HazeEst: Machine Learning Based Metropolitan Air Pollution Estimation From Fixed and Mobile Sensors

Ke Hu, Ashfaqur Rahman, *Senior Member, IEEE*, Hari Bhrugubanda, and Vijay Sivaraman, *Member, IEEE*

***Abstract*— Metropolitan air pollution is a growing concern in both developing and developed countries. Fixed-station monitors, typically operated by governments, offer accurate but sparse data, and are increasingly being augmented by lower fidelity but denser measurements taken by mobile sensors carried by concerned citizens and researchers. In this paper, we introduce HazeEst—a machine learning model that combines sparse fixed-station data with dense mobile sensor data to estimate the air pollution surface for any given hour on any given day in Sydney. We assess our system using seven regression models and tenfold cross validation. The results show that estimation accuracy of support vector regression (SVR) is similar to decision tree regression and random forest regression, and higher than extreme gradient boosting, multi-layer perceptrons, linear regression, and adaptive boosting regression. The air pollution estimates from our models are validated via field trials, and results show that SVR not only yields high spatial resolution estimates that correspond well with the pollution surface obtained from fixed and mobile sensor monitoring systems, but also indicates boundaries of polluted area better than other regression models. Our results can be visualized using a Web-based application customized for metropolitan Sydney. We believe that the continuous estimates provided by our system can better inform air pollution exposure and its impact on human health.**

***Index Terms*— Air pollution monitoring, machine learning, support vector regression, wireless sensor network, web application.**

I. Introduction

METROPOLITAN air pollution monitoring has to date been left to government agencies that typically commission and operate fixed-site stations housing air quality monitors [1]. These monitors, though very accurate, have high installation costs and large space requirements, which limits their number. For example, there are only 15 air pollution monitoring sites in the state of New South Wales, Australia [2]. They are separated by tens of kilometers (if not more), and the resulting coarse-grained spatial measurements do not accurately reflect the air pollution surface for a metropolitan area (such as Sydney) that has pockets of higher pollution. To address this shortcoming, many research groups around the world [3]–[6] have over the past few years developed systems comprising portable monitors that can "crowd-source" air pollution measurements (from citizens and/or public transport vehicles) with denser spatial granularity at low cost. However, little work has been done in the literature to fuse the data from these two sources, or to develop a model for estimating air pollution with high spatial granularity.

Fine-grained air quality estimation is needed for better understanding of the health impacts of pollution. It is recognized that air pollution data taken from (sparse) fixed stations is not representative of real exposure for patients of respiratory illnesses [7]–[9]. However, the use of (dense) mobile sensor data suffers from the problem that such data has neither been continuous nor long-term, since most projects for mobile air quality monitoring have operated for less than 5 years and produced sporadic data that is not systematically archived, limiting their utility for long-term medical studies.

In this paper we develop a novel solution to the above problem. We develop a system called *HazeEst*, which trains a machine learning model using existing pollution data from fixed and mobile sensors, and uses the model to estimate the fine-grained air pollution surface over long time periods. Our specific contributions are:

1) We design HazeEst, a system that uses historical data from both fixed and mobile sensors to learn air pollution profile at fine spatial granularity, and thereafter estimates the air pollution surface for any day/time in metropolitan Sydney. The surface can be visualized using our web application.
2) We compare and validate estimation accuracy across different regression models, and show that estimation accuracy of SVR (Support Vector Regression) is similar to DTR (Decision Tree Regression) and RFR (Random Forest Regression), but higher than XGB (Extreme Gradient Boosting), MLP (Multi-Layer Perceptrons), LR (Linear Regression) and ABR (Adaptive Boosting Regression).
3) We conduct field trials to validate our models, and show that our estimated surface matches well at high spatial resolution with the surface obtained from fixed/mobile sensor data, and further that the SVR model is able to clearly delineate the boundaries of polluted areas.

The rest of this paper is organized as follows: Section II describes related work, and Section III provides background on data sources, participatory sensing systems and details the seven regression models used in this study. Section IV describes the steps in our model, while model implementation and estimation accuracy is studied in Section V. Section VI presents results from our field trials in Sydney, and our

Manuscript received February 13, 2017; revised March 20, 2017; accepted March 20, 2017. Date of publication April 5, 2017; date of current version May 5, 2017. The associate editor coordinating the review of this paper and approving it for publication was Prof. Subhas Chandra Mukhopadhyay. *(Corresponding author: Ashfaqur Rahman.)*

K. Hu, H. Bhrugubanda, and V. Sivaraman are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia (e-mail: kehu@unsw.edu.au; z5039854@unsw.edu.au; vijay@unsw.edu.au).

A. Rahman is with ISSL, CISRO, Hobart, Tasmania 7001, Australia (e-mail: ashfaqur.rahman.omi@gmail.com).

Digital Object Identifier 10.1109/JSEN.2017.2690975

system implementation is discussed in Section VII. The paper concludes in Section VIII.

II. Related Work

Mapping urban air pollution using data from static stations and spatial interpolation methods has been studied for a long time. Authors developed a regression-based method to map traffic-related air pollution within a GIS environment in three European cities [10]. The air pollution data they used is from 80 fixed monitoring sites in each city, and they shown that their map produced good estimations compared to the monitored pollution levels. Another study which is taken in Milan shows that good air pollution concentration contour map can be generated using cokriging, which is an extension of kriging that has been commonly used for air pollution interpolation [11]. However, the air pollution data has been used in these studies is from low spatial resolution fixed monitoring stations, and the interpolation results cannot accurately reflect the air pollution surface.

