

# AirShot: Zero-shot Recommendation for Pollution Sensor Locations under Budget Constraints

Anonymous Author(s)

## Abstract

Air pollution is one of the biggest concerns faced by developing countries like India and the world at large. The capital of India, Delhi and the National Capital Region (NCR), sees life threatening air pollution levels. Pollution measurement at scale is critical, for correlation with causal factors and mitigation policy recommendations. Measurement is also necessary to inform citizens' decisions regarding suitability of outdoor exercise, use of face-masks or air purifiers, or choosing the least polluted routes for commute. 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 in Delhi-NCR, a mobile sensor network is engineered in this paper, mounting lower cost (and hence more noisy) PM 2.5 sensors on Delhi public buses. This novel PM 2.5 dataset is made public in this paper, for Machine Learning (ML) researchers and environmentalists. A crucial ML problem of recommending locations for incremental static sensor deployment is then analyzed. Using our mobile sensor dataset, our system AIRSHOT, trains an *inductive* model to strategically upgrade static sensor infrastructure, for minimal PM 2.5 estimation errors under budget constraints.

## 1 Introduction and Related Work

Air pollution has reached life-threatening levels in Delhi-National Capital (NCR) Region, India [IndiaToday, 2019], which is one of the most densely populated urban centers. The population of Delhi-NCR exceeds 46 million people [EconomicTimes, 2019] and it has been reported that 50% of all children staying in this region suffer from irreversible lung damage [ORF, 2021; Gizmodo, 2015; Wikipedia, 2021]. *Particulate Matter (PM)* is especially dangerous, since 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.

In the literature, several models have been built on predicting pollution levels at future time points [Gao and Li, 2021; Kurt *et al.*, 2008; Tsai *et al.*, 2018; Le *et al.*, 2020],

and identifying factors affecting pollution levels [JS *et al.*, 2011; Google, 2014; JS *et al.*, 2018; SE *et al.*, 2018]. A basic necessity for the functioning of all such models is access to fine-grained pollution data. Unfortunately, collecting pollution data is highly expensive (thousands of US Dollars per instrument). Hence, the density of pollution tracking sensors in developing countries is often low and insufficient for developing and validating sophisticated machine learning models.

To give concrete statistics, the *Central Pollution Control Board (CPCB)* and *Delhi Pollution Control Committee (DPCC)* have only 37 air pollution measurement centers in Delhi-NCR, which are thoroughly inadequate to cover the vast geography of 55,000 square Kms. There exists *interpolation models* [Qiao *et al.*, 2019; Rasmussen and Williams, 2005; Hamilton *et al.*, 2017] to reliably predict pollution levels at unseen locations based on a sufficient number of pre-installed sensors. However, interpolation models themselves are *supervised* in nature and hence cannot be designed in the absence of a sufficient amount of training data.

In this work, we aim to mitigate this data problem in a cost-effective manner. To elaborate, our goals are three fold. First, we would like to design a low-cost sensing mechanism (and potentially more noisy than high cost sensors) that allows us to collect PM data over a subset of the Delhi-NCR region at a fine spatio-temporal granularity. Second, use this data to not only train interpolation models, but also train an *inductive* model to strategically upgrade the high-cost (and therefore high-quality) PM sensing infrastructure in the covered region within a budget constraint  $b$ . The optimization problem here is to identify  $b$  strategic installation locations for high-cost sensors so that the performance of the interpolation model is maximally improved. Finally, apply the inductive model over the *entire* Delhi-NCR region to *infer* the optimal infrastructure upgradation strategy.

The proposed problem encompasses both engineering as well as algorithmic challenges. Creating the mobile PM dataset required us to design and implement our own embedded platform, choosing and calibrating appropriate sensors for maximum accuracy at low cost (see Fig. 1a and Fig. 1b). Packaging was challenging to securely mount the instruments in public buses, avoiding theft and ensuring enough ambient air to measure PM (see Fig. 5). Cellular connectivity was intermittent as the buses traversed the city, requiring us to augment real time data transfer when signal was present, with lo-

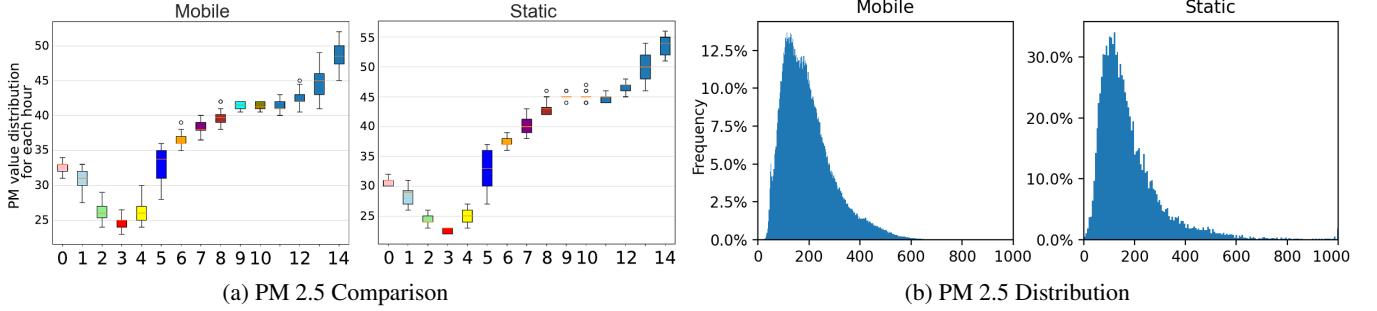


Figure 1: (a) PM values measured by our low-cost mobile PM sensor (USD 30) on the left vs. TSI DustTrak (USD 9500) on the right on a sample day Jul 24, 2021. The values are almost identical. (b) Distribution of PM 2.5 collected by Mobile and Static sensors. The distributions are similar across the two sets of instruments.

cal 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 the bus. Algorithmically, while several models exist to perform PM interpolation, none exist on mechanism to update infrastructure that maximally improves interpolation performance. In summary, the key contributions of our work are as follows:

- **Dataset:** We establish a vehicle-mounted PM sensing network and release the largest PM<sub>2.5</sub> dataset from one of the most polluted regions in the world. We 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 is of immense value to pollution researchers and environmental think tanks (§ 2).
- **Problem formulation:** We propose the novel problem of *learning a policy* for budget-constrained sensor placement for pollution monitoring (§ 3).
- **Algorithm:** We design a framework called AIRSHOT that uses *policy-gradients* to learn the optimal upgradation strategy. AIRSHOT is *inductive* and hence enables *zero-shot policy transfer* on unseen data (§ 4).
- **Experiments:** Through extensive experiments on the largest collection of real PM data collected over a span of three months, we demonstrate the efficacy of AIRSHOT and its superiority over baseline strategies (§ 5).

