Pedro Mariano Corresponding author
Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
Information Sciences and Technologies and Architecture Research Center (ISTAR-IUL), Lisboa, Portugal
Centro de Ciências e Tecnologias Nucleares (C2TN), Instituto Superior Técnico, Lisboa, Portugal
`plsmo@iscte-iul.pt`

Susana Marta Almeida
Centro de Ciências e Tecnologias Nucleares (C2TN), Instituto Superior Técnico, Lisboa, Portugal
`smarta@ctn.tecnico.ulisboa.pt`

Pedro Santana
Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
Information Sciences and Technologies and Architecture Research Center (ISTAR-IUL), Lisboa, Portugal
`pedro.santana@iscte-iul.pt`

On the Automated Learning of Air Pollution Prediction Models from Data Collected by Mobile Sensor Networks

Pedro Mariano*[1,2,3], Susana Marta Almeida[3], and Pedro Santana[1,2]

[1]Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
plsmo@iscte-iul.pt, pedro.santana@iscte-iul.pt
[2]Information Sciences and Technologies and Architecture Research Center (ISTAR-IUL), Lisboa, Portugal
[3]Centro de Ciências e Tecnologias Nucleares (C2TN), Instituto Superior Técnico, Lisboa, Portugal
smarta@ctn.tecnico.ulisboa.pt

Abstract

This paper addresses the problem of automated learning of air pollution predictive models that were trained using information gathered by a set of mobile low cost sensors. Concretely, fast to compute machine learning methods (Decision Trees and Support Vector Machines) were used to build regression models that predict air pollution levels for a given location. The models were trained using the data collected by the OpenSense project, in particular, number of particulate matter, particle diameter and lung deposited surface area (LDSA). We examined two different sets of attributes: one based on a geographical description of the location under analysis (e.g., distribution of households and roads), and another based on a time series of past air pollution observations in that location. Overall, we have found out that past measures lead to better pollution predictions. The best R^2 score was 0.751 obtained with the model that predicts LDSA and was trained with the data set with time series attributes, while the worst R^2 was 0.009 obtained with the geographical data set to predict number of particles. The performance of the best model is on par with similar air pollution systems. Moreover it can be used in a production system that requires frequent updates.

Keywords: machine learning, air pollution, time-series, land-use, decision tree, support vector machine.

Introduction

Air pollution is one of the most pressing problems in urban areas with very harmful impacts on the health and ecosystems. It is well documented that exposure to air pollution lead to adverse health effects, such as premature mortality and morbidity, mainly related to respiratory and cardiovascular diseases, asthma and allergies [35]. Moreover, the World Health Organization (WHO) has classified air pollution as carcinogenic to human beings [12]. More recently, the

*Corresponding author

latest report on air quality in Europe [5] shows that air quality implications are mainly due to high levels of particles suspended in the atmosphere (PM).

Airborne particulate matter sources range from coal combustion, transportation, power stations, among others. Its rise is linked to economical growth [10] and can trigger the development of different diseases [13]. Historically, air pollution is linked to industrialisation and urbanisation growth [37]. Part of these pollution sources are used to produce different types of energy which are fundamental to run an economy [9]. The impact in the environment is also well documented [20] ranging from over fertilisation, reduced photosynthesis, emission of green-house gases, among others. Tackling air pollution will require compromises in the energy sources that are used by our current society as well as current patterns of consumption.

The recognition of the sources of air pollution has lead to enactment of several laws to treat pollutants at the source [37]. Variations in anthropogenic emissions in Europe, especially in the last 30 years, have caused a decrease in the PM concentration values. However, several ecosystems and cities in Europe are still confronted with PM concentrations that go beyond European standards and, principally, the WHO Air Quality Guidelines. Rough calculations of the impacts of air pollution to health point out that PM2.5 (particles whose diameter is less than 2.5 µm) concentrations in 2018 were the cause of 417 000 deaths in the countries belonging to the European Union [5].

Air Pollution is perceived as the second biggest environmental concern for Europeans, after climate change [2] and, therefore, there is a growing political, public and media interest in air quality issues and increased public support for action. Moreover, exceeding European Standards and WHO guidelines leads to illnesses and to social and economic disruptions such as workplace slowdown, absenteeism and congested emergency rooms. China has experienced rapid industrial growth, which has lead to severe air pollution. This has raised the interest to tackle this issue [10]. Consequently, national, regional and local authorities have developed extensive air quality monitoring and information programs, whose objective is to increase the public's awareness of air quality state, especially with regard to health effects, so that individuals can modify their behaviour to protect their health.

It is vital to improve the systems used to predict air quality and warn the public of pending unhealthy conditions, so that direct exposure to ambient air pollution can be avoided during specific periods. Air pollution prediction is a complex and non-linear problem, once various factors affect air quality, such as the atmospheric conditions and geographical context like land use, traffic, topography, and location. Therefore, in the last decades, a wide variety of prediction techniques have be developed, which can be classified in two major categories: deterministic and statistical. One of the most used deterministic methods is the Gaussian Dispersion Model, where the physical and chemical processes that occur in the atmosphere are simulated in order to predict the quality of the air in a given place [33, 23]. Although the accuracy provided by dispersion models is good, we need reliable information regarding the places where pollutants appear, as well as the chemical and physical characteristics of the atmosphere, which can be difficult to obtain principally for large-scale applications. In addition, applying these models in a realistic setting, where the quantity of data is large, is very time consuming. These disadvantages have led to further development of the statistical methods to solve real-life problems, such as Machine Learning (ML) methods, which can model the complex relationships between air pollution and temporal and spatial variables [14], as these models are capable to approximate any complex non-linear function.

Due to decreasing costs, researchers have been using networks of sensors to monitor air quality. Having a set of mobile and low cost sensors (LCS) fits in the context of the Internet of things (IoT) as we are tapping in the available of cheap and accessible sensors to monitor air quality. This is a line of work that uses IoT to provide a sustainable society. In [21] the authors review the latest work on the usage of IoT technologies to provide a more sustainable society. One of the areas of IoT is on smart-cities where a network of sensors is deployed in order to monitor life quality indicators, manage critical systems, and provide real time information.

An example of a network of LCS to monitor environmental effects of air pollution is described in [1] where the authors report a network deployed in Salt Lake City, USA, which figures in the top most USA polluted regions. Their network has been used to study the exposure to air pollution. Another example is presented [34] where the authors describe how data collected by the sensor network was used to train a pollution classifier capable of tracking pollution on a hourly basis. The OpenSense project also deployed a network of mobile LCS in the city of Zurich. They were able to construct PM maps with high spatial and temporal resolutions, although maps with weekly or higher time scales were more accurate [11]. A final example is shown in [15], where the authors present a method to combine measures from their network of low cost sensors with the data collected by the Taiwan environmental monitoring stations that is able to improve spatial and temporal resolution of PM estimation.