The idea of using wireless sensor networks to get fine spatial resolution air pollution has been investigated by several projects around the world in the past few years. One among the very early projects which have implemented this idea realistically is the Mobile Environmental Sensing System Across a Grid Environment (MESSAGE) project [12] which is a 3 year research project costing 3.5 million pounds, led by the Imperial College, London since 2006. The project has evolved into the Cambridge Mobile Urban Sensing effort (CamMobSens). Low cost portable sensing units which can measure CO, CO_2 and NO_2 have been designed and calibrated in their project, and then deployed city-wide [13]. Data from static stations and mobile sensors is compared and proves that their sensors are a valid means to get dense and accurate pollution readings. Another early project is CommonSense [14] from Berkeley University and Intel. In this project, the researchers design a portable air pollution monitoring sensor prototype which can measure various pollutant concentrations. Data from the sensor can be uploaded to the server and is viewable on Google Maps via web application. Microsoft and Vanderbilt University also put effort into the Mobile Air Quality Monitoring Network (MAQUMON) project [15], in which they design a number of vehicle-mounted air pollution sensor nodes to measure O_3, NO_2 and CO/VOC concentrations. An air pollution contour map is also implemented using image overlays. However, it appears that this project has not been undertaken further in a large scale of deployment. In recent years, the most noteworthy project is OpenSense2 (inherited from OpenSense) [16] at EPFL Switzerland. They have successfully deployed ten air pollution monitoring sensor units on top of public buses city-wide in the city of Zurich and another twelve in Lausanne. A range of pollutants data is collected and stored in their central server for over two years. Statistical models such as region-based Gaussian model and land use regression model are also explored to produce high quality and fine-grained pollution maps.

In our prior work [17] and [18], we presented HazeWatch – an air pollution wireless sensing system which consists of four parts: (1) portable sensor units, (2) mobile application which can receive data from sensors via Bluetooth, and send data with timestamps and locations to the server in real-time, (3) a cloud-based server which stores all the data and interpolates the spatio-temporal estimates, and (4) mobile and web-based applications to visualize air pollution data. We designed and built our own sensor units for air pollution measurement. Metal oxide sensors (MOS) was used to develop the sensor units in the first place, and it allowed us to built our unit housing three sensors (CO, NO2 and O3) at a cost price close to $50, but posed many performance problems related to non-linearity and influence of temperature and pressure. Then we chose electrochemical (EC) sensors to build the second version of our units, which are sensitive, accurate, and linear, but expensive ($50-100 each). Concurrent to our development effort, we also used some other commercial sensing devices to get measurements, such as Node sensor (detailed information is given is section III). We validated our system with a number of trials and demonstrated that our system yields more accurate air pollution estimations than current systems based on government monitoring data.

These crowd-sourcing air pollution sensing systems highly increase the monitoring spatial resolution, however, data from such systems is not long-term, since most of these projects are operated for less than 5 years, and limits the usage for a better understanding of the health impacts of pollution. As a result, most medical researchers only use data from (sparse) static pollution monitoring sites to find associations between air pollution and respiratory diseases. We believe the conclusion they made could have been biased based on just sparse air pollution data.

To address the problem, many other research groups have started utilizing machine learning models and air pollution data from wireless sensor networks to estimate or predict air quality in the past two years. For example, authors compare air pollution forecasting performance among three machine learning algorithms using multi-gas sensing devices [19]. Their work uses three months of pollution data to predict temporal series data, and (sparse) fixed station locations as spatial factors to get air pollution forecasting maps. Compared to their work, our model not only considers both spatial and time-series aspects, but also uses historical data to improve the estimation accuracy.

Having similar goals to ours, the researchers from OpenSense project design their own air pollution mobile sensing system, and use the land use regression model to get fine spatial resolution pollution maps of ultrafine particles using both current and historical data [20]. Our model differs from their work in the specific method of estimating air pollution maps, for any given time in the past seven years.

III. Background

In this section we give a brief introduction of A) pollutant selection and data sources, and B) the seven regression algorithms we use and compare in our estimation model.

A. Pollutant and Data Sources

1) Pollutant: in this study, we monitor Carbon Monoxide (CO) as the pollutant because of its most well known

effect on the human body - reducing blood's oxygen holding capacity, which can cause oxygen delivery issues and lead to tissue and organ problems. Historical CO data is obtained from two sources - a government air pollution monitoring network and our own participatory sensing system [17].

2) Data source:
   i) *Government air pollution monitoring network*: there is a 15 static station network monitoring air quality, operated by the Office of Environment and Heritage (OEH) in New South Wales, Australia. All the readings which are updated hourly from the monitoring stations and are viewable online as both ambient concentrations and air quality index (AQI) values. Seven pollutants can be monitored within one station at the same time. Historical data in the past seven years can be obtained from their website [2]. In this paper, we specifically use CO concentration data from May 2009 to May 2016.
   ii) *Participatory sensing system*: as mentioned in section II, we have developed a participatory mobile sensing system which can crowd-source fine-grained spatial measurements of air pollution. In the past three years, we have collected 34,864 minutes of CO data using Node sensors [21], which is a commercial portable air pollution sensing device. The Node sensor platform is designed with plug-in modules mode. It comprises body platform part and interchangeable OXA gas sensor header part, and one pollutant can be measured with each separate sensor header. A range of pollutants can be monitored using Node sensors, such as Carbon Monoxide (CO), Carbon Dioxide (CO_2), Nitric Oxide (NO), Nitrogen Dioxide (NO2), etc. Smart phones can connect to the body platform with Bluetooth 4.0 up to 250 feet away. It has to be calibrated in six months by mobile app. The CO monitoring resolution of the Node sensor is less than 1.5 ppm, and the GPS shift of mobile phone is usually less than 15 meters.