## 2 Low-cost, High-granularity PM Sensing

To collect low-cost, high-granularity pollution data, we have mounted pollution tracking sensors on 13 public buses in Delhi for 3 months (Nov 1<sup>st</sup>, 2020 to Jan 31<sup>st</sup>, 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. The inside of our custom-made instrument comprising (a) PM sensor measuring PM 2.5, PM 10 and PM1, (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 low, (e) BME sensor to record temperature and relative humidity and (f) micro-controller to orchestrate the sense-store-communicate software (See Fig 5(a) in the appendix). The mounting location in the bus driver's cabin, next to two open windows to allow enough air-flow

(Fig 5(b)-(c)). Each bus commutes for 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 sq Km, along the main arterial roads in North-West, North, North-East and South-East Delhi.

The dataset has been cleaned and made available at *{hidden URL for author anonymity}* with proper documentation. This dataset complements the static sensor data available from the 37 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. Environmentalists can therefore use these complementary datasets to understand PM at tower heights using static sensor data, as well as near ground using our mobile sensor data. We next present statistics on the dataset quality and its uniqueness when compared to other PM datasets.

### 2.1 Data Quality

Fig 1a plots a low cost PM sensor (cost USD 30) on the left, and an industry grade reference instrument TSI DustTrak (cost USD 9500) on the right, showing hours of the day along x-axis and sensed PM 2.5 values along y-axis, for a sample day July 24, 2021. While the cost gap between the instruments is huge, the gap between their sensed PM 2.5 values, as seen in this graph, 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; Zheng *et al.*, 2019].

We next compare the distribution of PM values recorded by our mobile sensors vs. those by the high-cost static sensors. Fig. 1b reveal that both sets show similar PM distributions, in spite of the difference in heights they have been installed at, the difference in the amount of exposure of the sensor to direct smoke and dirt, and the difference in PM measurement technique. Further details on data quality analysis is provided in App. A.

### 2.2 Data Characteristics

Tables 1 and 2 summarize the various statistics of the dataset collected. While vehicle mounted air pollution sensing has been conducted [JS *et al.*, 2011; Google, 2014; JS *et al.*, 2018; SE *et al.*, 2018; Guo *et al.*, 2016; Adams and Corr,

Metric	Delhi-NCR	Canada
Total area	559 sq. km	1138 sq. km
Total samples	12,542,183	46080
Samples with PM2.5	12542183	12154
Pollutants covered	PM1, PM2.5 and PM10	CO, NO, NO <sub>2</sub> , SO <sub>2</sub> , O <sub>3</sub> , PM1, PM2.5 and PM10
Vehicles used	Public bus	Commercial van
Monitoring days	91	114

Table 1: Details of Delhi and Canada datasets.

Metric	Delhi-NCR			Canada		
	PM1	PM2.5	PM10	PM1	PM2.5	PM10
Mean	120.35	207.92	226.11	46.45	15.08	12.15
Std-dev	57.27	114.36	123.86	97.36	12.87	9.02
Missing %	0	0	0	72.24	73.62	71.71

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

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.

### 3 Budget-constrained Sensor Placement: Problem Formulation

A spatial location is denoted as  $\ell = (x_\ell, y_\ell, f_\ell)$ , where  $x_\ell$  and  $y_\ell$  denotes the location attributes such as latitude and longitude and  $f_\ell \in \mathbb{R}^d$ , is a  $d$ -dimensional feature vector characterising the location neighborhood (the region within  $r$  kms from  $\ell$ ) based on pollution-affecting features such as proportion of green cover, population density, industry presence, etc. A spatio-temporal point as  $p = (\ell, t)$  where  $t$  denotes the time. A pollution tracking sensor records tuple of the form  $\langle p, PM(p) \rangle$ , where  $p$  is a spatio-temporal point and  $PM(p)$  is the PM value at  $p$ . For a static sensor  $S$ , the spatial location is fixed, which we denotes using  $S_\ell$ .

**Definition 1** (Interpolation Model). *A PM interpolation model  $\mathcal{M}$ , trained on pollution data recorded by a collection of static PM sensors  $\mathcal{S}$ , takes as input a query spatio-temporal point  $q = (\ell, t)$  and predicts the PM value at  $q$ . We denote the instance of  $\mathcal{M}$  trained on  $\mathcal{S}$  as  $\mathcal{M}_\mathcal{S}$  and its prediction on  $q$  as  $\mathcal{M}_\mathcal{S}(q)$ .*

Several interpolation models for pollution data exists in the literature [Gao and Li, 2021; Le *et al.*, 2020; Qiao *et al.*, 2019; Rasmussen and Williams, 2005; Kurt *et al.*, 2008; Tsai *et al.*, 2018]. In general, interpolation models predict the PM at  $q$  as a function of the locations of nearby sensors and their PM values in that time slot. The loss on a prediction is quantified by typical loss functions such as RMSE, MAE etc. Without loss of generality, we assume RMSE to be the loss, i.e.,

$$\mathcal{L}(\mathcal{M}_\mathcal{S}, q) = \sqrt{\mathcal{M}_\mathcal{S}(q)^2 - PM(q)^2} \quad (1)$$

where  $PM(q)^2$  is the true PM value. In the context of our problem, the ground truth data is provided by the vehicle-mounted sensors and the static sensors correspond to the