The research reported here is part of the ExpoLIS project. Our goal in the ExpoLIS project is to deploy a network of mobile LCS to monitor air quality and to develop a set of software tools [28]. This paper is an extended version of our previous work [18] aiming at developing a pollution prediction software tool to assist the population. As such, in this paper we investigate how data collected from this type of network can be used to predict pollution levels. This contrasts with standard models where prediction is based on land-use, geographical and meteorological attributes. We improve the geographical data set (used on the previous paper) and we use a new data set based on a time series. This work improves on similar systems in that the prediction model is based on data collected by a network of mobile LCS and has to be updated on a regular basis.

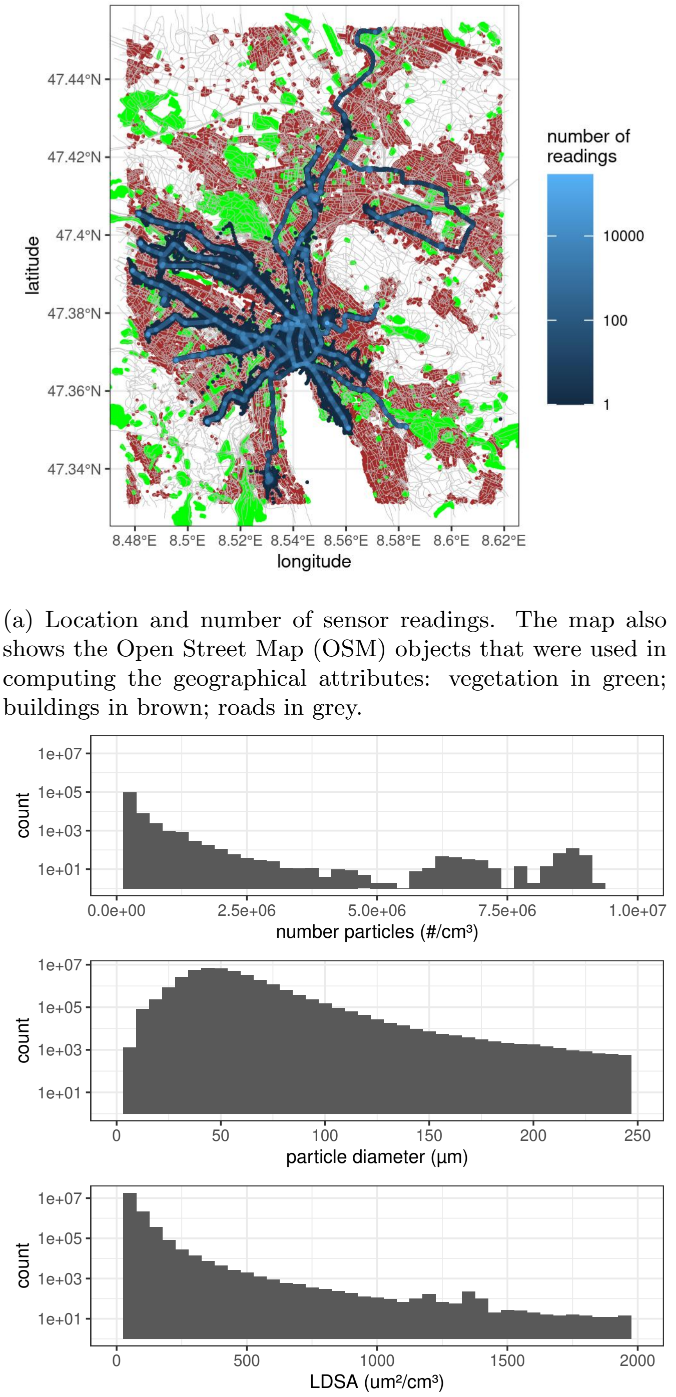
This paper is organised as follows. We start by presenting in Section *Material and Methods* the data sets and the ML methods that we used on the pollution prediction task. Next, in Section *Results*, we present the results that we obtained in terms of prediction performance and, in Section *Discussion*, we discuss and compare the predictions made by the models trained with the two data sets. We wrap up in Section *Conclusions* with concluding remarks and future avenues of research.

Materials and Methods

Data Set Preparation

As we have yet not deployed any sensors, we used the data collected by the OpenSense project [17] to train the ML models. They deployed a set of LCS in the trams of the city of Zurich. As we have mentioned, this data contains measurements of number of particles, particle diameter and LDSA. Besides air pollution data, they also stored the date and time with a precision of minutes, latitude, longitude, the GPS error, and the tram identification. The entire data set contains 36 818 362 sensor readings. Figure 1 shows an overview of the OpenSense data that we used.

For the purpose of the present work, the original data set was post-processed and extended to create two separate data sets, one based on geographical information and another based on temporal series. This separation has the goal of allowing investigating the impact that each information source may have on air pollution prediction error. The data set based on geographical information is a direct extension of our previous work [18].



(a) Location and number of sensor readings. The map also shows the Open Street Map (OSM) objects that were used in computing the geographical attributes: vegetation in green; buildings in brown; roads in grey.

(b) Histograms of collected sensor data.

Figure 1: Overview of OpenSense data.

Table 1: OSM tags and values used when computing a geographical attribute.

attribute	tag	value
building	`building`	*not empty*
greenery	`landuse`	`forest`
	`landuse`	`garden`
	`landuse`	`grass`
	`landuse`	`plant_nursery`
	`leisure`	`garden`
	`leisure`	`park`
	`natural`	`grass`
	`natural`	`grassland`
road	`highway`	*not empty*

Geographical Data Set

The geographical data set is stored in three databases. The raw sensor data, directly obtained from the OpenSense project, is stored in the first database, which is based on PostgreSQL with PostGIS extension.

The data that was used to compute geographical information is stored in a second database, also based on PostgreSQL with PostGIS extension. This data was fetched from an Open Street Map (OSM) provider[1]. We only selected the data located in a rectangle with geographical coordinates $(47^\circ17'\mathrm{N}, 8^\circ26'\mathrm{E})$ and $(47^\circ30'\mathrm{N}, 8^\circ39'\mathrm{E})$. To import the data from OSM to the database, the tool `osm2pgsql`[2] was used.

The third database contains the attributes used to train the pollution model. The attributes are the date, time, and geographical characterisation of the location where sensor data was collected by a tram. The date and time corresponds to the OpenSense data. The geographical characterisation is composed of the area of the buildings, road, and vegetation that are located in a circle around the location mentioned earlier. Compared to our previous work [18], the building area has been included in the present work. In an urban environment, buildings are a common feature, as well as urban canyons, which are known to reduce pollution dispersion. As such, building distribution is a crucial attribute that was missing from the previous paper.

We used the API of PostGIS to calculate the geographical characterisation. This API supplies functions that are used to filter objects on the surface of a sphere, for instance, objects within a circular region. We also selected the OSM objects that match the geographical attribute that we defined. This means setting conditions on OSM tags. Table 1 shows the conditions that were used in all three geographical attributes, namely, buildings, greenery/vegetation, and roads.

The geographical rectangle that we used is the same as the one from our previous paper [18], as it showed to encompass a number of OSM objects to be imported into the database that properly trades-off spatial coverage and computational efficiency. Concretely, this rectangle is wide enough to contain all relevant OSM objects that are at most 2 m from any sensor geographical location while simultaneously not resulting in excessive time required for a function to compute a single geographical attribute.

