Black carbon proxy sensor model for air quality IoT monitoring networks

Juan Paredes-Ahumada, Pau Ferrer-Cid, Jose M. Barcelo-Ordinas, Jorge Garcia Vidal
Universitat Politecnica de Catalunya, Spain
{juan.antonio.paredes, pau.ferrer.cid, jose.maria.barcelo, jorge.garcia}@upc.edu

Abstract

Most Internet of Things (IoT) air quality monitoring networks measure and report regulated pollutants such as O_3, NO, NO_2, SO_2 or CO and $PM_{2.5}$, PM_{10} particulates. However, there are pollutants such as black carbon that are not regulated by the authorities and are rarely measured, and if they are measured, the instrumentation is very expensive. One way to obtain measurements with cheaper equipment is to use the proxy concept, where from indirect measurements of other pollutants a virtual sensor is constructed using machine learning techniques. In this work, we design a machine learning-based proxy for black carbon based on low-cost sensor (LCS) nodes. We compare three techniques to build the proxy: support vector regression, random forest and a neural network. The LCSs have to be pre-calibrated also using machine learning techniques, linear or nonlinear. The results show, using data from a real deployment of IoT air quality sensor nodes, that the results obtained by the proxy with LCSs (R^2=0.72) using support vector regression are a good approximation in terms of performance to those obtained by a proxy using high-cost reference instrumentation (R^2=0.76).

Categories and Subject Descriptors

I.2.1 [**Computing Methodologies**]: Artificial Intelligence—*Applications and Expert Systems*

General Terms

Air quality monitoring sensor networks, IoT.

Keywords

Low-cost sensors, machine learning, proxy.

1 Introduction

Governmental air quality monitoring networks measure regulated pollutants such as O_3, NO, NO_2, SO_2 or CO and particles $PM_{2.5}$, PM_{10} with reference stations that have a high cost, using very accurate and expensive sensors. Reference stations measuring the mentioned pollutants cost around 100 KEuros. For this reason, in recent years there has been a great interest in the development and deployment of IoT air quality monitoring networks using low-cost sensors (LCSs) [14]. Such sensors are cheaper, in the tens of Euros, so that a measurement node, adding electronics, communications and an array of sensors, can be developed at a cost of around 1 KEuro. The challenge for such nodes is data quality, so there is an effort in the research community to improve the quality of the data reported by such IoT nodes using machine learning techniques. Another challenge in this type of monitoring networks is that there are pollutants that have a great impact on health, such as black carbon (BC), which is not regulated. This means that only very expensive BC sensors are available, with costs per sensor in the tens of KEuros, which results in very few BC measurements by the authorities. A novel way to measure this type of pollutant, since there are no low-cost sensors available, is by indirect measurements in what is called a *virtual proxy sensor* or simply a *proxy*. A virtual sensor is a mathematical artifact that estimates the values that a physical sensor would produce when no physical sensor is available. Examples of use are when a sensor malfunction, or when performing node maintenance or relocation of a sensor node. For example, Ferrer-Cid *et al.* [8] realized a virtual O_3 sensor using graph signal processing techniques involving nodes neighboring the target sensor. A proxy is a specific type of sensor in which the virtual sensor is realized from indirect measurements, in this case, sensors that do not measure BC but measure other pollutants such as O_3, NO_2, NO, PM_x, or submicron particle number concentration. In addition, environmental measurements such as temperature and relative humidity usually have an impact on the results, so the estimation has to take these environmental parameters into account. In this paper, we analyze a BC proxy sensor using IoT nodes including LCSs for air quality monitoring. More specifically:

- propose a machine learning-based BC proxy model using indirect IoT LCSs measures;
- calibrate the LCSs using multiple linear regression (MLR) and support vector regression (SVR);
- compare three possible BC proxy models using: i) support vector regression (SVR), ii) random forest (RF), and iii) an artificial neural network (ANN);

- evaluate the performance of the BC proxy with two real data sets obtained from real equipment. The first data set comes from indirect measurements taken by reference instrumentation that measure exact regulated pollutant measurements, and thus, acts as baseline proxy, and the second data set comes from real IoT wireless nodes that include LCSs. We compare the results of both proxies against measures taken by a high-cost BC sensor instrumentation.

The paper is organized as follows: section 2 gives the related work, and section 3 presents how to build a machine learning-based BC proxy. Sections 4 and 5 describe the data sets used and the performance of the BC proxy. Finally, section 6 concludes the paper.