B. Regression Models

1) Support Vector Regression (SVR): support vector machine is a thriving supervised model for regression analysis [22]. SVR aims to provide a non-linear mapping function to map a given training data set D: $\{(x_1, y_1),(x_2, y_2),\ldots,(x_i, y_i)\}$ to a high dimensional feature space. In this space a decision separating hyperplane can be defined which separates all the data points, with maximal functional margin.
2) Decision Tree Regression (DTR): decision tree [23] is a widely used machine learning method for classification and regression. The aim of decision tree learning is to create a model which can classify target values by learning decision rules from input features. Whether the decision tree is a classifier or regressor depends on if the output variable is categorical or numeric. Learned trees can be different based on different tree algorithms, such as ID3, C4.5, and C5.0. Classification and Regression Tree (CART) algorithm is used in this study.
3) Random Forest Regression (RFR): random forest [24] is an ensemble learning algorithm and based on decision tree learning and bootstrap aggregating. It can be used for classification, regression and other tasks. The basic concept of random forest is to fit a number of decision trees on random subsets of all the features and sub-samples of the dataset, and use averaging method to improve the prediction accuracy and avoid over-fitting. Specifically, it uses bootstrap aggregating (or named bagging) to repeatedly train decision or regression trees, with random feature subsets and sample subsets. It then predicts unseen input samples by averaging all the predictions from trees which have been trained. One advantage of random forest is that it can avoid high variance and high bias which is often happens using a single decision tree.
4) Extreme Gradient Boosting (XGB): gradient boosting [25] builds an ensemble of prediction models iteratively, and allows differentiable loss function optimization to obtain better prediction performance. Extreme gradient boosting [26] is an optimized gradient boosting library which provides a parallel tree boosting to solve both classification and regression problems. It is best known for its fast training speed and high prediction accuracy on a lot of real world problems.
5) Multi-Layer Perceptrons (MLP): multi-layer perceptron [27] is a feed-forward artificial neural network model which can learn a non-linear function and map a set of input features to a target. It comprises of perceptrons which are organized into layers. Between the input and output layer, there can be one or multiple hidden layers. It also uses back-propagation to update weights and minimize the loss function.
6) Linear Regression (LR): linear regression [28] focuses on finding the relationship between one or multiple inputs X and one output Y. In this paper we use ordinary least squares linear regression which aims to minimize the sum of the squares of the differences between real and predicted output Y.
7) Adaptive Boosting Regression (ABR): adaptive boosting [29] is a popular boosting algorithm and can resist overfitting better than many other algorithms. The core idea of adaptive boosting is to add a number of weak learners repeatedly to a single strong learner, and the final prediction is generated by the combination of predictions from all the weak learners. Different weightings for different weak learners are adjusted by the current prediction error.

IV. MODELING

Our estimation model consists of four main steps, which is shown in Fig. 1. We select nine input features to feed in our model: location row, location column, hour window, weekday/weekend, season, and values from four fixed stations for that particular hour. The output is one single CO value for the particular time and location. The detailed steps are as follows:

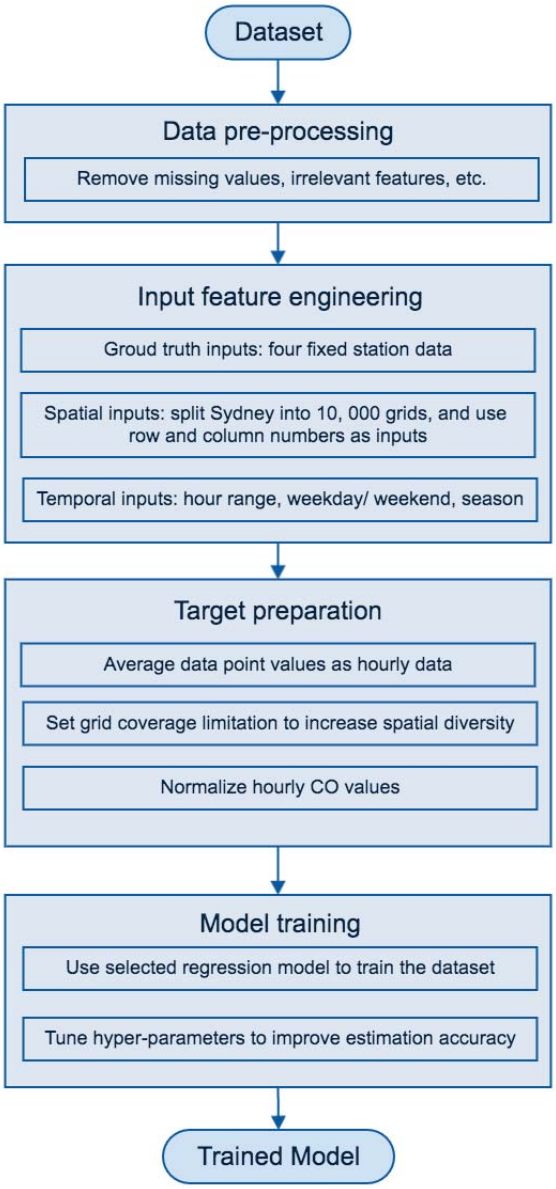


Fig. 1. Flowchart of our estimation model.

A. Data Pre-Processing

1) Remove all the missing values and irrelevant features from the database table, like other pollutant values.
2) Remove CO values which are outside of range 0 ppm to 60 ppm, to ensure the validity of the pollution data.

