Aagami: Active Learning for Air Quality Station Deployment

S. Deepak Narayanan
IIT Gandhinagar
deepak.narayanan@iitgn.ac.in

Apoorv Agnihotri
IIT Gandhinagar
apoorv.agnihotri@iitgn.ac.in

Nipun Batra
IIT Gandhinagar
nipun.batra@iitgn.ac.in

ABSTRACT

Recent years have seen a decline in air quality across the planet, with studies suggesting that air pollution is a significant cause of death. Governments have set up large scale air quality monitoring stations in various locations in their countries that can help aid them formulate policies to tackle this unforeseen decline in air quality. However, these air quality stations are expensive to install and have thus been often sparsely deployed. Motivated by sparse air quality monitoring and the expensive cost of air quality monitoring stations, we propose an active learning based solution to recommend locations to install air quality monitoring stations. We use a Gaussian Processes based approach for this purpose, motivated by their ability to encode prior knowledge using custom kernels. We empirically show that our proposed approach outperforms several baselines.

CCS CONCEPTS

• **Computing methodologies → Active learning settings**; • **Human-centered computing → Ubiquitous computing**.

KEYWORDS

air quality, gaussian processes, active learning, time-series data, sensor deployment, machine learning

ACM Reference Format:
S. Deepak Narayanan, Apoorv Agnihotri, and Nipun Batra. 2020. Aagami: Active Learning for Air Quality Station Deployment. In *KDD '20: August 22–27, 2020, San Diego, CA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent years have seen a decline in air quality across the planet, with studies suggesting that a significant proportion of the global population has reduced life expectancy by up to 4 years [2, 5, 22]. A recent report by the WHO suggests that 9 out of 10 people breathe polluted air and air pollution is responsible for more than 7 million deaths in a year [1]. To tackle this increasing growth in air pollution and its adverse effects, governments across the world have set up air quality monitoring stations that measure concentrations of various pollutants like NO_2, SO_2 and $PM_{2.5}$. $PM_{2.5}$ refers to the concentration of particles of diameter less than $2.5\mu m$ and is measured in $\mu g/m^3$. $PM_{2.5}$ has been shown to have a significant impact on health [31] and is used to measure air quality. One major issue with the deployment of these stations is the massive cost involved, since installing each one of these stations costs around a million dollars. Owing to the high installation and maintenance costs, the spatial resolution of air quality monitoring is generally poor. As an example, in India, a developing country, the current number of air quality stations is around 150, whereas the government pollution agency estimates the requirement to be 4000 stations. In Africa, the situation is even worse. Only 7 out of the 54 countries actually have real-time air quality monitoring stations according to a recent UNICEF report[2].

Given the sparse air quality sensor deployment, a natural question arises- *how best can we estimate the air quality at unknown locations?*. Estimating air quality is an inherently difficult task. Air quality is an extremely complex spatio-temporal phenomena. It is affected by a plethora of factors. Firstly, air quality is affected by various meteorological factors such as humidity, temperature, wind direction and wind speed and thus a location could be affected by distant sources. Second, there exist highly varied emission sources, like solid fuel used for domestic cooking, thermal plants, vehicular related, roadside dust, and construction activities among others. As a consequence of the above factors, it could be the case that two locations that are spatially far are closer in their air quality whereas two locations that are spatially close may be far in their air quality. Third, many prominent pollutants like $PM_{2.5}$ are complex in nature and are formed after complex interactions between more basic pollutants like SO_2 and NO_2. Such complex chemical reactions can often be extremely hard to model. Figure 1 shows the average daily air quality data for a given station in Beijing, China. It may be noticed that the air quality values are not smooth in time and show a lot of variations owing to a multitude of factors including meteorology amongst others.

Given the high cost involved in the installation and maintenance of an air quality monitoring station, it is important to be able to recommend station locations in a strategic or informative manner. One natural strategy would be to install stations uniformly to maximize spatial coverage. The factors mentioned above and prior work [12, 34] suggest that uniformly installing stations may not be optimal. This motivates the problem that we try to answer in this paper: *Given a set of air quality monitoring stations, where do we install the next set of air quality monitoring stations/sensors so that we can best infer the air quality at unknown locations?*

We propose an active learning [24] based method for optimizing sensor placement[3]. The optimality of sensor placement is non-trivial to define as it can be based on various objectives, some of which are confounding. The objectives include, but are not limited

[1] https://www.who.int/airpollution/en/

[2] https://uni.cf/2qh976E

[3] We use sensor deployment and station deployment interchangeably in this paper

KDD '20, August 22–27, 2020, San Diego, CA

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/1122445.1122456

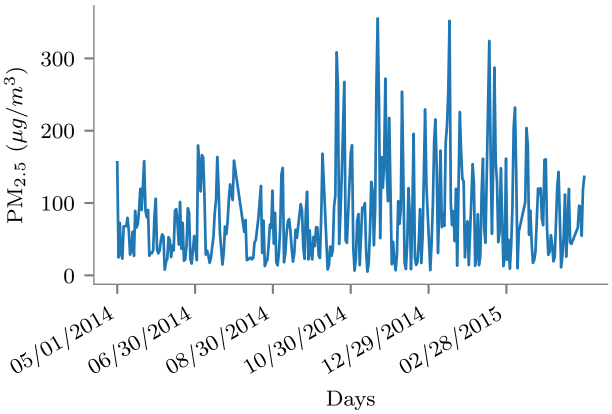

Figure 1: Daily Mean $\mathrm{PM}_{2.5}$ values of a particular air quality monitoring station in Beijing

to: 1) minimizing sensor cost; 2) maximizing prediction accuracy for unmonitored regions/times; 3) minimizing labor cost; 4) minimizing maintenance cost. In this current work we propose an online station location recommendation model that recommends a station to be installed every month. Our objective in this paper is to minimize prediction error at unmonitored locations.