2 Related work

Virtual sensors [12] are intelligent sensors that produce estimators of a physical phenomenon, and are used in place of real sensors to temporarily replace physical sensors. For example, Woo *et al.* [17] develop a virtual sensor to provide a micro-scale personal air pollution information services, using a CFD-based air quality modeling system. Zaidan *et al.* [18] propose the use of virtual sensors to calibrate CO_2 sensors and to estimate BC concentrations. On the other hand, Ferrer-Cid *et al.* [8] proposes to use a graph signal processing framework whose graph is constructed from the data and in which virtual sensors for air quality sensor networks can be developed, showing for example in [9] how to use signal reconstruction methods such as Laplacian interpolation or kernel-based graphical signal reconstruction models. Proxies have already been used as indirect sensors in other disciplines. For example, Coulby *et al.* [7] use a CO_2 IoT node to produce a ventilation proxy in indoor environments. There are few studies on how to develop a BC proxy. The main ones are the work of Zaidan *et al.* [19] where they propose a BC proxy based on white-box and black-box, using a Bayesian neural network for the black-box. This work shows for the first time that good results can be achieved for a proxy despite the complexity of the model used. Fung *et al.* [11] propose an input-adaptive BC proxy, using least squares linear regression but with the drawback of having to train several models with different features in case any of the sensors fail. More recently, the use of simpler machine learning techniques has been tested using data from reference stations to obtain the values of a BC proxy to test the exposure of people in the city of Barcelona, Spain [16]. In this paper, we investigate a BC proxy that uses LCS instead of only reference stations, which are much more expensive and compare three proxy candidates (SVR, RF and ANN) as possible machine learning models.

3 Black carbon (BC) proxy model

3.1 Calibration of low-cost sensors

The reference station sensors are perfectly calibrated and are recalibrated monthly. However, LCSs typically come uncalibrated and have to go through a calibration step. As an example, the alphasense OX-B431 (O_3) electrochemical sensors used in this paper measure O_3 and NO_2 simultaneously and are very sensitive to temperature and relative humidity, so to calibrate this sensor you have to train a supervised machine learning regression method, either linear or nonlinear, whose input is the sensor measurements of O_3 and NO_2, temperature and relative humidity. The regression uses as true values those obtained by the reference station in a calibration process that is called *in-situ calibration* [5, 13]. Once the model is trained, hyperparameters are obtained that allow estimating pollutant concentration values. The rest of the LCSs follow a similar methodology.

Thus, to calibrate these LCSs, the calibration model considers that an array of P sensors is involved. We define pairs $\{x_i, y_i\}_{i=1}^{N}$ where N is the number of measurements, $x_i \in \mathbb{R}^P$ is the i-th sensor measurement, y_i is the i-th reference value, and the goal of the supervised machine learning method is to find a model that approximates the reference value with a function that depends on the model parameters, i.e., learn the function f_{cal}: $\mathbb{R}^{\mathrm{P}} \rightarrow \mathbb{R}$, with P the number of sensors participating in the array:

$$y_i = f_{cal}(x_i) + \varepsilon_i;\ \forall i = 1,..,N, \tag{1}$$

where ε_i is the error assumed to be independent and identically distributed with zero mean and variance σ^2. There are several supervised machine learning methods to learn the function $f_{cal}(\cdot)$, among which we can find multiple linear regression, k-nearest neighbors, random forest, or support vector regression [5, 6, 10, 13]. We will choose a linear model (multiple linear regression, MLR) and a nonlinear model (support vector regression, SVR), which have been proven as good models [10], and compare which is the better calibration model for our LCSs.

For the case of the MLR, a linear function $f_{cal}(x_i) = \beta_0 + \beta^T x_i$ is used, where $\beta_0 \in \mathbb{R}$ (offset) and $\beta \in \mathbb{R}^{\mathrm{P}}$ (bias) are the coefficients to be learnt during the training phase. On the other hand, the SVR makes use of the "kernel trick" where the data is implicitly mapped to a higher dimension in order to find a better regression curve but doing all computations in input space via a kernel function $k(x, x')$, and the curve is fitted using $f_{cal}(x_i) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) k(x, x_i) + b$. The values for the parameters $\hat{\alpha}_i^*$, $\hat{\alpha}_i$ and b are found by solving a quadratic convex optimization problem. The objective function to solve is obtained with the dual formulation of the problem, minimizing an ε loss function and using a penalization term C. We have chosen to work with the radial basis function (RBF) kernel, as it has proven to work well in these air quality LCSs [10].

