Graph Signal Reconstruction Techniques for IoT Air Pollution Monitoring Platforms

Pau Ferrer-Cid, Jose M. Barcelo-Ordinas, Jorge Garcia-Vidal

***Abstract*—Air pollution monitoring platforms play a very important role in preventing and mitigating the effects of pollution. Recent advances in the field of graph signal processing have made it possible to describe and analyze air pollution monitoring networks using graphs. One of the main applications is the reconstruction of the measured signal in a graph using a subset of sensors. Reconstructing the signal using information from neighboring sensors is a key technique for maintaining network data quality, with examples including filling in missing data with correlated neighboring nodes, creating virtual sensors, or correcting a drifting sensor with neighboring sensors that are more accurate. This paper proposes a signal reconstruction framework for air pollution monitoring data where a graph signal reconstruction model is superimposed on a graph learned from the data. Different graph signal reconstruction methods are compared on actual air pollution data sets measuring O_3, NO_2, and PM_{10}. The ability of the methods to reconstruct the signal of a pollutant is shown, as well as the computational cost of this reconstruction. The results indicate the superiority of methods based on kernel-based graph signal reconstruction, as well as the difficulties of the methods to scale in an air pollution monitoring network with a large number of low-cost sensors. However, we show that the scalability of the framework can be improved with simple methods, such as partitioning the network using a clustering algorithm.**

***Index Terms*—Air Pollution Monitoring Networks, IoT Platform, Low-cost Sensors, Graph Signal Processing, Signal Reconstruction**

I. Introduction

OUTDOOR air pollution originates from natural and anthropogenic sources. Air pollution produces adverse health consequences, such as respiratory, neuronal, and cardiovascular problems, among others, as a result of short or long-term exposure. For all these reasons, in recent years much effort has been made to obtain pollution measures for their study and the application of prevention and containment measures. To increase the number of measurement points, the use of low-cost sensors, mounted on Internet of Things (IoT) nodes, has been proposed in conjunction with the values provided by reference stations, forming IoT platforms. Reference stations give much more precise values, but at a much higher price. On the other hand, the greatest difficulty in a deployment with low-cost sensors is the quality of the data it produces. To improve the quality of the low-cost sensors, it is necessary to calibrate [1], [2] these sensors in the field using supervised machine learning algorithms [3], [4], [5], [6]. Today, as the calibration of sensors seems to be feasible [7], [8], the study of sensor networks for pollution monitoring poses new challenges. Once the sensors are deployed, they suffer from drifts and aging [9], [10], so that the estimates made with the machine learning mechanisms become inaccurate. In addition, both low-cost sensors and reference stations lose data [11], which makes the supervised machine learning mechanisms difficult to estimate values.

The study of heterogeneous networks, which contain both reference stations and low-cost sensors, using graphs provides great flexibility, as the graphs can encode complex networks, as is the case when there is a reference station and a sensor in the same geographical location, a situation in which the geostatistical Kriging technique would fail. There is a wide variety of techniques for the estimation of the graph topology based on different methods such as statistical-based methods or graph signal processing (GSP) methods [12], [13].

There are a large number of direct applications of generating a graph with reference stations and low-cost sensors and overlaying a signal reconstruction mechanism in which several of the network nodes participate. The principle behind this technique is to obtain the relationships between the network sensors by means of a graph learned from the data, resulting in a smooth structure with respect to the measured data. Then, the data measured by the different sensor neighborhoods can be used by signal reconstruction methods to obtain estimates and maintain the quality of the network data. An example is the imputation of missing values. It is known that sensors are calibrated with supervised methods using arrays of sensors [3], [4], [5], [6]. If one of the array measurements is missing, the concentration of the pollutant cannot be estimated and a gap in the measurements occurs [11], [14], [15]. This measurement can be estimated using neighbors whose measures are highly correlated with the data from the sensor that has the missing value. Other examples where signal reconstruction methods can be applied overlaid on a graph constructed from the data are sensor drifts [5], [9], [16], the creation of virtual sensors [17] and the creation of proxies [14], [15], [18]. Virtual sensors are nodes in which it is difficult to deploy a physical sensor, and in which the value of the pollutant is estimated from values in the vicinity [19]. Thus, graph signal reconstruction can also help maintain network data quality by obtaining virtual sensor estimates when any of the nodes are under maintenance or have been relocated. Signal reconstruction is therefore a key technique to benefit from similar sensors, including reference stations, which can correct data for multiple applications.

Pau Ferrer-Cid (pau.ferrer.cid@upc.edu), Jose M. Barcelo-Ordinas (jose.maria.barcelo@upc.edu) and Jorge Garcia-Vidal (jorge.garcia@upc.edu) are with the Universitat Politecnica de Catalunya, Barcelona, Spain.

This work is supported by the National Spanish funding PID2019-107910RB-I00, by regional project 2017SGR-990, and with the support of Secretaria d'Universitats i Recerca de la Generalitat de Catalunya i del Fons Social Europeu.

In this paper, we propose a signal reconstruction framework to maintain air pollution sensor network data quality. Therefore, we study different methods for the reconstruction of the graph signal for several pollutants, studying their reconstruction capabilities when all the nodes have samples, and when various nodes are estimated at the same time. In this way, we demonstrate their ability to estimate air pollution and their scalability for use in IoT air pollution monitoring networks. More precisely, in this study we:

- propose a signal reconstruction framework based on graph signal reconstruction superimposed on a graph learned from the data. Thus, we compare the performance of three different graph signal reconstruction models applied to a set of Spanish reference stations that measure O_3, NO_2 and PM_{10} which could have missing values or be virtual sensors,
- analyze the scalability issues of the techniques for air pollution monitoring networks, and propose an application of clustering techniques to reduce these problems,
- show a low-cost sensor drift compensation example in a heterogeneous IoT air pollution platform with reference stations and low-cost sensors by using the proposed signal reconstruction framework.

The paper is organized as follows: section II covers the related work. Section III describes the graph learning and signal reconstruction techniques used, section IV describes the different Spanish reference station data sets employed. Section V shows the results for a set of experiments as well as the scalability study. Lastly, section VI concludes the paper.

II. Related Work

In-situ calibration and current monitoring networks: the use and calibration of low-cost sensors for air pollution monitoring have been intensively studied during the last years [1], [2]. The calibration process is carried out in uncontrolled environments, which means [2] that the sensor is calibrated in-situ in the field of deployment under real environmental conditions. Machine learning has been a key technique for sensor in-situ calibration, where studies include multiple linear regression [6], [5], random forest [4], support vector regression [20], [3], and artificial neural networks [3]. Once the sensors have been calibrated, they are deployed forming a network, and the nodes report the estimated data using a machine learning algorithm employed in the in-situ calibration phase. Some authors also make spatial estimates or create air pollution maps from this data. The most common form of spatial estimation of pollution is Kriging, a geostatistical method that extrapolates the known contamination values in some sources to other geographical locations forming a continuous surface [21], [16], [22]. For example, Schneider *et al.* [21] make use of Kriging to merge values from sensors and reference stations improving this way the estimation, and Barcelo-Ordinas *et al.* [16] use Kriging estimates to correct average concentrations based on reference stations measures.

Graph signal processing in air pollution monitoring networks: the study of these networks has also been carried out from a discrete point of view, modeling these networks as graphs. In [23] it is shown the use of Gaussian Markov random fields (GMRF) for the approximation of Gaussian fields and the extrapolation of environmental measurements. In addition, given the rise of the graph signal processing (GSP) field, which allows the use of classical signal processing techniques on defined signals in irregular structures such as graphs, many kinds of networks are modeled using graphs [13], [24]. The Laplacian matrix is the object of study of the GSP and is required to correctly describe the relationship between the network nodes, so there is a lot of literature available regarding the learning of the Laplacian matrix given a set of observations [13]. In the specific case of environmental monitoring networks, results have already been shown from the use of GSP on such networks. As examples, Jablonski [25] clusters tropospheric ozone measures in Poland using a GSP framework, and Ferrer-Cid *et al.* [11] compare three graph inferring methods for building a graph. Graph signal reconstruction techniques have also been used to estimate rainfall and temperature measurements [24], [26].

Limitations in current heterogeneous air pollution monitoring networks: in-situ calibration, where the signal from an individual sensor is estimated, has several limitations. One is its inability to estimate values at those moments in time when the sensor fails and is unable to give a reading [11], [14], [15]. Another is that as the sensor ages or the environmental conditions change from the time the calibration process was performed, sensors drift or age and the estimated data loses quality [16], [27], [9], [10]. Reference station data sets also report missing data [11]. The deterioration of the data quality of these sensors can be mitigated if in the estimation phase the signal of a problematic node is reconstructed using other sensors, which are potentially non-problematic, that were highly correlated during the graph learning stage. Thus, graph signal reconstruction for air pollution sensor networks is key to harnessing network information and maintaining data quality, where unobserved nodes (e.g., sensors with missing samples, drifting sensors, etc.) are reconstructed from a subset of observed nodes in the graph, also including places where there are no physical sensors (virtual sensors) [17], [14].