Our contributions in this paper are two fold. First, we propose a Gaussian Process Regressor model to predict air quality ($\mathrm{PM}_{2.5}$) values at unknown locations. We choose a Gaussian Process Regressor since it can help encode domain knowledge easily by supporting custom kernels. These custom kernels can encode spatial and temporal smoothness, relationship between meteorological factors and can help capture notions of periodicity that exists in spatio-temporal processes. Gaussian Processes are Bayesian non-parametric models and thus the model complexity can be tuned based on data availability. We can obtain the predictive posterior mean and variance owing to its Bayesian nature. For our model, we use this variance as a measure of uncertainty. Uncertainty sampling helps reduce the overall predictive variance of our model. In our case, since we use Gaussian Processes, the entropy is a monotonic function of the variance and thus our strategy to use uncertainty sampling is equivalent to decreasing the entropy the most [24]. Second, we propose "Aagami", meaning forthcoming in Sanskrit, as our station deployment strategy. We install stations by choosing the station with the maximum posterior variance. We install stations in an online manner: we install a station every month to the set of monitored stations and show that our model has very low predictive error at unmonitored locations compared to various baselines. To the best of our knowledge, this is the first work that addresses the problem of online air quality station deployment, where stations are installed one at a time. Our work is completely reproducible and the code is hosted at https://github.com/sdeepaknarayanan/activepm.

2 RELATED WORK

Our related work can broadly be classified into three categories: i) techniques for air quality estimation; ii) sensor network deployment; and iii) active learning. We now discuss each of these categories.

2.1 Techniques for air quality estimation

Various techniques to predict air quality have been employed in the past. These techniques can be broadly categorized into into physical model based techniques, spatial interpolation techniques, and data-driven learning based techniques.

Physical models model diffusion of air pollutants and have been used widely for estimating as well as forecasting air quality. These are numerical method based models that model the movement of air pollutants via differential equations, taking into consideration various emission sources and meteorological conditions. These models generally model the emissions within a domain and take into consideration different boundary conditions [7, 8, 19].

Spatial interpolation techniques have been widely used to predict air quality. As the name suggests, these methods only take into consideration spatial distribution of air quality while performing interpolation. Spatial interpolation techniques used in the air quality domain estimate the $\mathrm{PM}_{2.5}$ values at unobserved locations in space using only the $\mathrm{PM}_{2.5}$ values at observed locations. Kriging [18] is a spatial interpolation technique widely used in geostatistics [20] that has been used for estimating air quality. In Kriging, the observed values are assumed to be realizations of a random process, usually a gaussian process. The value at an unobserved location is predicted as a linear combination of the values at the observed locations. Kriging gives the best linear unbiased estimate at unobserved locations. Kriging gives us the estimated value at the unobserved location along with variance. Inverse Distance Weighted Interpolation (IDW) [28] interpolation is another widely used spatial interpolation technique. In IDW, the $\mathrm{PM}_{2.5}$ estimates at unobserved locations are estimated to be a weighted average of the $\mathrm{PM}_{2.5}$ at observed locations. The weights, which vary with each unobserved location, are inversely proportional to the distance between the observed and the unobserved locations. Spatial averaging is another method used for predicting air quality. KNN can be weighted or unweighted. In weighted KNN, data points are assigned weights inversely proportional to their distance from a test data point. Weighted KNN and IDW are equivalent, when K is the number of observed locations, and only the spatial features are considered. Wong et al. use all of IDW, Kriging, Spatial Averaging and KNN in their work to interpolate PM_{10} values in [30].

Recently, there has been an increased usage of data-driven techniques to predict air quality as these methods can incorporate various other features affecting air quality unlike spatial interpolation techniques. These models also capture non-linear relationships that exist between air quality and the various complex factors affecting air quality. Some of these features include weather conditions, meteorological factors, points of interests, traffic conditions and road networks [34, 36]. In [35], Zheng et al. proposed a co-training-based semi-supervised learning approach in 2013, in which they employ two classifiers to model the spatial and temporal factors influencing air quality. This model has its own limitations in that it uses an iterative training process susceptible to noise, which was overcome by Chen et al. [3]. For forecasting air quality, there have been quite a number of data-driven methods that have been used. Zheng et al. [36], proposed a fine-grained forecasting model that employs two independent predictors, a spatial predictor and a temporal predictor to forecast and then aggregates their independent results using meteorological data by employing a prediction aggregator. Recently, a deep distributed fusion network to forecast air quality values has also been proposed by Yi et al [32].

Non-parametric models have also been used for predicting air quality. Guizilini et al. [10] propose a Gaussian Process to predict air quality in an online manner. They introduce a custom kernel that can easily model short term and long term spatial trends present in the data as well as complex temporal trends. Their proposed kernel can model short term temporal trends as well as cyclical temporal trends. Non-parametric models based on Gaussian Processes are more interpretable since they can help capture domain knowledge via their kernels.

2.2 Sensor deployment

Sensor deployment in general has been a well studied problem. Krause et al. [16], propose an algorithm to simultaneously optimize the placement and the scheduling of sensors under constraints on the amount of power that is being consumed. Krause et al. [15], propose a sensor deployment model for the early detection of water contamination. The common aspect in both [15] and [16] is that they both deploy sensors from scratch, without having any previous sensors installed. Guestrin et al. [9], propose using mutual information, an optimization criterion to find the most information about the unsensed locations. Existing work has been largely limited in the domain of deployment of air quality monitoring stations. Hseih et al. [12], propose an incremental station deployment strategy for air quality station deployment. By incremental, we refer to a strategy where there are locations that already have air quality monitors. They propose a semi-supervised approach to infer air quality and then subsequently recommend a fixed number of locations for installing air quality monitoring stations. Their scheme proposes installing all the recommended air quality stations at once.

2.3 Active Learning

Active Learning is a sub field of machine learning in which the learning algorithm or learner chooses the data points (query) from which it learns to use minimal annotation or labelling for learning a good model [24]. Active learning has been used widely in various applications. Kapoor et al [13] use it for object categorization and Shen et al [27] use it for named-entity recognition. There are different scenarios in which the learners may chose to query data points and different ways in which they query the data points. Two majorly used methods include pool based sampling where there is a pool of data points to choose from, or a stream based selective sampling, wherein a data point is chosen or discarded from a stream of data [24]. All these learners have some notion of informativeness of the data points to make a decision and several strategies have been proposed in this regard. Common ones include uncertainty sampling and [17] query by committee [25]. In uncertainty sampling, the learner chooses the data point that it is most uncertain about. Entropy [26] is also widely used as a measure to quantify uncertainty and perform uncertainty sampling. In Query by Committee, a committee of different learners are maintained and the data point for which the committee disagrees the most is chosen as the data point to query [25].

