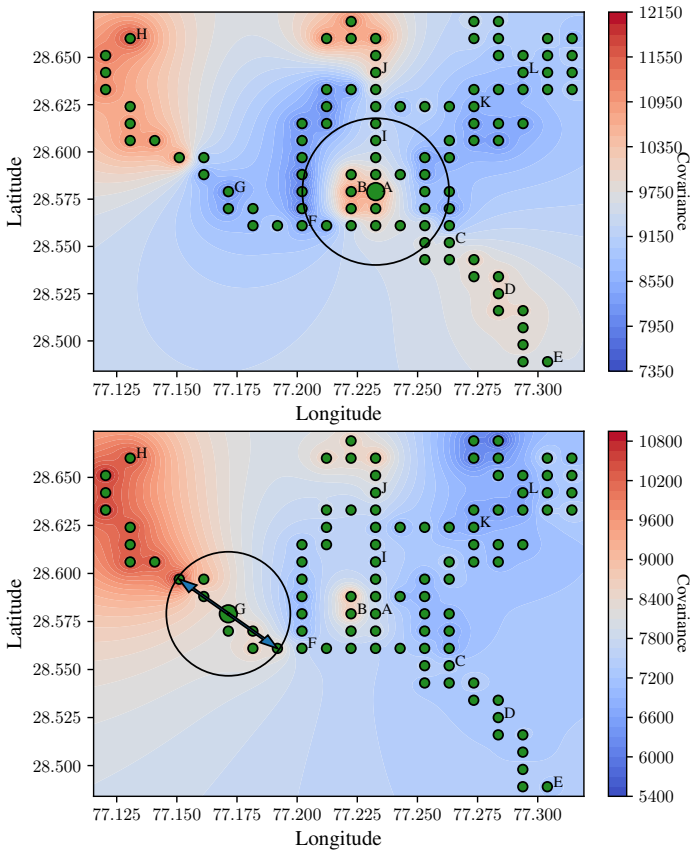


Figure 14: Interpolated empirical covariance between a base locations and other (99) grid locations in Delhi dataset. In the first plot, all the locations around the base location A has very low covariance with A. In the second plot, the locations at same distance from the base location G have different covariance w.r.t. G.

E Meteorological Factors (Temperature, Relative Humidity, Wind Speed) Analysis

A recent research Yang *et al.* [2020] focuses the effect of meteorological factors on the pollution traits in Shenyang, China to understand temporal-spatial characteristics of particles and analyze the causality factors. Similarly, we analyzed the given dataset for the impact of the collected meteorological parameters - Temperature and Relative Humidity (RH).

We also purchased Wind Speed (WS) data for the 3 months (Nov 2020 - Jan 2021) from *www.windfinder.com*, and received data with half-hourly frequency for IGI Airport. Data has wind speed in knots varying from 0 to 19, with maximum daily average being $\sim$7 knots. This wind data is not very fine, it has compromised precision with integer values only with some values missing. Still we could use this to analyze the effect of wind speed on the pollution.

Table 10: Correlation of Temperature, RH and WS with PM values.

	Temperature	Relative Humidity (RH)	Wind Speed (WS)
PM_1	-0.305	0.323	-0.508
$PM_{2.5}$	-0.303	0.332	-0.480
PM_{10}	-0.317	0.335	-0.477

Table 10 shows the correlation of temperature, humidity and WS with PM values. Overall we observe a negative correlation of PM with temperature, positive correlation of PM with RH, and negative correlation of PM with WS. Other researchers, like Yang *et al.* [2020]; Liu *et al.* [2020]; Avdakovic *et al.* [2016], have also observed similar positive and negative correlations, which is intuitive as temperature expands the air reducing PM per unit volume, humidity/moisture intensifies it, and wind blows it away. However, at a given time, which meteorological factor among these three will dominate needs more complex ML modeling.