B. Input Feature Engineering

1) Ground truth data: we select hourly CO values from four fixed stations (Liverpool, Chullora, Rozelle and Prospect) as the ground truth data feature. Because the high quality government static monitoring stations are more accurate than our portable sensors, introducing values from these sites to our model can reduce the bias values from our sensing system. Also, these four stations are the only static CO monitoring stations which are distributed within the greater Sydney region.
2) Spatial feature: split the greater Sydney area (one grid) into 100×100 cells, and convert the original latitude and longitude locations into these 10,000 value cells.
3) Temporal features: Fig. 2 shows that average hourly CO readings change based on different hours, day of the week and season in the past seven years. As a result, we use the following temporal features:
   i) *Hour*: from Fig. 2(a), we can see that average hourly CO concentrations increase from 08:00, and reach the peak at 11:00, then decrease till 15:00. After that, the values rise a little bit before dropping to the bottom at 23:00. Therefore, we classify the hours into four time windows (8:00-11:00, 12:00-15:00,16:00-19:00,20:00-23:00) as the hour feature.
   ii) *Day of the week*: it can be seen from Fig. 2(b) that average hourly concentrations are stable among different days of the week. However, pollution levels start to decrease from Saturday, except Rozelle, and all reaches the floor on Sunday. Hence, we use weekday/weekend as one temporal feature.
   iii) *Season*: Fig. 2(c) shows that summer has the lowest average hourly CO concentrations, while winter is the worst season for CO exposure. Season is another indicative feature for pollution estimation.

C. Target Preparation

Data from the sensor network is uploaded every 5 seconds. We need to convert target values into hourly data before we train the model.

1) If there is more than one data point from the sensor network in a particular cell in one hour, use the average value of all the data points in that cell to represent the hourly CO value for that cell in that particular hour.
2) Remove data from the database table where sensing data points coverage is less than 10 cells in an hour to increase spatial diversity.
3) Normalize hourly CO values of all cells using standard score method (also known as Z-score standardization) to make the mean and standard deviation of the values to 0 and 1 respectively.

D. Model Training

Feed normalized data and features into one regression model, and optimize the estimation performance by tuning hyper-parameters.

V. Model Implementation and Estimation Results

A. Model Implementation

All the regression models are implemented using Python and scikit-learn [30] (except XGB), which is an open source machine learning library for the Python programming language. All hyper-parameters are tuned using ten-fold cross validation method and the GridSearchCV function. GridSearchCV function can exhaustively search over specified parameter values defined by the user and automatically detect

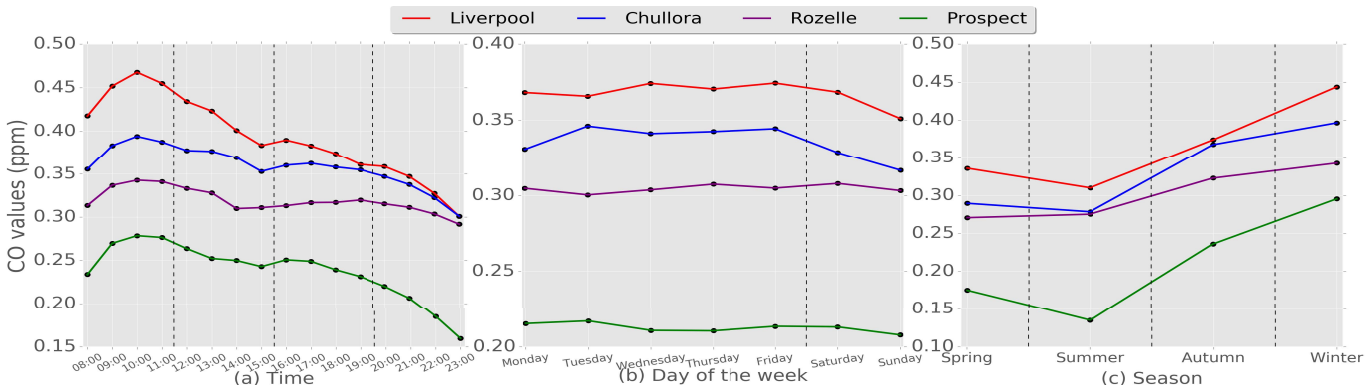


Fig. 2. Four static stations CO readings from May 2009 to May 2016 based on (a) different hours from 8:00 to 23:00, (b) day of the week from Monday to Sunday, and (c) seasons from spring to winter.