Our work mainly differs from related literature in that we do not deploy all the air quality monitoring stations at once; we rather install a station, use air quality data from this installed station, and then install the next station. Such a choice was motivated by the fact that our deployment scheme is highly appropriate and useful for a realistic deployment where there may not be sufficient funds to deploy all the stations at once.

3 PROBLEM STATEMENT

In this paper we attempt to solve two problems. The first problem is to estimate air quality at unknown locations and the second problem is to choose locations to install air quality stations in a strategic manner to improve air quality inference. We describe both the problems more formally below.

Problem 1: Given a set of air quality monitoring stations S, along with information about their $PM_{2.5}$ values, weather and meteorological conditions over a period of time $\{t_0, t_1, \ldots, t_k\}$, where each t_i denotes a timestamp, estimate the air quality values at unmonitored locations for day k, leveraging both the current and the historical data for the monitored locations.

Problem 2: Given a set of air quality monitoring stations S, along with information about their $PM_{2.5}$ values, weather and meteorological conditions over a period of time $\{t_0, t_1, \ldots, t_k\}$, where each t_i denotes a timestamp, deploy air quality monitoring stations at a few candidate locations, every f timestamps, beginning on day t_{k+1}, such that estimation of air quality at unmonitored locations improves the most across timestamps beginning t_{k+1}. In this problem formulation, once an air quality monitoring station is deployed, its $PM_{2.5}$ data is readily available from the day after the deployment onwards.

4 APPROACH

Before discussing the details of the approach, we would like to give a few key intuitions behind our approach. We make use of weather and meteorology since air quality is affected by these factors. Weather has a natural role to play in pollutant concentrations. In winters typically when the weather is cold, the $PM_{2.5}$ concentration tends to rise since cold air is dense and the particles are suspended lower ue to reduced planetary boundary layer height (PBLH), close to the ground. During summers, the reverse happens, with warm air carrying the pollutants and hence a general reduction in pollution. This trend is visible in Figure 1, where the days on the X-axis denote the days starting from the month of May onward. Also, the change in weather conditions is smooth, since there is usually a monsoon season that prevails during the months between summer and winter. Meteorological factors like wind speed and direction affect the movement of suspended particulates since a high speed wind in a particular direction will likely move the particulates away from a particular location. Further it is natural to expect pollution values to increase during specific hours of a day or specific days of the week. This could be during the peak traffic hours when people go to work. These patterns tend to be periodical, i.e., they tend to repeat over time. In our approach we wish to capture these temporal patterns. There could be long term and short term dependencies that may exist among different locations spatially. These dependencies could arise because of local factors such as a factory emitting smoke or by global factors like wind speed and direction carrying particulate matter. Our approach is motivated by taking into consideration all these naturally occurring events and encodes this domain knowledge by using a custom kernel described below.

4.1 Gaussian Process Regression

We provide a very brief primer to Gaussian Processes here. Gaussian Processes (GPs) are a non-parametric model that induce a distribution over functions. In any Gaussian Process model, we have a prior mean function $\mu : \mathbb{R}^d \rightarrow \mathbb{R}$ and a prior covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. These covariance functions quantify the similarity among different data points. They are also called kernels and are used to build the covariance matrix. The covariance matrix K has entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where k is the covariance function and $\mathbf{x}_i$ and $\mathbf{x}_j$ are two data points in $\mathbb{R}^d$. We use the following notation in the section: Let $X \in \mathbb{R}^{n \times d}$ denote all the training data points and let $y \in \mathbb{R}^n$ be the training labels. K_{XX} refers to the covariance matrix. $\hat{K}_{XX} = K_{XX} + \sigma_n^2 I_n$ is the covariance matrix added with zero-mean Gaussian noise of variance σ_n^2, I_n is the identity matrix of order n. $K_{X\mathbf{x}^*}$ refers to the vector that is formed by calculating the covariance function between any test point $\mathbf{x}^*$ and all the train points. Once the model is trained to fit the data, we obtain the predictive posterior distribution. For a test point $\mathbf{x}^*$ and its corresponding predictive distribution $y*$, we have

$$\mathbb{E}[y^*|X, \mathbf{y}] = \mu(\mathbf{x}^*) + K_{X\mathbf{x}^*}^T \hat{K}_{XX}^{-1} \mathbf{y}$$

$$\mathrm{Var}[y^*|X, \mathbf{y}] = k(\mathbf{x}^*, \mathbf{x}^*) - K_{X\mathbf{x}^*}^T \hat{K}_{XX}^{-1} K_{X\mathbf{x}^*}$$

where $\mathbb{E}[y^*|X, \mathbf{y}]$ and $\mathrm{Var}[y^*|X, \mathbf{y}]$ are the expectation and the variance of the posterior distribution respectively [23].

In GPs, there are usually a number of standard kernels that are used for a variety of problems. The addition or multiplication of kernels still results in a valid kernel and hence they are combined in a variety of ways to create custom kernels that are typically used to encode domain knowledge and capture complex dependencies between features. In this paper we use a few standard kernels and combine them to create a custom kernel described below. We use a combination of three standard kernels in our work - the Matérn kernel, the radial basis function (RBF) kernel and the periodic kernel. We use the matern kernel with $\nu = 3/2$ (Matérn32). We choose this particular value of ν to account for less smoothness in the approximation function. These three kernels are described below. We choose Matérn52 to our to allow for a little more smoothness as compared to $\nu = 3/2$ but not as much as an RBF kernel.

$$k_{Matern32} = \sigma^2 \left(1 + \frac{\sqrt{3}d}{\theta_l^2}\right) exp\left(-\frac{\sqrt{3}d}{\theta_l^2}\right)$$

$$k_{RBF} = \sigma^2 exp\left(\frac{-d^2}{2\theta_l^2}\right)$$

$$k_{Periodic} = \sigma^2 exp\left(-\frac{sin^2\left(\frac{\pi d}{\gamma}\right)}{2\theta_l^2}\right)$$

where $d = ||x_i - x_j||$ is the eucledian distance between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$. In all of the above kernels, the σ, θ_l and γ are hyperparameters that are optimized while fitting the GP. θ_l is the lengthscale hyperparameter, σ is the variance hyperparameter and γ, used only in the periodic kernel is the period hyperparameter.