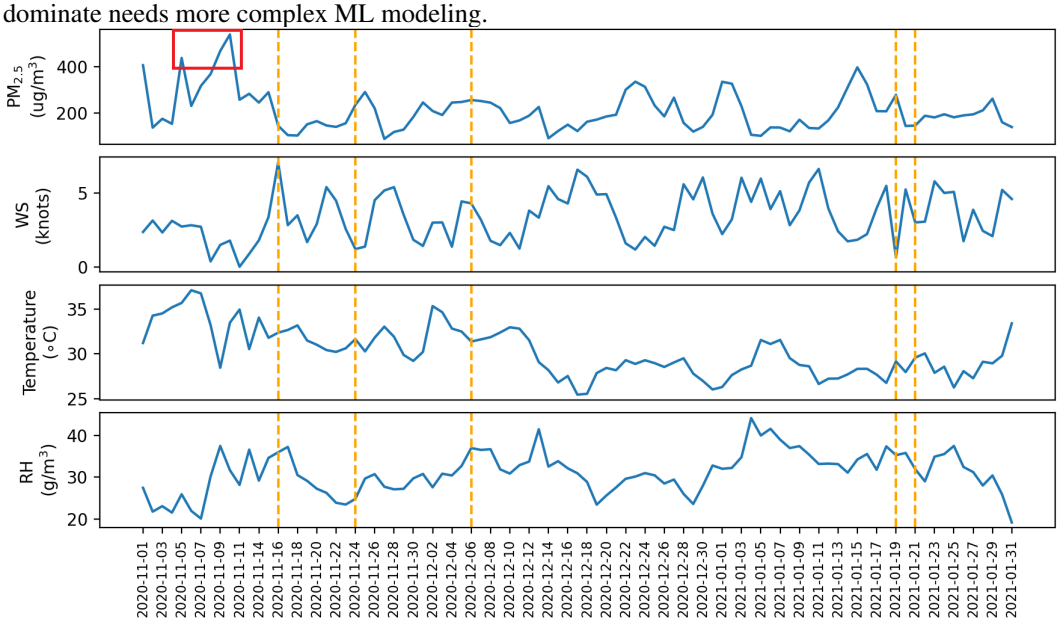


Figure 15: Meteorological factors vs $PM_{2.5}$ over the 3 months. The orange vertical lines show the relation of high or low $PM_{2.5}$ with WS, temperature and RH. The red box shows the situation of very high $PM_{2.5}$ which is explained with non-meteorological causes.

Fig.15 shows the average wind speed, temperature and humidity for the 3 months with the corresponding average $PM_{2.5}$ values. We observe high WS with low PM on Nov 16 and low WS with high PM on Nov 24 and Jan 19, which matches the intuition. But there are adverse situations of high WS with high PM on Dec 6, and low WS with low PM on Jan 21 which doesn't match the intuition.

The initial days of November show very high pollution spikes with stable winds. As per Weather [2020], there was *high moisture, calm winds and stubble burning around the beginning of November 2020*. As per HT [2020], in 2020, Delhi had six consecutive *severe* days from November 5-10, the longest *severe* spell seen in the city since 2016. A combination of multiple factors affected this, including a prolonged and intensive stubble burning season that started early on and in high incidence, the firecracker bursting festival and unfavourable meteorological conditions.

Due to such severe pollution situation in Delhi-NCR around winters, CAQM [2022] revised their action plan for **(a)** *Very Poor* Air Quality to avoid dust generating construction activities during months of October to January, and **(b)** *Severe* Air Quality to instruct individual house owners to provide electric heaters to security staff to avoid open burning.

Besides the meteorological parameters provided with the dataset, external data like below can be useful while modeling -

1. **Meteorological data from ERA5:** `https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels`
2. **NASA Fire count (VIIRS):** `https://firms.modaps.eosdis.nasa.gov/active_fire`
3. **Pollution Data from other sources:** OpenAQ: (`openaq.org`), CPCB: (`cpcb.nic.in`)
4. **Photochemical modelling:** `www.camx.com`
5. **Planetary boundary layer height:** `www.nrsc.gov.in/readmore_atmosphere_planet`
6. **Traffic Data from the dataset duration:** `delhi-trafficdensity-dataset.github.io`
7. **ClimateLearn: state-of-the-art climate-data/ML-models framework:** Nguyen *et al.* [2023b]
8. **ClimaX: flexible and generalizable weather/climate deep learning model:** Nguyen *et al.* [2023a]

F Bus Route Analysis

We analyzed the Delhi dataset based on the different (13) bus routes available in the data. This is shown in Fig. 16 with the overall $PM_{2.5}$ colour-map.