Graph signal reconstruction techniques: there are methods from the field of semi-supervised learning such as the Laplacian interpolation (Laplacian regularization from Belkin *et al.* [28]) that use the Laplacian of the graph to extrapolate the observed values to those that are not, maximizing the smoothness of the resulting signal. Other techniques, based on signal processing, use the Fourier transform assuming certain properties of the signal to recover the full signal given some measurements [29]. Some works address the question from the point of view of the kernel methods, kernelizing the signal reconstruction and obtaining the kernel ridge regression (KRR) [26]. Matrix completion methods are transductive methods that complete the missing entries of the data matrix assuming that it has a lower rank representation; e.g. the kernelized probabilistic matrix factorization (KPMF) [30] that uses graph kernels to factor the matrix. In addition, another recent field that has benefited from graph signal processing tools is graph neural networks (GNN).Among the networks developed we find ChebyNet [31], which approximates convolution filtering through a Chebyshev polynomial filter, or the inductive graph neural network Kriging (IGNNK) [32], a 2-layer diffusion convolution neural network that allows reconstructing any of the nodes through a subnetwork selection scheme during the training. But the two last techniques need to be further developed, since matrix completion methods do not naturally fit into the problem of signal reconstruction, as for each new sample it needs to be re-trained. Besides, there not exist much literature about graph neural networks architectures for graph signal reconstruction in cases where data is limited. We restrict the comparison of reconstruction methods to linear models, since non-convex models usually require more data, which are often unavailable in low-cost sensor deployment environments. Moreover, non-convex models are more difficult to adapt to environments where any subset of nodes can have missing data.

Graph signal reconstruction in air pollution monitoring networks: recently, research has been carried out by applying graph signal reconstruction to PM2.5 data sets [33], [34]. The matrix completion problem has also been tackled using variational graph autoencoders (VGAE) applied to NO_2 and PM data sets [35]. Graph convolutional recurrent neural networks have also been used for PM data [36]. However, as mentioned above, the matrix completion approach to signal reconstruction, or the need for graph neural networks for large amounts of training data, can make it difficult to use these models in sensor network deployments. In this paper, we opt for the use of linear graph signal reconstruction models on a graph learned from the data, given their low data and computational requirements. This allows extending the use of these techniques to both sensor deployment environments where the data may be scarce, and to environments where any node could fail and the signal reconstruction model would need to be recalculated. Furthermore, in most cases graph signal reconstruction models for air pollution are applied to graphs created from the geodesic distances of the nodes [33], [35], [36]. However, given the effectiveness of graphs learned from data in air pollution sensor networks [11], we chose to learn the graph from the data in our signal reconstruction framework.

Our work: we propose a framework composed of graph signal reconstruction superimposed on a graph learned from the data to maintain the data quality of a sensor network. Since the data quality maintenance is proportional to the quality of the signal reconstruction, we compare the accuracy of three graph signal reconstruction methods to reconstruct air pollution monitoring signals with missing values for O_3, NO_2 and PM_{10}; Laplacian interpolation, GSP low-pass reconstruction and kernel-based graph signal reconstruction. We propose a cluster-based methodology to split the problem of graph learning and signal reconstruction in order to overcome scalability issues for large air pollution sensor networks. And finally, we show a case study on data quality maintenance where signal reconstruction is used to correct a drifting low-cost sensor.

TABLE I
NOTATION SUMMARY.

SYMBOL	MEANING
$N \mid P$	Number of nodes | Number of observations
$G \mid \mathcal{V} \mid \mathcal{E}$	Graph | Set of nodes | Set of edges
$\mathbf{A} \mid \mathbf{W}$	Graph adjacency matrix | Graph weight matrix
$\mathbf{D} \mid \mathbf{L}$	Graph degree matrix | Graph Laplacian matrix
$\mathrm{x}_i \mid \mathrm{N}(i)$	ith vertex | ith vertex neighborhood
$\mathbf{\Sigma} \mid \mathbf{\Theta}$	Covariance matrix | Precision matrix
$\mathrm{tr}(\cdot) \mid \mathrm{det}(\cdot)$	Trace of a matrix | Determinant of a matrix
$\|\cdot\|_1 \mid \|\cdot\|_F$	l_1-norm of a matrix | Frobenius norm of a matrix
$\alpha \mid \beta$	Graph learning hyperparameters
$\mathbf{U} \mid \mathbf{K}$	Laplacian eigenvector matrix | Graph kernel matrix
$\mathcal{M} \mid \mathcal{U}$	Set of observed | unobserved or missing nodes
$\mathrm{M} \mid \mathrm{U}$	Number of observed nodes | unobserved nodes
$\lvert\mathcal{M}\rvert \mid \mathbf{U}^{\dagger}$	Cardinality of a set | Matrix pseudoinverse
$\mathbf{0} \mid \mathbf{1}$	Vector of zeros | Vector of ones

III. Graph signal reconstruction framework

This section presents the framework to learn the graph topology given a set of observations and introduces the signal reconstruction techniques used in the study. Bold uppercase letters denote matrices, bold lowercase letters denote vectors and lowercase letters denote scalars. Table I summarizes the notation used throughout the paper. Figure 1 represents the proposed signal reconstruction framework where a graph signal reconstruction model is superimposed on a graph learned from the sensor network data. The following sections describe the framework's two main pieces; graph learning and signal reconstruction.

A. Graph learning

The graph learning task consists in learning the graph triplet G=$\{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ given a set of observations $\mathbf{X}\in\mathbb{R}^{\mathbf{P}\times\mathbf{N}}$, where P corresponds to the number of observations and N corresponds to the number of nodes. $\mathcal{V}$ is a prefixed set of nodes, $\mathcal{E}$=$\{e_{ij}$:$v_i, v_j \in \mathcal{V}\}$ are the learned edges between nodes and $\mathbf{W}$ is the weight matrix whose entries encode the edges weights. Furthermore, the Laplacian matrix $\mathbf{L}$ also defines the connectivity of the resulting graph. The Laplacian matrix is defined as $\mathbf{L}$= $\mathbf{D}$-$\mathbf{W}$, where $\mathbf{D}$ is the diagonal degree matrix with values D_{mm}= $\sum_n W_{mn}$. For an undirected graph (our case), the Laplacian matrix is symmetric, $\mathbf{L}$= $\mathbf{L}^T$. This graph topology can then be used to flow information between similar sensors using signal reconstruction techniques. The nodes that make up the graph are reference stations that provide accurate data, and low-cost sensor array nodes, which provide the estimation of values already calibrated in-situ with supervised machine learning methods.

In recent studies, several approaches have been studied to obtain the weight matrix $\mathbf{W}$ or the Laplacian matrix $\mathbf{L}$, from statistical approaches, such as the graphical lasso algorithm, to graph signal processing approaches with distance-based and smoothness-based methods. As most of the reconstruction methods rely on the signal smoothness assumption -vertices connected with large weights tend to have similar values [24], [13]- the smoothness method shown by Dong *et al.* [37] is used. Dong *et al.* formulate a jointly non-convex problem

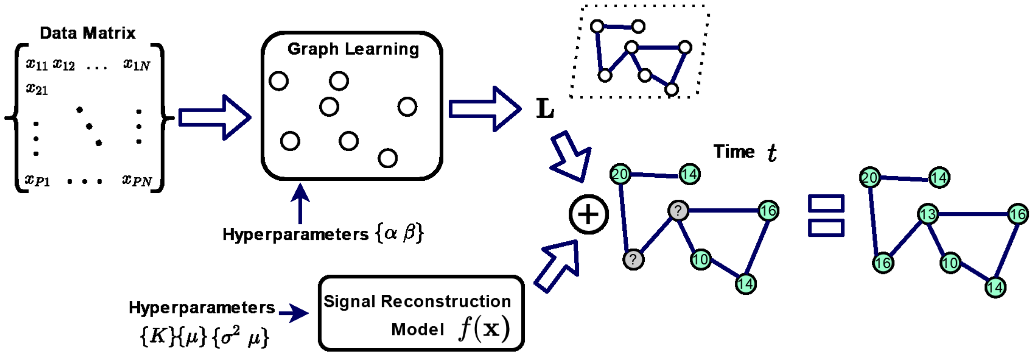

Data Matrix
x_{11} x_{12} ... x_{1N}
x_{21}
x_{P1} ... x_{PN}
Graph Learning
Hyperparameters $\{\alpha\ \beta\}$
L
Time t
20
14
?
16
?
10
14
20
14
13
16
16
10
14
Hyperparameters
$\{K\}\{\mu\}\{\sigma^2\ \mu\}$
Signal Reconstruction Model $f(\mathbf{x})$

Fig. 1. Signal reconstruction framework. Given a set of graph learning and signal reconstruction hyperparameters, the Laplacian can be learned from the data, and network measures can be reconstructed using both the graph and a signal reconstruction model.

to find a filtered version $\mathbf{Y}$ of the observations $\mathbf{X}$ and the Laplacian matrix $\mathbf{L}$:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{Y}} \quad & \underbrace{\|\mathbf{X}-\mathbf{Y}\|_F^2}_{data\ \ fidelity} + \underbrace{\underbrace{\alpha\, tr(\mathbf{Y}^T\mathbf{L}\mathbf{Y})}_{smoothness} + \beta\|\mathbf{L}\|_F^2}_{sparsity} \\ \text{s.t.} \quad & tr(\mathbf{L}) = n, \\ & L_{ij} = L_{ji} \leq 0, \qquad i \neq j, \\ & \mathbf{L}\cdot\mathbf{1} = \mathbf{0}. \end{aligned} \tag{1}$$

This problem is solved by alternating the minimization on $\mathbf{Y}$ and $\mathbf{L}$. Two scalar hyperparameters need to be provided to the algorithm; α and β. Hyperparameter α controls the smoothness of the solution, $\mathbf{L}$ smooth with respect to $\mathbf{Y}$, while hyperparameter β penalizes the Frobenius norm $||\cdot||_F^2$ of the Laplacian matrix, so controlling the number of non-zero entries of the matrix. As the authors mention, the computational complexity scales quadratically with the number of vertices in the graph.