3.2 BC proxy model

In order to build the proxy, we use supervised nonlinear regression machine learning algorithms due to the nonlinear nature of the data [19]. We denote by $\mathbf{y}_{BC} \in \mathbb{R}^{\mathrm{N}}$ the BC values provided by the reference instrumentation, where N is the number of samples. Then, we can group the set of LCS calibrated measurements into a sensor matrix $\mathbf{X}_{\mathrm{S_{cal}}} \in \mathbb{R}^{\mathrm{N} \times \mathrm{P_S}}$, where $\mathrm{P_S} = |\mathcal{S}|$ is the dimension of the calibrated sensor set $\mathcal{S}$. It is possible that there is overfitting in the model due to the high number of sensors that can participate in the creation of the proxy. For this reason, the set of predictors was iteratively reduced using a feature elimination mechanism. The backward feature elimination (BFE) algorithm consists of the machine learning model starting with the entire set of

predictors, and at each iteration, the predictor that has the least impact on the model is removed from the model. The process is repeated until no other predictor can be removed without a statistically significant loss of fit. We can select the best subset of $\mathcal{S}_{\text{FS}}$ sensors to use as predictors, where $\mathcal{S}$ is the set of available sensors:

$$\mathcal{S} \underset{BFE}{\Rightarrow} \mathcal{S}_{\text{FS}} \subset \mathcal{S} \tag{2}$$

Now, the data matrix involved in the design of the proxy model is given by $\mathbf{X}_{\text{S}_{\text{FS}}} \in \mathbb{R}^{\text{N} \times \text{P}_{\text{FS}}}$, where $\text{P}_{\text{FS}} = |\mathcal{S}_{\text{FS}}|$ is the dimension of the sensor array selected by the BFE algorithm. The data-driven proxy model, then, can be defined as:

$$\mathbf{y}_{BC_i} \approx f_{proxy}(\mathbf{x}_{FS_i}),\ i = 1, \ldots, N \tag{3}$$

where f_{proxy}: $\mathbb{R}^{\text{P}_{\text{FS}}} \rightarrow \mathbb{R}$ is the function that estimates the BC concentrations. For modeling the proxy function $f_{proxy}(\cdot)$, we propose to compare three non-linear models: support vector regression (SVR), random forest (RF), and an artificial neural network (ANN).

Algorithm 1 Proxy sensor model for black carbon estimation.

Input: $\{\mathbf{S}, \mathbf{X_S}, \mathbf{Y_{ref}}, \mathbf{f_{cal_s}}(\cdot), \mathbf{y_{BC}}, \mathbf{f_{proxy}}(\cdot)\}$

▷ Obtain LCS calibrated data for the proxy
1: **for** s$\in\mathcal{S}$ **do**
2: **if** s is calibrated **then**
3: $\mathbf{x_{s_{cal}}} \leftarrow$ Get_Sensor($\mathbf{X_S}$)
4: **else**
5: $\mathbf{y_s} \leftarrow$ Get_Ref($\mathbf{Y_{ref}}$)
6: $\mathbf{Z_s} \leftarrow$ Select_Features($\mathbf{X_S}$)
7: $\mathbf{x_{s_{cal}}}, \Theta_s \leftarrow$ Calibrate_LCS($Z_s, y_s, f_{cal_s}(\cdot)$)
8: **end if**
9: $\mathbf{X_{S_{cal}}} \leftarrow$ Add_To_Proxy_Training_Matrix($\mathbf{x_{s_{cal}}}$)
10: **end for**

▷ Train proxy model
11: $\mathcal{S}_{FS}, \Theta_{proxy} \leftarrow$ BFE($\mathbf{X_{S_{cal}}}, \mathbf{y_{BC}}, \mathbf{f_{proxy}}(\cdot)$)

▷ BC proxy estimation for new measurements
12: **while** $\mathbf{x}_{new}$ **do**
13: **for** s$\in\mathcal{S}_{\mathcal{FS}}$ **do**
14: $\mathbf{x}_{new} \leftarrow \text{f}_{cal_s}(\mathbf{x}_{new_s}, \Theta_s)$
15: **end for**
16: $\tilde{\mathbf{x}}_{BC} \leftarrow \text{f}_{proxy}(\mathbf{x}_{new}, \Theta_{proxy})$
17: **end while**