In the dataset that we used in this work, we have the following as features: latitude, longitude, weather and meteorological factors humidity, pressure, wind speed and wind direction. We use the following kernels in our Gaussian Process Regressor.

$$k_{longitude,latitude} = k_{Matern32} + k_{Matern32} \tag{1}$$

$$k_t = k_{Matern32} + \sum_{i=1}^{5} k_{Matern32} \times k_{Periodic} \tag{2}$$

$$k_{Temp,Hum} = k_{Matern32} + k_{Matern52} \tag{3}$$

$$k_{Windspeed} = k_{RBF} \tag{4}$$

$$k_{Weather} = k_{RBF} \tag{5}$$

$$k_{Pressure} = k_{RBF} \tag{6}$$

The final kernel that we employ is

$$\prod_{f \in Features} k_f$$

where $Features = \{(Longitude, Latitude), Time, (Temperature, Humidity), Windspeed, Weather, Pressure\}$. In our kernel, we use the same spatiotemporal kernel ($k_{longitude,\ latitude}$ and k_t) that was proposed by Guizilini et al [10]. Our rationale for using their kernel is as follows: (1) The two $Matern32$ kernels naturally can account for short term and long term spatial trends that we wish to capture (Equation 1) (2) Their temporal kernel Equation 2 has a Matérn kernel multiplied with a periodic kernel added to a Matérn kernel. The first term accounts for cyclical delays while the second captures long term or short term trends. In Equation 3, we use the domain knowledge that temperature and humidity are related and appropriately capture their variations together. The choice of two Matérn kernels allows the regressor to learn non smooth relationships as well as differing trends. For modeling wind speed, pressure and weather, we used standard RBF kernels to account for a more smoother variation in their values.

4.2 GPs and Kriging

Kriging Interpolation is very similar to Gaussian Process Regression and is sometimes even referred to as Gaussian Process Regression. They have a few fundamental differences that we explain here. Kriging gives the best linear unbiased predictions for the unobserved locations whereas in GPs, we have the conditional distribution after observing the data, whose mean is taken to be the prediction. Yet another major difference that exists between both Kriging and GPs is in their use of kernels. In Kriging, there is use of a variogram cloud, which is a plot of the semivariance versus the lag. The semivariance is defined for a given lag and it is half the variance of the differences between all possible points with a given lag. Variograms are then fit to this plot. Variograms have been shown to be equivalent to covariance functions [14], and this naturally allows for only a specific set of functions to be variograms. In practice, parametric variograms that satisfy these conditions are used and are fit via weighted least squares or restricted MLEs [6]. The final predictions in Kriging depend on the values that this parametric variogram takes for queried points. Kriging is limited in scope because of its restrictive nature of the parametric variograms it can use. GPs, as already mentioned in Section 1 can support custom kernels to help encode domain knowledge. GPs overall offer a lot more flexibility and explain-ability than Kriging.

5 DATASET

To evaluate our proposed approach we use the dataset released by Zheng et al [33, 34, 36]. We now summarise the data set by describing the present attributes.

(1) Air Quality data: The dataset contains hourly $PM_{2.5}$ measurements for a total of 36 air quality monitoring stations in Beijing from 1st May 2014 to 30th April 2015.

(2) Weather and meteorological data: In addition to the $PM_{2.5}$ data, the dataset also consists of weather and meteorological data. Meteorological data includes humidity, wind speed, wind direction, pressure and temperature. We choose the closest meteorological station for obtaining the meteorological data for each air quality monitoring station. We create two new features out of the data for wind speed and wind direction. We create a feature WS_x, that gives the magnitude and direction of wind speed along East and WS_y, that gives the magnitude and direction of wind speed along North. We created these two features to jointly capture the effect of wind speed and wind direction.

6 DATASET RESAMPLING

We downsampled the data to a single measurement per day because of the following reasons. (1) Missing $PM_{2.5}$ data: In our dataset we had around 13.3 % of $PM_{2.5}$ data missing. We also had a significant amount of data missing in the meteorological and weather data with missing percentages going nearly to 29.6% and 40.1% for wind speed and humidity respectively. (2) Also, city authorities often look at 24 hour exposures before deciding to take actions.

7 EXPERIMENTAL SETTINGS

We consider two experiments in this work. In the first experiment we estimate air quality at unknown locations in the first experiment. In the second experiment we incrementally deploy air quality monitoring stations one by one using active learning. Before discussing the two sets of experiments in detail, we first discuss some common time-series specific evaluation strategies common to both the experiments.

7.1 Time-series specific evaluation strategies

In all our experiments we maintain the temporal integrity present in the dataset, i.e., we ensure that air quality values from the future are not used to predict air quality values in the past. This choice allows us to perform all our experiments in a real-world setting. We present a modified cross-validation approach here. We split the data based on stations to create the train, test and validation splits. For stations in the validation set, we use only the current day's data and for stations in the train set, we use historical and current day's data. For a specific test set, we choose the optimal hyperparameters across all the different validation sets by performing a thorough nested cross validation [11], in the manner described above. We predict on the same day as the current day.

7.2 Problem 1: Estimation of Air Quality at unknown locations

We now consider Problem 1 stated in Section 3 in this experiment. For tuning the hyperparameters of all our models, except in the case of Gaussian Processes, we use the method described above. In the case of Gaussian Processes, we maximize the log marginal likelihood to optimize the hyperparameters, which, because of its balance between complexity and data-fit prevents overfitting. [23]. We perform a six fold outer cross validation and a 5 fold inner cross validation. To demonstrate the importance of adding features such as weather and meteorological factors, we create two different feature sets. Feature set A consists of only latitude, longitude and time as features. Feature set B consists of latitude, longitude, time, weather and the remaining meteorological factors. We evaluate our approach on both the feature sets and report our results in Section 9. In all our experiments, we scale all the features that we use to have values between 0 to 1 via standard min-max scaling. Since we leverage historical data, we choose to use the data from the last k days. We choose k from the following set: $\{10, 20, 30, 50\}$. We choose $k \geq 10$ to provide sufficient context for the model to learn temporal patterns present in the data. We also predict once every two days. As will be shown in the results section, most of the models could not make use of data present beyond 30 days and thus we chose k to be from the set mentioned above. Our choice of predicting every other day was due to the expensive nature of computation involved in performing nested cross validation. We enumerate our baselines below.

(1) Ordinary Kriging (OK): Ordinary Kriging is a spatial interpolation technique used widely in geostatistics. Ordinary Kriging provides the best linear unbiased predictor [18, 21].