Another factor that has an impact on computing geographical attributes is the number of locations. The OpenSense data set that we used has 21 019 480 distinct geographical points. Such high number of points has an impact on the time needed to compute all geographical attributes. To cope with this challenge, a rectangular grid with grid cells of 2 m was used. A grid cell of

[1] `https://overpass-api.de`

[2] `https://github.com/openstreetmap/osm2pgsql`

Table 2: Attributes used in the pollution prediction based on geographical data task.

attribute	values
minute of day	$\{0, 1, \ldots, 60 \cdot 24 - 1\}$
day of week	$\{0, 1, \ldots, 6\}$
week of year	$\{0, 1, \ldots, 52\}$
building area	$]0, +\infty[$
vegetation area	$]0, +\infty[$
road area	$]0, +\infty[$

2 m spans a latitude of $0°0'0.0972''$ and a longitude of $0°0'0.0648''$. This approach results in 489 478 grid cells and a gain of 93.98% in terms of time to compute the geographical attributes. Figure 1a shows the result of building this rectangular grid where we can see grid cells and how many readings are per cell. This figure also presents the OSM objects that were used to compute the geographical attributes. Table 2 presents an overview of the attributes used in the pollution prediction based on geographical data task.

The histograms of air pollution data, presented in Figure 1b, clearly show a prevalence of low values. Number of particles shows a long tail that decays exponentially to outliers. It also has the greatest range of values compared to particle diameter and LDSA.

Time Series Data Set

Before creating the time series that was used to build the prediction model, the OpenSense data was checked for mistakes, outliers, or invalid values. A set of 44 records were found to be outside the places reached by trams. There is one record with a negative particle diameter value. In [7], the authors discuss the performance of the sensors used to collect the OpenSense data. They mention that for particles larger than 250 nm the sensor readings are not reliable. This lead us to exclude sensor readings whose particle diameter were greater than 250 nm. This condition only excludes 13 654 records from the time series creation. Examining the box plots of the sensor readings excluding the aforementioned data, it is possible to see that there are still outliers. A possible solution to remove these outliers would be to filter out sensor readings that are located above the third quartile plus 1.5 times the inter quartile range (computed in the original data). However, this procedure would exclude around 6%, 3.1% and 5.2% records considering only the number of particles, particle diameter, and LDSA, respectively. In alternative, we opted to exclude records whose number of particles value is greater than 10^6, which is a higher threshold than the inter quartile range criteria. This condition only excludes 2565 records. Combining all the conditions, a total of 16 263 records were discarded for the time series creation.

To create a time series, the spatial area under analysis is first divided in a rectangular grid with a cell size of c meters. In turn, time is divided in a sequence of slots $s_1, s_2, \ldots$, each with a duration of d minutes. For each cell g and slot s we compute the average of air pollution data yielding a set of x_{gs} values. If there is no data during a slot s in a cell g, then x_{gs} is undefined. The number of points of a time series is defined by l. A time series is thus described by:

$$\langle x_{gs_{t+1}}, x_{gs_{t+2}}, \ldots, x_{gs_{t+l}} \rangle \quad , \tag{1}$$

and is only created if all the $x_{gs_{t+i}}$, with $1 \leq i \leq l$, are defined.

Different values for the time series length (i.e. the number of points) and the duration of time slot were considered, as this allows us to build time series in order to generate predictions, given the last $l - 1$ hours or the last $l - 1$ days. Table 3 shows the time series generator parameters

Table 3: Parameter values used to create time series from the OpenSense data: l time series length, d duration of a time series slot (in minutes), c grid cell (in meters). The two right most columns show how many time series were created and how many geographical cells have time series.

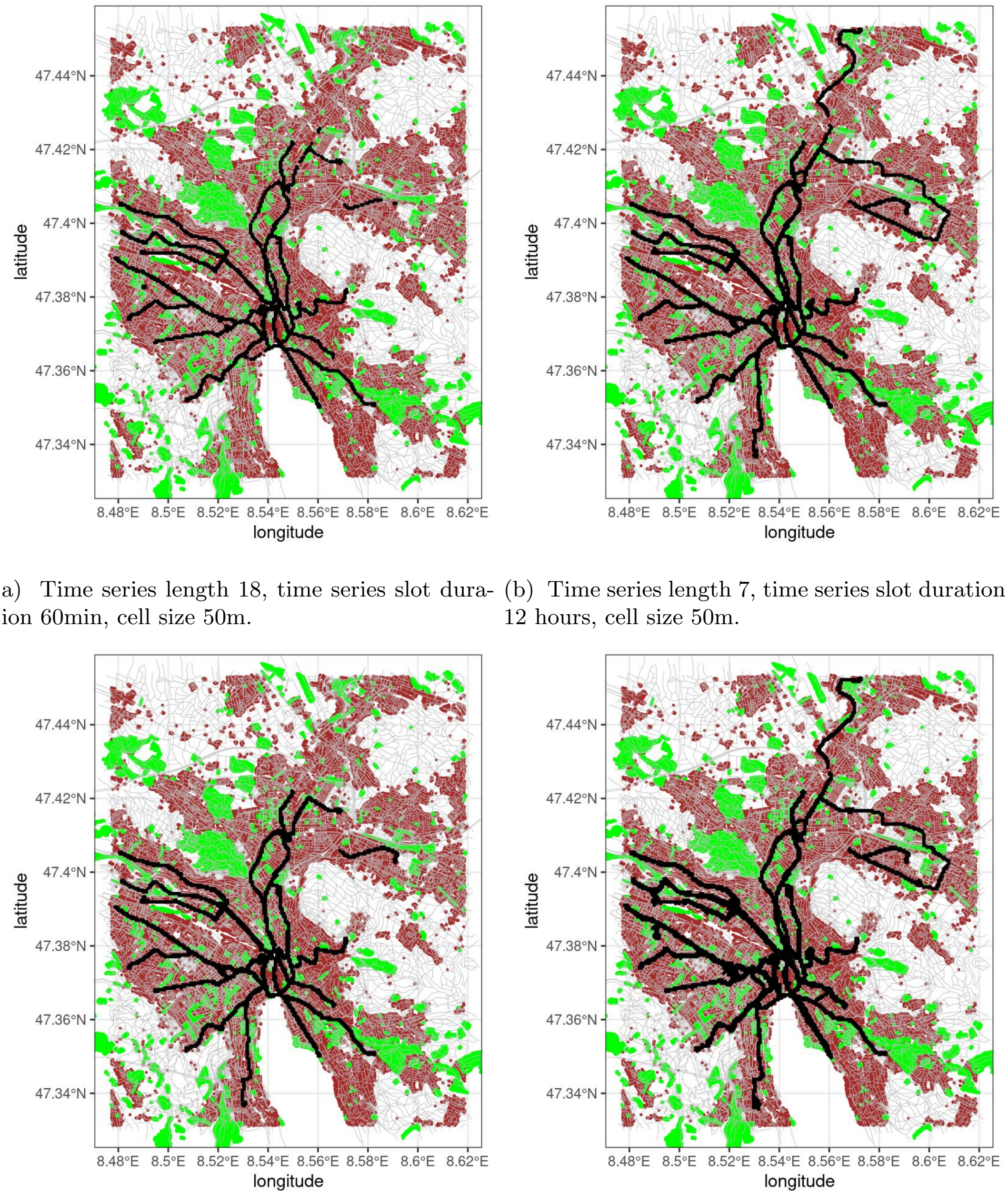
l	d	c	number of time series	number of cells	
2	60	50	4 498 529	2919	
3	60	50	3 543 084	2412	
6	60	50	2 153 097	1947	
12	60	50	854 828	1483	
18	60	50	170 529	1303	⋆
24	60	50	5192	168	
2	120	50	3 099 352	3036	
3	120	50	2 489 213	2571	
6	120	50	1 233 259	2094	
12	120	50	9502	570	
24	120	50	857	13	
2	180	50	2 221 138	3123	
3	180	50	1 671 246	2626	
8	180	50	62 421	1448	
16	180	50	7925	488	
2	720	50	866 444	3198	
3	720	50	653 907	2704	
7	720	50	293 631	2136	⋆
2	1440	50	455 255	3269	
3	1440	50	324 337	2737	
7	1440	50	120 587	1838	⋆