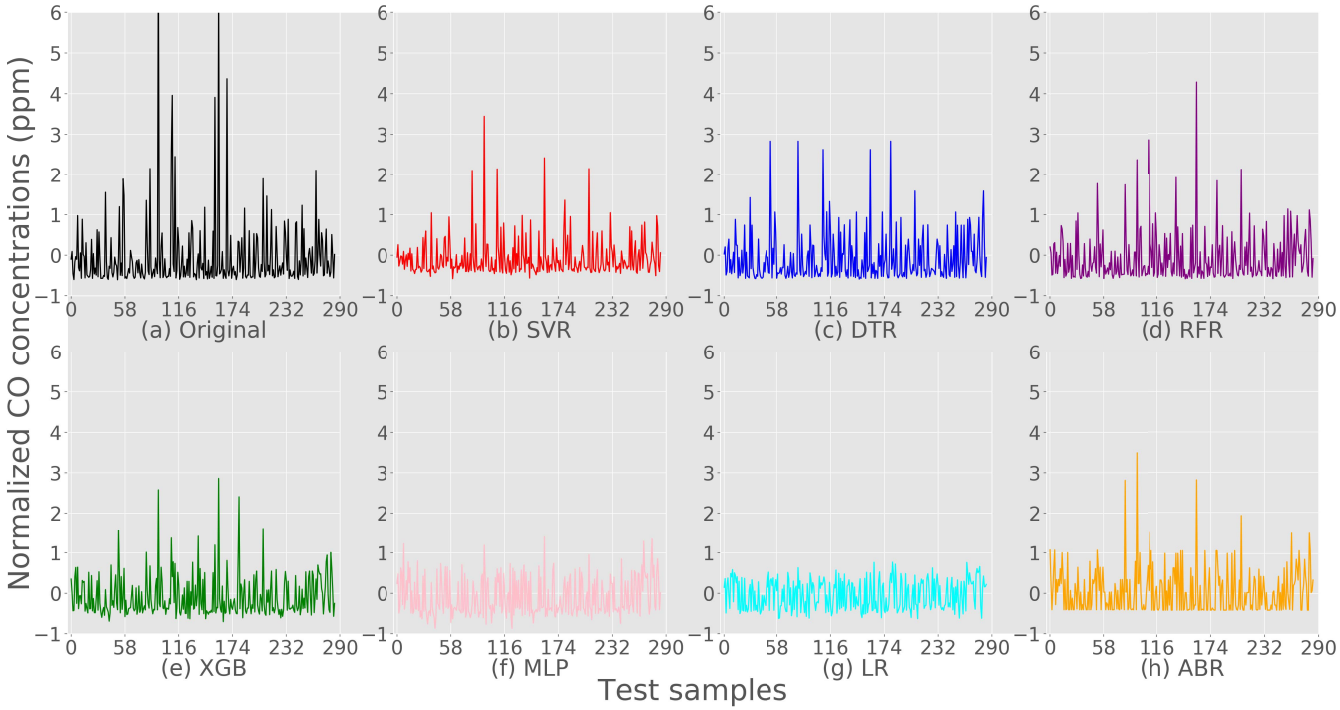


Fig. 3. Comparison of (a) original values in test dataset and (b)–(h) test estimation values based on seven regression models.

TABLE I
DATASET USED IN THE EXPERIMENTS

Number of instances	Location row range	Location column range	Fixed station CO range (ppm)	Normalized CO range (ppm)	CO Mean value (ppm)	CO Standard deviation
2844	22 - 98 (77)	18 - 98 (81)	0 - 0.7	-0.60659 - 8.48916 (9.09575)	3.48652	5.74640

the hyper-parameter which has the best score. To avoid over-fitting, which is a common problem in machine learning algorithms, we limit the minimum sample numbers per leaf on DTR and RFR, and ensure the estimation accuracy is similar based on both training and test dataset.

B. Estimation Accuracy Comparison

The entire dataset we use in the estimation is shown in Table I. We can see that CO values from fixed stations are very low with a range of 0.7ppm, while the normalized CO value for each cell has a value range around 9ppm. For ten-fold cross validation method, mean absolute error (MAE) and root mean squared error (RMSE) are used to validate the model output accuracy.

First we use ten-fold cross validation method to train and test the entire dataset using seven different regression algorithms. The results are shown in Table II. According to the table, SVR, DTR, and RFR have similar MAE and RMSE errors, which are lower than results from XGB, MLP, LR, and ABR. RFR achieves the lowest MAE and RMSE.

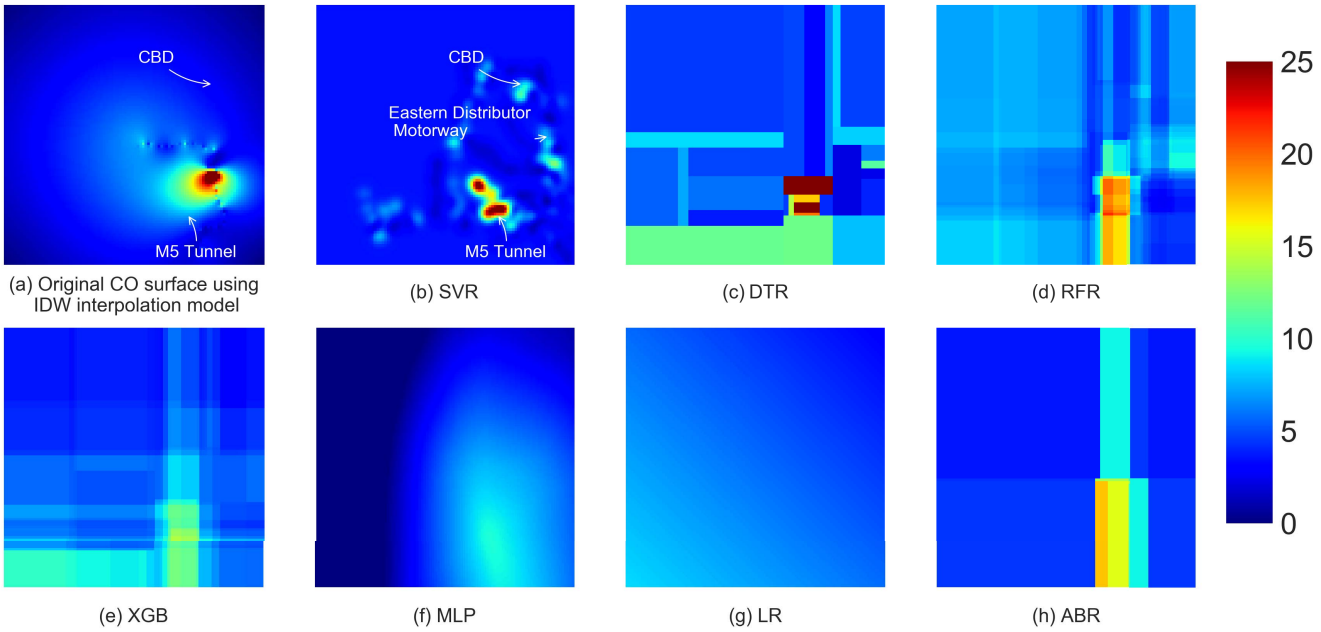


Fig. 4. Comparison of air pollution contour map based on (a) interpolated values based on real data from both static stations and wireless sensor network, and (b)–(h) regression model estimations at 16:00 on 03/05/2016.