B. Signal reconstruction

Signal reconstruction is key to maintaining the data quality of a sensor network. Estimates can be obtained for a sensor that may present problems (e.g., missing value, malfunction, drift, etc.) using information from its neighboring sensors. In addition, in a worst case, such estimation can also be performed when the information is partial, i.e., there is more than one sensor whose signal needs to be reconstructed or presents missing data. We consider the problem of having a subset of nodes (vertices of the graph) with samples and we would like to estimate the signal of the graph in the other vertices so that the resulting signal is smooth. This can be seen as a signal reconstruction problem that can be solved using methods from various fields. In this paper we consider the following; Laplacian interpolation, GSP low-pass based graph signal reconstruction, and kernel-based graph signal reconstruction[1]. Laplacian interpolation [28] is a graph-based semi-supervised learning algorithm whose goal is regression with graph regularization assuming smoothness with respect to the Laplacian matrix. This method regresses a function $f\colon\mathcal{V}\to\mathbb{R}$ over the graph G, assuming partial information, it is to say, information for M nodes. GSP low-pass based graph signal reconstruction [29] is a graph signal processing reconstruction method that considers subsampling low-pass graph signals, thus assuming a sparse Fourier coefficient vector. Finally, kernel-based graph signal reconstruction [26] formulates signal reconstruction as a regression task on reproducing kernel Hilbert spaces of graph signals. In the following, we briefly describe the three methods and their correspondence to a linear model.

1) Laplacian interpolation: Also known as graph interpolated regularization by Belkin *et al.* [28], this method minimizes the quadratic form of the Laplacian matrix with respect to the graph signal $\mathbf{x}$, which is a measure of signal smoothness, given that the observed measurements $\{x_m\colon \forall m{\in}\mathcal{M}\}$ remain unchanged. This reconstruction results in a linear combination of the observations weighted by the Laplacian matrix entries L_{ij}.

$$\begin{aligned} &\underset{y}{Min} \quad \mathbf{y}^T\mathbf{L}\mathbf{y} \\ &\text{s.t.} \qquad y_m = x_m, \qquad \forall m \in \mathcal{M}, \end{aligned} \tag{2}$$

2) Graph Signal Processing (GSP) low-pass reconstruction: This technique [29] recovers a set of unobserved nodes $\{x_u\colon \forall u{\in}\mathcal{U}\}$ given that the graph discrete Fourier transform of the complete signal is sparse and of low-pass nature, meaning that it has K nonzero components corresponding to the lowest frequencies (smallest eigenvalues λ_i of the Laplacian matrix). Given that the Laplacian matrix admits the eigendecomposition $\mathbf{L}{=}\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, the graph discrete Fourier transform (GDFT) of a graph signal $\mathbf{x}$ can be computed as:

$$\mathbf{X} = \mathbf{U}^{-1}\mathbf{x} \tag{3}$$

Now, a K-sparse GDFT coefficient vector of the following form is to be recovered:

$$\mathbf{X} = (X(0), \ldots, X(K-1), 0, \ldots, 0)^T \tag{4}$$

For this purpose a subset of measurements $\mathcal{M}$ are used to recover the sparse coefficient vector by solving the following system:

$$\mathbf{x}_{\mathcal{M}} = \mathbf{U}_{\mathcal{M}K}\mathbf{X}_K \tag{5}$$

[1] All Python code implementing the methods explained can be downloaded here at the research group http://sans.ac.upc.edu/?q=node/231.

Since the system is overdetermined, the solution of the above system in the least squares sense is given by $\mathbf{X}_K = \mathbf{U}^{\dagger}_{\mathcal{M}K}\mathbf{x}_{\mathcal{M}}$, where $\mathbf{U}^{\dagger}_{\mathcal{M}K} = (\mathbf{U}^T_{\mathcal{M}K}\mathbf{U}_{\mathcal{M}K})^{-1}\mathbf{U}^T_{\mathcal{M}K}$ is the matrix pseudo-inverse of $\mathbf{U}_{\mathcal{M}K}$; the nonzero coefficients are obtained, and after appending the corresponding zero coefficients, the inverse graph discrete Fourier transform (IGDFT) $\mathbf{x}=\mathbf{U}\mathbf{X}$ is computed to obtain the complete set of measurements $\mathbf{x}$ at all vertices.

3) Kernel-based graph signal reconstruction: Romero *et al.* [26] introduced the kernelized reconstruction of graph signals. Given a set of noisy observations $\{x_m = y_0(v_m) + \epsilon_m : \forall m \in \mathcal{M}\}$, the kernel regression estimates the underlying function y_0 in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, which is a space of functions $f : \mathcal{V} \to \mathbb{R}$:

$$\mathcal{H} := \{f : f(v) = \sum_{n=1}^{N} \alpha_n k(v, v_n), \alpha_n \in \mathbb{R}\} \tag{6}$$

Where $k : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is a kernel map that defines some similarity between nodes. After some manipulation and the application of the representer theorem, which states that the solution can be expressed as a linear combination of the kernel map values of the observed nodes $\hat{f}_0(v) = \sum_{m\in\mathcal{M}} \hat{\bar{\alpha}}_m k(v, v_m)$, we can define the kernel ridge regression problem:

$$\hat{\bar{\boldsymbol{\alpha}}} := \underset{\hat{\bar{\alpha}}\in R^M}{arg\,min} \underbrace{\frac{1}{M}||\bar{\mathbf{y}} - \bar{\mathbf{K}}\bar{\boldsymbol{\alpha}}||^2}_{MSE} + \mu \underbrace{\bar{\boldsymbol{\alpha}}^T\bar{\mathbf{K}}\bar{\boldsymbol{\alpha}}}_{RKHS\ norm} \tag{7}$$

Where the mean squared error (MSE) is used as loss function, the regularization term is the RKHS norm of the solution f_0, $\bar{\boldsymbol{\alpha}} = \boldsymbol{\Phi}\boldsymbol{\alpha}$, $\bar{\mathbf{K}} = \boldsymbol{\Phi}\mathbf{K}\boldsymbol{\Phi}^T$ and $\boldsymbol{\Phi}$ is the sampling matrix. This problem has closed-form solution:

$$\hat{\bar{\boldsymbol{\alpha}}} = (\bar{\mathbf{K}} + \mu M \mathbf{I}_M)^{-1}\bar{\mathbf{y}} \tag{8}$$

The key to the successful application of kernel-based graph signal reconstruction is in the kernel selection. Thus, based on the assumption that the signal evolves smoothly over the graph, graph kernels that capture such prior information can be used, a common choice is the diffusion kernel (KRR-DIFF) where $r(\lambda) = e^{\sigma^2\lambda/2}$:

$$\mathbf{K} = r^{\dagger}(\mathbf{L}) = \mathbf{U} r^{\dagger}(\boldsymbol{\Lambda})\mathbf{U}^T \tag{9}$$

Vertex-covariance kernel (KRR-COV) can also be used, which is based on the covariance instead of graph structure, which turns out to be the local linear minimum mean squared error (LMMSE) estimator on a Markov random field (MRF) [26]. Since the actual covariance matrix is unknown, the graphical lasso has been used to estimate the covariance matrix $\hat{\boldsymbol{\Sigma}}$. Given the presence of multicollinearity in air pollution data [11] and that the covariance matrix has to be estimated, the result is suboptimal.

4) Linear models: Given the analytical form of the solution of the methods described above, it can be noticed how the reconstruction of any of the methods corresponds to a linear combination $\mathbf{x}_{\mathcal{U}} = \boldsymbol{\beta}\mathbf{x}_{\mathcal{M}}$ of the observed nodes $\{x_m : \forall m \in \mathcal{M}\}$. Thus, Table II shows how the $\boldsymbol{\beta}$ coefficients are calculated for the reconstruction. As it can be seen, the main operations correspond to a matrix inversion and multiplication; in the case of Laplacian interpolation the $\boldsymbol{\beta}$ are calculated by the Laplacian $\mathbf{L}$, in the case of GSP by the Laplacian eigenvector matrix $\mathbf{U}$, in the case of the kernelized ridge regression with diffusion kernel by a pre-computed kernel matrix $\mathbf{K}$, and in the case of the kernelized ridge regression with vertex-covariance kernel by the data covariance matrix $\hat{\boldsymbol{\Sigma}}$.

TABLE II
EQUIVALENCE OF THE MODELS TO A LINEAR MODEL.

MODEL:	$\mathbf{x}_{\mathcal{U}} = \boldsymbol{\beta}\mathbf{x}_{\mathcal{M}}$
Lap.Int	$\boldsymbol{\beta} = -\mathbf{L}^{-1}_{\mathcal{U}\mathcal{U}}\mathbf{L}_{\mathcal{U}\mathcal{M}}$
GSP	$\boldsymbol{\beta} = \mathbf{U}_{\mathcal{U}K}(\mathbf{U}^T_{\mathcal{M}K}\mathbf{U}_{\mathcal{M}K})^{-1}\mathbf{U}^T_{\mathcal{M}K}$
KRR-DIFF	$\boldsymbol{\beta} = \mathbf{K}_{\mathcal{U}\cdot}\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\mathbf{K}\boldsymbol{\Phi}^T + \mu M\mathbf{I}_M)^{-1}$
KRR-COV	$\boldsymbol{\beta} = \hat{\boldsymbol{\Sigma}}_{\mathcal{U}\cdot}\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\hat{\boldsymbol{\Sigma}}\boldsymbol{\Phi}^T + \mu M\mathbf{I}_M)^{-1}$

IV. DATA SETS

This section introduces the data sets used for the comparison of the reconstruction methods described above. Nowadays, there are many open data initiatives to improve transparency and encourage research. The Spanish government carries out the measurement of pollution levels by means of reference stations, which are worth thousands of euros given their high accuracy, and makes such data public. Therefore, data captured by reference stations[2] in the area of Catalonia, Spain, over an area of 32,108 km^2 have been selected for three pollutants; tropospheric ozone (O$_3$), nitrogen dioxide (NO$_2$) and particulate matter 10 (PM$_{10}$).