As explained before, SVR is a kernel method that maps the data in a higher feature dimensional space and makes use of the kernel trick to find a better regression curve but doing all the calculations in the input space through a kernel function. The RF method differs from SVR in that it combines several decision trees by sampling the data set via bootstrapping. Finally, the ANN algorithm consists of layers of interconnected units, in which a node of a given layer receives as input a linear combination of the values of the nodes in the previous one, which is then mapped via a non-linear activation function. We use a fully connected feed-forward neural network with as many nodes per layer as the number of predictors and two hidden layers at most. Both hyperbolic tangent and rectified linear unit were tested as activation functions. To avoid overfitting we set an early-stopping if there is no improvement in the MSE after 10 epochs.

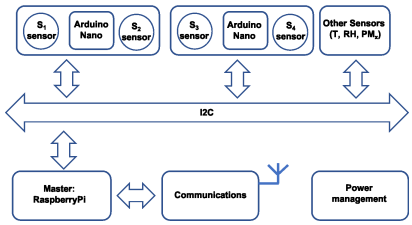
The BFE mechanism is linked to the supervised machine learning mechanism used. For each supervised mechanism the BFE result can be different and therefore a different set $\mathcal{S}_{\text{FS}}$ can be chosen. When the set $\mathcal{S}_{\text{FS}}$ is fixed, the trained model will set the hyperparameters to be used in the estimation process.

Algorithm 1 describes the steps followed in the design of the BC proxy: the input of the algorithm is the set of sensors S, the sensor raw data X_S, the reference sensor data Y_{ref}, the sensors calibration function (per each sensor of set S) $f_{cal_s}(\cdot)$, the reference BC data y_{BC}, and finally the BC proxy function $f_{proxy}(\cdot)$. Lines 1-10 describe the calibration of the available LCS and obtains the hyperparameters Θ_s for each calibration model and the calibrated data that will participate in the proxy model; line 11 trains the BC proxy model with the calibrated sensor values using a BFE algorithm. The output are the set of selected sensors $\mathcal{S}_{FS}$ participating in the proxy model and the hyperparameters Θ_{proxy} of the proxy model. Finally, lines 12-17 estimate new BC concentrations. First, each sensor concentration is estimated using the sensor calibration hyperparameters, and then with these values the BC concentration is estimated with the proxy function.

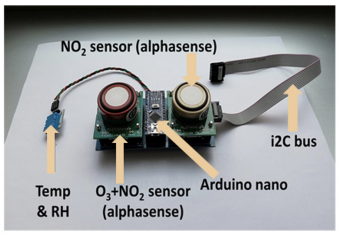
4 Data set

We consider two types of measurements, those obtained by reference instrumentation and those obtained with nodes deploying LCS. The reference values, Table 1, were measured at the reference station located in located in Palau Reial (41°23′14″N, 2°6′56″E, 80 m.a.s.l.), Barcelona, Spain. Reference BC mass concentrations were monitored using a multiangle absorption photometer (MAAP, Thermo ESM Andersen Instrument) fitted with a PM_{10} inlet, operating on a 1 min time resolution. The total particle number concentrations (N) was measured with a water-based condensation particle counter (WCPC TSI 3785) with 5 min time resolution. The temporal resolution of these reference measurements are given by the commercial equipment. The reference station is equipped with high-cost instrumentation for measuring the particulate matter concentrations (PM_1, $\text{PM}_{2.5}$, PM_{10}), tropospheric ozone (O_3), dioxide of nitrogen (NO_2) and monoxide of oxygen (NO). Meteorological variables (temperature and relative humidity), which are used only as a corrector in the proxy using reference stations, were obtained from a meteorological station located on the roof of the Faculty of Physics of the Univ. of Barcelona, about 400 m from the Palau Reial station. All data was aggregated at 10 min resolution.

Two IoT nodes with LCS were deployed: the first is a commercial node called PurpleAir PA-II node [15], and that measures PM_1, $\text{PM}_{2.5}$ and PM_{10}. This node is equipped with a PMS5003 dual laser particle counter. Built-in WiFi enables the air quality measurement device to transmit data to the real-time PurpleAir Map, which is stored and made available to any smart device. The data can be downloaded with a 10 min resolution. The second node, Captor node, is an experimental prototype developed at Universitat Politecnica de Catalunya (UPC) for real IoT deployments. This node includes three electrochemical Alphasense sensors; one OX-B431 O3 sensor [3], one NO2-B43F NO2 sensor [2] and one



(a) Captor node scheme



(b) Arduino model

Figure 1. Captor node scheme (a) and a picture of the Arduino model with an NO_2 sensor, a O_3 sensor and a temperature and relative humidity sensor.