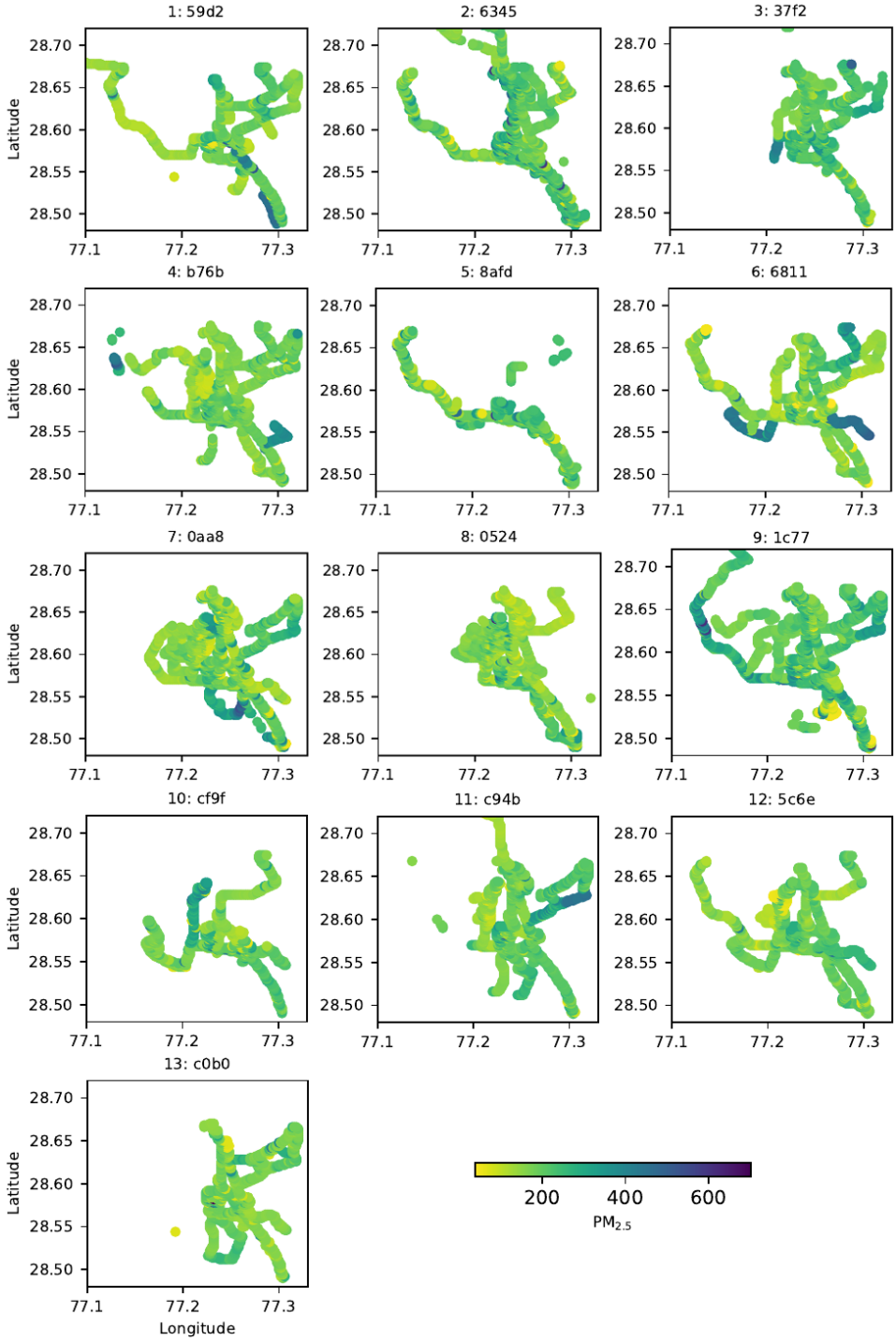


Figure 16: $PM_{2.5}$ distribution for the bus-routes followed over 3 months.

The title of each subplot shows the last 4 characters of the bus-route number from the dataset, with Latitude on Y-axis and Longitude on the X-axis. In different routes, there are sub-routes which showed high $PM_{2.5}$ concentrations in this duration, which can be due to high traffic concentrations

and or other local factors. Such route level $PM_{2.5}$ concentrations can be used to recommend people which routes to choose for commute/travel. Such and many other pollution traits can be further analyzed by using our dataset in conjunction with other open source data.

We further checked the daily $PM_{2.5}$ concentrations across the 13 routes. Fig. 17 shows the mean $PM_{2.5}$ values per day, with $PM_{2.5}$ levels on the Y-axis and day number on the X-axis. The number on the right axis in each subplot denotes the bus-route number.