values considered in the context of this article. This table also shows the number of time series that were created with the given time series generator parameters, and the number of unique geographical cells that have a time series associated. Overall, a decrease on the number of time series and geographical cells covered is observed when the time series length and the time slot duration is increased. The main reason for this being daily gaps in measuring activity and absence of trams during the night.

Different ML methods have different requirements in terms of time complexity to train a model. Coupled with finding a set of hyper-parameters with reasonable performance, this increases the time spent finding a suitable model. This aspect is made more clear in the following section where we present the ML that we used in this paper. With this in mind, we selected ML methods with low time complexity to investigate which time series resulted in the best possible prediction model (see below). ML methods with higher time complexity were only tested in three time series (the ones marked with a star in Table 3). In Figure 2 we display the time series grid cells that were used with the more computationally time intensive ML methods. For comparison, this figure also shows time series generator parameters that produced the time series with the highest number of locations.

Prediction Models

To build air pollution prediction models, we used different machine learning methods, known to be fast to compute, namely Decision Trees (DTs), Neural Networks (NNs), Support Vector



(a) Time series length 18, time series slot duration 60min, cell size 50m.

(b) Time series length 7, time series slot duration 12 hours, cell size 50m.

(c) Time series length 7, time series slot duration 1 day, cell size 50m.

(d) Time series length 2, time series slot duration 60min, cell size 50m.

Figure 2: Locations of selected time series: a to c used in time intensive Machine Learning methods; d time series generator parameters that produced highest time series count.

Machiness (SVMs), and k-nearest neighbourss (KNNs). These methods belong to two of the five main fields of machine learning [4]. All the algorithms that build these methods have at least one hyper-parameter. The *scikit-learn* software library [22] was used to build the prediction models and to find a suitable set of hyper-parameters for them.

Overview of Prediction Models

A DT is a ML method that produces a set of if-then rules organised as a tree [19, 25]. Each inner node of the tree consists in a condition involving one attribute of the data set (e.g., geographical location). A leaf node is tagged with a class or in the case of regression a linear function of a subset of the attributes. While learning an optimal DT is a hard problem, greedy algorithms are usually used; these are very fast in creating a DT. Whenever the algorithm has to split a leaf node into an inner node, it looks for the attribute that is able to discriminate best the subset of the training samples at that leaf node. Afterwards the subset of training samples is divided among the child nodes of the newly created inner node, and then the algorithm examines the next leaf node. A leaf node is not split, if its subset of training samples have the same class or the same value (in the case of regression), or the size of the subset is small enough, or the depth of the node is high enough. Typical hyper-parameters are the depth of the tree, and the threshold (on the size of the subset samples of a leaf node) that blocks the algorithm from splitting a leaf node. In this paper we will refer to the last parameter as *min-samples-leaf.* These parameters control the over-fitting of the DT to the training set. There are other parameters that can be adjusted. We did not discuss them here, however they can be consulted in the *scikit-learn* documentation[3].

NNs are models inspired in the organisation of the neurons in the brain. A NN is composed of units called neurons, organised in layers. Each neuron receives as inputs the data from the previous layer, it performs a linear combination of the input, and then feeds the result to an activation function to produce the neuron's output [19, 25]. With the advent of *deep learning*, the popularity of NNs has risen due to its success in a variety of tasks [31]. Regarding hyper-parameters, we have to specify the activation function used in the middle and output layers, the number of layers, the number of neurons in the hidden layers (the size of the input and output layers are determined by the problem), and the number of training iterations. NNs are capable of approximating any function given the correct set of hyper-parameters and a sufficiently large model capacity (i.e., the number of neurons in the inner layers). The *scikit-learn* API provides other parameters that have minor impact on the produced, that again we point to the *scikit-learn* documentation. Training a single NN varies polynomial with the number of neurons, the number of records, and the number of iterations, which means that finding the best combination of hyper-parameters can be time consuming.

Before the rise of deep learning, SVM were very popular as they can also approximate any function but with more efficient training and fewer hyper-parameters, compared with NN [29]. A SVM depends on a user-specified kernel function (e.g., linear, polynomial, radial basis) that is used to map the attribute space to a higher order space, in which a linear combination can be efficiently computed that is able to classify or to obtain a regression model. The hyper-parameters of a SVM are related to the kernel function; please refer to complete *scikit-learn* documentation for remaining parameters.

Building a Prediction Model

A good prediction model is one that has a high score when fed with data unseen during training. That is to say, the model has to be to able to generalise. Moreover the algorithms to train a

[3] `https://scikit-learn.org/`

prediction model depend on a set of hyper parameters. This has lead to the following procedure where the data set is divided into three subsets: the *training* and *test* sets are used when searching for a suitable set of hyper-parameters; the *validation* set is used to assess the performance of the selected set of hyper-parameters [19].

One method to obtain a model capable of generalisation and to evaluate the hyper-parameter space and choose one set of values is to perform k-fold cross validation. This means splitting the data set in two sets: the *train/test* set and the *validation* set. The *train/test* is divided in k folds. To evaluate one set of hyper-parameters, the model is trained in $k-1$ folds and then tested in the left-out fold. This procedure is repeated k times and the score of the set of hyper-parameters is the average of the score obtained in all the k iterations. Afterwards, the set of hyper-parameters that scored the highest is evaluated in the *validation* set in order to assess how good and how capable is the model to generalise to unseen data.

After a hyper-parameter has been selected to build a prediction model, the importance of the attributes in the result needs to be analysed. One way to achieve this is to perform an ablation test, in which a prediction model is built but without some of the attributes. The performance of the resulting model is then compared with the performance of the model which uses all attributes. If the ablated model performs similarly to the full model, then one may conclude that the discard attribute is irrelevant to the regression problem.