NO-B4 NO sensor [1], and one DHT1 Grove air temperature (T) and air relative humidity (RH) sensor to measure the internal box environmental temperature and relative humidity. The Captor node, Figure 1, uses a Raspberry Pi as central processing unit connected via an I2C bus to the sensing subsystems. A pair of Alphasense electrochemical sensors with their individual sensor boards (ISBs, [4]) are connected to an Arduino Nano microcontroller unit (MCU) that sends the collected data to the Raspberry pi central processor. The temperature and relative humidity sensor are connected to the I2C bus. The Raspberry pi polls the sensor subsystems on a round-robin basis and finally sends the data to a server in the cloud via a wireless communication radio unit. The sampling rate can be reconfigured, and for our experiments it was set to 1 s for the powered node version, while it was set to 5 min with a duty-cycle mechanism for the battery-powered node version. The data are aggregated at a resolution of 10 min to match those of the reference station.

5 Results

We divide the results section into LCS calibration and BC proxy construction. The methodology to train the sensor calibration methods and to train the proxy are the following: i) a randomly selected fraction of the data set (75%) was used for training the model, and the remaining fraction (25%) for validating the model, ii) a 10-fold cross-validation strategy was used to obtain the model's hyperparameters by averaging the root-mean square error (RMSE), iii) then, to ensure that the calibration or proxy model is trained on a diverse range of input variable concentrations, the data sets were randomized. This approach prevents the model from learning patterns that are specific to a particular time or season, and instead challenges it with a broad range of concentrations, reducing the effect of variations due to different seasons or times of day.

Table 1. Data sets used in the BC proxy model.

VAR.	# SAMPLES	PERIOD	RESOL.	MEASUREM. SOURCE
BC	186367	2021/08/31 - 2022/01/31	1 min	Ref. Stat.
O_3	25924	2021/08/31 - 2022/01/25	10 min	Ref. Stat.
NO_2	25924	2021/08/31 - 2022/01/25	10 min	Ref. Stat.
NO	25924	2021/08/31 - 2022/01/25	10 min	Ref. Stat.
N	73436	2021/08/01 - 2022/05/19	5 min	Ref. Stat.
PM_{10}	24837	2021/08/01 - 2022/01/31	10 min	Ref. Stat.
T	17712	2021/08/31 - 2021/12/31	10 min	Met. Stat.
RH	17712	2021/08/31 - 2021/12/31	10 min	Met. Stat.
O_3	7373793	2021/08/31 - 2022/01/25	1 s	LCS
NO_2	7373793	2021/08/31 - 2022/01/25	1 s	LCS
T	7373793	2021/08/31 - 2022/01/25	1 s	LCS
RH	7373793	2021/08/31 - 2022/01/25	1 s	LCS
NO	7410741	2021/08/31 - 2022/01/25	1 s	LCS
PM_1	136593	2021/10/19 - 2022/01/31	2 min	LCS
$PM_{2.5}$	139515	2021/10/19 - 2022/01/31	2 min	LCS
PM_{10}	136593	2021/10/19 - 2022/01/31	2 min	LCS
N	130515	2021/10/19 - 2022/01/31	2 min	LCS

5.1 Calibration of LCS

The accuracy of the proxy model depends on the quality of the captured data. Thus, before building the proxy it is necessary to calibrate the LCS, and test how accurate they are with respect to the values obtained by the reference values. We calibrate the sensors using a linear (MLR) and a nonlinear (SVR) method. Linear methods work fine for calibrating O_3, NO_2, and NO with R^2 ranging from 0.81 to 0.86. The nonlinear method improves the R^2 by 4-6% (R^2 between 0.86-0.9). In the case of PM_{10}, the linear method produces an R^2 of 0.70. The nonlinear method improves the R^2 by approximately 12% (R^2=0.79). Figures 2.a), b), c) and d) compare the values predicted with SVR with the values of the reference station for O_3, NO_2, NO and PM_{10} respectively, showing the good performance of the predictions in terms of high values of R^2 (≥ 0.79) and low values of RMSE. In the case of O_3 and NO_2, which are less local phenomena than NO and PM, a very good prediction is observed over the whole range of values. On the other hand, both NO and PM_{10} are more local physical phenomena that have peaks, which makes these peaks more difficult to estimate, impacting the R^2 and RMSE. Since the SVR calibration results are the best, we will use these estimated values as input values for the BC proxy.