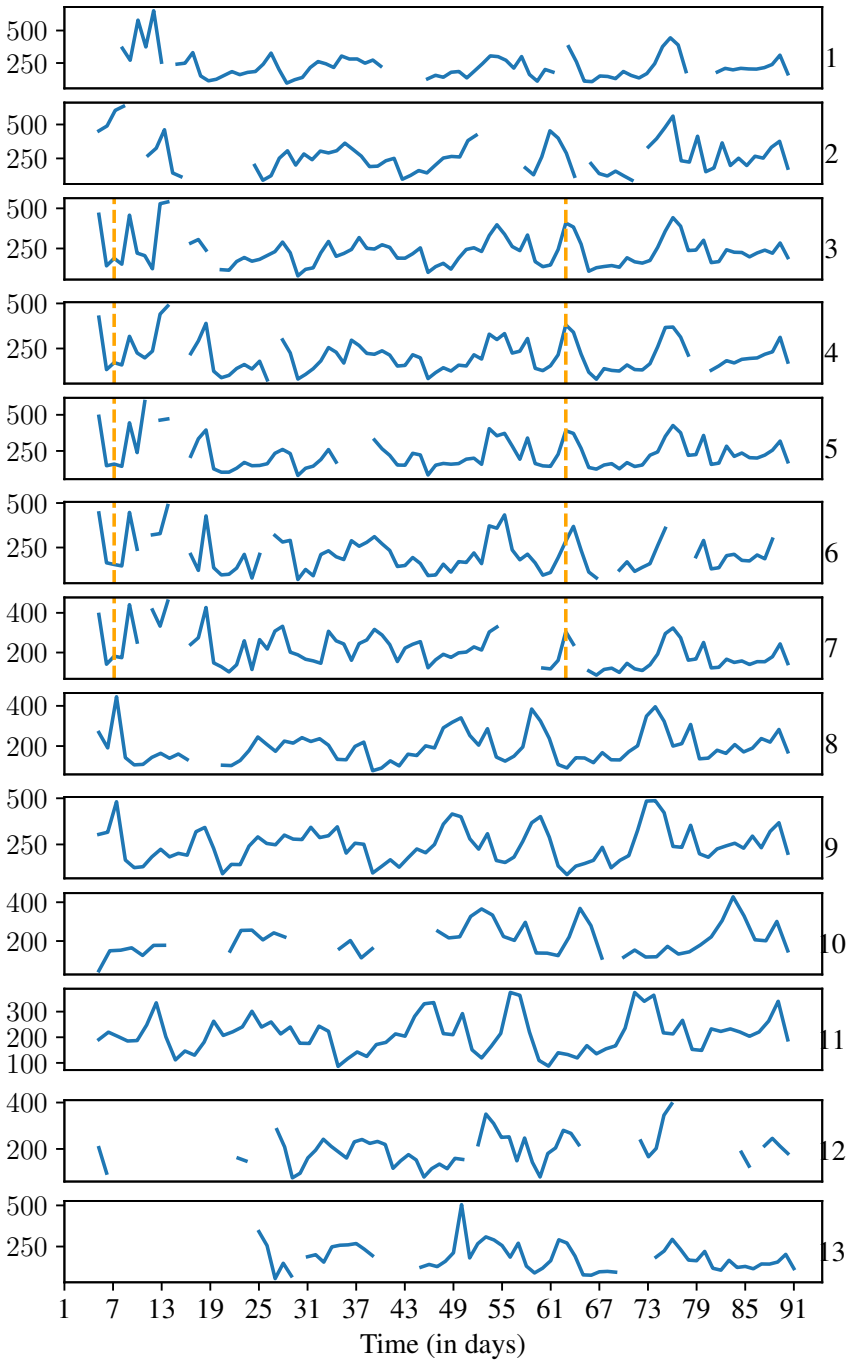


Figure 17: Daily $PM_{2.5}$ averages (denoted on left axis) for the different routes (mentioned on right axis) for the 3 months (x-axis). We observe some common low and high pollution periods, due to sharing of sub-routes by the buses.

As observed around the vertical lines in orange colour in Fig. 17, the different bus-routes seem to exhibit similar pollution traits due to the common paths tracked by them. We also observe the difference among different routes and hence a formal correlation analysis was performed to get better characteristics on the behaviour.