These pollutants exhibit different spatial behavior, which allows studying the signal reconstruction under various spatial conditions. In addition, the use of reference stations over a large area makes it possible to investigate the feasibility of building a large network and the application of these methods. Table III shows the characteristics of the data sets used, and Figure 3 presents the location of the reference stations for the data sets. Moreover, Table III shows the means for the pollutants as well as the pooled standard deviation to better interpret the error measures in the following sections. These three data sets will be used to show the ability to reconstruct signals in an air pollution monitoring network in a regional area such as the metropolitan area of Barcelona, Spain.

The fourth data set corresponds to the deployment of a heterogeneous IoT network for the European project H2020 CAPTOR during the summer of 2017 [38]. In this case, we set the data set to two reference stations and six nodes with low-cost sensors. These nodes are composed of Arduino as the processing unit, temperature, humidity, and five SGX Sensortech MICS 2614 metal oxide ozone sensors. The data are sent using a cellular network to a repository for off-line processing. This data set will be used to show a concrete application of signal reconstruction in a heterogeneous network when an sensor drifts.

V. RESULTS

The methodology used to learn the graph and train the signal reconstruction method is as follows; i) we use 100% of the data

[2] These data are available at http://mediambient.gencat.cat/ca/05_ambits_dactuacio/atmosfer

TABLE III
STATISTICS OF THE DATA SETS USED FOR THE SIGNAL RECONSTRUCTION EXPERIMENTS.

DATA SET	POLLUTANT	# NODES	# SAMPLES	PERIOD	MEAN ($\mu gr/m^3$)	POOLED STD. ($\mu gr/m^3$)
1	O_3	46	1155	2019/01/01 - 2019/05/31	66.84	28.61
2	NO_2	60	983	2019/01/01 - 2019/05/31	23.46	16.52
3	PM_{10}	33	709	2019/01/01 - 2019/05/31	20.11	11.51
4	O_3	8	2612	2017/06/18 - 2017/09/16	64.92	34.84

for cross-validation (CV), ii) given the 2 hyperparameters of the graph learning method and the possible hyperparameters of the signal reconstruction method, we apply a 5-fold CV, iii) in each of the folds we learn the graph using the training data, we reconstruct each one of the nodes in the test and report the average root-mean squared error (RMSE) between all the nodes, and iv) given the hyperparameters corresponding to the lowest CV error we obtain the graph topology and the trained reconstruction method, section V-A. From now on, all performance metrics reported throughout the paper are cross-validation metrics.

Three experiments are carried out for comparing the performance of the graph signal reconstruction methods. The first, section V-B, consists in reconstructing all nodes given that all neighbors are available during the CV, thus obtaining the average CV error to show the reconstruction power of the methods. The second, section V-C, simulates that a variable percentage of nodes need to be estimated simultaneously (e.g. some nodes may have missing data or are faulty nodes), this shows the ability of the methods to reconstruct the signals in a semi-supervised learning setting. The last experiment, section V-D, analyzes the scalability of the framework including both graph learning and signal reconstruction models. Finally, to conclude the results part, section V-E shows a practical example of the signal reconstruction framework when the network compensates for sensor drift. The experiments in sections V-A to V-D use data sets 1, 2 and 3, while the experiment in section V-E uses data set 4.

A. Learning an air pollution monitoring graph

Figure 1 shows the usual procedure for graph inference and signal reconstruction. First, given a data matrix $\mathbf{X}$ and the graph learning method with its hyperparameters α and β (eq. 1), we get the Laplacian matrix $\mathbf{L}$ that describes the relationships between the nodes. Then, given a graph signal reconstruction model and its hyperparameters - none for the Laplacian interpolation, K for the GSP, μ and σ^2 for KRR-DIFF and μ for KRR-COV - we can reconstruct a graph signal at any time t given a subset of observed nodes $\mathcal{M}$. Actually, this is the methodology performed during the cross-validation, where the graph is learned using the training set of that fold and the nodes in the test set, of the same fold, are reconstructed. The four graph signal reconstruction models are transductive, it is to say, the coefficients for the reconstruction can be calculated given a set of observed nodes to reconstruct the unobserved ones, but if the set of observed nodes changes (e.g. some nodes have missings) the model needs to be recalculated to reconstruct the new set of unobserved nodes. Algorithm 1 gives a detailed description of the signal reconstruction framework process, where $\mathbf{f}(\cdot)$ is the graph signal reconstruction model, C is the number of clusters, and $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{hyp}\}$ are the hyperparameters for the graph learning and graph signal reconstruction models. Lines 2 to 5 partition the set of nodes $\mathcal{V}$ into C disjoint clusters, where $\mathcal{C}_i \subseteq \mathcal{V}$ are the nodes indexes belonging to the i-th cluster, and learn a separate Laplacian $\mathbf{L}_i$ per cluster. We elaborate more on this idea in the scalability section V-D, for the moment we assume that we do not partition the set of nodes (i.e., C=1).

Algorithm 1 Cluster-wise graph signal reconstruction.

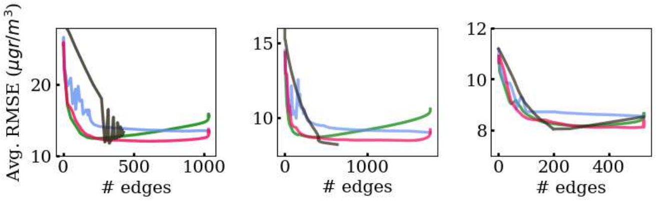
```
Input: $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{X}, \mathbf{f}(\cdot), \mathbf{hyp}, C\}$
 1: $\bar{\mathbf{X}} \leftarrow$ Standardization$(\mathbf{X})$
 2: $\mathcal{C}_1, ..., \mathcal{C}_C \leftarrow$ Clustering$(\bar{\mathbf{X}}, C)$        ◁ ↑ Scalability
 3: for $i = 1$ to $C$ do
 4:     $\mathbf{L}_i \leftarrow$ Graph_Learning$(\alpha_i, \beta_i, \bar{\mathbf{X}}_{\mathcal{C}_i})$
 5: end for
 6: while $\mathbf{x}_{new}$ do        ◁ New sample collected
 7:     $\mathcal{U} \leftarrow$ Get_Unobserved_Nodes$(\mathbf{x}_{new})$
 8:     $\mathcal{M} \leftarrow \mathcal{V} \setminus \mathcal{U}$
 9:     $\bar{\mathbf{x}}_{new} \leftarrow$ Standardization$(\mathbf{x}_{new})$
10:     for $i = 1$ to $C$ do
11:         $\bar{\mathbf{x}}_{new_{\mathcal{C}_{i_{\mathcal{U}}}}} \leftarrow \mathbf{f}(\bar{\mathbf{x}}_{new_{\mathcal{C}_{i_{\mathcal{M}}}}}, \mathbf{L}_i, \mathbf{hyp}_i)$ ◁ Reconstruction
12:     end for
13:     $\mathbf{x}_{new} \leftarrow$ Unstandardization$(\bar{\mathbf{x}}_{new})$
14:     return $\mathbf{x}_{new}$
15: end while
```

There are then two blocks of hyperparameters; one that corresponds to graph learning and the other to signal reconstruction. Therefore, it is possible that for a given graph, with a certain number of edges, two methods perform in a different way. For this reason, cross-validation (CV) is made with all the hyperparameters, both for the graph learning and for the signal reconstruction, since the performance of the reconstruction method can be associated with the obtained graph. To illustrate this, Figure 2 shows the average CV RMSE for each of the methods given the graph obtained. That is, the average RMSE of CV for each α and β and the hyperparameters of the signal reconstruction method that have obtained the lowest error (the best K for GSP, the best μ and σ^2 for KRR-DIFF and the best μ for KRR-COV). The case of the kernelized ridge regression with vertex-covariance kernel is a special case, where the covariance matrix is estimated using the graphical Lasso algorithm with different λ hyperparameter values, thus obtaining an adjacency matrix (given the precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$) but not a Laplacian matrix.

The same trends can be seen in the three plots in Figure 2. Laplacian interpolation gets its best error for a low number of edges, increasing later as the number of edges in the graph grows. The KRR-DIFF finds its lowest RMSE, similar to that of the Laplacian interpolation, but with a larger number of

TABLE IV
CV PERFORMANCE METRICS FOR THE DATA SETS AND RECONSTRUCTION METHODS.

METHOD	O_3				NO_2				PM_{10}			
	RMSE	MAE	R^2	#EDGES	RMSE	MAE	R^2	#EDGES	RMSE	MAE	R^2	#EDGES
Lap.Int.	12.41	9.51	0.66	216.80	8.72	6.44	0.42	369.80	8.16	5.66	0.26	297.60
GSP	13.43	10.44	0.56	1035.00	9.03	6.73	0.26	1761.20	8.54	5.99	0.16	528.00
KRR - DIFF	12.02	9.21	0.69	614.40	8.49	6.27	0.46	1514.00	8.10	5.61	0.29	481.80
KRR - COV	11.72	8.98	0.71	328.80	8.23	6.04	0.50	638.40	8.03	5.51	0.30	198.00



(a) O_3 data set.

(b) NO_2 data set.

(c) PM_{10} data set.