5.2 BC proxy using reference instrumentation and LCS

To evaluate the performance of a BC proxy we have built one proxy from data obtained from high-quality sensor instrumentation included in the reference station. We begin with a proxy built with all the variables available in the reference station (O_3, NO_2, NO, PM_{10}, N), where here N is the total particle number concentrations, and the meteorological station (T and RH) as correcting factors. We use the BFE algorithm on the dataset composed of the reference instrumentation sensor (reference stations), with the objective of creating a baseline BC proxy, i.e. a proxy with high-quality data. Table 2 compares the best subset of predictors selected by SVR, RF, and ANN in terms of R^2 (Ref. Station column) using the BC reference instrumentation. We note that the BFE algorithm chooses different features when using dif-

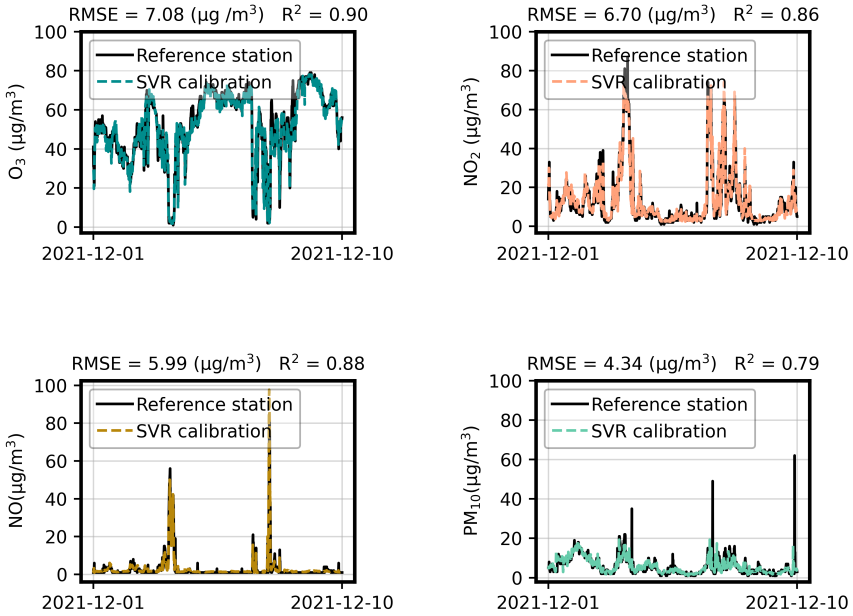


Figure 2. Ten-day time series for O_3, NO_2, NO, and PM_{10} after being calibrated by a SVR model. The reference station values are included for comparison. RMSE and R^2 are included for every calibrated pollutant.

ferent machine learning models. For example SVR chooses as optimal features $\{O_3, PM_{10}, N, RH, T\}$ and RF does not choose the RH, $\{O_3, PM_{10}, N, T\}$. On the other hand, ANN adds NO and NO_2 as features to its optimal set $\{O_3, NO_2, NO, PM_{10}, N, T\}$. Thus, a first conclusion is that proxy performance is sensitive to the features used depending on the model selected.