### Algorithm 1 Training pipeline of AIRSHOT.

**Require** Data from static sensors  $\mathcal{S}$ , mobile sensors  $\mathcal{Q}$ , interpolation model  $\mathcal{M}_\mathcal{S}$

- 1:  $\mathcal{P} \leftarrow \{(\mathcal{S}_1, \mathcal{S}_2) | \mathcal{S}_1 \subseteq \mathcal{S}, \mathcal{S}_2 \subseteq \mathcal{S}\}$   $\triangleright$  Random partitioning instances of  $\mathcal{S}$  into  $\mathcal{S}_1$  and  $\mathcal{S}_2$
- 2:  $\Pi \leftarrow$  initialize with random parameters
- 3: **while**  $\Pi$  has not converged **do**
- 4:     **for all**  $P = (\mathcal{S}_1, \mathcal{S}_2) \in \mathcal{P}$  **do**
- 5:          $\mathcal{A}^0 \leftarrow \emptyset$
- 6:          $\mathcal{M}_{\mathcal{S}_1} \leftarrow$  Interpolation model trained on  $\mathcal{S}_1$ .
- 7:         Sample  $b \sim [1, |\mathcal{S}_2|]$
- 8:          $t \leftarrow 1$
- 9:         **for**  $t \in [1, b]$  **do**
- 10:              $S^* \leftarrow \arg \max_{S \in \mathcal{S}_2} \Pi(S, \mathcal{M}_{\mathcal{S}_1 \cup \mathcal{A}}, \mathcal{Q})$
- 11:              $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \setminus \{S^*\}$
- 12:              $\mathcal{A}^t \leftarrow \mathcal{A}^{t-1} \cup \{S^*\}$
- 13:             Train  $\mathcal{M}_{\mathcal{S}_1 \cup \mathcal{A}^t}$
- 14:     Backpropagate to minimize loss using Eq. 9
- 15: **Return**  $\Pi$

high-cost sensors installed in area covered by the vehicle-mounters sensors. Hence, the total loss  $\mathcal{E}(\mathcal{M}(\mathcal{S}))$  is:

$$\mathcal{L}(\mathcal{M}_\mathcal{S}) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathcal{E}(\mathcal{M}_\mathcal{S}, q) \quad (2)$$

where  $\mathcal{Q} = \{q_1, \dots, q_n\}$  is the set of all PM2.5 recordings from vehicle-mounted sensors.

**Problem 1** (Budget-constrained infrastructure upgradation).

**Training:** *Given pollution data from static sensors  $\mathcal{S}$  and high-granularity vehicle sensor data  $\mathcal{Q}$ , learn an inductive active-learning model  $\Pi$  such that:*

**Inference:** *Given pollution data from a set of unseen sensors  $\mathcal{S}'$ , candidate locations  $\mathcal{C} = \{\ell_1, \dots, \ell_n\}$  where PM monitoring is desired and therefore static sensors may be installed, and a budget  $b$  where  $b \ll |\mathcal{C}'|$ , use  $\Pi$  to predict  $b$  locations from  $\mathcal{C}$  for installation of new sensors such that the marginal decrease in expected loss is maximized. Formally,*

$$\begin{aligned} \Pi(\mathcal{M}, \mathcal{S}', \mathcal{C}, \mathcal{Q}, b) &= \arg \max_{\mathcal{A} \subseteq \mathcal{C}, |\mathcal{A}|=b} \{\mathcal{L}(\mathcal{M}_{\mathcal{S}'}) - \mathcal{L}(\mathcal{M}_{\mathcal{S}' \cup \mathcal{A}})\} \\ &= \arg \min_{\mathcal{A} \subseteq \mathcal{C}, |\mathcal{A}|=b} \{\mathcal{L}(\mathcal{M}_{\mathcal{S}' \cup \mathcal{A}})\} \end{aligned}$$

### 4 AIRSHOT

Fig. 2 presents the basic training framework of AIRSHOT, which is detailed in Alg. 1. First, we split the sensors in the train set into two halves uniformly at random to create  $\mathcal{S}_1$  and  $\mathcal{S}_2$  (line 1). The pollution data of  $\mathcal{S}_2$  is hidden from us and we are only aware of their locations. Thus,  $\mathcal{S}_2$  simulates the candidate locations  $\mathcal{C}$  in the inference phase. We select a budget  $b \in [1, |\mathcal{S}_2|]$  and iterate for  $b$  steps (lines 7-13). In each iteration we select a sensor from  $\mathcal{S}_2$  based on a policy  $\Pi$  and retrain the interpolation model with this added sensor (lines 10-13). This process is then repeated over multiple partitions of  $\mathcal{S}$  (line 4). The key task here is in learning policy  $\Pi$ , where the optimal policy minimizes the loss over all partitions of  $\mathcal{S}$  through backpropagation. We next discuss the details of this procedure.

#### 4.1 Graph-based Modeling of Space

We model the geographical space over which prediction is being performed as an undirected, weighted graph  $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ . Specifically, we first collect all unique locations of PM recordings from  $\mathcal{S}$  and  $\mathcal{Q}$ . Each unique location corresponds to a node  $v \in \mathcal{V}$ . The edge set is defined as  $\mathcal{E} = \{e = (u, v) \mid$

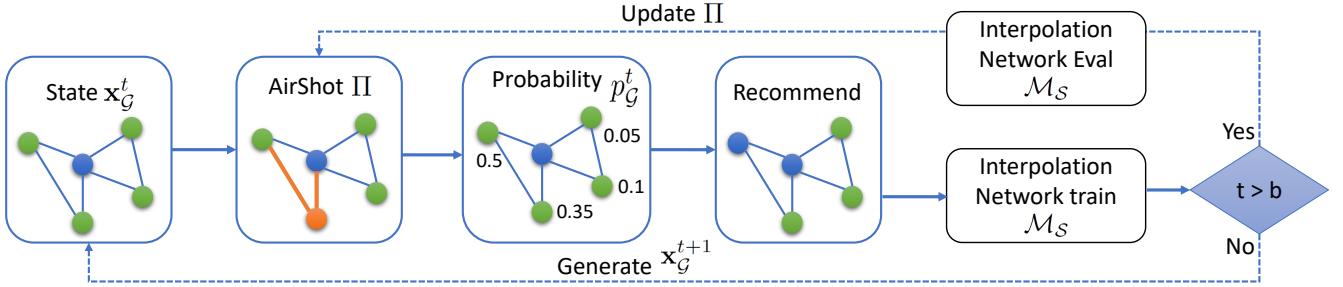


Figure 2: AIRSHOT framework. Blue are the sensor nodes from  $\mathcal{S}_1$ , and Green are from  $\mathcal{S}_2$ . For all nodes from  $\mathcal{S}_2$ , AIRSHOT gives recommendation probabilities, allowing  $b$  recommendations in total.