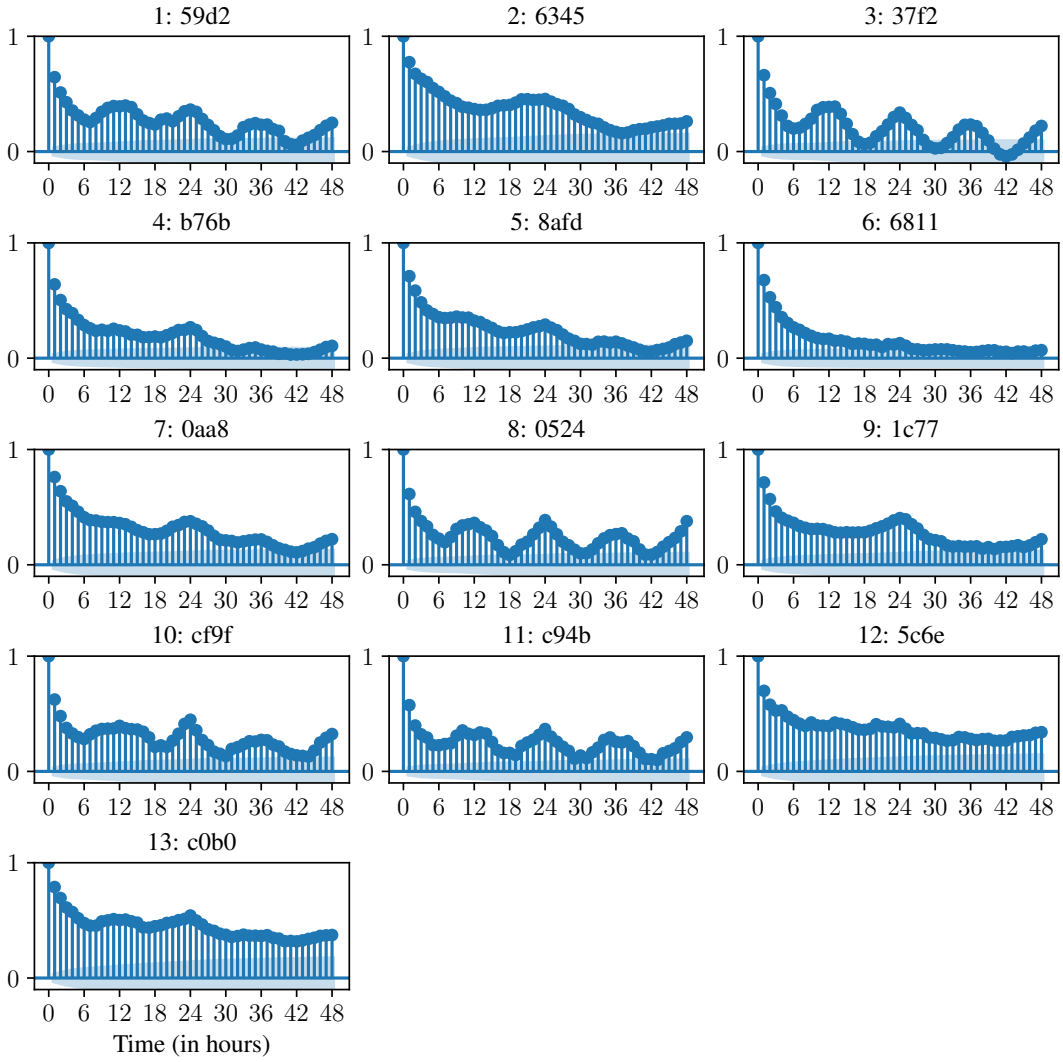


Figure 18: autocorrelation for the 13 bus-routes, for upto 2 days (48 hours) lags over hourly $PM_{2.5}$ averages. We observe positive autocorrelation in the bus-routes.

We performed the autocorrelation for the 13 bus-routes separately for hourly $PM_{2.5}$ averages. Similar to the combined route analysis done in Appendix D, we observe autocorrelation for separate bus-routes for 24 hour lags. As seen in Fig. 18, the level of autocorrelation is not same for all bus-routes, highlighting influence of local factors affecting pollution in their transit paths. Similar to overall correlation in Appendix D, we also find correlations at sub 12 hour lags as well for some of the bus-routes. autocorrelation for the 3 types of bus-routes, for upto 7 days lags over hourly $PM_{2.5}$ averages is shown in Fig. 19. We observe both positive and negative correlation in bus-route 3, almost no long-term correlation in bus-route 6 and varying but positive correlation in bus-route 11.

In contrast to the grid level autocorrelation analysis performed in Appendix D, we observed less negative autocorrelation for most of the bus-routes, which seems due to the same paths being traversed at same time each day by the buses.

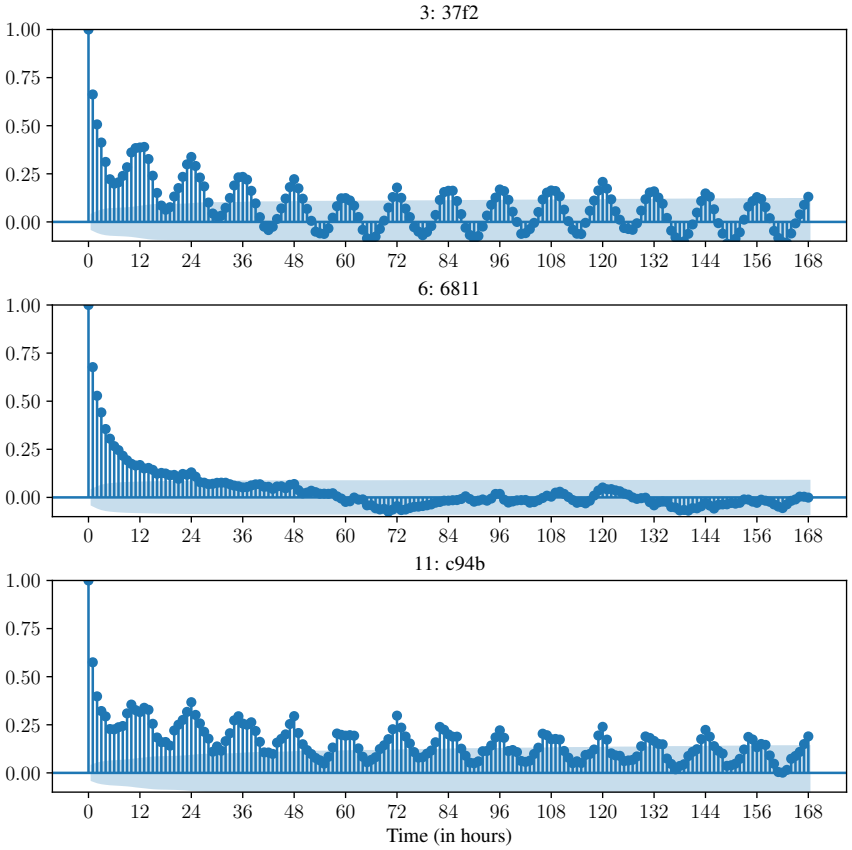


Figure 19: autocorrelation for the 3 types of bus-routes, for upto 7 days lags over hourly $PM_{2.5}$ averages. Both positive and negative correlation in bus-route 3, almost no long-term correlation in bus-route 6, varying but positive correlation in bus-route 11.