Now, we construct a BC proxy using LCSs. The RMSE and R^2 are calculated, again, using the reference BC instrument (column LCSs). In this case, we do not perform the BFE, as we want to compare the BC proxy using LCSs against the baseline BC proxy using reference data. We observe that the BC proxy using LCS performs close to the BC proxy using features measured by the reference station, which is the best proxy we can obtain given a given machine learning model. Among the models used, we see that SVR offers the best performance with an RMSE=0.37 $\mu g/m^3$ and R^2=0.76 if we use the data from the reference stations versus an RMSE=0.41 $\mu g/m^3$ and R^2=0.71 if we use LCS. These results are in agreement with the results obtained in the same area during 2 years (2018 and 2019), with reference stations, and where seasonality was also studied [16]. Finally, we also observe whether some overfitting is present in the BC proxy calculations with LCS because the optimal model using BFE is calculated on the reference station data which does perform a cross-validation process to avoid overfitting. The point of using the reference data is that it acts as a baseline case since we know that they are accurate data, whereas if we perform a BFE on the LCS data, the selection of the BFE will be very dependent on the quality of each sensor at every moment and the set of sensors in the data set. We have run a BFE with SVR on the LCS sensors to see how different the choice of features is, doing cross-validation as is done with the BFE on reference data. The results of this experiment showed that the optimal set of predictors is comprised of $\{O_3, NO_2, N_{0.5}, N_1, T$, and RH$\}$, where in the set appears the count suspended particle sizes $N_{0.5}$, N_1 instead of total count N and PM_{10}, obtaining a slightly better performance than when using the set obtained from the reference station RMSE=0.39 $\mu g/m^3$, R^2=0.77. However, it is worth mentioning that the LCS uses counts of suspended particle sizes to derive PM_x concentrations, so particle counts and particle concentrations are related in the values reported by the LCS and may participate interchangeably. Nevertheless in the reference station data set we had only the aggregated value of particle number N, and not the individual channels. One would expect the BC proxy with reference data to outperform the LCS BFE selection if we had access to $PM_{2,5}$ or PM_1 data. However, the reference station did not provide such data.

Table 2. BC proxy comparison after backwards feature selection.

	Predictors subset	Ref. Station		LCSs	
		RMSE ($\mu g/m^3$)	R^2	RMSE ($\mu g/m^3$)	R^2
SVR	O_3, PM_{10}, N, T, RH	0.37	0.76	0.41	0.71
RF	O_3, PM_{10}, N, T	0.41	0.71	0.47	0.68
ANN	O_3, NO_2, NO, PM_{10}, N, T	0.41	0.71	0.41	0.71

Figure 3 provides the time series of both proxies and compares them with the values obtained by the BC reference instrument. Even though both models follow BC instrument trend, the BC proxy model trained with the reference sta-

tion measurements is better at predicting high values than the LCS proxy model. The greatest difficulty is in predicting local BC peaks that occur occasionally. These peaks are a challenge and one of the lines of research to follow, in order to improve their prediction.

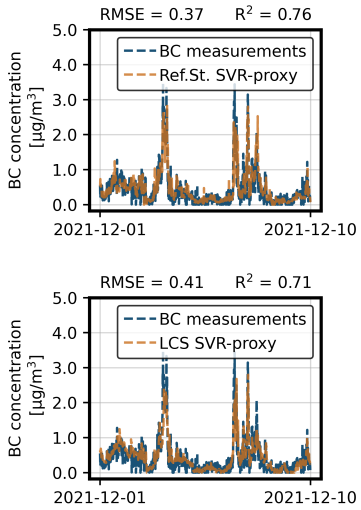


Figure 3. Time series for BC proxy using SVR as model.

6 Conclusions

In this paper, we have described the design of a proxy for BC using LCSs. Proxies are virtual sensors that substitute for real sensors, when these are not available, by means of indirect measurements. To test the performance of the proxy, we have deployed LCS IoT nodes that measure particle number concentrations (N), PM_1, $PM_{2.5}$, PM_{10}, O_3, NO_2, NO, temperature and relative humidity. We have first calibrated with supervised linear and nonlinear machine learning models the LCS showing the quality of the estimates in terms of RMSE and R^2. We found that the nonlinear SVR model gave the best estimation results. We then trained the BC proxy model using three techniques, RF, SVR, and ANN. First, we performed a BFE to reduce the number of sensors participating in the proxy. We have first compared a BC proxy created only with indirect measurements captured by a reference station. This experiment shows us the best we can do with as accurate as possible measurements taken by instrumentation used to report official data from government agencies. The proxy is evaluated in terms of RMSE and R^2 with data from a high-cost BC sensor that gives accurate values, giving, the best model (SVR), a R^2 of 0.76. The second experiment is to create the BC proxy using data taken by IoT nodes deploying LCS. In this case, we observe that the BC proxy using LCS approximates (R^2=0.71) the BC proxy using data measured by the reference station, showing that the approximation is quite good. We believe, as unresolved lines of work, that it is necessary to investigate how to obtain a more robust proxy taking into account the loss of data in the different features participating in the proxy, using for example missing value imputation methods.

Acknowledgments

This work is supported by projects PID 2019-107910RB-I00, 2021 SGR-01059, CDTI MIG-20221061.