$u, v \in \mathcal{V}, d(e) \leq \theta\}$ , where  $\theta$  is a distance threshold and  $d(e)$  denotes the Haversine distance between its endpoints  $u$  and  $v$ . The edge weight function  $\mathcal{W} : \mathcal{E} \rightarrow \mathbb{R}$  is defined as  $\frac{1}{1+d(e)}$ . In our implementation, we set  $\theta = 1km$ . The pollution levels in two locations may be correlated if they are close to each other. The graph-based modeling of space aims to model this aspect by connecting nearby regions through edges.

## 4.2 Learning $\Pi$ as an MDP

The task of  $\Pi$  is to iteratively select  $b$  nodes (locations) that minimize the loss of  $\mathcal{M}$ . We model this task as a *Markov Decision Process (MDP)*. Specifically, the current performance of  $\mathcal{M}$  on  $\mathcal{S}_1$  corresponds to a *state*. The selection of a node from  $\mathcal{S}_2$  by  $\Pi$  is the *action*, and the decrease in loss is the *reward*. We next formalize this idea.

**State:** The state of the system at any time  $t$  is the collection of all node representations at time  $t$ . Recall from § 3 that each node (location)  $v$  is characterized a feature vector  $f_v$  characterizing the spatial neighborhood features. We further supplement  $f_v$  with four features:

1. **PageRank [Brin and Page, 1998].** The PageRank of a node  $v$  allows us to capture its importance in the context of the graph. A node with a high PageRank is likely to be centrally located and a sensor installed here may predict the PM in a large number of other nodes accurately.
2. **Interpolation loss:** The interpolation loss at  $v$  at time  $t$  is

$$IL^t(v) = \sum_{\forall q \in \mathcal{Q}, loc(q)=v} \mathcal{L}(\mathcal{M}_{\mathcal{S}_1 \cup \mathcal{A}^t}(q)) \quad (3)$$

$loc(q)$  denotes the node corresponding to PM record  $q$ . The higher the loss, the more is the need for a static sensor at this location.

3. **Loss discrepancy:** The loss discrepancy at  $v$  is the average deviation of loss at  $v$  with its neighbors. Specifically,

$$LD^t(v) = IL^t(v) - \frac{1}{|N_v|} \sum_{\forall u \in N_v} IL^t(u) \quad (4)$$

Here,  $N_v = \{u \mid (u, v) \in \mathcal{E}\}$ . A high Loss discrepancy indicates  $v$  is not well-correlated with its neighbors.

4. **Indicator variable:** The indicator variable indicates where a node  $v$  has been selected for sensor installation, i.e.,  $IV^t(v) = \mathbf{1}\{v \in \mathcal{A}\}$

The representation of a node  $v$  at time  $t$  is:

$$\mathbf{x}_v^t = f_v \parallel PageRank(v) \parallel IL^t(v) \parallel LD^t(v) \parallel IV^t(v) \quad (5)$$

Finally, the state of  $\mathcal{G}$  is  $\mathbf{x}_G^t = \{\mathbf{x}_v^t \mid v \in \mathcal{V}\}$ . While we use the above four features, one may use additional features as relevant for the application.

**Action:** At time  $t$ , the action corresponds to selecting a node  $v \in \mathcal{S}_2 \setminus \mathcal{A}^t$  from  $p_v^t \sim \Pi(v \mid \mathbf{x}_G^t)$ . We will discuss the computation of  $p_v^t$  in § 4.3.

**Reward:** Since we would like to minimize the loss of the interpolation model, the reward of the set of selected nodes  $\mathcal{A} = \mathcal{A}^b$  is the negative of the current loss, i.e.,

$$R(\mathcal{A}) = -\mathcal{L}(\mathcal{M}_{\mathcal{S}_1 \cup \mathcal{A}}) \quad (6)$$

**State transitions:** Once a node is selected for inclusion in  $\mathcal{A}^t$ , the interpolation model  $\mathcal{M}$  is updated based on the new static sensor. Following this update, the time dependent features of each node, i.e.,  $IL^t(v)$ ,  $LD^t(v)$ , and  $IV^t(v)$  are updated and the next round of node selection is performed.

## 4.3 Neural Architecture for Policy Training

To learn  $p_v^t$ , we take the initial feature representations  $\mathbf{x}_v^t$ , and pass it through an  $K$ -layer *Graph Convolutional Network* (GCN) [Hamilton *et al.*, 2017] to learn a distribution over all nodes. Formally, let the  $\mathbf{h}_v^0 = \mathbf{x}_v^t$ . In each layer  $k \in [1, K]$ , we perform the following transformation:

$$\mathbf{h}_v^k = \sigma \left( \mathbf{W}^k \left( \mathbf{h}_v^{k-1} \parallel \sum_{\forall u \in N_v} w(e = (u, v)) \mathbf{h}_u^{k-1} \right) \right) \quad (7)$$