Results

Geographical Data Set

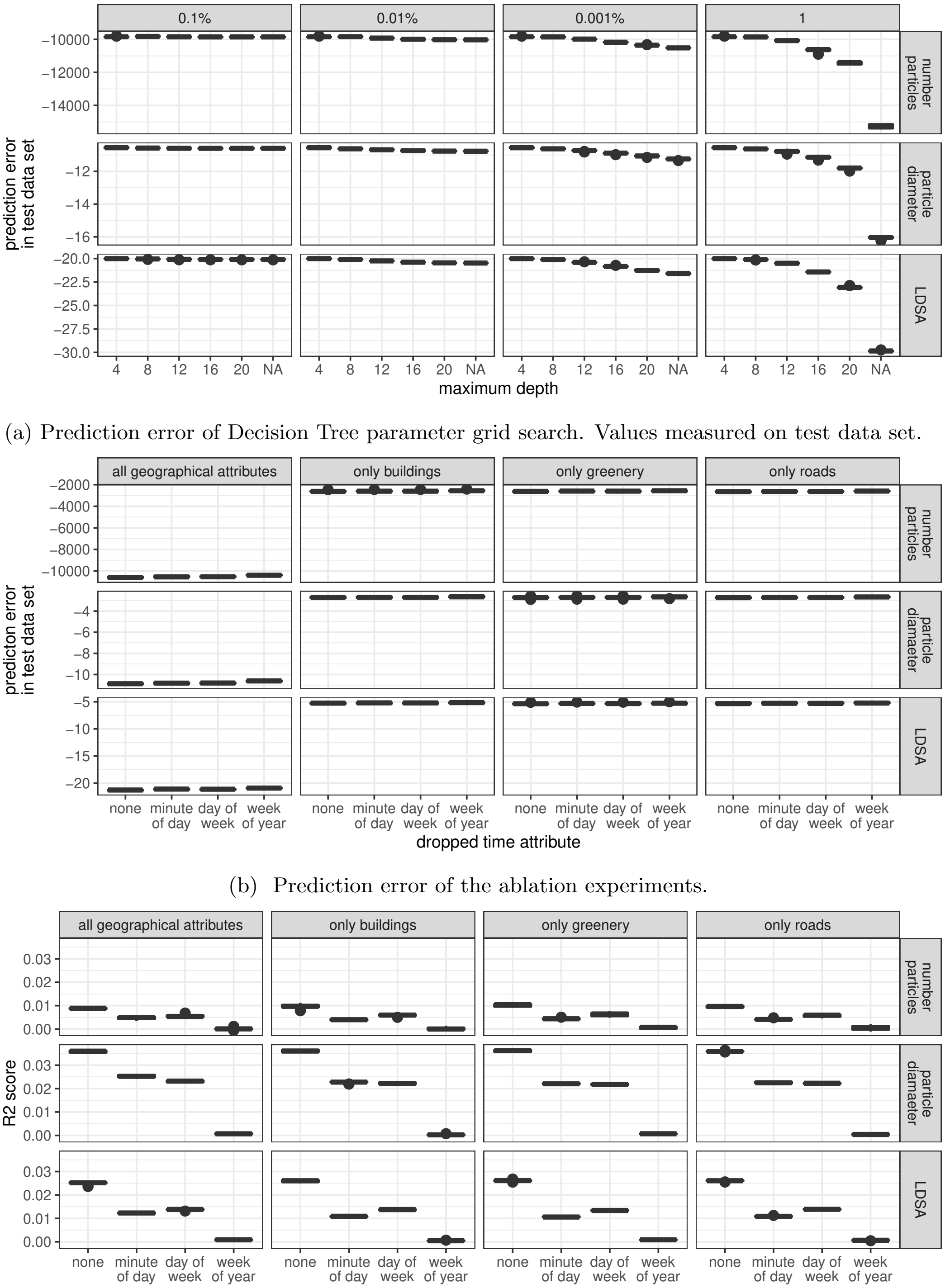
We begin by presenting the results of searching a set of hyper-parameters for DTs training through cross validation (see Figure 3a). The DT hyper-parameters maximum *tree length* (shown on the horizontal axis) and the *min-samples-leaf* (varied horizontally across the plots) were tested. The maximum tree length varied between 4 to 20. We also allowed the tree to grow as needed (marked as NA in the horizontal axis labels). As mentioned in the previous section, parameter *min-samples-leaf* controls the growth of the DT during the training algorithm. At each time step, the algorithm examines a tree leaf and must decide if it should expand it or leave it as is. This depends on the subset of the training samples that are assigned to the leaf. As for the values that we chose for this parameter, a percentage means that whenever the size of the subset of training samples relative to the size of the test set was higher than this number, the algorithm kept splitting the leaf nodes of the tree. When *min-samples-leaf* is equal to one, the algorithm keeps splitting until the subset size becomes one.

The prediction error in the test data set is based on how different are the predicted and true pollution levels (see the vertical axis of Figure 3a). The following equation shows how prediction error is computed:

$$-\frac{1}{n}\sum_{i}|f(\mathbf{x_i}) - y(\mathbf{x_i})|, \tag{2}$$

where $\mathbf{x_i}$ is the ith attribute vector, $y(\mathbf{x_i})$ is the pollution recorded by the sensor network (number of particles, for instance) that corresponds to ith attribute vector, $f_i(\mathbf{x_i})$ is the pollution value as predicted by the DT model, and n is the data set size. With this expression, the best value is zero, while negative values mean the DT is not predicting well.

Figure 3a shows that, as the trees get deeper, the performance on the test data set degrades. This effect is greater when parameter *min-samples-leaf* forces the algorithm to keep splitting until the subset of records is small. The left most plots in Figure 3a show a steep decline as the tree depth increases.



(a) Prediction error of Decision Tree parameter grid search. Values measured on test data set.

(b) Prediction error of the ablation experiments.

(c) R^2 score of the ablation experiments.

Figure 3: Results using the geographical data set.

We decided to perform the ablation experiments using DTs that were learned without imposing a limit to their depth. The prediction error obtained when using all the attributes or when some attributes were removed is shown in Figure 3b. It shows that, if only one of the geographical attributes is used, then the prediction error decreases considerably. Regarding the time attributes, the effect is not as visible as for the geographical attributes. However, Figure 3c shows that if any time attribute is removed, then the R^2 score approaches zero, meaning that the prediction is random. If only one geographical attribute is used, then the R^2 score increases a small amount compared to using all the geographical attributes. Regarding the different air pollution data, particle diameter has the highest R^2 score. Overall, the obtained score is far from one, meaning DTs with the geographical data set fare poorly when predicting air pollution.

Time Series Data Set

Figure 4a shows the R^2 score for the DT trained on all time series. Overall, the score increases as the time series length increases, and as the time series slot duration decreases. Regarding air pollution data, LDSA has the best score followed closely by particle diameter. As for number of particles the score is good only when we use a time series with 18 slots of 60 minutes. However, even in this case the R^2 score is lower than 0.5.

As for the results using SVM, as this method requires quadratic space on the input size, only the time series generator parameters that did not produced a high number of time series were tested. The tested parameters are marked with a star in Table 3. Figure 4b shows the R^2 score of these experiments, which, as expected, is higher when compared to the results when using DT. Again, air pollution data number of particles exhibited the worst performance. Although only two different time series length were tested, it is noticeable that the longer time series showed better performance, excepting in what regards the number of particles sensor data.