Fig. 2. Average CV RMSE obtained for different number of edges and the best signal reconstruction hyperparameters. Line in green corresponds to the Lap.Int., blue line corresponds to GSP, and red and black lines correspond to KRR-DIFF and KRR-COV respectively.

edges. The GSP reconstruction shows great instability for very sparse graphs, but its best error is obtained as the density of the graph increases. Finally, the case of KRR-COV shows great instability due to the multicollinearity present in the data, in many cases, it is not possible to obtain the complete graph with the graphical Lasso due to this problem. Also for similar graphs, the error can vary a lot, but the best error seems to be obtained with 50% of the edges. So, for a low number of edges the Laplacian interpolation and KRR-DIFF are the best, the KRR-COV seems to obtain its best error with a relatively small number of edges, and the low-pass based reconstruction needs a large number of edges (around 100%) to obtain its best performance. Although the KRR-COV is the best linear estimator for all three data sets, Figure 2 shows how for sparse graphs (<25% of the edges) the other methods produce a lower error. This is because the smoothness-based graph learning method is able to find a better set of neighbors per node for signal reconstruction than graphical Lasso, so although KRR-COV is optimal given a dense graph, it is not able to obtain the best performance for sparse graphs. In summary, each method behaves differently for a graph, so graph learning must be coupled to the signal reconstruction model in the cross-validation procedure.

B. Signal reconstruction: All nodes available

Let us see the result of learning the graph and performing signal reconstruction given that the data of all node neighbors are available. The average CV error of all reference stations is used as performance metric. Table IV shows the minimum average CV RMSE between stations, the average CV mean absolute error (MAE), as well as the corresponding average CV coefficient of determination R^2, and the average number of edges for the data sets and signal reconstruction methods. The methods appear to perform similarly, with KRR-COV being the best and GSP the worst. For the O_3 we can observe how the GSP is the worst method obtaining an R^2 of 0.56 with a complete graph (1035 edges) and the KRR with vertex-covariance kernel is the best model obtaining R^2 of 0.71 and 328.8 edges on average, followed by the KRR-DIFF that obtains a similar performance with a denser graph (614.40 edges on average) and the Laplacian interpolation whose best performance is obtained with a sparse graph. The same trend is observed for the NO_2 where the KRR-COV obtains the best performance with an R^2 of 0.50 and 638.40 edges on average and followed by the KRR-DIFF and Laplacian interpolation that obtain a denser and a sparser graph respectively. Finally, the reconstruction methods applied to the PM_{10} data set obtain the worst results with R^2s around 0.16-0.3, but this time the KRR-COV performs the best with an average R^2 of 0.30 and the sparsest graph with 198.0 edges on average.

As it has been observed, O_3 and NO_2 can be estimated quite well by reconstructing the graph signal, but PM_{10} can not. The reason is that PM_{10} is more heterogeneous in the area of study than O_3 and NO_2, which makes it difficult to estimate using neighboring nodes. Among the methods, the KRR-COV obtains the best results as it can be interpreted as the local minimum linear mean squared error (LMMSE) estimator on a Markov random field. Nevertheless, as the covariance matrix is estimated and the data suffer from multicollinearity, the result is suboptimal and the graphical Lasso has not been able to obtain the covariance matrix with all dependencies (100% of the edges). However, the KRR-DIFF performs almost as well as KRR-COV, followed by Laplacian interpolation which also shows good results.

The Laplacian interpolation has been able to obtain an error close to the optimal local linear estimator (KRR-COV) since the smoothness of the Laplacian matrix with respect to the training data is a criterion of the graph learning optimization problem. As for the resulting graphs, the GSP method obtains the best performance with 100% of the edges. On the other hand, the Laplacian interpolation obtains its best performance with 21% of the edges (with respect to the complete graph) for O_3, 21% of the edges for NO_2 and 56% of the edges for PM_{10}. This is an interesting result, since these methods allow obtaining a low RMSE with a small number of edges, which makes the graph sparse when the number of nodes is increased. Furthermore, in the previous section V-A, it has been observed that the vertex-covariance method is no longer the best model for sparse graphs, with approximately less than 25% of the edges.

Figure 3 shows the graphs obtained by the cross-validation procedure using the Laplacian interpolation method as reconstruction method. The color of the nodes denotes the average

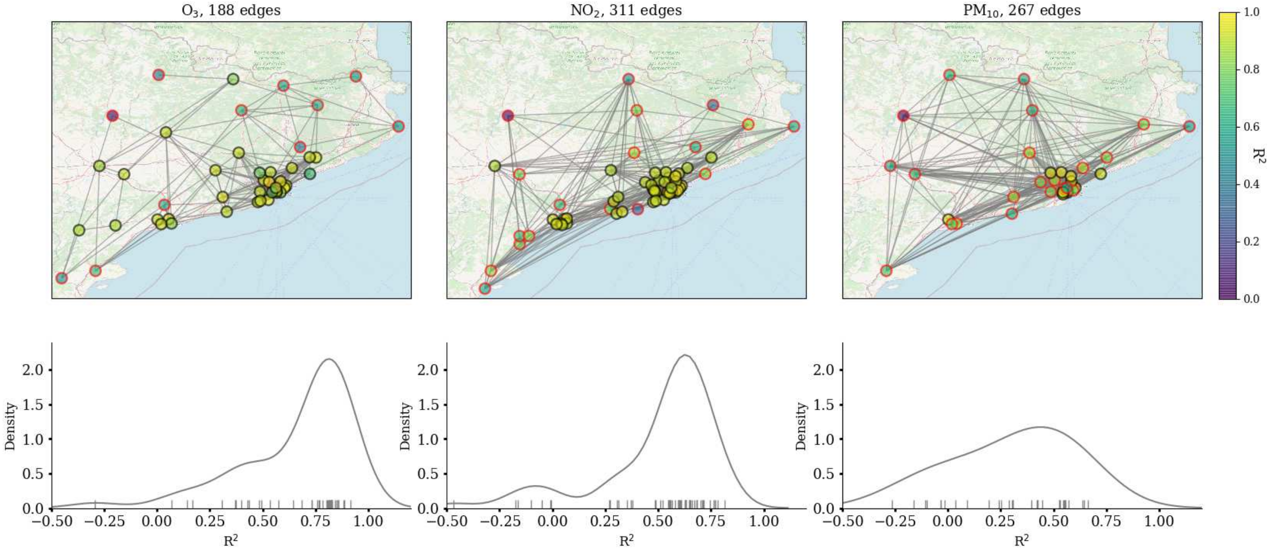


Fig. 3. Plots representing the CV result for the data sets using the Laplacian interpolation method. Top figures show the graphs obtained with node color denoting the average cross-validation R^2 of each station. The nodes highlighted in red denote reference stations with a CV R^2 below 0.5. Bottom figures show the empirical R^2 distribution of the graphs above.

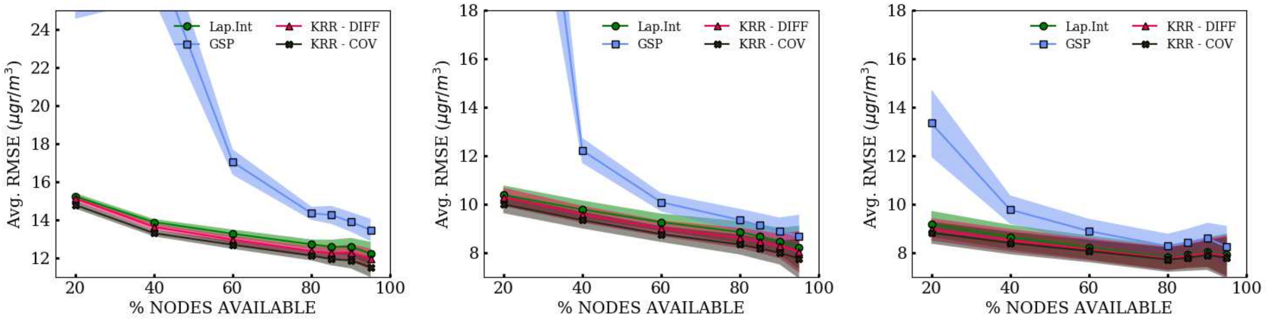
CV R^2 of the given stations. As shown in table IV, despite having an average R^2 of 0.66, many of the stations in areas with a high density of stations (e.g. Barcelona area) have an R^2 around 0.8. Thus, ozone values in denser areas of similar reference stations can be estimated effectively. The same is observed for NO_2, although it presents a lower spatial correlation, some reference stations obtain a coefficient of determination larger than 0.7. Finally, PM_{10} can not be approximated quite well using similar reference stations as most stations get an R^2 smaller than 0.6.

The results show a good prediction for most of the stations. Figure 3 shows the location of the nodes, some of them are far away from others or far from zones with density of stations. In the same figure, we can observe the reference stations highlighted in red, which are those with a CV R^2 smaller than 0.5. In the case of O_3, these are the most distant stations, and only 10 out of 46. In the case of NO_2, they already represent a larger percentage of the network stations with 19 stations out of 60 and are more distributed. Finally, in the case of PM_{10} the bad reconstructed stations represent the majority, 22 stations out of 33. This represents the idea that O_3 and NO_2 are more predictable due to their homogeneity. This result indicates that if we look at the individual RMSE of the stations instead of the global one, those stations that are in a high-density area achieve a high R^2 and, therefore, a good estimate, while those that are more distant and, therefore, do not have many neighbors, do not take advantage of the nodes of the network as much. As an example, if we considered as outliers the stations with low R^2 and kept them out of the calculation, the average R^2 with the KRR-DIFF would improve from 0.69 to 0.78 for O_3 and from 0.46 to 0.65 in the case of NO_2. This shows the existence of a possible grouping of nodes into those that can be predicted well with their neighbors (group of similar nodes), and into other nodes that can not be predicted well (group of not similar nodes). This idea will be examined in more detail in the scalability section V-D, where similar nodes will be grouped using a clustering technique in order to improve the scalability.