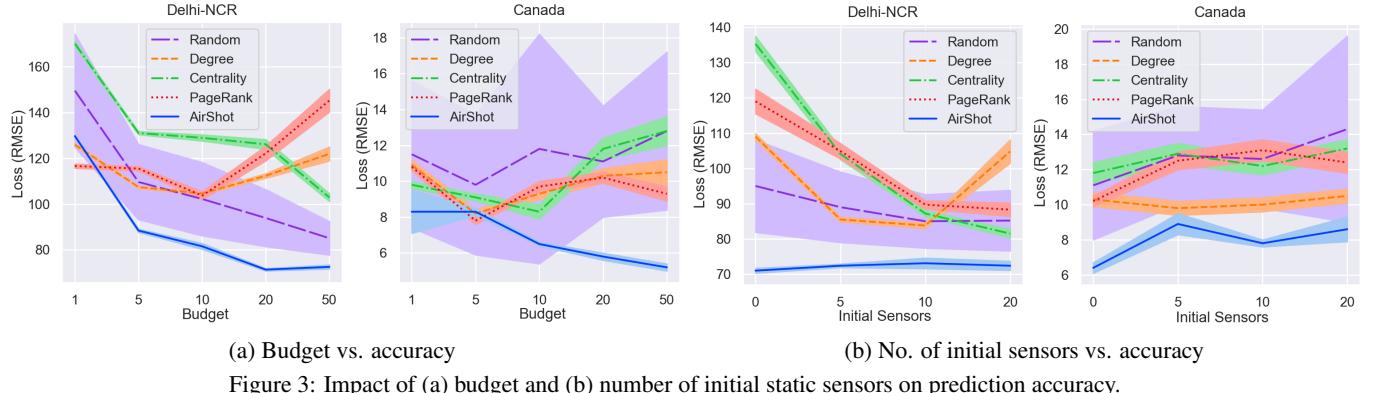
$\sigma$  is an activation function. In our implementation, we use ReLU.  $w(e = (u, v))$  is the edge weight of edge  $e = (u, v)$ .  $\mathbf{W}^k \in \mathbb{R}^{2d^{k-1} \times d^k}$  is a learnable weight matrix where  $d^k$  is the hyper-parameter denoting the representation dimension in hidden layer  $k$ . Nearby locations may display correlation in their PM values. The topology of the graph models possible correlation flows in the geographical space. Through GCN, we allow incorporation of these topological features in the node representations. Finally, we apply a linear layer to convert  $\mathbf{h}_v^K$  into a scalar, which is then passed through a SOFT-MAX layer to obtain a distribution. Mathematically,

$$z_v = \mathbf{w} \cdot \mathbf{h}_v^K, \quad p_v^t = \frac{e^{z_v}}{\sum_{\forall u \in \mathcal{V}} e^{z_u}} \quad (8)$$

Here,  $\mathbf{w} \in \mathbb{R}^{d^K}$  is a learnable vector.

To learn all parameters of model  $\Pi$ , our objective is to maximize the sum of expected rewards. Hence, we minimize the following loss function.

$$\mathcal{J}(\Pi) = \sum_{\forall P \in \mathcal{P}} \mathbb{E}_{p(\mathcal{A}_P; \theta)} \mathcal{R}(\mathcal{A}_P) \quad (9)$$



(a) Budget vs. accuracy

(b) No. of initial sensors vs. accuracy

Figure 3: Impact of (a) budget and (b) number of initial static sensors on prediction accuracy.

Here  $\mathcal{P}$  denotes the set of all partitions of  $\mathcal{S}$  (Recall, line 1 in Alg. 1) and  $\mathcal{A}_p$  denotes the final set of selected nodes for partition  $P \in \mathcal{P}$ . We utilize REINFORCE [Williams, 1992], a classical *policy gradient* method, to train  $\Pi$ .

#### 4.4 Inference: Zero-shot Policy Transfer

Given  $\mathcal{S}$  and  $\mathcal{C}$ , we construct a graph on the corresponding spatial points using the method outlined in § 4.1. We then perform  $b$  rounds of node selection, where in each round we do a forward pass through  $\Pi$  to sample a node from  $p_v^t \sim \Pi(v \mid \mathbf{h}_g^t)$ . Following the selection, the node features are updated and the process repeats till  $b$  steps. It is worth noting that in inference, high-granularity PM data is not needed and therein lies the main impact of AIRSHOT.

### 5 Empirical Evaluation

In this section, we benchmark AIRSHOT and establish its efficacy in zero-shot, budget-constrained sensor placement. The sample dataset<sup>1</sup> and codebase<sup>2</sup> are anonymously available. We are unable to share the entire dataset anonymously due to storage bottlenecks.

#### 5.1 Experimental Setup

All our experiments have been performed on a machine running Ubuntu 18.04.5 LTS, with Python 3.7, over Xeon(R) CPU E5-2698 v4 @ 2.20GHz CPU, and Nvidia Tesla V100 GPU, and 256GB RAM.

**Datasets:** We use datasets from Delhi-NCR and Canada [Adams and Corr, 2019]. Their properties are listed in Tables 1 and 2.

**Baselines:** No work exists on budget-constrained sensor placement for PM monitoring. Hence, as baselines, we consider selecting the top- $b$  nodes based on the following factors: (1) Degree, (2) Between Centrality, (3) PageRank, and (4) Random. The first three features favor selecting locations in dense neighborhoods based on various centrality-dependent factors in the resultant graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ . Finally, Random represents the most naïve placement selection policy.

**Train-Validation-Test:** To train, we sample a random collection of 18 days over the 3 month span. All PM recordings from these days form our train set. The validation and test sets comprise of a random sample of 3 days and 6 days. We

always ensure that the three sets are disjoint. This process is repeated over three samples of train, validation and test sets and we report the mean and standard deviation of the metrics being measured.

**Parameters:** We set the number of graph convolutional layers in AIRSHOT at 2. The hidden dimension is set to 14, converging to 8 features after the convolutional layers. These 8 features are passed through 2 layers of MLP (Eq. 8), giving the *logits* for each node. The *logits* are further normalized via a SOFTMAX layer allowing us to select the optimal node.

**Interpolation Model:** GCN-based interpolation models have been successful in air quality prediction Zhan *et al.* [2021]; Liu *et al.* [2021]. Hence, we use GCN as the interpolation model (See App. C for details). Note that the GCN for interpolation is independent of the GCN used within AIRSHOT. AIRSHOT treats the interpolation model as a black-box and is compatible with any model that consumes static sensor data as input and predicts PM level in a query location.

