Pollution Prediction Model Using Data Collected by a Mobile Sensor Network

Pedro Mariano[1,2,3] Susana Marta Almeida[1] Pedro Santana[2]
[1] *Centro de Ciências e Tecnologias Nucleares, Instituto Superior Técnico, Lisboa, Portugal*
[2] *Instituto Universitário de Lisboa (ISCTE-IUL), Instituto de Telecomunicações, Lisboa, Portugal*
[3] *University of Lisboa, Faculty of Sciences, BioISI - Biosystems & Integrative Sciences Institute, Lisboa, Portugal*

***Abstract*—In this paper we investigate how to build a model to predict pollution levels using geographical information. By focusing on this kind of attributes we hope to contribute to an effective city management as we will find the urban configurations that conduct to the lowest pollution levels. We used decision trees to build a regression model. We performed a parameter grid search using cross validation. Ablation analysis where some attributes were removed from training showed that geographical based attributes impact the prediction error of decision trees.**

***Index Terms*—machine learning, air pollution, geographic information system**

I. Introduction

Over the last decades, exposure to airborne particulate matter (PM) has been identified as an important risk factor for human mortality, and negative health outcomes have been observed at concentrations usually experienced in cities. Even though the air quality in Europe has been improving due to emission control strategies, PM concentrations are still exceeding the EU limit values and the WHO air quality guidelines in many cities, conducting to more than 400000 premature deaths annually [1]. Consequently, prompt action through efficient air quality management is required, not only to ensure that the legal limits are not exceeded, but also to guarantee that the consequences of poor air quality are controlled and minimized.

The levels of PM in the cities depend on a combination of factors, such as emissions, meteorology and dispersion conditions, which are affected by the topography, green infrastructures and geometry of the streets and buildings. The management of air quality requires quantitative estimates about the impact of these factors in the air quality.

In this paper we investigate how we can build a regression model for air pollution using only geographical information and data from a mobile sensor network. We focus on machine learning methods such as decision tree (DT) in contrast to other approaches on pollution modeling. We analyse which attributes are more relevant to give a correct prediction. This work is part of the ExpoLIS project, which aims at deploying a sensor network on buses.

Since the hardware from ExpoLIS is still under development [2] we use the dataset collected by the OpenSense project [3], which also used a mobile sensor network. This dataset has 2.5 years of measurements of number of particles, particle diameter, and lung deposited surface area (LDSA). This data has been used to construct air pollution maps [4].

There are different approaches on pollution prediction and modeling: Gaussian plume models, computational fluid dynamics [5], Krigging [6], land use regression [7], and neural networks [8]. For a review on particle dispersion see [9]. Within these approaches, different variables are used as parameters to tune the models. In [7] the authors use distance to roads, in [5] they focus on urban canyons formed by buildings along side a road geometry. Another attribute that affects pollution is topography [8]. Time (week day and month) also affects pollution levels as it was one of the most effective attribute to predict air pollution [8].

Meteorological conditions such as wind speed and direction are known to also affect air pollution [10]. Nevertheless, we are interested in investigating the accuracy of a pollution regression model that does not rely on wind speed and direction but focuses in urban geographical information. By focusing on this type of information, we hope to find the best urban configuration (building geometry, presence of vegetation, number of roads) with lowest pollution levels.

II. Methods

A. Dataset Preparation

The dataset collected by the OpenSense project [3] was used. The data was gathered during the period April 2012 to December 2014 by a mobile sensor network. The measuring hardware was mounted on the top of 10 streetcars that operated in the city of Zurich, Switzerland. The dataset contains time of day, geographical location, number of particles, average particle diameter, and LDSA. The sensor data was stored in a PostgreSQL database with PostGIS extension.

The computation of geographical information was based on data provided by Open Street Map, in particular a user defined rectangle bounded by coordinates $(47°17'\mathrm{N}, 8°26'\mathrm{E})$ and $(47°30'\mathrm{N}, 8°39'\mathrm{E})$ was used.[1] This information was stored in a database using the tool `osm2pgsql`.[2] This is also a PostgreSQL/PostGIS database.

This work has been funded by FCT - Foundation for Science and Technology, I.P., within the framework of the project ExpoLIS (LISBOA-01-0145-FEDER-032088).