C. *Signal reconstruction: Semi-supervised scenario*

The previous section V-B showed the CV error of reconstructing the signal of a reference station when the others were available. In this section, we experiment with a semi-supervised learning setting, estimating multiple nodes at the same time, since we want to reconstruct several stations at once or we want to reconstruct some nodes that have missing data. This scenario is of special interest in the air pollution monitoring paradigm, since in a network there will always exist malfunctioning nodes, nodes with loss of samples, nodes in maintenance, and even the presence of virtual sensors. That is why the graph signal reconstruction model needs to be flexible to couple with the absence of data in neighboring sensors and estimate the signal in all nodes given a learned graph topology. Given a fixed set of hyperparameters, those found in the previous section, we calculate the CV error selecting a random incremental percentage of nodes to be estimated simultaneously, as if they were virtual sensors during the validation set, and perform ten repetitions.

Figure 4 shows the mean CV error with bands indicating its 95% confidence interval for the signal reconstruction methods and data sets. Figure 4.a) shows the results for the O_3 data set. First of all, we can observe how the GSP low-pass based reconstruction is the method that has the highest error with a 95% of nodes available with 13.45 μgr/m^3 followed by a large difference by the Laplacian interpolation method with 12.22 μgr/m^3. In the previous section (100% of nodes available), the Laplacian interpolation (12.41 μgr/m^3) already outperformed the GSP method (13.43 μgr/m^3). Moreover, as the percentage of available nodes decreases, the GSP low-pass based method is the reconstruction method whose error increases faster,



(a) O_3 data set.

(b) NO_2 data set.

(c) PM_{10} data set.

Fig. 4. Average CV error and its 95% confidence interval for several percentage of available nodes and 10 repetitions.

rising from 60% of nodes available. Thus, this method seems to be unable to generalize to the semi-supervised setting. The Laplacian interpolation and the kernelized ridge regressions are performing the best with a large difference. With 95% of the nodes available, the kernelized ridge regression with a diffusion kernel obtains a CV RMSE of 11.91 μgr/m^3 and the KRR with the vertex-covariance kernel obtains a CV RMSE of 11.50 μgr/m^3, clearly the KRR-COV is the lower-bound of KRR-DIFF and Laplacian interpolation, and the three follow the same trend as the percentage of available nodes increases. In these latter cases, the error is slightly smaller than in the previous section probably due to the random selection of subsets of nodes, but it is noticed that with a high percentage of nodes available 95-80% the error increases from 11.91 to 12.34 μgr/m^3 in the KRR-DIFF case, and from 11.86 to 12.27 μgr/m^3 in the vertex-covariance case. Finally, the error of all the reconstruction methods seems to grow significantly from 40% of available nodes.

In Figure 4.b) the same results can be seen for the NO_2 data set. The GSP low-pass based method is the worst performing method, 8.66 μgr/m^3, with a high percentage of nodes available (95%). The two KRR methods obtain similar errors, 8.03 μgr/m^3 and 7.75 μgr/m^3 respectively, in the case of NO_2, the difference between the two is even less than in the case of O_3. The kernel ridge regressions and the Laplacian interpolation methods show a similar pattern, the error with 95% of the nodes (57 nodes) is less than the average error of all the nodes (subsection V-B) due to the random node subset selection. As the percentage of available nodes decreases the error increases, although the increase is moderate, the GSP low-pass method degrades considerably from 60% nodes available, and in the kernel-based methods the increase is produced from 40 percent to 20 percent, going from 9.57 μgr/m^3 to 10.19 μgr/m^3 and 9.33 $\mu gr/m^3$ to 9.98 μgr/m^3 respectively. In summary, the kernelized ridge regressions and the Laplacian interpolation obtain similar values, the NO_2 has a smaller average concentrations (see section IV) and these methods get lower average R^2 for the NO_2 data, since its concentrations are less smooth in an area than the O_3.

Finally, figure 4.c) shows the results for the particulate matter data set. Again, the methods follow the same trend, where as the percentage of available nodes increases the error decreases. It is worth remembering that in subsection V-B the PM_{10} has been observed not to be predictable using neighboring concentrations, achieving a maximum coefficient of determination of 0.3 with the KRR-COV. Thus, the same results are maintained in the semi-supervised case where the worst method is GSP and the best the KRR-COV followed closely by its similar with diffusion kernel and the Laplacian interpolation. With a 20 percent of nodes available the methods get an error of; 13.32 μgr/m^3 with GSP, 9.17 μgr/m^3 with Laplacian interpolation, 8.95 μgr/m^3 with KRR-DIFF and 8.81 μgr/m^3 with KRR-COV.

In summary, we can conclude that Laplacian interpolation and kernel-based methods are robust and efficient in reconstructing the signal when an acceptable percentage (>60%) of nodes are available, and that efficiency decreases as fewer nodes become available.

D. Scalability

This section studies the scalability of the signal reconstruction framework, distinguishing between the graph learning task and the reconstruction of a graph signal $\mathbf{x}$ at a time instant t. Table V shows the costs and the required hyperparameters for the methods. It is important to mention that the learning of the graph is linked to the training of the reconstruction method, since their hyperparameters are selected by means of a 5-fold CV. In this way, the more hyperparameters, the more dimensions the grid search will have and the more cost the cross-validation will have. The graph learning task is solved by iteratively addressing a convex optimization problem that scales quadratically with the number of nodes N, so that, it is the most expensive task in the learning process. On the other hand, learning the graph for signal reconstruction with kernel ridge regression and diffusion kernel will require four hyperparameters (α, β, μ and σ^2) and, therefore, will be the one that the grid search will take the longest.

Once the hyperparameters are selected and thus the graph learned, it is not necessary to use a training set to train the graph signal reconstruction method, since these are transductive methods, where only the values of the set of observed nodes $\mathbf{x}_{\mathcal{M}}$ are needed to interpolate the unobserved ones $\mathbf{x}_{\mathcal{U}}$ at instant t. In the test set, the predictions will have the cost shown in the Table V given the reconstruction models seen in section III-B4. Generally, all four reconstruction methods require a matrix inversion or multiplication, whose cost is

TABLE V
METHODS' COST ALONG WITH THEIR HYPERPARAMETERS.

METHOD	COST	HYPERPARAMETERS	OBSERVATIONS
Graph learning	Iterative algorithm, requires solving iteratively a quadratic program and a matrix $\mathbb{R}^{N \times N}$ inverse	α, β	Easy to tackle cluster-wise
Lap. Int	Matrix inversion or multiplication (depending on if $\|\mathcal{U}\| << \|\mathcal{M}\|$)	None	$\mathbf{L}_{\mathcal{U}\mathcal{U}}$ may be singular
GSP	Matrix inversion or multiplication (depending on if $K << \|\mathcal{M}\|$)	K	Needs $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathbf{T}}$ decomposition once: $\mathcal{O}(N^3)$
KRR-DIFF	Matrix inversion or multiplication (depending on if $\|\mathcal{M}\| << \|\mathcal{U}\|$)	μ, σ^2	Kernel matrix $\mathbf{K}$ only needs to be computed once: $\mathcal{O}(N^3)$
KRR-COV	Matrix inversion or multiplication (depending on if $\|\mathcal{M}\| << \|\mathcal{U}\|$)	μ	Kernel matrix $\mathbf{K}$ only needs to be computed once: $\mathcal{O}(N^3)$

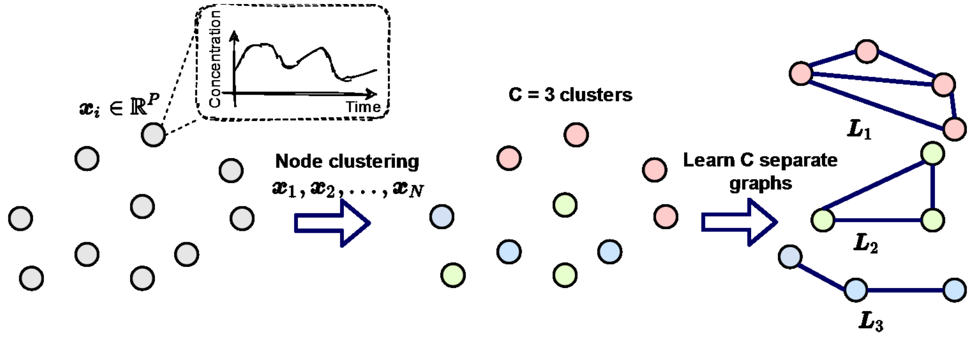
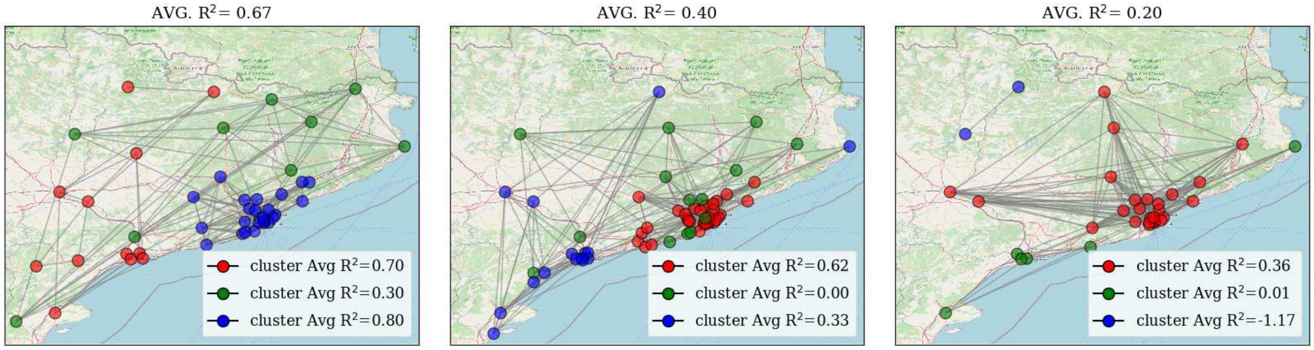


Fig. 5. Example of air pollution monitoring graph learning process when dealing with large graphs, where C=3 clusters. The problem can be split and solved for C disjoint subgraphs.