#### 5.2 Efficacy

**Impact of Budget:** First, we measure how the RMSE improves as we add more sensors based on AIRSHOT and the other considered baselines. Fig. 3 presents the results in Delhi-NCR and Canada datasets. In this experiment, we start with zero static sensors. Thus, the experiment not only reveals the performance on the setup where some sensing infrastructure exists, but also the scenario where the infrastructure needs to be built up from scratch. As visible in Fig. 3, AIRSHOT outperform all baselines significantly in both Delhi-NCR and Canada. In addition, the standard deviation of AIRSHOT is also much lower indicating stable performance. While it may be surprising to note RANDOM as the closest baseline in Delhi-NCR, we observe that all other baselines choose locations that are concentrated in the same neighborhood. Hence, they are not effective over the entire region. Also note that RANDOM consistently has the highest standard deviation in both Delhi-NCR and Canada, indicating its lack of reliability.

**Impact of initial sensors:** Next, we measure how the number of existing static sensors impact the efficacy of our recommendations. In this experiment, we keep the budget fixed at 20. Fig. 3(b) presents the results. We notice that in Delhi-NCR the RMSE of AIRSHOT remains stable, whereas for the other baselines, there is a decreasing trends. This shows that the 20 sites selected by AIRSHOT are enough on their own.

<sup>1</sup><https://anonymous.4open.science/r/PMDistDelhi>

<sup>2</sup><https://anonymous.4open.science/r/AirShot>

However, the sites selected by other baselines are not as effective and hence we the number of initial sensors is increased, their RMSEs improve.

### 5.3 Impact of Training Data

**Training Volume:** First, we study the impact of training data volume on the quality of recommendations. In Fig. 4(a), we vary the volume of training data and plot the corresponding RMSE. The trend is as expected; it decreases and the gradient of the decrease slows as the training volume increases.

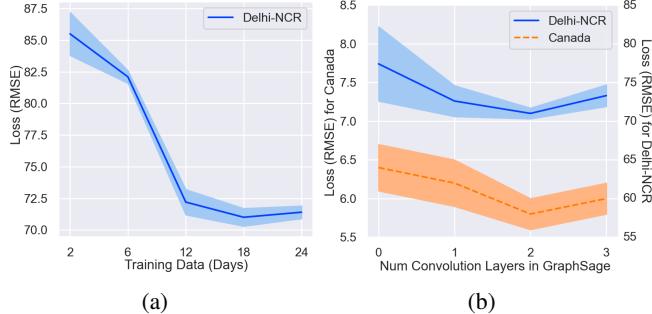


Figure 4: Impact of (a) training data volume and (b) number of graph convolutional layers on prediction accuracy. Note that the axis scales for Canada and Delhi are different (left and right sides respectively).

**Stability of recommendations:** Neural algorithms, by nature, are stochastic. Thus, even if AIRSHOT is run on identical input data multiple times, the recommended sites may change. Furthermore, how sensitive are the recommendations on the training data itself? If the sensor sites change dramatically across runs, then framing policies is problematic. Our next experiment investigates these questions.

To analyze the stability of our recommendations, we use *bipartite matching* analysis, across different sets of recommendations. Specifically, we form a complete bipartite graph using two sets of recommended sites across two different runs. The edge weights correspond to the *Haversine* distance between nodes. Performing a *minimum weight bipartite matching* allows us to form the *bijection* across nodes with the smallest cumulative distance. The mean distance of the matched edges represent the stability of the matching, and hence the stability of the recommendations. The lower the mean value, the better is the consistency.

Table 3a presents the results when run on a budget of 20 at Delhi-NCR. We use three random samples of training data. The diagonal entries of the table answers the first question of stability across the same training data. As visible, the average distance is within 170m. The rest of the entries reveal, that even when the training data itself is different, the mean distance is within 1.5km on average. Given that the entire region covers 559 sq. Km, an average distance of 1.5Km indicates stable recommendations.

### 5.4 Impact of Features and Topology

**Feature ablation study:** In our formulation of MDP in § 4.2, we utilize three features in capturing the state, namely, PageRank (*PR*), Interpolation loss (*IL*), and Loss discrepancy (*LD*). In the next analysis, we investigate the importance of each feature and various subsets. Table 3b presents the

Dataset	Features			Performance
	PR	IL	LD	
1	0.47	0.99	1.88	$79.0 \pm 0.4$
2	0.99	0.10	1.56	$80.7 \pm 2.5$
3	1.88	1.56	0.01	$80.4 \pm 1.9$
(a) Stability of recommended sensor sites.	✓	✓	✓	$83.5 \pm 3.7$
	✓	✗	✓	$80.7 \pm 2.1$
	✗	✓	✗	$87.3 \pm 5.7$
	✗	✗	✓	$84.9 \pm 3.8$

(b) Feature ablation study.

Table 3: Stability and impact of features.

RMSE corresponding to various combination of features. As expected, when all features are used, the RMSE is the lowest. We observe that PageRank is the most important feature, whose ablation leads to largest deterioration in RMSE. It is worth noting that simply using the top- $b$  highest PageRank nodes is not effective as already outlined in Fig. 3(a). Thus, while PageRank is important, it also requires strategic treatment, which the MDP learns.

**Importance of topology:** While AIRSHOT uses a GCN to learn rewards, is the topological information important? Specifically, what would be the impact if the GCN is skipped and the input features are fed straight to the MLP (Eq. 8). We investigate this question next. In Fig. 4(b), we vary the number of graph convolutional layers, starting from 0, and measure its impact on the prediction loss. We note that at 0 layers, i.e., without a GCN, the performance is the worst, and thereby clearly highlighting the importance of topology. The accuracy peaks at 2 and deteriorates at 3. This behavior is known in the literature due to the over-smoothing problem with GCN [Klicpera *et al.*].