[1]The data provider at https://overpass-api.de was used.
[2]https://github.com/openstreetmap/osm2pgsql

```
CREATE FUNCTION attribute_area_greenery_in_a_circle (
   IN position GEOGRAPHY,
   IN radius INTEGER
)
RETURNS FLOAT
LANGUAGE SQL
AS
$$
   SELECT
      SUM (
         ST_Area (
            ST_Intersection (
               ST_Buffer (
                  position,
                  radius,
                  ''
               ),
               way
            )::geography
         )
      )
   FROM __table_polygon__
   WHERE
      ST_DWithin (
         way,
         position,
         radius,
         TRUE
      )
      AND (
         landuse = 'forest' OR
         landuse = 'garden' OR
         landuse = 'grass' OR
         landuse = 'plant_nursery' OR
         leisure = 'garden' OR
         leisure = 'park' OR
         "natural" = 'grass' OR
         "natural" = 'grassland'
      )
$$
;
```

Fig. 1. SQL function used to compute vegetation area attribute.

The attributes used in the learning task were stored in a third database. These are divided in date/time and geographical. Date/time attributes are minute of day $\{0, 1, \ldots, 60 \cdot 24 - 1\}$, day of week $\{0, 1, \ldots, 6\}$, and week of year $\{0, 1, \ldots, 52\}$. Geographical attributes are road and vegetation area, all within a circle with radius r. All circles are centred in a grid cell centre. Each entry in this database corresponds to a sensor reading in the first database.

Computation of the geographical attributes was done in SQL using the API of PostGIS. Figure 1 shows an example of the function to compute the attribute vegetation area within a circle with radius r. It takes as parameters the geographical coordinates of a given location and r. For efficiency, the geographical objects within r units of the position are filtered (first condition in the `WHERE` clause). Each Open Street Map has a set of tags that describe the objects, which are used in the second part of the `WHERE` clause to select the objects that are characterised as vegetation.

The time complexity of the functions that compute the geographical attributes increases linearly with the number of Open Street Map objects that are imported in the database.

TABLE I
NUMBER OF GRID CELLS AND GAIN FOR DIFFERENT GRID CELL LENGTHS.

grid cell length (m)	number of grid cells	gain (%)
1	1 225 260	84.92
2	489 478	93.98
5	139 263	98.29
10	52 671	99.35
20	19 684	99.76

The smallest rectangle that contains all the sensor readings locations could be used, but Open Street Map data providers leave out objects that are not entirely contained in a rectangle. This would affect the result of the computation of geographical attributes, that is, the resulting value would be underestimated. We could use regional or city maps that are provided by a couple of Open Street Map data providers, but in the case of Zurich it contains too much objects. As a compromise, the rectangle mentioned above was chosen.

The database has around $36 \cdot 10^6$ sensor readings that correspond to 21 019 480 unique geographical locations. If the computation of a geographical attribute for a single location takes $1\,\mathrm{second}$, then computing the geographical attributes for all unique locations would take 243 days. In order to reduce the time needed to compute all geographical attributes, a rectangular grid was considered. Table I shows how many grid cells with sensor readings there are for different grid cell lengths. Column *gain* shows the expression $1 - c_i/u$, where u is the number of unique geographical locations and c_i is the number of grid cells (with sensor readings) when using a grid cell length of i. This column is thus the speed up gain in computing a geographical attribute.

A grid cell of $2\,\mathrm{m}$ was chosen as it was a good compromise between number of grid cells and time to compute geographical attributes. These cells span a latitude of $0°0'0.0972''$ and a longitude of $0°0'0.0648''$. Figure 2 shows the locations and number of sensor readings that were used.

Figure 3 shows the histograms of collected sensor data: number of particles, diameter of particle and LDSA. Examining these histograms and comparing with the error of the regression model, we can measure how good was the prediction.

B. Model Parameters

We have used the *scikit-learn* python package [11] to perform parameter exploration, model learning and sensitivity. This package provides different methods (DTs, neural networks (NNs), Gaussian processes (GPs), k-nearest neighbours (KNN)) to build a regression model. DTs were selected due to their fast training speeds.