TABLE II
ENTIRE DATASET ESTIMATION ACCURACY

Model	Attributes	Test process	MAE	RMSE
SVR	9	10-fold cv	0.314419	0.637581
DTR	9	10-fold cv	0.299986	0.634661
RFR	9	10-fold cv	**0.295037**	**0.611891**
XGB	9	10-fold cv	0.350042	0.671155
MLP	9	10-fold cv	0.427556	0.794201
LR	9	10-fold cv	0.494788	0.926375
ABR	9	10-fold cv	0.533723	0.845920

TABLE III
TEST DATASET ESTIMATION ACCURACY

Model	MAE	RMSE
SVR	0.301532	0.670076
DTR	**0.276770**	0.695081
RFR	0.280306	**0.642000**
XGB	0.332278	0.667065
MLP	0.397027	0.789698
LR	0.493440	0.926362
ABR	0.508495	0.912644

Then the entire dataset is split into 2559 training samples and 285 test samples. Every model is trained using the training set, and estimations are generated based on trained model and the test set inputs afterwards. The estimation accuracy is shown in Table III. Similar to the previous results, SVR, DTR, and RDR also has the lower MAE and RMSE than MLP, LR and ABR. XGB has a higher MAE but a lower RMSE than SVR. DTR has the lowest MAE while RFR gets the lowest RMSE. A visualization of estimation results and original test data can be found in Fig. 3, from which one can see that estimations from SVR, DTR, RFR, XGB and ABR correspond well to the original test output. In contrast with results from the above four models, estimations from MLP, LR cannot indicate significant values, and stay at a low level all the time.

VI. TRIAL RESULT AND DISCUSSION

From the previous section, we can see that SVR, DTR and RFR have better estimation performance across all seven regression algorithms. In this section, we evaluate our approach with field trials and show the estimation performances between different algorithms.

We conducted trials and utilized an interpolation model to generate hourly air pollution data, based on CO readings from both static sites and the mobile sensor network. This is compared to the data which is based on estimation values from seven regression models. One of these trials was conducted at 16:00 on 03/05/2016. During the trial, several participants were asked to bring the Node sensors which have been mentioned in §III to collect real-time air pollution data near the CBD and M5 tunnel, which usually have high pollution concentration values in Sydney. All the data was uploaded to the server using our mobile application, via mobile networks. After one hour of data collection, we then used the inverse distance weighting (IDW) interpolation model to get CO pollution surface values using data from both fixed stations and mobile sensors. The contour map is shown in Fig. 4 (a).

Based on seven regression trained models and input features, which follow the feature engineering rules for that particular hour, we estimate 10,000 cell values for each algorithm. After reverting the normalized CO estimations back to normal values, we have the seven contour maps as shown in Fig. 4(b) - (h). One immediate observation can be made – the contour map based on SVR corresponds well with the sensing interpolation map, and can clearly highlight the most

TABLE IV
ESTIMATION ACCURACY FOR A PARTICULAR HOUR

Model	Cells	Max value	MAE	RMSE
SVR	10,000	29.442001	**1.949668**	**3.167807**
DTR	10,000	**32.464006**	3.788206	5.286464
RFR	10,000	21.642411	4.085328	4.906636
XGB	10,000	14.242757	2.731489	3.772054
MLP	10,000	9.714869	3.532728	4.518752
LR	10,000	8.539178	2.943455	3.721933
ABR	10,000	17.533830	2.559300	3.746430
IDW	10,000	48.009810	-	-

polluted area in greater Sydney. Results from DTR, RFR, XGB and ABR can indicate polluted areas to some extent, however, polluted area boundaries are indicated by large pixels instead of smoothly by small pixels. MLP can present the polluted area smoothly, however, the polluted area is quite large compared with the original one. Moreover, from the figure, we can see that LR results cannot show the pollution contour map correctly.

Estimation accuracy can be found in Table IV. These max values, MAE and RMSE are based on values that have been reverted from the normalized values, which are different from the cross validation and testing accuracy results which are based on normalized values. From Table IV, we can see that the maximum values from original data are around 48 ppm, and SVR cannot capture the peak values as accurate as DTR. However, SVR has the lowest estimation MAE and RMSE, which is 1.95 and 3.17, while MAE and RMSE of DTR is higher at 3.79 and 5.29 respectively. Based on contour maps and estimation accuracy results, we can see that SVR has the best estimation performance among these seven regression algorithms.

VII. SYSTEM IMPLEMENTATION

In this section we briefly describe the pollution estimation system implementation. As shown in Fig. 5, the system comprises three main parts: data collection, cloud server and web-based application for users to visualize the pollution contour map.

A. Data Collection

The first component of the system is data collection. As described in Section III, air pollution is collected both from a fixed-based (static) and mobile sensor network.

We have used scripts which are written in Perl to get fixed stations data from the NSW Office of Environment and Heritage (OEH) government website [2] since March 2010. We also partnered with OEH who provided us data between May 2009 and March 2010. Carbon Monoxide data is updated hourly from eight monitoring sites around whole New South Wale area.

Data from the sensor network is collected and uploaded using Node sensors and a mobile application every 5 seconds. A few data contributors collected data while they were commuting between home and workplace. All the sensors are calibrated every half year, using calibration devices which are supported by OEH.

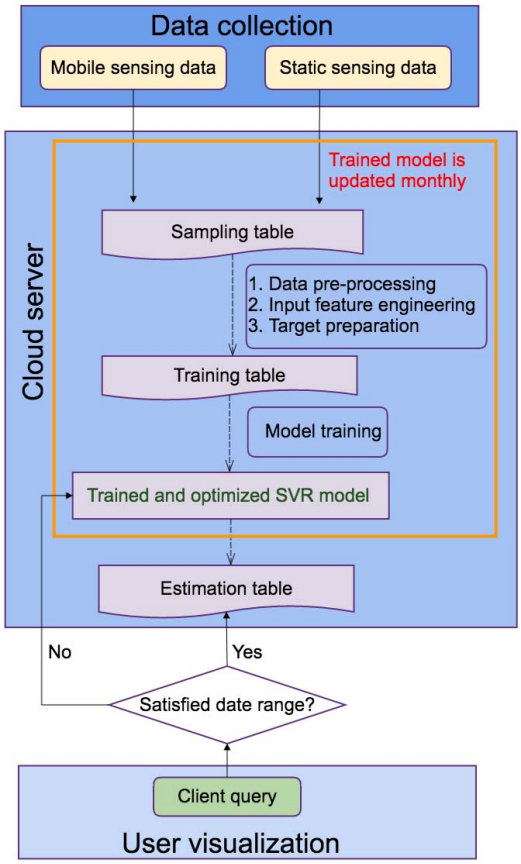


Fig. 5. System flowchart.