## 6 Conclusions

Delhi-NCR is infamous for air pollution, endangering health of its 46 million odd citizens. We release a PM dataset from this region in this paper, collected using low cost IoT devices deployed in public buses. Environmental researchers and medical practitioners can use this dataset, to understand PM exposure of commuters at ground levels, as well as PM temporal variations over days, weeks and spatial variations across locations. Such a low cost mobile monitoring system, in the longer term, can augment the highly expensive static sensor network in the city, to inform citizens’ day to day decisions about local PM levels – whether it is safe to exercise outdoors, whether to use face-masks or air purifiers, or which route or transportation mode to use for commute to minimize PM exposure. The current static sensor network is sparse and inadequate in Delhi, 37 static sensors covering 50,000 sq. Km. Gradual upgradation of this static network is needed for reliable PM measurement and calibration of low cost sensors. Our system AIRSHOT, can effectively recommend static sensor locations under budget constraints. Our overall methodology of low cost mobile sensor deployment in public transport, and using that data to recommend installation sites for expensive and less noisy static sensors, can be replicated in other urban areas. We release our code and data, so that any developing country facing budget constraints and environmental sustainability issues can build on this work.

## References

- Matthew D. Adams and Denis Corr. A mobile air pollution monitoring data set. *Data*, 4(1), 2019.
- Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *COMPUTER NETWORKS AND ISDN SYSTEMS*, pages 107–117, 1998.
- 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.
- EconomicTimes. Delhi could be the world’s most populous city by 2028. but is it really prepared?, 2019.
- 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.
- Gizmodo. India’s air pollution is so bad it’s causing lung damage in kids, 2015.
- Google. Mapping the invisible: Street view cars add air pollution sensors, 2014.
- 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.
- William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *31st NeurIPS Conference*, 2017.
- IndiaToday. Deadly air pollution may cut 17 years from a delhiite’s life, 2019.
- Apte JS, Kirchstetter TW, Reich AH, Deshpande SJ, Kaushik G, Chel A, Marshall JD, and Nazaroff WW. Concentrations of fine, ultrafine, and black carbon particles in autorickshaws in new delhi, india. *Atmospheric Environment*, 45, 2011.
- Apte JS, Messier KP, Gani S, Brauer M, Kirchstetter TW, Lunden MM, Marshall JD, Portier CJ, Vermeulen RCH, and Hamburg SP. High resolution air pollution mapping with google street view cars: Exploiting big data. *Environmental Science & Technology*, 51, 2018.
- Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 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.
- Xinwei Liu, Muchuan Qin, Yue He, Xiwei Mi, and Chengqing Yu. A new multi-data-driven spatiotemporal pm2.5 forecasting model based on an ensemble graph reinforcement learning convolutional network. *Atmospheric Pollution Research*, 12(10):101197, 2021.
- ORF. Delhi is failing its children, air pollution is choking their future, 2021.
- 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 Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- Alexeef SE, Roy A, Shan J, Liu X, Messier K, Apte JS, Portier C, Sidney S, and van den Eeden SK. High-resolution mapping of traffic related air pollution with google street view cars and incidence of cardiovascular events within neighborhoods in oakland. *Environmental Health*, 2018.
- 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.
- Wikipedia. Air pollution in delhi, 2021.
- Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.
- Choujun Zhan, Wei Jiang, Qiaoling Zhen, Haoran Hu, and Wei Yuan. Daily pm2. 5 forecasting using graph convolutional networks based on human migration. In *International Conference on Neural Computing for Advanced Applications*, pages 717–727. Springer, 2021.
- 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 Data Collection Details

Our mobile sensors spatially cover more area compared to static sensors. As shown in Fig 5(d), the static sensors do *point* measurements where the location icons are, while the buses cover the entire region shown as location heatmaps. However, the static sensors get values for the point location at all times, whereas mobile sensors collect some values at a location and move. Thus, we compare the overall temporal volume of data collected by the two sensor types to compare their data quantities.

For static sensors, an active hour must have at least one value reported, as their sampling frequency is low. For mobile sensors, a device is said to be active in an hour if it sends at least 800 data points in that hour. This indicates the mobile device being active for at least 40 minutes in the hour with ideal sampling rate of 20 data points in a minute. Fig. 6(a)-(b) show cumulative active hours per device from 16<sup>th</sup> November 2020 to 30<sup>th</sup> January 2021 for static and mobile sensors. Each curve belongs to a different instrument. When a curve is flatter and more parallel to  $x$ -axis, then the number of active hours for that instrument is less on that particular day. Similarly, if the curve is steeper and is more parallel w.r.t  $y$ -axis, then there are more active hours that day.

From the plots, we observe that most of the static sensors are active for most days though they have much less values per hour (1 to 5 readings). The mobile sensors are anomalous initially, when the mobile network was being setup, until around 10<sup>th</sup> of December as indicated by the lower slopes of the curves before that date. After Dec 10<sup>th</sup>, the mobile sensors become ideal for the remaining days, with more active hours per day and also with many more samples per hour (800 to 1200 readings) compared to static sensors. Thus, the mobile dataset adds significant quantity of data for the environmentalists to analyze, once the mobile network stabilizes. Fig. 6(c) shows the average PM 2.5 values for Jan 2021, after the mobile network stabilizes.

### B Measured PM Compared to Static Sensors

We next correlate static and mobile sensor readings more directly. We first locate the mobile sensors that were close (less than or equal to a distance of 150m), to any static sensor. We found three static sensors satisfying this criteria, which were installed at CRRI Mathura Road, Delhi, Jawaharlal Nehru Stadium, Delhi and ITO, Delhi respectively (referred to as CRRI, JNS and ITO respectively in Fig 7). We compute the correlation between the hourly mean of all PM2.5 values recorded by a static sensor and its nearby mobile sensors. Since the PM values should roughly move from one hour to the other hour in the same direction (increase or decrease), we expect to see a high positive correlation between both the hourly averages. Fig. 7 (a) shows the daily correlation values of all three locations from 1<sup>st</sup> November 2020 to 30<sup>th</sup> January 2021. Region corresponding to JNS lacks some values, as static sensor data was unavailable for comparison in that period. We observe a high correlation across most days. We also found 15 instances where the correlation was found to be negative. Fig. 7 (b) shows a common example of the type

of instances we found to give out negative correlation. Correlation does not take into account how close in magnitude the recorded PM values are. These were cases where PM values were extremely close in magnitude and thus, their trend became trivial. Overall, the correlation with static sensors is remarkable, given the positioning and measurement technique differences. We had 177 total correlation values out of which 103 were found to be significant at  $\alpha = 0.05$ . The mean correlation of this subset was 0.85. Only one significant negative correlation was found.