$\mathcal{O}(T^3)$, where T is the matrix dimension, but of different dimension. Thus, the cost comes from the dimension that dominates the operations (e.g. whether the number of unobserved nodes $|\mathcal{U}|$ is less than the observed ones $|\mathcal{M}|$). To speed up the reconstruction, if the set of nodes does not vary, the signal reconstruction coefficients can be calculated once and applied at different time instants t. However, if this set changes, which is a common situation where some of the nodes may have missing values and need to be reconstructed, the signal reconstruction coefficients must be recalculated. If cross-validation becomes extremely computationally expensive, a greedy approach can be carried out, where a graph with a desired level of sparsity is first obtained, and then cross-validation is performed only for the hyperparameters of the reconstruction method.

The most demanding task is graph learning since it involves iteratively solving a convex problem, and the complexity of this problem scales quadratically with the number of nodes N. However, as we are dealing with air pollution data, there exist spatial patterns and correlation between sensors, so the graph learning problem can be split to learn a number of disjoint subgraphs without having a large impact in the performance. Thus, we propose to find the approximation of a large graph as a set of disjoint subgraphs with a small impact on the signal reconstruction error. In a similar problem, Stein *et al.* [39] propose a cluster-based methodology to solve the Kriging scalability issues, learning a single Kriging model per cluster. Therefore, given the correlations present in our data, a clustering-based approach to partitioning the graph learning problem is a good strategy to cope with the scalability problem.

Given the spatial correlation of certain pollutants (e.g. ozone tropospheric and nitrogen dioxide) and the assumption that stations with different variability patterns should be weakly connected, we divide the reference stations into C clusters and learn C Laplacians independently, resulting in C disjoint subgraphs. Figure 5 shows the graph learning process for large graphs, where using the normalized time series of each of the nodes $\mathbf{x_i} \in \mathbb{R}^{\mathbf{P}}$, which can correspond to the training data, a clustering algorithm can be applied to the N nodes to find C clusters. The number of clusters can be decided in several ways, among them using unsupervised metrics such as the Calinski-Harabasz or the Silouhette index [40]. Depending on the specific data set the size of the clusters may vary, but the computational complexity improvement of the problem will always depend on the size of the largest cluster, the smaller the greater the improvement, as the subgraph learning can be solved in parallel. The performance of the solution will always depend on the localization of the pollutants and the dependencies present in the data. In addition, the resulting Laplacians can be treated independently or joined together in a Laplacian in the form of a block-diagonal matrix. Algorithm 1 in section V-A shows this cluster-wise procedure, where a different Laplacian $\mathbf{L}_i$ is learned for each cluster of nodes $\mathcal{C}_i$, with their corresponding α and β hyperparameters. Once the Laplacians are learned, the graph signal reconstruction $\mathbf{f}(\cdot)$ can be applied in a cluster-wise manner, since nodes from different clusters are not connected.



(a) O_3 data set. (b) NO_2 data set. (c) PM_{10} data set.

Fig. 6. Results for a clustering example, with C=3, applied to all three data sets using the Laplacian interpolation.

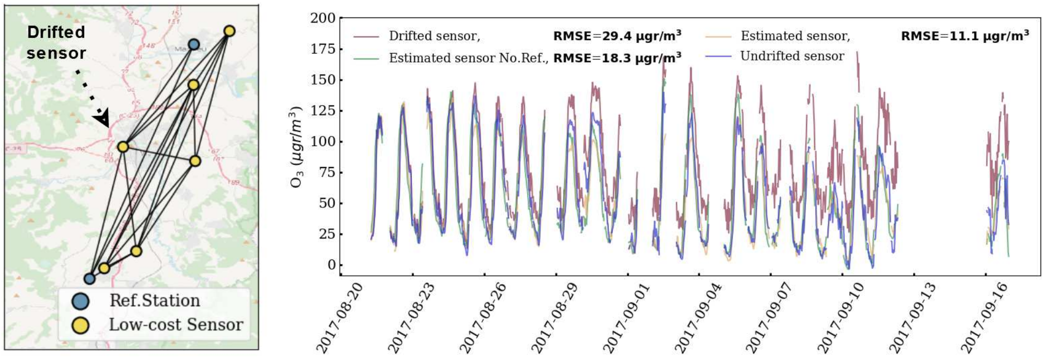
TABLE VI
COMPARISON OF THE AVERAGE R^2 USING THE WHOLE GRAPH AND USING CLUSTERS.

Data set	Original R^2	Cluster R^2	Problem size reduction
O_3	0.66	0.67	↓ 47.83%
NO_2	0.42	0.40	↓ 48.33%
PM_{10}	0.26	0.20	↓ 24.24%

Figure 6.a) shows the results for the first experiment (section V-B) learning three Laplacians (one per cluster) with the O_3 data set using the hierarchical clustering algorithm (colors denote the clusters obtained), which is a widely studied algorithm for clustering air pollution time series [40], using the euclidean distance and the ward linkage criterion. There is a cluster that has a very good average CV R^2 0.80, another that has a good average R^2 with 0.7, and another set of stations that cannot be reconstructed well (R^2 of 0.30). Looking further into the results, we see how the cluster that is best predicted (blue cluster) is the one with the highest density of stations in an area and the second one (red cluster) has a few stations nearby. If we compare the results with those shown in Figure 3, where we learned the graph with all the stations we see that the average CV R^2 is as good as the R^2 obtained learning the whole graph (0.66, see Table IV). Moreover, it can be observed how in the clusters obtained for each pollutant the clustering algorithm is able to group significantly the nodes that do not benefit from the neighborhood information. In the case of NO_2 and PM_{10}, Figures 6b) and c) respectively, we see how the R^2 has worsened a little from 0.42 to 0.40 and from 0.26 to 0.20 with respect to learning the whole graph. In the case of PM_{10}, it is observed that in one of the clusters R^2=-1.17, with a negative value, which means that the model does not follow the trend of the data, and therefore fits worse than if there was a null model. These clusters correspond to nodes that cannot be reconstructed given their location and lack of influential neighboring information. Table VI shows the average R^2 without using clustering and using clustering to improve the scalability. The coefficients of determination are mostly maintained or are slightly lower, so the scalability is improved at almost no signal reconstruction cost. The problem size reduction label indicates the reduction in the size of the resulting graph learning with respect to the original case, in the cluster-based learning case the size is defined by the size of the largest cluster. Thus, the problem size is reduced around a 24-48%, meaning that the largest cluster is of size 75-52% of the original graph, in almost all cases. As a summary, it is suggested that the problem of graph learning can be broken down into C problems with C clusters with the most similar reference stations, thus eliminating connections between subgraphs without a large impact on the error.

E. Application: Drift compensation

In this section, we show an example of signal reconstruction applied to a heterogeneous network, where a sensor drift is simulated by adding Gaussian noise of increasing magnitude. Signal reconstruction allows to maintain the quality of the drifting sensor by reconstructing its sensor values. We show how the compensation works when there are reference stations in the sensor neighborhood and when there are only other low-cost sensors. This small network containing eight nodes can be seen as a subgraph resulting from the split of a larger network using the clustering procedure shown in section V-D. Figure 7.a) shows the disposition of the nodes, the learned graph, the position of the drifted sensor and its six neighbors (two reference stations and four low-cost sensors). For illustration purposes, Laplacian interpolation has been used as a reconstruction method and the best graph has been learned performing cross-validation over the training set, 66% of the data, and in the test set a drift has been simulated in the target sensor. Figure 7.b) shows the error made by the sensor with drift with respect to what would have been its normal behavior, 29.4 μgr/m^3. It is also observed the reconstruction made using all its neighbors, including the two reference stations, 11.1 μgr/m^3, showing the benefits from having precise instrumentation nearby to correct possible seasonal changes. Finally, as an interesting example, it has been simulated the unavailability of the two reference stations. Even in this case, the graph signal reconstruction is able to reduce the RMSE from 29.4 to 18.3 μgr/m^3 only using neighboring low-cost sensors, indicating the possibility of correcting drifts using only neighboring low-cost sensors. Alternatively, performing the same experiment using the kernelized ridge regression with vertex-covariance kernel and a 28-edge graph learned with the



(a) 19-edge graph learned from the H2020 Captor data.

(b) Results for simulated drift compensation in two scenarios; using reference stations and low-cost sensors, and only using low-cost sensors.

Fig. 7. Example of a simulated drifted sensor corrected with a heterogeneous sensor network using the signal reconstruction framework.

graphical lasso, the signal reconstruction error obtained using both reference stations and low-cost sensors is 10.89 μgr/m^3 and the error using only low-cost sensors is 14.84 μgr/m^3.

VI. Conclusions

In this article, we have investigated the use of a signal reconstruction framework for air pollution monitoring data, where signal reconstruction techniques superimposed on a graph learned from the data can be used to maintain the data quality of this type of network. For this purpose, we have used O_3, NO_2 and PM_{10} data from reference stations in Catalonia, Spain, and we have compared three reconstruction techniques from fields such as Laplacian interpolation (graph-based semi-supervised learning), low-pass graph signal reconstruction (signal processing) and kernel ridge regression with diffusion kernel and vertex-covariance kernel (kernel methods).