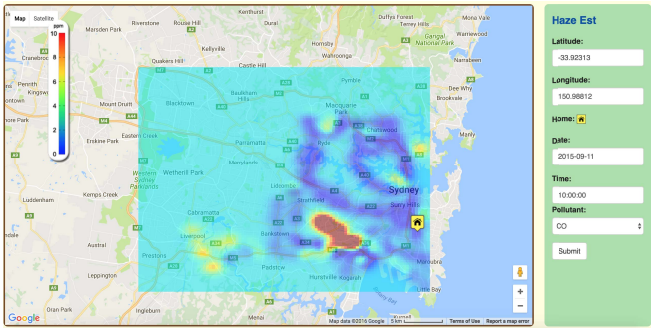
B. Cloud Server

The cloud server is the central component, which is hosted in our data center. We use a relational MySQL database, for its efficiency and reliability while searching over a large dataset. Collected sensing data is stored in a table named Samples which have fields like latitude, longitude, location_name (for the fixed stations), user_id, and device_id, etc. Data from Samples is processed and fed into the estimation model to filter and generate values. These are then inserted into a table for training. Model training is executed on these to get trained and an optimized SVR model. New sensing data is uploaded every day to our database, therefore we execute this data filtering and training occur every month to train an updated SVR model.

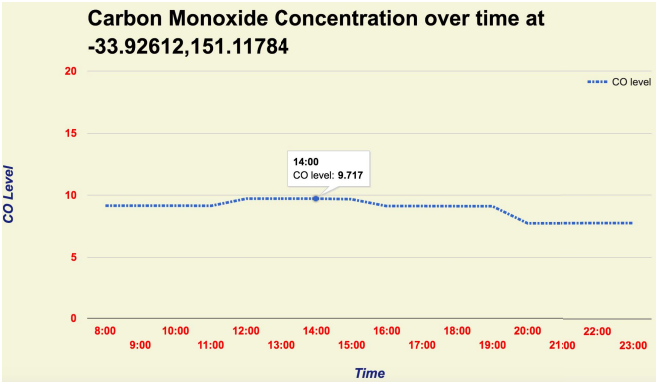
To reduce the estimation query response time, we have generated one-year estimation data from May 2015 to May 2016 and saved it in an estimation table. There are around 8700 hours of data in the estimation table, with each hour having 10, 000 values for the grid.

C. Web Application

The last component is web application [31], with which users can visualize pollution estimations in Sydney in any particular hour in the past seven years. We set a date detection step between user query and database, to speed up querying time. If the query date is between May 2015 to May 2016, our system retrieves data from the database directly. If not, it analyzes the user query, extracts the input features, and feeds



(a)



(b)

Fig. 6. (a) Contour map of CO estimation concentrations on Google map, and (b) plot of 8:00 am–11:00 pm CO concentrations at a particular location.

them into the trained model to get 10, 000 estimation values for the particular hour.

Sample screen shots of our web application are shown in Fig. 6. It comprises two parts – the first part as shown in Fig. 6(a) is the air pollution contour map visualization part, and the second part is a chart which shows whole day concentration variations in a particular location as shown in Fig. 6(b). The panel on the right of the first part allows users to input data such as location (latitude/ longitude), date, time and pollutant (CO is the only available pollutant at present). Location information is needed for the second part and all the rest of the inputs are required to generate estimations. The right side of the first part shows the contour map along with the pollution level indicator bar. The second part shows the concentration variations in a particular location which can be set via input or dragging the yellow home location icon on the map. Variation trends will change based on the selection of different locations in real-time. Users do not need to refresh the web page to get a new pollution variation chart, which makes the web application more user-friendly.

VIII. Conclusion

In this paper, we introduce a novel machine learning based dense air pollution estimation system which utilizes historical data both from (sparse) government monitoring sites and (dense) wireless sensor network. We choose seven regression models and compare the estimation performances, and then select SVR as the machine learning algorithm. This is applied in our system for its optimal air pollution surface estimation performance. A web application is also developed for users to visualize air pollution contour maps in Sydney for any given hour in the past seven years. In the model training part, we show that the estimation accuracy of SVR, DTR, and RFR has better estimation performances among all seven regression algorithms, using both entire dataset and test dataset. During the entire dataset training, which uses ten-fold cross validation, RFR has the lowest MAE and RMSE which is 0.295037 and 0.611891 respectively. In the test dataset validation, DTR has the lowest MAE which is 0.276770 while RFR has the best RMSE which is 0.642000. In the air pollution surface estimation part, estimates using SVR correspond well with the sensing interpolation map, and can more clearly highlight the most polluted area in greater Sydney compared with other regression models. These results indicate that our system can generate accurate air pollution estimations, and highly increase the air pollution map resolution. We believe our system is useful for long-term air pollution related disease research. In the future we aim to introduce meteorological factors into this system, such as weather and wind speed, to increase the estimation accuracy.

References

[1] AirNow. *Air Quality Index*, accessed on Dec. 10, 2016. [Online]. Available: https://www.airnow.gov/

[2] Australia-NSW-Environment and Heritage. *Air Quality Index*, accessed on Dec. 10, 2016. [Online]. Available: http://www.environment.nsw.gov.au/aqms/aqitable.htm

[3] D. Hasenfratz *et al.*, "Deriving high-resolution urban air pollution maps using mobile sensor nodes," *Pervasive Mobile Comput.*, vol. 16, pp. 268–285, Sep. 2015.