The *scikit-learn* package allows to perform parameter exploration. In a DT we have explored: the maximum depth of the tree (higher values produce a DT that is specialised in the training set), called *max depth* henceforth; and the threshold used in expanding a tree node that depends on the number of

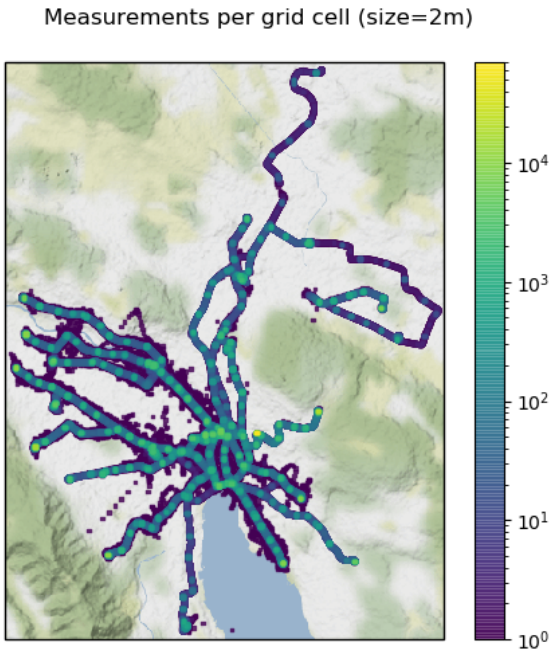

Fig. 2. Location and number of sensor readings.

samples, called *min samples* henceforth. For other parameters we refer the reader to the *scikit-learn* documentation.[3]

Parameter exploration was done using grid search with cross validation. Afterwards, the best parameters were tested in a validating dataset (not used during cross validation).

To assess if an attribute is relevant to the prediction tasks, we have performed an ablation analysis. This analysis is usually done with NN and consists in removing input neurons. The new network is feed the test data set and the results are compared with the unmodified network. In the case of DTs we opted to train a new DT but using fewer attributes: road and vegetation, only roads, only vegetation, without minute of day, or day of week, or week of year. For the best DT parameters found and attribute set, we performed 10-fold cross validation.

III. Results

Table II shows the result of the DT parameter grid search using cross validation. The prediction error shown in each entry of the table is the absolute difference between the predicted and true pollution levels:

$$\frac{1}{n}\sum_{i}|f_i(x_i) - y_i(x_i)|, \tag{1}$$

where x_i is the ith data sample, $y_i(x_i)$ is the pollution ground-truth (either number of particles, particle diameter, or LDSA) of data sample x_i, $f_i(x_i)$ is the predicted pollution, and n is the number of data samples. Rows NL mean that no limit was imposed on the maximum depth of the learned DT. Columns 0.1% and 0.05% mean that if the fraction of samples

[3]https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

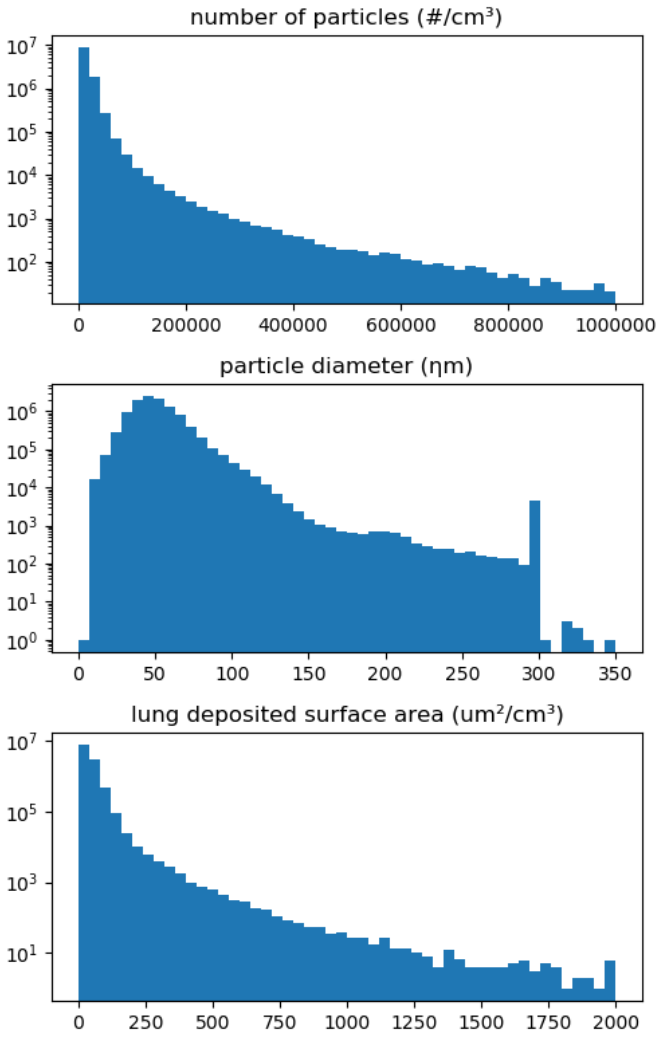

Fig. 3. Histograms of collected sensor data.