(2) Inverse Distance Weighted (IDW) Interpolation: IDW [28] is a spatial interpolation technique widely used in geostatistics. and takes a weighted average of the $PM_{2.5}$ values of the train set to predict on the test set.

(3) XGBoost Regressor (XGB): XGB [4] is a boosting ensemble model that is used in regression settings. It makes use of multiple decision trees to predict $PM_{2.5}$.

(4) K-Nearest Neighbors (KNN) Regressor: KNN [1] usually predicts the $PM_{2.5}$ values as the average of the $PM_{2.5}$ values of the K closest data-points from the training data. In our case we assign weights to the data points depending on the distance in a manner similar to IDW.

(5) Lasso: It is a linear model that incorporates feature selection via implicit regularization [29].

(6) Spatiotemporal Gaussian Process Regressor ($SptGPR$): Spatiotemporal Gaussian Processes makes use of latitude, longitude and time as features for training the model. We use the kernel employed in Guizilini et al [10].

(7) Support Vector Regressor (SVR): In this particular baseline, we use Support Vector Machines (SVMs) for regression. Prior work [12] has shown that SVR is capable of outperforming various state-of-the-art methods [35, 37] for air quality prediction.

7.3 Problem 2: Active Learning for Air Quality Station Deployment

We now consider Problem 2 stated in Section 3 in this experiment. For this purpose we maintain three sets of stations - the train set S_{train}, the test set S_{test} and the pool set S_{Pool}. S_{train} contains the air quality stations that are currently monitored, S_{test} contains air quality stations where we wish to estimate the air quality, and S_{Pool} contains the set of candidate air quality monitoring stations (locations) to be installed every month. For evaluating the performance of our model, we use the ground truth data available at S_{test}. We predict every day for the current day using the exact same settings as in Section 7.2, making use of the data for the last k days including the current day. We choose that value of k for which our model performs the best in the first experiment described in Section 7.2 We query a station from the pool set to be added to the train set once every month. Note that this queried station is no longer a part of the pool set. More formally, let s_q be the queried station. Then $S_{train} = S_{train} \cup \{s_q\}$ and $S_{pool} = S_{pool} \setminus \{s_q\}$. From the day of querying onwards, the PM$_{2.5}$ values are available for the pool stations since they were now added to the train set. In this experiment, motivated by sparse air quality monitoring stations, we use 6 stations for S_{train}, 24 stations for S_{pool} and the remaining 6 stations for S_{test}. This results in a total of 6 different test sets, each with 5 different training and pool sets. We vary the choice of S_{train}, S_{pool} and S_{test} by splitting the dataset in the same manner as was done for the nested cross validation procedure described earlier.

7.3.1 *Models.*

Our intuition is that a model that can estimate air quality well will be able to choose data points in a more informative and useful manner. We thus choose the top 4 best performing regressors from the first experiment to perform active learning.

7.3.2 *Active Learning Strategies.*

(1) Query By Committee [25]: In Query by Committee (QBC) we maintain a committee consisting of multiple learners, all of which are trained on the same train dataset. In our setting, we create a committee by using the same learners initialized with different hyperparameters. These hyperparameters were chosen using the first experiment results. We describe the selection process in more detail in 7.3.3. We look at the pool station (location) with which this committee disagrees the most. We use the standard deviations in the predictions of these learners to quantify the disagreement and choose the station in the pool set having maximum disagreement as the station to be added to the train set. In all our experiments we use a committee size of 5.

(2) Random Sampling : Random sampling is a sampling method in which we randomly choose a station from the pool set and add it to the train set. It is widely used in Active Learning as a baseline [24]. We use 5 different random seeds and report the mean and standard deviation in our predictions of PM$_{2.5}$.

(3) Uncertainty Sampling [24]: Gaussian Process Regressor provides us with the posterior predictive mean and variance. This variance gives us the confidence that the GPR has in its predictions. As mentioned earlier in Section 1, in this particular case, choosing the station location with highest variance is equivalent to choosing the location with highest entropy. We therefore choose the pool station that the model is most uncertain about and add it to the train set when querying from the pool set.

7.3.3 *Choice of Hyperparameters for QBC Regressor.*

To choose the regressors for our committee (for KNN and XGB) in QBC, we use the regressors with the best 5 hyperparameters from experiment 1. We chose those hyperparameters that were most consistently chosen via nested cross validation across all folds and across all timestamps. This ensures that the chosen hyperparameters are robust to change in time and also across different train datasets. Another way to choose the best hyperparameters would be to choose those hyperparameters that have the lowest test errors across all the different folds and different timestamps. But a natural issue that can arise while choosing these hyperparameters is that the hyperparameters may have a very low error, but was probably only chosen for very few timestamps. We empirically verified such an occurrence and appropriately chose the hyperparameters to significantly strengthen our baselines.

7.3.4 *Reproducibility.*

Our entire codebase, baselines, dataset, analysis and experiments will be released upon acceptance.

8 EVALUATION METRICS

We denote the prediction of PM$_{2.5}$ at station i at time t by $\hat{y_t^i}$. We consider three evaluation metrics: Root Mean Squared Error of a station i, $RMSE^{Station}(i)$ and Mean Root Mean Squared Error (MeanRMSE) for computing the errors in the prediction of PM$_{2.5}$ and Win Loss ratio to compare different active learning strategies. RMSE and $MeanRMSE$ are given below.

$$RMSE^{Station}(i) = \sqrt{\frac{\sum_{i=0}^{T}(\hat{y_t^i} - y_t^i)^2}{T}} \tag{7}$$

$$MeanRMSE = \frac{\sum_{i=0}^{|S|} RMSE^{Station}(i)}{|S|} \tag{8}$$

In equation 7, T is the total number of timestamps for station i. $\hat{y_t^i}$ and y_t^i refer to the predicted value and the ground truth value for the t^{th} timestamp of the i^{th} station. In equation 8, i refers to a station and |S| denotes the total number of stations. In our case, $|S| = 36$. For experiment 1 described earlier, we use $RMSE^{Station}(i)$ to compute the error for a given station across all timestamps and subsequently we use $MeanRMSE$ to compute the error across all timestamps and all stations in the test set. Note that computing the $MeanRMSE$ is equivalent to computing the mean error across all the 6 different folds used in our cross validation.