[4] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.*, 2013, pp. 15:1–15:8.

[5] B. Predić, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer, "Exposuresense: Integrating daily activities with air quality using mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2013, pp. 303–305.

[6] A. Marjovi, A. Arfire, and A. Martinoli, "High resolution air pollution maps in urban environments using mobile sensor networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst.*, Jun. 2015, pp. 11–20.

[7] T. Schikowski *et al.*, "Association of ambient air pollution with the prevalence and incidence of COPD," *Eur. Respiratory J.*, vol. 44, no. 3, pp. 614–626, 2014.

[8] O. Raaschou-Nielsen *et al.*, "Air pollution and lung cancer incidence in 17 European cohorts: Prospective analyses from the European study of cohorts for air pollution effects (ESCAPE)," *Lancet Oncol.*, vol. 14, no. 9, pp. 813–822, 2013.

[9] O. Raaschou-Nielsen *et al.*, "Particulate matter air pollution components and risk for lung cancer," *Environ. Int.*, vol. 87, pp. 66–73, Sep. 2016.

[10] D. J. Briggs, *et al.*, "Mapping urban air pollution using GIS: A regression-based approach," *Int. J. Geogr. Inf. Sci.*, vol. 11, no. 7, pp. 699–718, 1997.

[11] V. Singh, C. Carnevale, G. Finzi, E. Pisoni, and M. Volta, "A cokriging based approach to reconstruct air pollution maps, processing measurement station concentrations and deterministic model simulations," *Environ. Model. Softw.*, vol. 26, no. 6, pp. 778–786, 2011.

[12] M. E. S. S. A. A. Grid Environment. *Node+ Platform*, accessed on Dec. 17, 2016. [Online]. Available: http://www.commsp.ee.ic.ac.uk/~wiser/message/

[13] M. I. Mead *et al.*, "The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks," *Atmos. Environ.*, vol. 70, pp. 186–203, May 2013.

[14] P. Dutta *et al.*, "Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *Proc. SenSys Demonstration*, Berkeley, CA, USA, Nov. 2009, pp. 349–350. [Online]. Available: http://www.communitysensing.org/

[15] Vanderbilt University. *MAQUMON*, accessed on Dec. 17, 2016. [Online]. Available: http://www.isis.vanderbilt.edu/projects/maqumon
[16] EPFL. *Opensense*, accessed on Dec. 17, 2016. [Online]. Available: http://http://opensense.epfl.ch/wiki/index.php/OpenSense_2
[17] K. Hu, V. Sivaraman, B. G. Luxan, and A. Rahman, "Design and evaluation of a metropolitan air pollution sensing system," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1448–1459, Mar. 2016.
[18] V. Sivaraman, J. Carrapetta, K. Hu, and B. G. Luxan, "HazeWatch: A participatory sensor system for monitoring air pollution in Sydney," in *Proc. IEEE 38th Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Oct. 2013, pp. 56–64.
[19] K. B. Shaban, A. Kadri, and E. Rezk, "Urban air pollution monitoring system with forecasting models," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2598–2606, Apr. 2016.
[20] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Sep. 2014, pp. 69–77.
[21] V. Inc., *Node+ Platform*, accessed on Jan. 14, 2017. [Online]. Available: http://www.variableinc.com
[22] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.
[23] L. Rokach and O. Maimon, *Data Mining With Decision Trees: Theory and Applications*. Singapore: World Scientific, 2014.
[24] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[25] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[26] T. Chen and T. He. *eXtreme Gradient Boosting*, accessed on Jan. 21, 2017. [Online]. Available: https://github.com/dmlc/xgboost
[27] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, nos. 14–15, pp. 2627–2636, 1998.
[28] J. Neter *et al.*, "Applied linear statistical models," *Chicago Irwin*, vol. 4, p. 318, Sep. 1996.
[29] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 23–27.
[30] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Sep. 2011.
[31] UNSW. *Haze Est*, accessed on Mar. 20, 2017. [Online]. Available: http://www.hazewatch.unsw.edu.au

Ke Hu received the B.Eng. degree from Beijing Jiaotong University in 2007 and the M.Eng. degree from Xinjiang University in 2012, where he is a Research Assistant. He is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. He was with TE Connectivity as a Market Engineer and as a Project Manager with China Mobile in China. His research interests include wireless sensor network, machine learning, statistical modeling, and air pollution.

Ashfaqur Rahman (SM'15) received the Ph.D. degree in information technology from Monash University, Australia. He is currently a Senior Research Scientist and the Team Leader with the Data61 Division of CSIRO. His key research areas are machine learning and data mining. He and his team are involved in multidisciplinary research teams with a strong focus on agriculture and mining. He has published more than 80 peer-reviewed journal articles, book chapters, and conference papers. He was involved in organizing a number of key events including the IEEE DICTA 2013, MSLDA 2014, and MLSDA 2016.

Hari Bhrugubanda received the B.Sc. degree majoring in physics and computer science from the University of Sydney in 2014. He is currently pursuing the master's degree in information technology, with a specialization on data science, software engineer on a mobile payments platform. He designed LTE fixed wireless systems for several years in Australia, as a Network Engineer. His interests include technology and applications of machine learning.

Vijay Sivaraman received the B.Tech. degree from the Indian Institute of Technology, New Delhi, India, in 1994, the M.S. degree from North Carolina State University in 1996, and the Ph.D. degree from the University of California at Los Angeles, Los Angeles, CA, USA, in 2000. He was with Bell Labs as a Student Fellow, in a Silicon Valley start-up manufacturing optical switch-routers, and a Senior Research Engineer with the CSIRO, Australia. He is now an Associate Professor with the University of New South Wales, Sydney, Australia. His research interests include optical networking, packet switching and routing, network architectures, and sensor networks for the environment, health-care, and sports monitoring.