(compared to the dataset size) in a tree node had at least the previous percentage, then it was expanded. Rightmost column $1s$ means that if the number of samples in a tree node was higher than one, then it was expanded. As the depth of a DT gets higher, the prediction error gets lower (towards zero). Bigger DTs can lead to over-fitting and poorer generalisation capabilities. However, even if we allow a DT to grow as large as possible (NL rows) or tree nodes are introduced when the number of samples is higher than one, the error does not reach zero. This is due to imprecision in the sensors.

Table III shows the prediction error on the validating set using the DT parameters that had the best result during grid search. As can be seen, the error is similar to the prediction error obtained during grid search. Moreover, if we compare with the range of sensor values shown on the histograms in Figure 3, the error is small compared to the range of values: 0.5%, 1.3% and 0.4% for number of particles, particle diameter and LDSA, respectively. Notice that this percentage is computed against the range of measured sensor values (the

TABLE II
RESULTS OF DECISION TREE PARAMETERS GRID SEARCH. PREDICTION ERROR OF REGRESSION MODEL.

		number of particles		
		min samples		
		0.1%	0.05%	1s
max depth	10	7371.915	7281.015	7139.713
	12	7219.262	7074.611	6776.82
	14	7123.724	6933.215	6429.622
	16	7075.938	6847.629	6093.787
	NL	7041.493	6774.615	**4946.988**
		particle diameter		
		min samples		
		0.1%	0.05%	1s
max depth	10	8.465	8.296	8.078
	12	7.99	7.672	7.172
	14	7.759	7.353	6.469
	16	7.619	7.146	5.868
	NL	7.561	7.007	**4.662**
		LDSA		
		min samples		
		0.1%	0.05%	1s
max depth	10	13.801	13.5	13.132
	12	13.386	12.9	11.991
	14	13.134	12.488	10.873
	16	13.035	12.289	9.933
	NL	12.993	12.163	**7.787**

horizontal axis in Figure 3, and the fifth column in Table III). It is not the range of values that the sensor electronics is capable of measuring.

As the DT unlimited *max depth* parameter and *min samples* equal to one where the ones that showed the lowest prediction error, they were used in the ablation experiment. Figure 4 shows the prediction error for all combinations of removed attributes for all collected sensor data. The effect of using only the vegetation geographical attribute improved the prediction error of DT in respect to number of particles. Regarding the other sensor data (particle diameter and LDSA), only when we removed one of the time attributes we observed an increase in prediction error.

IV. DISCUSSION

Examining the prediction error shown in Figure 4, if we compare the effect of only dropping a time attribute (left column in the figure), the prediction error increases significantly (p-value with significance level at 95%) in all cases. If in addition only roads are used (middle column), again the prediction error increases. If only vegetation is used and a time attribute is dropped, then the prediction error increases in all cases except: day of week for number of particles, and minute of day for particle diameter.

Regarding the geographical attribute, if we compare the effect of only using one of them, then the prediction error decreases if only vegetation is used. If in addition a time attribute is also dropped, then again only when vegetation is used, the prediction error is statistically different. This means that vegetation is more important to obtain a reliable regression model.

The models obtained with DT can be interpreted but there are known limitations in the models that can be built. The learning algorithm also performs poorly when the dataset is unbalanced regarding the number of classes or in our case the distribution of the target function. As such it is interesting to consider other types of models.

NN can approximate any function. However, in this work, the time complexity of backpropagation is a problem. If there are n data samples, m attributes, k hidden layers each containing h neurons, and o output neurons, then backpropagation has a time complexity of $O(n \cdot m \cdot h^k \cdot o \cdot i)$, where i is the number of iterations until convergence. In this work, the time complexity expression is dominated by the number of data samples (as has been said the number of sensor readings is around $36 \cdot 10^6$). Preliminary tests using NN did not produced regression models with better prediction errors. Doing parameter exploration grid search on the learning parameters of NN is very time consuming.