For our active learning experiment, apart from RMSE, we use Win Loss Ratio as a metric. Win Loss Ratio of two vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ denoted as $WinLoss(\mathbf{x}, \mathbf{y})$ is computed as follows:

$$WinLoss(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} 1 - X_i} \tag{9}$$

where $X_i = 1$, if $x_i < y_i$ and 0 else. In our case, we use $WinLoss(\mathbf{x}, \mathbf{y})$ specifically for active learning to compare the performance of our model with other models. We use this particular metric because over large number of stations and time stamps, aggregating errors (via RMSE) may not lead to conclusive results.

9 EXPERIMENTAL RESULTS

9.1 Results of Experiment 1

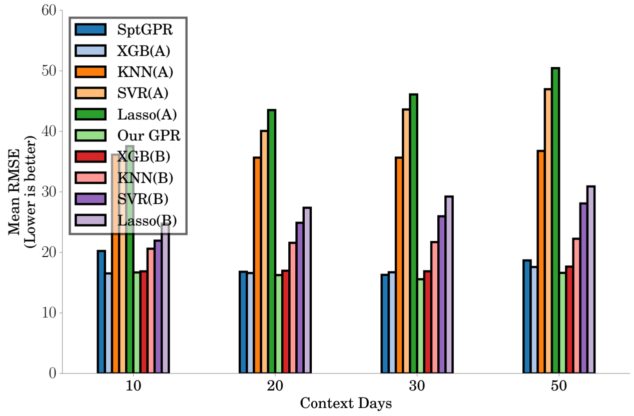

Figure 2: ***MeanRMSE* computed for all stations using feature set *A* and *B*. Note that Our *GPR*, performs vary favourably when compared with the other regressors. It outperforms all the baselines when $k = 30$.**

In Figure 2 we report the errors on feature set *A* and *B*. We observe that XGB and *SptGPR* outperform the other regressors for feature set *A* and our proposed Gaussian Process Regressor (GPR) described earlier in section 4, clearly outperforms all the competing regressors on feature set *B* and *A*. In feature set *A*, XGB has an overall improvement of **5.67**% over *SptGPR*. In feature set *B*, we have an average improvement in prediction for our GPR of **4.73 %** and a maximum improvement in prediction of **7.77 %** over XGB across all the different values of $k \in \{10, 20, 30, 50\}$. Upon addition of features, apart from XGB, all the other models are able to learn better and hence we have the subsequent decrease in their *MeanRMSE*'s. It is still unclear to us as to why XGB's performance does not improve upon the addition of new features. Our proposed GPR consistently outperforms XGB upon addition of new features. Our proposed model is more interpretable as compared to XGB because of the custom kernels we use to encode domain knowledge. We can clearly observe that encoding domain knowledge at a very rudimentary level results in consistently better performances as opposed to other baselines. While the graphs show the results of learning based methods, we present the results of spatial interpolation techniques in Table 1.

Table 1: The *MeanRMSE* for spatial baselines

Interpolation Method	Mean RMSE
Ordinary Kriging	**14.66**
IDW	16.99

The mean RMSE across all days across all the different train test sets in our dataset. The above mentioned methods are spatial baselines.

It is important to note that Kriging interpolation performs better than our GPR. Our proposed GPR is at least as applicable as Kriging, if not more, since it gives us an additional option to forecast into the future owing to its temporal component. In earlier work [12], it has been shown that SVR outperforms state-of-the-art methods [35, 37]. SVR also outperforms both *OK* and *IDW* convincingly in [12]. But this was not true in our case. We believe that this could be because of the fact that [12] used air quality data of a higher temporal resolution, i.e., hourly data of $PM_{2.5}$, while we used a downsampled dataset, owing to the fact that the dataset that we used had a large number of hourly missing entries as described in Section 6. We also believe that our proposed GPR will perform much better when the data is used at an hourly granularity, owing to the 5 temporal kernels that we used, which can find finer temporal patterns within a single day, rather than just across days.

Once the hyperparameters of the GPR have been tuned, the learnt hyperparameters can further be used to give insights. As an example, we observed some of the periods learnt by the temporal kernel in our GPR and were able to infer repetitive patterns occurring every ≈ 7 and ≈ 29 days. These offer good insights into the behavior of complex spatio-temporal phenomena, suggesting roughly weekly patterns and monthly patterns learnt from the data. This explainability also helps in preparing for inputs that can cause potential failures instead of silently failing on some adversarial input for complex models like neural networks.

From experiment 1, we find that Kriging provides a very good estimate of $PM_{2.5}$. While it is not entirely clear how to perform active learning in a time series setting using Kriging, we choose the following experiments to perform active learning using Kriging. Kriging provides the mean of the prediction along with the corresponding variance. We use this variance as a measure of uncertainty akin to how we use it with Gaussian Processes. We maintain the same experimental setting as was present for experiment 2, adding a station every month and predicting every day. We no longer use historical data for prediction using Kriging and use the variance of the predictions at the pool station locations to be queried for only the day of query.

9.2 Results of Experiment 2

As can be seen from figure 2, the results obtained were the best for $k = 30$. We use this choice of k for all our active learning experiments as explained in section 7.3. Since we performed active learning over multiple different train sets and test sets, we report the Win Loss ratio across these sets in Table 3. We report report the RMSE in daily prediction across all the different sets of data in Table 2. We report the improvement in % that our Gaussian Process Regressor provides over different baselines. We compute win loss ratio for a given train set and test set. We have a total of 30 such splits in our dataset. For each of these splits, we perform active learning beginning on day 30 on wards and predict at test locations to obtain an error for each day of prediction. To compute win loss, we take the mean of this error computed everyday, across all timestamps, for a given train test set for a given regressor. We then compare this with another regressor's RMSE for the same train set and test set.

Our GPR performs very favourably when compared with the various active learning baselines we used in our experiments, as can be seen from Tables 2 and 3. From Table 2, we can see that our GPR has the least RMSE across all the timestamps and splits of the data when compared to active learning baselines. From Table 4, we can also see that our method provides up to **40 %** improvement in prediction over the best random method baseline and also an average improvement of up to **14%** across both the random baselines. We observe that Kriging performs better than our GPR in experiment. In this particular case, it has a low RMSE of **16.66** as opposed to our GPR with 20.67 and the other active learning baselines.