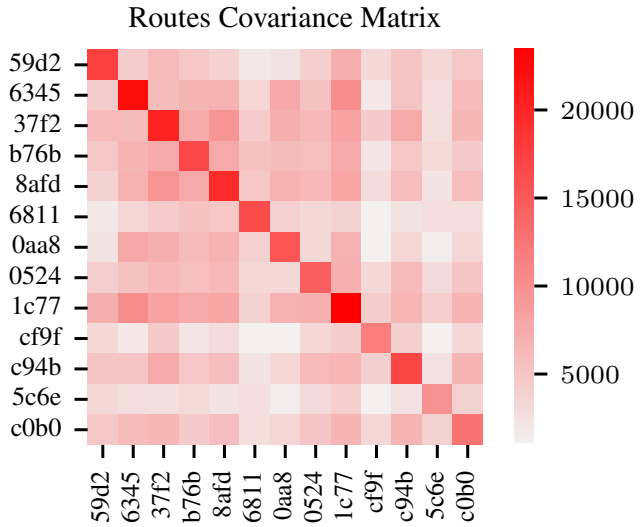
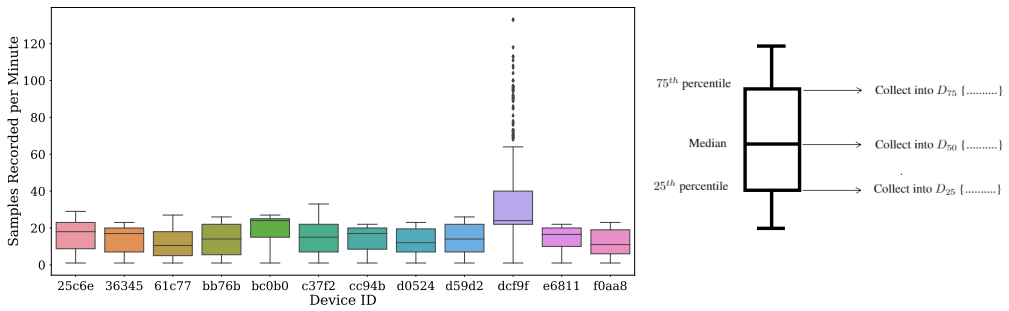


Figure 20: Buses share sub-routes hence some level of correlation exists between different bus routes.

Encouraged with the autocorrelation among different routes, we check the covariance among them, taking hourly $PM_{2.5}$ averages similar to autocorrelation. In Fig. 20, we can observe some average covariance among the different routes, while some of them show very low covariance. As Buses transit through different areas of the city, their corresponding data may contain peculiar characteristics for the area. A thorough analysis to understand those peculiar characteristics in contrast with other areas can reveal significant spatial traits for the Delhi region.

G Anomaly Detection

This dataset has been created using a novel IoT network with low cost sensor platform, deployed in public buses in a developing country, the first of its kind. There are many points of faults — sensors can be faulty, internet connection can be shaky, buses might be down .... the faults can affect the quantity of data as well as quality. Detecting such anomalies for quick fixes is a necessity. We applied statistical analysis to detect anomalies, which involves many heuristics with manually tuned thresholds. Our findings can serve as anomaly ground truth for this dataset. Automating this process with ML based methods (instead of manually tuned thresholds) can open up new avenues of anomaly detection in mobile and IoT networks. ML researchers can try and automate the fault detection process using our dataset and the ground truth anomalies. They can also modify our released code, to change our empirical thresholds for more or less aggressive anomaly definition. Additionally unsupervised learning methods perhaps would change the anomalies detected manually by us. We describe next the different anomaly metrics that we compute on the dataset using statistical analysis and empirically determined thresholds.



(a) (b)

Figure 21: (a) Sampling rate i.e. number of samples recorded per minute for 3.12.2020. This helps us in finding out if any device isn't sampling properly. (b) Illustration of a sample box plot and process of collecting median, 25^{th} percentile and 75^{th} percentile in metric 1 and 2.

Anomaly metric 1: Samples recorded per minute: This metric checks for faulty devices which might be sampling more or less than expected rate. Fig. 21a shows ideal samples collected per minute should be around 20. If it deviates too much, that device is anomalous. The amount of deviation allowed is calculated statistically by observing the distributions for several days. Our algorithm (detailed in the supplementary section) finds the upper bounds and lower bounds of the median (Θ^L_{50}, Θ^U_{50}), 25^{th} percentile (Θ^L_{25}, Θ^U_{25}) and 75^{th} percentile (Θ^L_{75}, Θ^U_{75}) of the expected distribution. Anomaly is reported if any two of the three bounds are violated.