### C Details of Interpolation Model

To facilitate the interpolation of PM values across the span of the dataset, we deploy a simple GCN with 2 convolutional layers followed by a single MLP layer. The node features are extracted from the input graph (which is prepared from the pollution dataset via preprocessing). The Latitude, Longitude, and Time features of the nodes are one-hot-encoded utilizing a suitable quantization to generate  $F$  features (298 for Delhi-NCR and 347 for Canada). The  $F$  input features are translated to 64 and then to 10 by the convolutional layers. The MLP layer finally predicts the expected PM per given node. Besides being used for generating the RMSE loss for back-propagation, the GCN prediction is also used to generate the *Interpolation loss* (Eq. 3) and *Loss discrepancy* (Eq. 4) features of the State for AIRSHOT.

**Impact of GCN Layers in the Interpolation model:** For this analysis, we modify the number of Graph Convolution layers to observe its impact on the RMSE convergence. Table 4 shows the performance for our Delhi dataset for 0, 1, 2 and 5 convolutional layers. We find that the deployed Interpolation model is performing good with 2 such layers.

NumLayers	RMSE
0	$195.7 \pm 0.2$
1	$70.7 \pm 0.2$
2	$64.0 \pm 0.6$
5	$64.5 \pm 1.7$

Table 4: Impact of Convolution layers



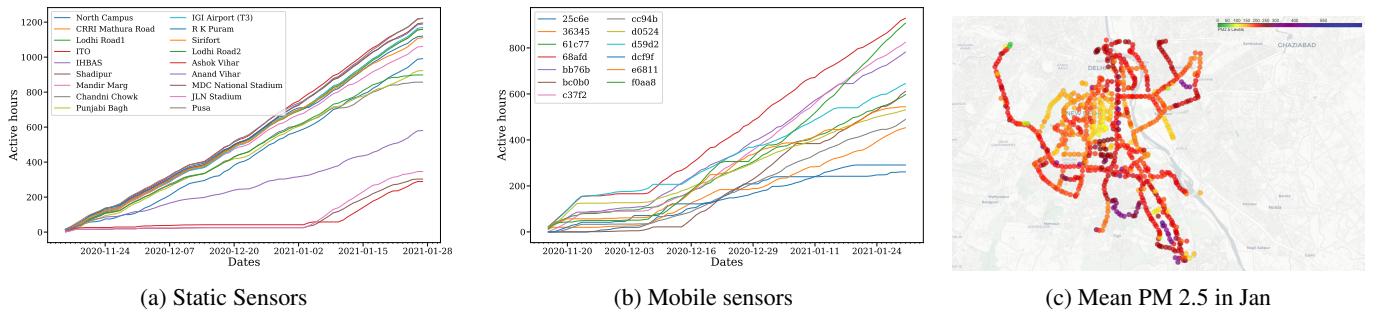
(a) Measurement device

(b) Mounting location

(c) Mounted device

(d) Bus trajectories

Figure 5: (a) Shows the inside of our PM measuring IoT unit. (b) Shows the mounting location in bus driver's cabin in non air-conditioned public bus (below the existing white box). (c) Shows a mounted IoT unit in the bus (below the existing white box). (d) We show the government deployed static sensors installed in and around our bus trajectories, as location icons. We overlay a heatmap of PM data collection by the buses over the three months, in the same figure.



(a) Static Sensors

(b) Mobile sensors

(c) Mean PM 2.5 in Jan

Figure 6: Cumulative active hours for (a) static and (b) mobile sensors between 16<sup>th</sup> Nov to 30<sup>th</sup> Jan. (c) Average PM 2.5 in Jan.

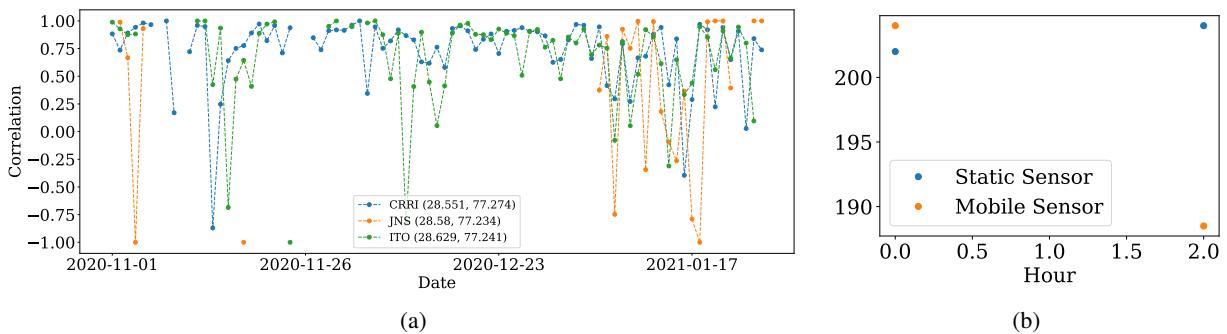


Figure 7: (a) Correlation between hourly averages of static and mobile sensors from 1<sup>st</sup> November 2020 to 30<sup>th</sup> January 2021. The three line plots correspond to three different locations where we consistently found the mobile sensors to be passing by the static sensor. We see that most of the instances have a positive correlation indicating reliability of our low-cost sensors. (b) Shows a common example of negative correlation between the static and mobile PM values. The correlation here is -1. The PM values recorded by both the sensors are extremely close in magnitude and thus the negative correlation can be ignored.