GPs can provide interpretable models but suffer from cubic time and quadratic space complexities on the number of training samples. In our case this would require storing a data structure with $(36 \cdot 10^6)^2 = 1\,156\,000\,000\,000\,000 \approx 10^{15} = 1$ Peta. There is ongoing research to tackle large datasets [12].

KNN is a parameter less method that has a wide success in prediction and regression [13]. The models that are built use the data that is more similar to the target pollution level. In this work, time complexity is also an issue as it is necessary to find for every data sample the nearest one. A naive implementation of KNN can take quadratic time on the number of training samples, but with appropriate data structure this can be reduced to $O(n \log n)$.

V. CONCLUSIONS

We have presented preliminary results of building a pollution prediction model based on data collected by a mobile sensor network using as attributes geographical information. Although there is time lag between sensor collection and the computation of geographical attributes, the prediction error of the model was around 1% of the range of measured sensor values.

Regarding the usage of geographical information to predict pollution data, we observed that the presence of attribute vegetation reduced prediction error only with number of particles pollution data. For the other two data, the attributes that depend on time are more important to reduce prediction error.

As future work we plan to investigate how to reduce data size in order to use other regression models. Decision trees have known limitations on the models that can be obtained. Preliminary experiments with NN, GP and KNN failed either because of the time needed to build a model or due to memory restrictions. One candidate avenue is to reduce considerably the size of the training dataset.

TABLE III
PREDICTION ERROR ON VALIDATING SET.

sensor data	max depth	min samples	prediction error	value range	fraction
number of particles	NL	1s	4704.374	1 000 000	0.5%
particle diameter	NL	1s	4.446	350	1.3%
LDSA	NL	1s	7.421	2000	0.4%