Anomaly metric 2: Number of minutes each device is active in an hour: A device can be active for all the 60 minutes of an hour or less based on time of the day/lunch break, stoppage at bus depots etc. So again we tried plotting the box plots for the distributions across the days and devices. We observe that ideally device should be active for 60 minutes of an hour, if the bus was taking a trip in that hour. So we used the same technique used in Metric 1: find the upper bounds and lower bounds of the median (Θ^L_{50}, Θ^U_{50}), 25^{th} percentile (Θ^L_{25}, Θ^U_{25}) and 75^{th} percentile (Θ^L_{75}, Θ^U_{75}) of the expected distribution. Anomaly is reported if any two of the three bounds are violated.

Anomaly metric 3 : Number of active hours in a day: An active hour for a particular sensor is any hour in which the sensor sends at least a fixed number(γ) of samples. The number of active hours should ideally be greater than a threshold value(τ). But it is hard to fix one τ across all sensors, as different buses have different schedules and frequencies, which also can change over time. This is shown by the bar plots of number of active hours in Fig 22.

So, we define τ or ideal number of active hours for every sensor as the maximum of 10 active hours and 15^{th} percentile of the sensor's previous 15 days' active hours. This ensures that τ is appropriately chosen for every sensor depending on its particular bus's recent schedule. If the number of active hours for a sensor on any day doesn't satisfy the threshold (τ), it is reported as anomalous for the day.

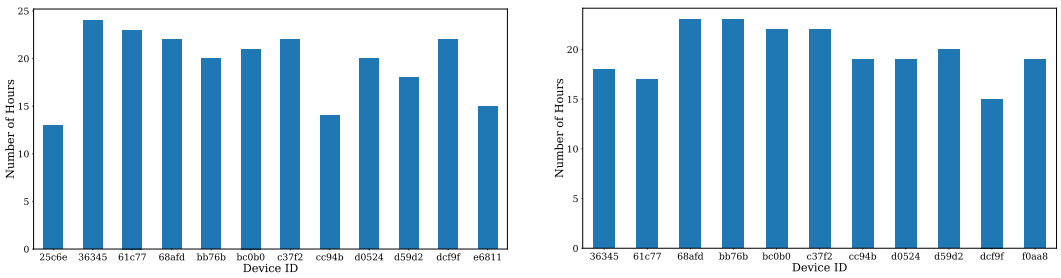
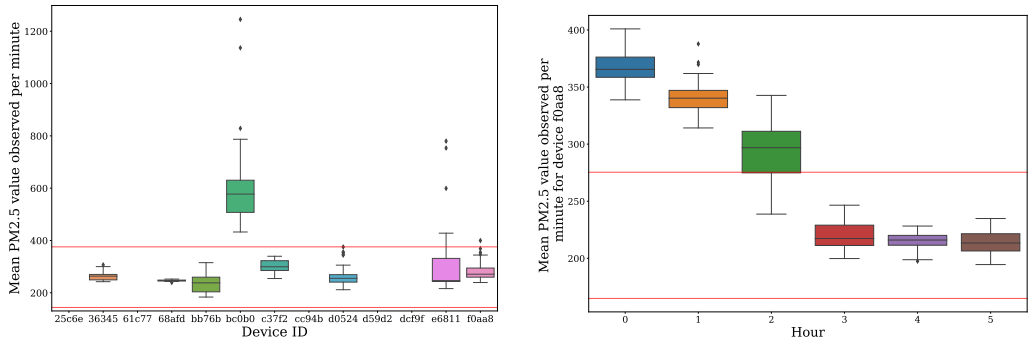


Figure 22: Number of active hours for different devices on two different days as shown by two different plots. We can see that the number of active hours on both days for each device are different.

Anomaly metric 4: Samples recorded per region: It is important to check whether daily around same number of data points are collected or not in an area. This metric detects situations where bus may not complete its scheduled trip due to mechanical breakdowns or high traffic resulting in less recording of data points in some areas. We divided the area covered by buses into 16 square regions. Given total number of data points collected in each region on a day, a region is reported as anomalous if its value deviates from past seven days average of that region by at least $\delta\%$. The value of δ is calculated by observing the data of several days.