Firstly, the performance of each method has been studied by reconstructing each of the reference stations present in the network, assuming only missings in the node to be reconstructed. In this case, the kernel ridge regression methods and the Laplacian interpolation have demonstrated their superiority by outperforming the other method in the three data sets. Furthermore, it has been observed that O_3 and NO_2 can be correctly predicted with these methods as opposed to PM_{10}. The reason is the higher correlation between reference station data for O_3 and NO_2, while PM_{10} data are more local and with lower correlation between stations. The kernel ridge regression with vertex-covariance kernel is the method that has worked best because it is the minimum local linear estimator, but the Laplacian interpolation and kernel ridge regression with diffusion kernel work better for sparse graphs. Secondly, we have studied the reconstruction with an increasing percentage of nodes to be estimated at the same time (e.g. nodes with missings) to simulate the semi-supervised learning paradigm. Again, the kernel methods and Laplacian interpolation outperform the other method and produce very good results when 0-40% of the nodes are missing.

Thirdly, we have studied the scalability of proposed signal reconstruction framework, where the most expensive task is graph learning. Thus, since the cost of learning the graph involves a quadratic program that scales quadratically in the number of nodes in the network, the use of a clustering technique is proposed to approximate the problem of learning and reconstructing a large graph in the learning and reconstruction of C smaller graphs. Finally, the signal reconstruction framework has been successfully applied to a heterogeneous network to compensate for a drifting low-cost sensor. As future work, it would be interesting to study these ideas with other pollutants and the automatic application of signal reconstruction to correct faulty sensors in a sensor network deployment. In addition, other clustering algorithms or other partitioning techniques can be applied to allow high scalability of signal reconstruction methods on IoT platforms.

References

[1] B. Maag, Z. Zhou, and L. Thiele, “A survey on sensor calibration in air pollution monitoring deployments,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4857–4870, Dec 2018.

[2] 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*, vol. 88, pp. 142–159, 2019.

[3] E. Esposito, S. De Vito, M. Salvato, G. Fattoruso, and G. Di Francia, “Computational intelligence for smart air quality monitors calibration,” in *International Conference on Computational Science and Its Applications*. Springer, 2017, pp. 443–454.

[4] N. Zimmerman, A. A. Presto, S. P. Kumar, J. Gu, A. Hauryliuk, E. S. Robinson, A. L. Robinson, and R. Subramanian, “A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring,” *Atmospheric Measurement Techniques*, vol. 11, no. 1, pp. 291–313, 2018.

[5] 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*, vol. 6, no. 6, pp. 9563–9571, Dec 2019.

[6] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide,” *Sensors and Actuators B: Chemical*, vol. 215, pp. 249–257, 2015.

[7] D. E. Williams, “Low cost sensor networks: How do we know the data are reliable?” *ACS sensors*, vol. 4, no. 10, pp. 2558–2565, 2019.

[8] A. Ripoll, M. Viana, M. Padrosa, X. Querol, A. Minutolo, K. M. Hou, J. M. Barcelo-Ordinas, and J. García-Vidal, “Testing the performance of sensors for ozone pollution monitoring in a citizen science approach,” *Science of the Total Environment*, vol. 651, pp. 1166–1179, 2019.

[9] S. De Vito, E. Esposito, N. Castell, P. Schneider, and A. Bartonova, “On the robustness of field calibration for smart air quality monitors,” *Sensors and Actuators B: Chemical*, vol. 310, p. 127869, 2020.

[10] G. Miskell, K. Alberti, B. Feenstra, G. S. Henshaw, V. Papapostolou, H. Patel, A. Polidori, J. A. Salmond, L. Weissert, and D. E. Williams, "Reliable data from low cost ozone sensors in a hierarchical network," *Atmospheric Environment*, vol. 214, p. 116870, 2019.

[11] P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal, "Graph learning techniques using structured data for iot air pollution monitoring platforms," *IEEE Internet of Things Journal*, 2021.

[12] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[13] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.

[14] 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*, vol. 20, no. 1, p. 182, 2020.

[15] M. A. Zaidan, L. Dada, M. A. Alghamdi, H. Al-Jeelani, H. Lihavainen, A. Hyvärinen, and T. Hussein, "Mutual information input selector and probabilistic machine learning utilisation for air pollution proxies," *Applied Sciences*, vol. 9, no. 20, p. 4475, 2019.

[16] J. M. Barcelo-Ordinas, P. Ferrer-Cid, J. Garcia-Vidal, A. Ripoll, and M. Viana, "Distributed multi-scale calibration of low-cost ozone sensors in wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2503, 2019.

[17] M. Matusowsky, D. T. Ramotsoela, and A. M. Abu-Mahfouz, "Data imputation in wireless sensor networks using a machine learning-based virtual sensor," *Journal of Sensor and Actuator Networks*, vol. 9, no. 2, p. 25, 2020.

[18] M. A. Zaidan, N. H. Motlagh, P. L. Fung, D. Lu, H. Timonen, J. Kuula, J. V. Niemi, S. Tarkoma, T. Petäjä, M. Kulmala *et al.*, "Intelligent calibration and virtual sensing for integrated low-cost air quality sensors," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 638–13 652, 2020.

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

[20] P. Ferrer-Cid, J. M. Barcelo-Ordinas, J. Garcia-Vidal, A. Ripoll, and M. Viana, "Multisensor data fusion calibration in iot air pollution platforms," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3124–3132, 2020.

[21] 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.

[22] K. Shukla, P. Kumar, G. S. Mann, and M. Khare, "Mapping spatial distribution of particulate matter using kriging and inverse distance weighting at supersites of megacity delhi," *Sustainable Cities and Society*, vol. 54, p. 101997, 2020.

[23] H.-R. Song, M. Fuentes, and S. Ghosh, "A comparative study of gaussian geostatistical models and gaussian markov random field models," *Journal of Multivariate analysis*, vol. 99, no. 8, pp. 1681–1697, 2008.

[24] G. B. Ribeiro and J. B. Lima, "Graph signal processing in a nutshell," *Journal of Communication and Information Systems*, vol. 33, no. 1, 2018.

[25] I. Jabłoński, "Graph signal processing in applications to sensor networks, smart grids, and smart cities," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7659–7666, 2017.

[26] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, 2016.

[27] B. Mijling, Q. Jiang, D. de Jonge, and S. Bocconi, "Field calibration of electrochemical no_2 sensors in a citizen science context," *Atmospheric Measurement Techniques*, vol. 11, no. 3, pp. 1297–1312, 2018.

[28] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *International Conference on Computational Learning Theory*. Springer, 2004, pp. 624–638.

[29] L. Stanković, E. Sejdić, S. Stanković, M. Daković, and I. Orović, "A tutorial on sparse signal reconstruction and its applications in signal processing," *Circuits, Systems, and Signal Processing*, vol. 38, no. 3, pp. 1206–1263, 2019.

[30] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information," in *Proceedings of the 2012 SIAM international Conference on Data mining*. SIAM, 2012, pp. 403–414.

[31] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[32] Y. Wu, D. Zhuang, A. Labbe, and L. Sun, "Inductive graph neural networks for spatiotemporal kriging," *arXiv preprint arXiv:2006.07527*, 2020.

[33] K. Qiu, X. Mao, X. Shen, X. Wang, T. Li, and Y. Gu, "Time-varying graph signal reconstruction," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 870–883, 2017.

[34] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, "Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting," in *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 2020, pp. 163–166.

[35] T. H. Do, E. Tsiligianni, X. Qin, J. Hofman, V. P. La Manna, W. Philips, and N. Deligiannis, "Graph-deep-learning-based inference of fine-grained air quality from mobile iot sensors," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8943–8955, 2020.

[36] V.-D. Le, "Spatiotemporal graph convolutional recurrent neural network model for citywide air pollution forecasting," *TechRxiv*, 2021.

[37] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[38] J. M. Barcelo-Ordinas, P. Ferrer-Cid, J. Garcia-Vidal, M. Viana, and A. Ripoll, "H2020 project captor dataset: Raw data collected by low-cost mox ozone sensors in a real air pollution monitoring network," *Data in Brief*, vol. 36, p. 107127, 2021.

[39] B. Van Stein, H. Wang, W. Kowalczyk, M. Emmerich, and T. Bäck, "Cluster-based kriging approximation algorithms for complexity reduction," *Applied Intelligence*, vol. 50, no. 3, pp. 778–791, 2020.

[40] P. Govender and V. Sivakumar, "Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)," *Atmospheric Pollution Research*, vol. 11, no. 1, pp. 40–56, 2020.

Pau Ferrer-Cid is a PhD student at the Statistical Analysis of Networks and Systems (SANS) research group, Universitat Politecnica de Catalunya (UPC). He holds a B.Sc in Computer Science and a M.Sc in Data Science by the UPC. His main research interests are the applications of novel data analysis methods to sensor data coming from IoT platforms and the analysis of other kinds of data from fields like biology and computer vision.

Jose M. Barcelo-Ordinas is an Associate Professor at Universitat Politecnica de Catalunya (UPC) from 1999. He holds a PhD and B.Sc+M.Sc in Telecommunication Engineering and a B.Sc+M.Sc in Mathematics. He has participated in many European projects such as WIDENS, EuroNGI, EuroNFI, EuroNF NoE and H2020 CAPTOR. His currently research areas are wireless sensor networks, mobility patterns, and the statistical analysis of sensor data.

Jorge Garcia-Vidal is since 2003, full professor at the Computer Architecture Department of UPC, and since 2012 responsible of the Smart Cities Initiative at Barcelona Supercomputing Center (BSC-CNS), coordinating the H2020 CAPTOR project or being the BSC-CNS responsible of the H2020 project ASGARD. His main current research interest is in problems related with the capture, processing and statistical analysis of sensor data.