Win Loss Ratio for Kriging: It is noteworthy that Kriging had a 100 % win with all the <regressor, active learning strategy> pairs except for when the pair was <Our GPR, Uncertainty Sampling>. It had a win loss ratio of 29 and had a win % of 96.67. Kriging's mean for random sampling across 5 different seeds resulted in an error of **17.11 ± 2.14**. One thing to note here is that Kriging does not seem to be selective. Kriging seems to be a very good interpolator for this particular dataset. It does not seem selective since Kriging is able to gain only **3.62** % over a random sampling method for choosing stations, by using its own predictive variance, whereas GPR's uncertainty sampling gains **15.22** % over a random sampling method.

9.3 Insight - Locations chosen by Kriging and our GPR

Though Kriging may have outperformed our GPR in terms of RMSE, we were able to draw insights into station locations chosen by GPR. For this purpose, we chose to observe the station locations chosen by our GPR and Kriging on their best performances and their worst performances. In both the cases, our GPR initially tries to *maximize* spatial coverage, suggesting the high influence of location as a factor, and then becomes selective, choosing stations based on other features. This was supported by the results that we observed with Kriging, as it was able to outperform our GPR in terms of predictions. Surprisingly, the locations chosen by Kriging did not offer a lot of interpretability and resembled random selection of station locations. This substantiates our point of *explainability* and *interpretability* that our GPR offers.

Table 2: Mean RMSE at the test stations with all <Regressor, Active Learning Strategy> pairs

Regressor	Active Learning Strategy	Mean RMSE
GPR	Uncertainty Sampling	**20.67**
GPR	Random Sampling	24.38 ± 5.08
XGB	QBC Sampling	22.67
XGB	Random Sampling	24.13 ± 1.71
KNN	QBC Sampling	30.68
KNN	Random Sampling	30.54 ± 1.23

The above table shows the mean RMSE across all days across all different train test sets in our dataset. As can be seen, our GPR with uncertainty sampling has the lowest RMSE.

10 FUTURE WORK

In our current setting we only install a single stations at the end of every month. One natural extension to this problem would be allowing for an installation of k stations every month so that the predictive error decreases the most at unmonitored locations. This is a non-trivial extension if we use predictive variance as our query scheme. This is because, in such a setting, selecting the top-k unmonitored stations with highest uncertainty would not take into account potentially underlying correlation between stations. In our current setting, we avoid this problem by only installing a single station every month. Since we have sufficient data from the newly installed station in the current formulation, we avoid selecting highly correlated stations. Additionally, using the last 30 days of station data allowing our model to have equal amount of data from all the installed stations for selecting the next location for station installation.

Table 3: Win Loss Ratio, Win % and Loss % across all the different train test splits

Regressor	Active Learning Strategy	Win - Loss Ratio
XGB	QBC	5 (25:5)
KNN	QBC	29 (29:1)
XGB	Random	6.5 (26:4)
KNN	Random	∞ (30:0)
GPR	Random	4 (24:6)

The above table shows the win loss ratio across the different train test splits we used in our experiments for our GPR. Our GPR performs better than all the active learning baselines at estimating air quality across multiple different test sets demonstrating its robustness across all train test splits

Table 4: Relative improvement in predictions compared to different random strategies

	Max. %		Mean %	
	GPR (US)	XGB (QBC)	GPR (US)	XGB (QBC)
GPR (Rd)	33.57	33.79	14.27	4.96
QBC (Rd)	40.77	23.89	13.73	5.54

We report the best % improvement and the mean % improvement in predictions of our GPR and XGB when compared with a random method. Note that our GPR provides the maximum improvement in predictions on average, clearly beating the other regressors. Note: US - Uncertainty Sampling and Rd - Random Sampling

In our current setting, we assume the cost of installing stations is the same at any location. This assumption might not hold true. Installations of station at different locations could entail different costs. Additionally, for a particular location it might be costly to install a station early on than to install a sensor once the infrastructure (eg. road networks) around the location is developed. This imposes additional constraints on top of our current formulation. The key to solving this problem will be balancing the cost of station installation and the uncertainty reduction.

As part of future work we would like to use the hourly data in a systematic manner to improve upon our models. We believe that at the hourly level, there are more finer patterns that our temporal kernel can capture. This may include pollution spikes during specific timings of a day and patterns within a week.

11 CONCLUSIONS

In this paper, we address the problem of air quality station deployment in an online setting. To the best of our knowledge, this is the first work addressing this problem of online station deployment. In our work we propose a Gaussian Process Regressor (GPR) that can encode domain knowledge by supporting custom kernels to choose locations. We showed that our GPR outperforms several baselines. Our GPR was also shown to be *selective* in terms of choosing locations, since a) our GPR initially maximizes spatial coverage and b) uncertainty sampling was critical to GPR attaining the lowest RMSE when compared with regressors apart from Kriging in Experiment 2.

REFERENCES

[1] N. S. Altman. 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician* 46, 3 (1992), 175–185. https://doi.org/10.1080/00031305.1992.10475879

[2] Kalpana Balakrishnan, Sagnik Dey, Tarun Gupta, RS Dhaliwal, Michael Brauer, Aaron J Cohen, Jeffrey D Stanaway, Gufran Beig, Tushar K Joshi, Ashutosh N Aggarwal, et al. 2019. The impact of air pollution on deaths, disease burden, and life expectancy across the states of India: the Global Burden of Disease Study 2017. *The Lancet Planetary Health* 3, 1 (2019), e26–e39.

[3] Ling Chen, Yaya Cai, Yifang Ding, Mingqi Lv, Cuili Yuan, and Gencai Chen. 2016. Spatially Fine-grained Urban Air Quality Estimation Using Ensemble Semi-supervised Learning and Pruning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 1076–1087. https://doi.org/10.1145/2971648.2971725

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *CoRR* abs/1603.02754 (2016). arXiv:1603.02754 http://arxiv.org/abs/1603.02754

[5] Yuyu Chen, Avraham Ebenstein, Michael Greenstone, and Hongbin Li. 2013. Evidence on the impact of sustained exposure to air pollution on life expectancy from China's Huai River policy. *Proceedings of the National Academy of Sciences* 110, 32 (2013), 12936–12941.