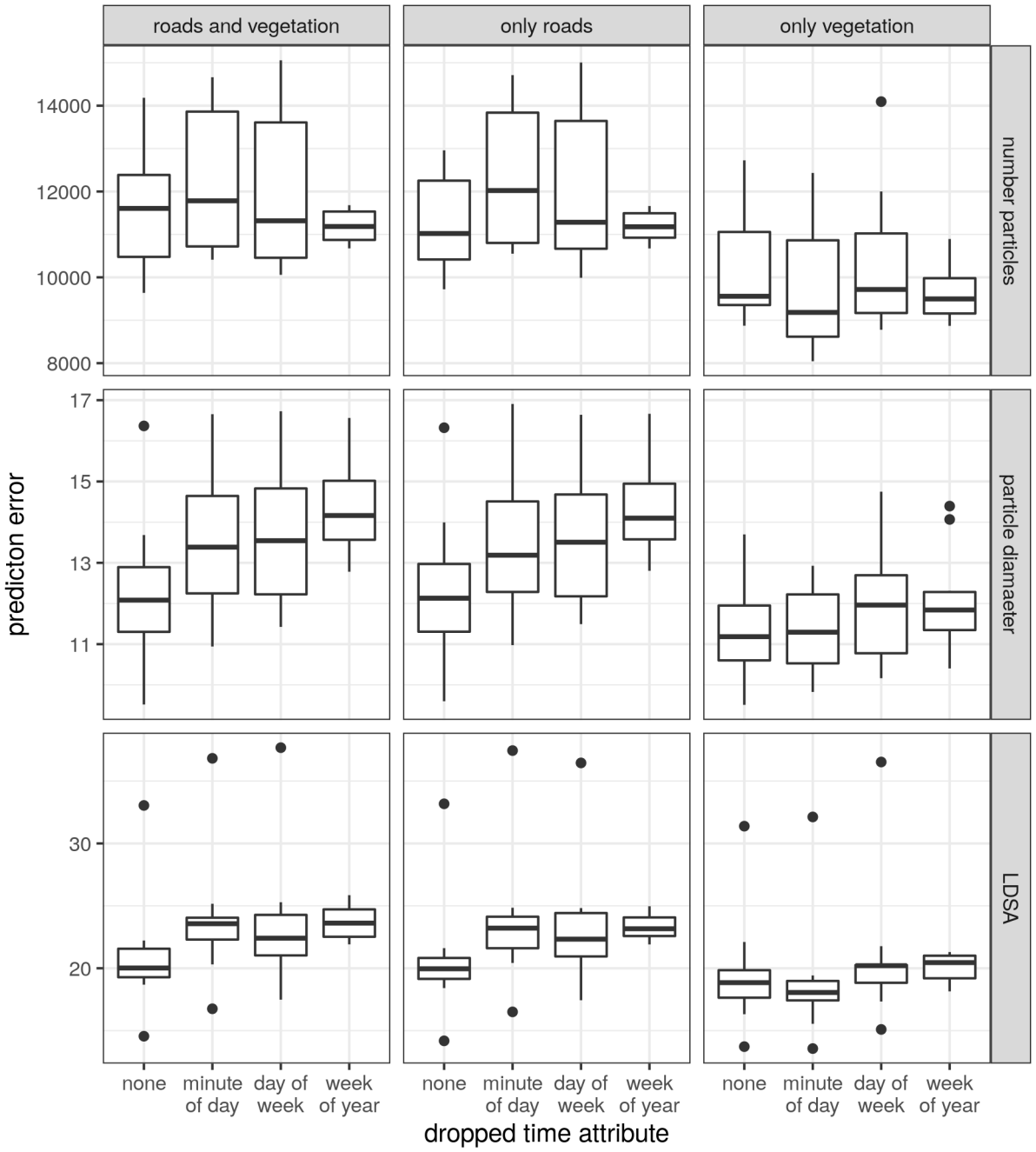

Fig. 4. Prediction error of ablation test.

REFERENCES

[1] "Air quality in europe – 2019 report," Report No 10/2019, European Environment Agency, ISSN 1977-8449.

[2] P. Santana, A. Almeida, P. Mariano, C. Correia, V. ania Martins, and M. Almeida, "An affordable vehicle-mounted sensing solution for mobile air quality monitoring," To appear in SpliTech 2020 - 5th International Conference on Smart and Sustainable Technologies, 2020.

[3] B. Maag, D. Hasenfratz, O. Saukh, Z. Zhou, C. Walser, J. Beutel, and L. Thiele. Ultrafine particle dataset collected by the opensense zurich mobile sensor network. [Online]. Available: http://doi.org/10.5281/zenodo.1415369

[4] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, T. Arn, J. Beutel, and L. Thiele, "Deriving high-resolution urban air pollution maps using mobile sensor nodes," *Pervasive and Mobile Computing*, vol. 16, pp. 268–285, 2015, selected Papers from the Twelfth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2014). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001928

[5] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling air quality in street canyons: a review," *Atmospheric Environment*, vol. 37, no. 2, pp. 155–182, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1352231002008579

[6] P. Schneider, N. Castell, M. Vogt, F. R. Dauge, W. A. Lahoz, and A. Bartonova, "Mapping urban air quality in near real-time using observations from low-cost sensors and model information," *Environment International*, vol. 106, pp. 234–247, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0160412016310741

[7] P. H. Ryan, G. K. LeMasters, P. Biswas, L. Levin, S. Hu, M. Lindsey, D. I. Bernstein, J. Lockey, M. Villareal, G. K. K. Hershey, and S. A. Grinshpun1, "A comparison of proximity and land use regression traffic exposure models and wheezing in infants," *Environ Health Perspect.*, vol. 115, no. 2, p. 278–284, Feb 2007.

[8] M. R. Delavar, A. Gholami, G. R. Shiran, Y. Rashidi, G. R. Nakhaeizadeh, K. Fedra, and S. Hatefi Afshar, "A novel method for improving air pollution prediction based on machine learning approaches: A case study applied to the capital city of tehran," *ISPRS International Journal of Geo-Information*, vol. 8, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2220-9964/8/2/99

[9] P. Kumar, M. Ketzel, S. Vardoulakis, L. Pirjola, and R. Britter, "Dynamics and dispersion modelling of nanoparticles from road traffic in the urban atmospheric environment—a review," *Journal of Aerosol Science*, vol. 42, no. 9, pp. 580–603, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021850211000887

[10] Z. Ghaemi, A. Alimohammadi, and M. Farnaghi, "LaSVM-based big data learning system for dynamic prediction of air pollution in Tehran," *Environmental Monitoring and Assessment*, vol. 190, no. 5, may 2018.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] H. Liu, Y. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–19, 2020.

[13] G. H. Chen and D. Shah, "Explaining the success of nearest neighbor methods in prediction," *Foundations and Trends in Machine Learning*, vol. 10, no. 5-6, pp. 337–588, 2018. [Online]. Available: http://dx.doi.org/10.1561/2200000064