Anomaly metric 5: Inter-sensor PM values variation: Ideally the PM values measured by different sensors should lie in a close range if the measurements were carried out at the same location and time. Every night from 0 AM IST to 5 AM IST all the buses remain parked at the same bus depot. We have used the PM value data from this time period to find devices whose PM 2.5 value measurements deviate from the general PM value trend of majority devices. For each hour, we have a box plot describing PM value distributions of all the devices as shown in Fig. 23a. Let Θ_{25} and Θ_{75} represents 25th and 75th percentile of distribution of PM values of a device during an hour. Interquartile range (IQR) is defined as (Θ_{75}-Θ_{25}). Given a box plot, a device is flagged for possible anomalous behaviour if it's IQR is very high (e.g. device e6811 in Fig. 23a). In order to define how much IQR should be considered high to be flagged, we define a threshold max_IQR which is set as 90 percentile of all the IQRs of all the devices in training set. Secondly, if a box (middle 50% data) of a device in the plot varies in a range different than the range of other devices then also the device is flagged (e.g. device bc0b0 in Fig 23a). To find such anomalies, we first find a range in which boxes of majority devices lie, then all those devices which are out of this range are flagged. A parameter called 'buffer' is defined statistically based on training data for finding the range. Given a PM value distribution of an hour, we iteratively calculate candidate range as [Θ_{25}-buffer, Θ_{75}+buffer] for each device. The candidate range which contains the boxes of maximum number of devices is considered as the final range. The devices whose box does not fit completely in this range are flagged for that hour. Finally a device is reported as anomalous if its get flagged for at least three hours in a day.



(a) (b)

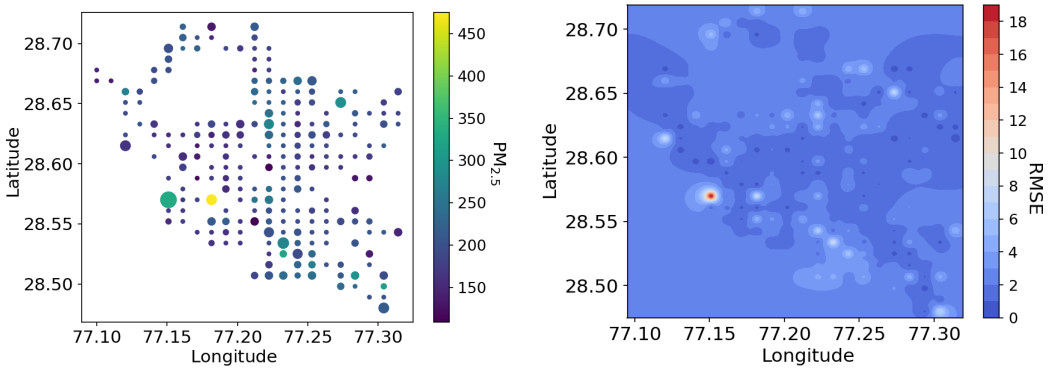
Figure 23: (a) shows a box plot of PM value distributions of various devices on 2020-12-20 4:00 AM IST. Red lines indicate the majority PM range. Device bc0b0 is flagged as it is out of majority PM range while device e6811 is flagged as its IQR exceeds max_IQR. They will be declared as anomaly if they show this behaviour in at least two more hours. (b) shows PM value distributions of the device f0aa8 on 2021-01-16. The device is reported as anomaly as its PM value measurements are highly varying across several hours.

Anomaly metric 6: Intra-sensor PM values variation: Similar to the above metric, intra sensor analysis verifies that the variation in a device's PM value recordings across consecutive hours is not very high. Given a device's PM value recordings during 0 AM IST to 5 AM IST, it is flagged for further checks if IQR of the device during any hour is greater than max_IQR or if at least three boxes lie out of majority PM range. The majority PM range is the PM value range which contains the maximum number of boxes computed similarly as described in the above metric except that the value of buffer here is computed based on intra senor PM value distributions. Finally a device is reported as anomalous only if its get flagged for at least three hours in a day. Figure 23b shows one such anomaly.

We detail the heuristics for computing the above six anomaly metrics, the thresholds and summary statistics of all anomalies found, in the supplementary section and the website. The anomalies found in the paper were cross-checked with the platform vendor Aerogram and the deployment partner, the public bus company DIMTS, for correctness and usefulness. All cases on inter-sensor and intra-sensor variations (metrics 5 and 6) were caused by local electrical maintenance work in a particular bus at the depot, whose sensor readings deviated from other buses in the depot. Lack of samples per minute or per hour (metrics 1 and 2) are helping to understand 4G networking issues. Finally the metrics for active hours per day and spatial coverage consistency (metrics 3 and 4) are helping to gain insights on unpredictable public bus behavior in Delhi, especially during Covid-19 induced lockdowns, where bus schedules and routes are seeing significant variations. Thus all these anomalies are highly important to gain insights about a live IoT network deployment. Defining these metrics and the multiple thresholds for them has been cumbersome, and more automated ML methods using this dataset and our findings as ground-truth, will be immensely valuable.

H Miscellaneous

We also observed the output of Spatio-temporal Interpolation using Random Forest (RF) algorithm to see the distribution of RMSE for the predictions. Fig. 24 shows some locations with large error, which needs indicates need of special handling.



(a) (b)

Figure 24: RMSE of the Spatio-temporal Interpolation using RF. (a) bigger circle denotes bigger RMSE for the spatial location while modeling. (b) shows the RMSE distribution. Few locations are hard to model.