[6] Noel A. C. Cressie. 1993. *Statistics for Spatial Data.* John Wiley & Sons, Inc. https://doi.org/10.1002/9781119115151

[7] Claudio Gariazzo, Camillo Silibello, Sandro Finardi, Paola Radice, Antonio Piersanti, Giuseppe Calori, Angelo Cecinato, Cinzia Perrino, Fabio Nussio, Marco Cagnoli, Armando Pelliccioni, Gian Paolo Gobbi, and Patrizia Di Filippo. 2007. A gas/aerosol air pollutants study over the urban area of Rome using a comprehensive chemical transport model. *Atmospheric Environment* 41, 34 (Nov. 2007), 7286–7303. https://doi.org/10.1016/j.atmosenv.2007.05.018

[8] Timothy M. Gaydos, Rob Pinder, Bonyoung Koo, Kathleen M. Fahey, Gregory Yarwood, and Spyros N. Pandis. 2007. Development and application of a three-dimensional aerosol chemical transport model, PMCAMx. *Atmospheric Environment* 41, 12 (April 2007), 2594–2611. https://doi.org/10.1016/j.atmosenv.2006.11.034

[9] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. 2005. Near-optimal Sensor Placements in Gaussian Processes. In *Proceedings of the 22Nd International Conference on Machine Learning* (Bonn, Germany) *(ICML '05)*. ACM, New York, NY, USA, 265–272. https://doi.org/10.1145/1102351.1102385

[10] Vitor Guizilini and Fabio Ramos. 2015. A Nonparametric Online Model for Air Quality Prediction. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 651–657. http://dl.acm.org/citation.cfm?id=2887007.2887098

[11] Trevor Hastie. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics).* Springer.

[12] Hsun-Ping Hsieh, Shou-De Lin, and Yu Zheng. 2015. Inferring air quality for station location recommendation based on urban big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 437–446.

[13] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. 2007. Active Learning with Gaussian Processes for Object Categorization. In *2007 IEEE 11th International Conference on Computer Vision*. 1–8. https://doi.org/10.1109/ICCV.2007.4408844

[14] Kitanidis. 1997. *Introduction to Geostatistics (Stanford-Cambridge Program).* Cambridge University Press.

[15] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. 2008. Efficient Sensor Placement Optimization for Securing Large Water Distribution Networks. *Journal of Water Resources Planning and Management* 134, 6 (Nov. 2008), 516–526. https://doi.org/10.1061/(asce)0733-9496(2008)134:6(516)

[16] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin. 2009. Simultaneous placement and scheduling of sensors. In *2009 International Conference on Information Processing in Sensor Networks*. 181–192.

[17] David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland) *(SIGIR '94)*. 3–12.

[18] Andreas Lichtenstern. 2013. *Kriging methods in spatial statistics.* Technical Report. Department of Mathematics, Technische Universitat Munchen. https://mediatum.ub.tum.de/doc/1173364/1173364.pdf

[19] Benjamin N. Murphy and Spyros N. Pandis. 2009. Simulating the Formation of Semivolatile Primary and Secondary Organic Aerosol in a Regional Chemical Transport Model. *Environmental Science & Technology* 43, 13 (2009), 4722–4728. https://doi.org/10.1021/es803168a

[20] M.A. Oliver and R. Webster. 2014. A tutorial guide to geostatistics: Computing and modelling variograms and kriging. *CATENA* 113 (Feb. 2014), 56–69. https://doi.org/10.1016/j.catena.2013.09.006

[21] M. A. OLIVER and R. WEBSTER. 1990. Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information Systems* 4, 3 (1990), 313–332. https://doi.org/10.1080/02693799008941549

[22] C Arden Pope III, Majid Ezzati, and Douglas W Dockery. 2009. Fine-particulate air pollution and life expectancy in the United States. *New England Journal of Medicine* 360, 4 (2009), 376–386.

[23] Carl Edward Rasmussen. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series).* The MIT Press.

[24] Burr Settles. 2009. *Active Learning Literature Survey.* Computer Sciences Technical Report 1648. University of Wisconsin–Madison. http://burrsettles.com/pub/settles.activelearning.pdf

[25] H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by Committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (Pittsburgh, Pennsylvania, USA) *(COLT '92)*. ACM, New York, NY, USA, 287–294. https://doi.org/10.1145/130385.130417

[26] C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (July 1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[27] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Multi-criteria-based Active Learning for Named Entity Recognition. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics* (Barcelona, Spain) *(ACL '04)*. Association for Computational Linguistics, Stroudsburg, PA, USA, Article 589. https://doi.org/10.3115/1218955.1219030

[28] Donald Shepard. 1968. A Two-dimensional Interpolation Function for Irregularly-spaced Data. In *Proceedings of the 1968 23rd ACM National Conference (ACM '68)*. ACM, New York, NY, USA, 517–524. https://doi.org/10.1145/800186.810616

[29] Robert Tibshirani. 1994. Regression Shrinkage and Selection Via the Lasso. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 58 (1994), 267–288.

[30] David W Wong, Lester Yuan, and Susan A Perlin. 2004. Comparison of spatial interpolation methods for the estimation of air quality data. *Journal of Exposure Science & Environmental Epidemiology* 14, 5 (Sept. 2004), 404–415. https://doi.org/10.1038/sj.jea.7500338

[31] Yu-Fei Xing, Yue-Hua Xu, Min-Hua Shi, and Yi-Xin Lian. 2016. The impact of PM2.5 on the human respiratory system. *Journal of Thoracic Disease* 8, 1 (2016). http://jtd.amegroups.com/article/view/6353

[32] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2018. Deep Distributed Fusion Network for Air Quality Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 965–973. https://doi.org/10.1145/3219819.3219822

[33] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban Computing: Concepts, Methodologies, and Applications. *ACM Trans. Intell. Syst. Technol.* 5, 3, Article 38 (Sept. 2014), 55 pages. https://doi.org/10.1145/2629592

[34] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. 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*. ACM, 1436–1444.

[35] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-Air: When Urban Air Quality Inference Meets Big Data. In *Proceedings of the 19th SIGKDD conference on Knowledge Discovery and Data Mining (KDD 2013)* (proceedings of the 19th sigkdd conference on knowledge discovery and data mining (kdd 2013) ed.). https://www.microsoft.com/en-us/research/publication/u-air-when-urban-air-quality-inference-meets-big-data/

[36] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting Fine-Grained Air Quality Based on Big Data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. ACM, New York, NY, USA, 2267–2276. https://doi.org/10.1145/2783258.2788573

[37] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML* (Washington, DC, USA) *(ICML'03)*. 912–919.