7 References

[1] Alphansense. No-b4 sensor datasheet. [Online] https://www.alphasense.com/products/nitric-oxide-safety/. [Accessed: 13/04/2023].

[2] Alphansense. No2-b43f sensor datasheet. [Online] https://www.alphasense.com/products/nitrogen-dioxide/. [Accessed: 13/04/2023].

[3] Alphansense. Ox-b431 sensor datasheet. [Online] https://www.alphasense.com/products/ozone/. [Accessed: 13/04/2023].

[4] Alphansense. Support circuits (ppb): Isb individual sensor board datasheet. [Online] https://www.alphasense.com/products/support-circuits-air/. [Accessed: 13/04/2023].

[5] J. M. Barcelo-Ordinas, M. Doudou, J. Garcia-Vidal, and N. Badache. Self-calibration methods for uncontrolled environments in sensor networks: A reference survey. *Ad Hoc Networks*, 88:142–159, 2019.

[6] F. Concas, J. Mineraud, E. Lagerspetz, S. Varjonen, X. Liu, et al. Low-cost outdoor air quality monitoring and sensor calibration: A survey and critical analysis. *ACM Transactions on Sensor Networks*, 17(2), may 2021.

[7] G. Coulby, A. K. Clear, O. Jones, and A. Godfrey. Low-cost, multimodal environmental monitoring based on the internet of things. *Building and Environment*, 203:108014, 2021.

[8] P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Data reconstruction applications for iot air pollution sensor networks using graph signal processing. *Journal of Network and Computer Applications*, page 103434, 2022.

[9] P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Graph signal reconstruction techniques for iot air pollution monitoring platforms. *IEEE Internet of Things Journal*, 9(24):25350–25362, 2022.

[10] P. Ferrer-Cid, J. M. Barcelo-Ordinas, J. Garcia-Vidal, A. Ripoll, and M. Viana. A comparative study of calibration methods for low-cost ozone sensors in iot platforms. *IEEE Internet of Things Journal*, 6(6):9563–9571, 2019.

[11] P. L. Fung, M. A. Zaidan, S. Sillanpää, A. Kousa, J. V. Niemi, H. Timonen, J. Kuula, E. Saukko, K. Luoma, T. Petäjä, et al. Input-adaptive proxy for black carbon as a virtual sensor. *Sensors*, 20(1):182, 2019.

[12] L. Liu, S. M. Kuo, and M. Zhou. Virtual sensing techniques and their applications. In *2009 International Conference on Networking, Sensing and Control*, pages 31–36. IEEE, 2009.

[13] B. Maag, Z. Zhou, and L. Thiele. A survey on sensor calibration in air pollution monitoring deployments. *IEEE Internet of Things Journal*, 5(6):4857–4870, 2018.

[14] N. H. Motlagh, E. Lagerspetz, P. Nurmi, X. Li, S. Varjonen, J. Mineraud, M. Siekkinen, A. Rebeiro-Hargrave, T. Hussein, T. Petaja, et al. Toward massive scale air quality monitoring. *IEEE Communications Magazine*, 58(2):54–59, 2020.

[15] PurpleAir. Classic air quality monitor. [Online] https://www2.purpleair.com/products/purpleair-pa-ii. [Accessed: 13/04/2023].

[16] J. Rovira, J. A. Paredes-Ahumada, J. M. Barcelo-Ordinas, J. Garcia-Vidal, C. Reche, Y. Sola, P. L. Fung, T. Petäjä, T. Hussein, and M. Viana. Non-linear models for black carbon exposure modelling using air pollution datasets. *Environmental research*, 212:113269, 2022.

[17] J. Woo, S. An, K. Hong, J. Kim, S. Lim, H. Kim, and J. Eum. Integration of cfd-based virtual sensors to a ubiquitous sensor network to support micro-scale air quality management. *Journal of Environmental Informatics*, 27(2), 2016.

[18] M. A. Zaidan, N. H. Motlagh, P. L. Fung, et al. Intelligent calibration and virtual sensing for integrated low-cost air quality sensors. *IEEE Sensors Journal*, 20(22):13638–13652, 2020.

[19] M. A. Zaidan, D. Wraith, B. E. Boor, and T. Hussein. Bayesian proxy modelling for estimating black carbon concentrations using white-box and black-box models. *Applied sciences*, 9(22):4976, 2019.