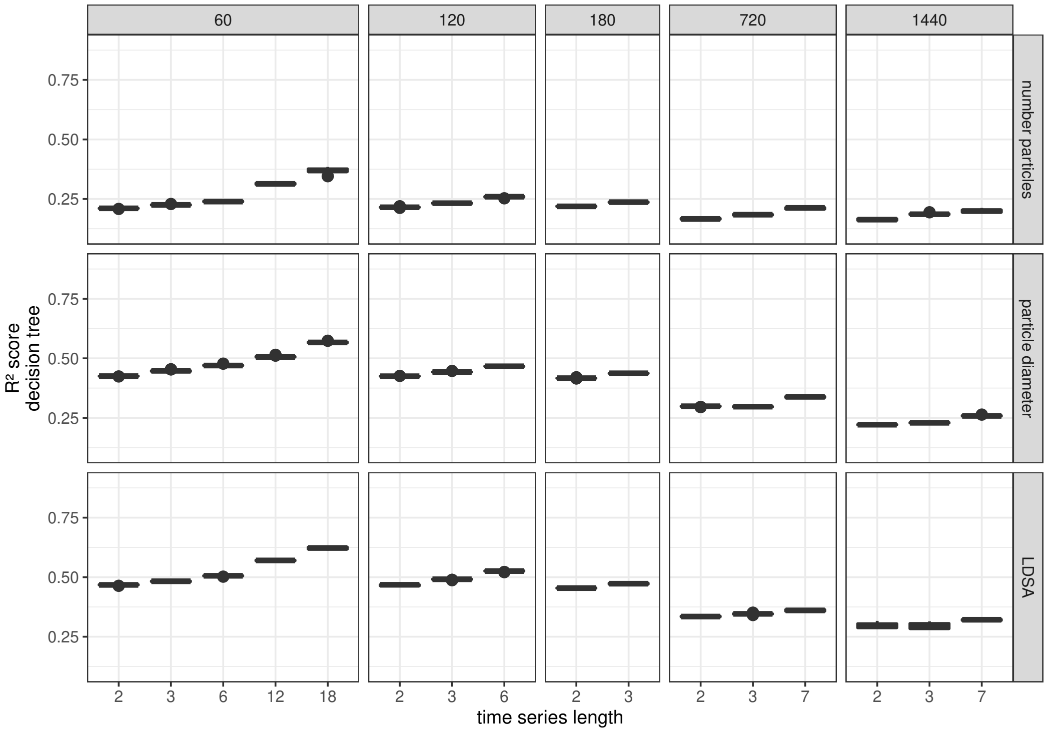
Discussion

Geographical Data Set

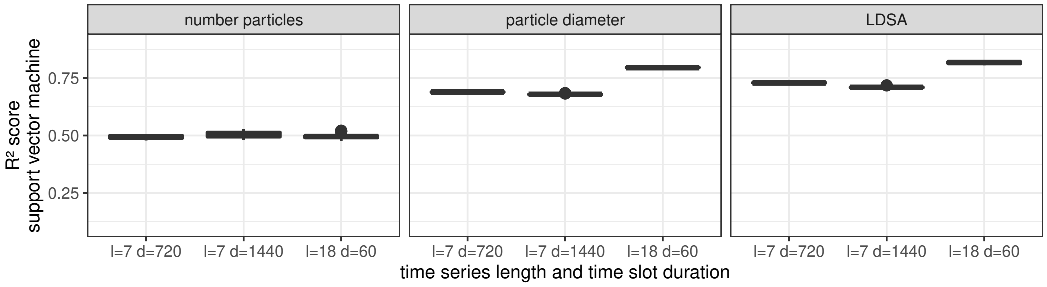
The prediction error depicted in Figure 3b shows that, when we remove either the attribute minute of the day, or day of the week, or week of the year, the prediction error does not changes much. However, if only one geographical attribute is used, the prediction error decreases, and the final value is the same independently of which geographical attribute is considered.

Overall, the obtained R^2 score is very low, with air pollution data particle diameter and LDSA showing the best values. Even then, it is less than 0.1, which means that the model is casting random guesses. The results also show that the effect of only using a geographical attribute does not affect the R^2 noticeable, compared to the effect of dropping a time attribute. These time attributes have a higher effect on the R^2 score compared to the geographical attributes, with week of year having the greatest impact when removed.

Overall, we believe that the low results obtained with the models based on DT are a direct consequence of the limited capacity and representation power of the models that can be obtained with this method. In previous work [18] we considered alternative methods, such as NNs, but these exhibit a major drawback, which is their time complexity on the number of samples. With data sets as big as the ones considered in the present work, exploring the space of hyper-parameters would be too time consuming. Another issue that may be hampering the results is the unbalanced nature of the used data set. As can be seen from the histograms (depicted in Figure 1b), the data are skewed towards low values.



(a) Prediction error of DT for all time series data set.



(b) R^2 score of Support Vector Machines for selected time series generator parameters.

Figure 4: Results using the time series data set.

One possibility to reduce the prediction error based on geographical attributes is to reduce the data set size by considering not only cell size but also a time slot duration. If we increase cell size, we reduce the number of geographical cells that have to be considered. Similarly, using time slot duration parameter, we reduce the size of the data set, which opens the possibility to use ML with higher time complexity.

Time Series Data Set

The low R^2 score obtained in the geographical data set score lead us to test other attributes when building air pollution models, in particular time series. The obtained results show that the R^2 score obtained with models trained with both geographical and time series information is higher than when no time series is considered. Figures 3a and 4a show the R^2 score when using DTs. It can be seen that even the worst DT has a R^2 score of around 0.15, which is considerably higher than the one obtained by the best DT on geographical data set, which achieved a score of around 0.04.

If a different ML method was considered, we would probably improve the R^2 score of the air pollution prediction model. To assess this possibility, in this work, we focused on SVM due to their small number of hyper parameters. The experiments that were performed with SVM showed promising results in terms of prediction accuracy for different time scales. We tested a time series with 7 days and another with 18 hours. Both resulted in high R^2 score for the particle diameter (0.68 and 0.8 for 7 days and 18 hours, respectively) and LDSA (0.72 and 0.83 for 7 days and 18 hours, respectively) sensor data. We also tested a time series using 7 half-days, which performed similarly to the 7 days time series.

Comparison

Regarding the two data sets that we present in this work, the time series data set was the one that induced the models with the best results (compare the R^2 score shown on Figures 3c and 4a). Regarding our previous work [18], we were able to improve the performance of the air pollution prediction model based on data collected by a network of mobile LCS. The geographical and time series data sets obtained better scores compared to the work presented in [18].

We considered other ML methods, such as NN and KNN, but these were discarded due to the unpractical training time they would require, given the large data set at hand. Overall, using DT to select a particular data set for further investigation was proven successful. With DT, we managed to quickly gain insight of which time series generator parameter or geographical attribute is more favourable, given practical constraints on the available computational budget required to handle a large data set such as one that a mobile sensor network can produce.

This work is an alternative to existing approaches for air pollution prediction, such as physically modelling the behaviour of air fluids [32], Krigging [30], taking into account the geographical information around the land where sensor data is collected and using it to build a regression model [27], and NN [3]. Our approach on using geographical information is similar to the one presented in [27] where the authors use distance to roads, to the one presented in [32] where the authors also focus on roads and specifically on urban canyons, and the one presented in [3] where the authors consider terrain topography in general. Regarding time, in [3] the authors found out that it had considerable success in the task of prediction the value of pollution data.

Air pollution prediction using time series is yet another approach. Different methods have been, used such as SVM [36] and NN [26, 16, 6]. These methods are able to better capture non-linearity in the problem when compared to more traditional statistical methods, which assume that variables are linearly dependent. Several authors also mention the time complexity of these methods [8].

There are several NN architectures such as Recurrent Neural Networks that we could have used. Deep NN have rise in popularity due to their success [31]. However they suffer from a high number of hyper-parameters.

When we need to build a pollution prediction model we are faced with the task of selecting a set of suitable attributes and of selecting a set of training model hyper-parameters. We can reduce the time required if we use a ML with low time complexity such as DT to select a set of attributes. This allows us to save time and energy in this phase and focus on the task of improving the performance of the pollution prediction model by tuning its hyper-parameters.

The results that we obtained were based on two types of data sets. The one based on geographical attributes compares with the work of [11] (land-use and traffic), [34] (only land-use), [26] (weather). As we have said, the performance obtained with this data set was not good. Moreover, each of the aforementioned work obtained better results. Regarding the time series data set, it compares favourably in terms of time span and time series length with the work: of [26] where they use two time instants but only for two weather attributes, and the best results where obtained when the data points were one hour apart compared to the 24 hour lag; of [6] they use a set of meteorology and pollution variables from the previous 3 months to forecast pollution values for the next 24 hours.

The best set of R^2 scores was obtained with the time series data set, with the highest being 0.83. In [6] the authors reported a maximum value of 0.91 obtained with a recurrent NN to predict SO_2. In [26] they report a low root mean square error with their NN. The OpenSense project has produced air pollution maps (with different time-scales) using land-use regression[11]. The maps with yearly and monthly resolutions had the best R^2 score, 0.38, when compared to measured values.

While not all the research that we reviewed uses the R^2 score, the results obtained with the time series data set is promising. The best score we obtained is higher than the results of [11], the research group that produced the data we used. The time series data set that we used to train the SVM does not cover all the geographical points as it is limited to locations where there is a tram line (see Figure 2). Even in locations where there is tram traffic, time series with hourly precision may be difficult to obtain when tram traffic is low (this problem is similar to the poor performance of hourly pollution maps reported in [11]). One way to circumvent this, is to allow missing data in the pollution prediction model. We can also include geographical and time (minute of day, day of week, and week of year) attributes in the time series data set.

Conclusions

In this paper we presented two pollution prediction models using two different data sets. One was based on geographical information and another on time series. The latter obtained the best results as the average R^2 score was 0.227, 0.390, and 0.438, for number of particles, particle diameter and LDSA, respectively. These values contrast with the average score obtained by the data set based on geographical information was 0.009, 0.036, and 0.025. When we used SVMs the R^2 score improved to 0.498, 0.721 and 0.752 for particle diameter and LDSA, respectively. The geographical information data set may have produced worst results due to fine spatial resolution, no aggregation in time, or differences in the geographical characterisation of sensor locations as sensor data was collected between 2012 and 2014, and geographical information used data that was introduced after 2014. In effect no single geographical attribute stand out when compared to time attribute.

As the time series based pollution prediction model provided better accuracy, it is more suited to be used as a tool that predicts regularly occurring pollution. When we compared our results with other models, while each uses different techniques to predict air pollution and they differ to the ground truth used (ranging from high fidelity and costly measurement stations to low cost sensors), the R^2 score obtained with our methodology is promising. We can improve on its accuracy by including attributes used in the aforementioned research. Thus, future work will consist on introducing more attributes in the time series data set.

Having a system that is capable of predicting air pollution is an important tool in order to improve population's health. This system can be used by people to avoid polluted areas in their daily routines. In the case of predicting periodic pollution events, this system can bring awareness to the dangers of air pollution. In this way, citizens may demand from their politicians solutions to the curb down pollution sources. Ultimately this will lead to changes on the energy sources that are used to run the modern world and to our relation with the environment.

Acknowledgements

This work was developed in the scope of the project ExpoLIS (LISBOA-01-0145-FEDER-032088) funded by FEDER and by national funds, through FCT – Foundation for Science and Technology, IP. C2TN authors also acknowledge the support of FCT for funding the strategic project UIDB/04349/2020.

Biographical Note

Pedro Mariano has a PhD in Informatics Engineering from the University of Lisbon obtained in 2006. Currently he is working as a researcher in the ExpoLIS project. His research interest include machine learning and artificial intelligence applied to games.

Susana Marta Almeida has a PhD in Environmental Sciences from the University of Aveiro. Currently she is Principal Researcher at C2TN from Instituto Superior Técnico where she develops work on air quality field.

Pedro Santana received a PhD in Computer Science from University of Lisbon in 2011. Currently he is an Assistant Professor at ISCTE - University Institute of Lisbon and a Researcher at ISTAR-Information Sciences and Technologies and Architecture Research Centre, Lisbon.

References

[1] Thomas Becnel et al. "A Distributed Low-Cost Pollution Monitoring Platform". In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 10738–10748. DOI: `10.1109/JIOT.2019.2941374`.

[2] European Commission. *Special Eurobarometer 468: Attitudes of European citizens towards the environment.* 2017. URL: `http://data.europa.eu/euodp/en/data/%20dataset/S2156_88_1_468_ENG`.

[3] Mahmoud Reza Delavar et al. "A Novel Method for Improving Air Pollution Prediction Based on Machine Learning Approaches: A Case Study Applied to the Capital City of Tehran". In: *ISPRS International Journal of Geo-Information* 8.2 (2019). ISSN: 2220-9964. DOI: `10.3390/ijgi8020099`.

[4] Pedro Domingos. *The Master algorithm: How the Quest for the Ultimate Machine Learning Algorithm Will Reshape the World.* Basic Books, 2015.

[5] EEA. *Air quality in Europe – 2020 report.* EEA Report No 09/2020. European Environment Agency, 2020. URL: `https://www.eea.europa.eu/publications/air-quality-in-europe-2020-report`.

[6] Rui Feng et al. “Recurrent Neural Network and random forest for analysis and accurate forecast of atmospheric pollutants: A case study in Hangzhou, China”. In: *Journal of Cleaner Production* 231 (2019), pp. 1005–1015. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2019.05.319.

[7] Martin Fierz et al. “Design, Calibration, and Field Performance of a Miniature Diffusion Size Classifier”. In: *Aerosol Science and Technology* 45.1 (2011), pp. 1–10. DOI: 10.1080/02786826.2010.516283.

[8] Zeinab Ghaemi, Abbas Alimohammadi and Mahdi Farnaghi. “LaSVM-based big data learning system for dynamic prediction of air pollution in Tehran”. In: *Environmental Monitoring and Assessment* 190.5 (May 2018). ISSN: 15732959. DOI: 10.1007/s10661-018-6659-6.

[9] Charles A.S. Hall and Kent Klitgaard. *Energy and the Wealth of Nations. An Introduction to Biophysical Economics*. Springer, 2012. ISBN: 978-1-4419-9397-7. DOI: 10.1007/978-1-4419-9398-4.

[10] Jiming Hao et al. “Air pollution and its control in China”. In: *Frontiers of Environmental Science & Engineering in China* 1.2 (May 2007), pp. 129–142. ISSN: 1673-7520. DOI: 10.1007/s11783-007-0024-2.

[11] David Hasenfratz et al. “Deriving high-resolution urban air pollution maps using mobile sensor nodes”. In: *Pervasive and Mobile Computing* 16 (2015). Selected Papers from the Twelfth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2014), pp. 268–285. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2014.11.008.

[12] IARC. *Outdoor air pollution a leading environmental cause of cancer deaths IARC*. press release (17-10-2013). 2013. URL: https://www.euro.who.int/en/health-topics/environment-and-health/urban-health/news/news/2013/10/outdoor-air-pollution-a-leading-environmental-cause-of-cancer-deaths.

[13] Andrew Kibble and Roy Harrison. “Point sources of air pollution”. In: *Occupational Medicine* 55.6 (Sept. 2005), pp. 425–431. ISSN: 0962-7480. DOI: 10.1093/occmed/kqi138.

[14] Yun-Chia Liang et al. “Machine Learning-Based Prediction of Air Quality”. In: *Applied Sciences* 10.24 (2020). ISSN: 2076-3417. DOI: 10.3390/app10249151.

[15] Yuan-Chien Lin, Wan-Ju Chi and Yong-Qing Lin. “The improvement of spatial-temporal resolution of PM2.5 estimation based on micro-air quality sensors by using data fusion technique”. In: *Environment International* 134 (2020), p. 105305. ISSN: 0160-4120. DOI: 10.1016/j.envint.2019.105305.

[16] Jun Ma et al. “Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques”. In: *Atmospheric Environment* 214 (2019), p. 116885. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2019.116885.

[17] Balz Maag et al. *Ultrafine Particle Dataset Collected by the OpenSense Zurich Mobile Sensor Network*. Zenodo, Sept. 2018. DOI: 10.5281/zenodo.1415369. URL: https://doi.org/10.5281/zenodo.1415369 (visited on 25/01/2021).

[18] Pedro Mariano, Susana Marta Almeida and Pedro Santana. “Pollution Prediction Model Using Data Collected by a Mobile Sensor Network”. In: *5th International Conference on Smart and Sustainable Technologies* (virtual conference, 23rd–26th Sept. 2020). Ed. by Joel J. P. C. Rodrigues and Sandro Nižetić. 2020. ISBN: 978-953-290-100-9.

[19] Tom M. Mitchell. *Machine Learning*. McGraw Hill Education, 1997.

[20] Luisa T. Molina and Bhola R. Gurjar. “Regional and Global Environmental Issues of Air Pollution”. In: *Air Pollution. Health and Environmental Impacts.* Ed. by Bhola R. Gurjar, Luisa T. Molina and Chandra S.P. Ojha. CRC Press, 2010. Chap. 17, pp. 493–518. ISBN: 978-1-4398-0963-1. DOI: 10.1201/EBK1439809624.

[21] Sandro Nižetić et al. “Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future”. In: *Journal of Cleaner Production* 274 (2020), p. 122877. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2020.122877.

[22] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[23] Laura Ranzato et al. “A comparison of methods for the assessment of odor impacts on air quality: Field inspection (VDI 3940) and the air dispersion model CALPUFF”. In: *Atmospheric Environment* 61 (2012), pp. 570–579. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2012.08.009.

[24] Joel J. P. C. Rodrigues and Sandro Nižetić, eds. *5th International Conference on Smart and Sustainable Technologies* (virtual conference, 23rd–26th Sept. 2020). 2020. ISBN: 978-953-290-100-9.

[25] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Fourth edition. Prentice Hall, 2020.

[26] Ana Russo, Frank Raischel and Pedro G. Lind. “Air quality prediction using optimal neural networks with stochastic variables”. In: *Atmospheric Environment* 79 (2013), pp. 822–830. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2013.07.072.

[27] Patrick H. Ryan et al. “A Comparison of Proximity and Land Use Regression Traffic Exposure Models and Wheezing in Infants”. In: *Environ Health Perspect.* 115.2 (Feb. 2007), pp. 278–284. DOI: 10.1289/ehp.9480.

[28] Pedro Santana et al. “An Affordable Vehicle-Mounted Sensing Solution for Mobile Air Quality Monitoring”. In: *5th International Conference on Smart and Sustainable Technologies* (virtual conference, 23rd–26th Sept. 2020). Ed. by Joel J. P. C. Rodrigues and Sandro Nižetić. 2020. ISBN: 978-953-290-100-9.

[29] Nicholas I. Sapankevych and Ravi Sankar. “Time Series Prediction Using Support Vector Machines: A Survey”. In: *IEEE Computational Intelligence Magazine* 4.2 (2009), pp. 24–38. DOI: 10.1109/MCI.2009.932254.

[30] Philipp Schneider et al. “Mapping urban air quality in near real-time using observations from low-cost sensors and model information”. In: *Environment International* 106 (2017), pp. 234–247. ISSN: 0160-4120. DOI: 10.1016/j.envint.2017.05.005.

[31] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489.

[32] Sotiris Vardoulakis et al. “Modelling air quality in street canyons: a review”. In: *Atmospheric Environment* 37.2 (2003), pp. 155–182. ISSN: 1352-2310. DOI: 10.1016/S1352-2310(02)00857-9.

[33] Laura E. Venegas, Nicolás A. Mazzeo and Mariana C. Dezzutti. “A simple model for calculating air pollution within street canyons”. In: *Atmospheric Environment* 87 (2014), pp. 77–86. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2014.01.005.

[34] Lena Weissert et al. "Low-cost sensor networks and land-use regression: Interpolating nitrogen dioxide concentration at high temporal and spatial resolution in Southern California". In: *Atmospheric Environment* 223 (2020), p. 117287. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2020.117287.

[35] J. Jason West et al. "What We Breathe Impacts Our Health: Improving Understanding of the Link between Air Pollution and Health". In: *Environmental Science & Technology* 50.10 (2016). PMID: 27010639, pp. 4895–4904. DOI: 10.1021/acs.est.5b03827.

[36] Wentao Yang et al. "Prediction of hourly PM2.5 using a space-time support vector regression model". In: *Atmospheric Environment* 181 (2018), pp. 12–19. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2018.03.015.

[37] Junfeng (Jim) Zhang and Jonathan M. Samet. "Chinese haze versus Western smog: lessons learned". In: *Journal of Thoracic Disease* 7.1 (2015). ISSN: 2077-6624. DOI: 10.3978/j.issn.2072-1439.2014.12.06.