



HAL
open science

Gas Sources Parameters Estimation Using Machine Learning in WSNs

Sandy Mahfouz, Farah Mourad-Chehade, Paul Honeine, Joumana Farah,
Hichem Snoussi

► **To cite this version:**

Sandy Mahfouz, Farah Mourad-Chehade, Paul Honeine, Joumana Farah, Hichem Snoussi. Gas Sources Parameters Estimation Using Machine Learning in WSNs. IEEE Sensors Journal, 2016, 16 (14), pp.5795 - 5804. 10.1109/JSEN.2016.2569559 . hal-01965049

HAL Id: hal-01965049

<https://hal.science/hal-01965049>

Submitted on 4 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gas Sources Parameters Estimation Using Machine Learning in WSNs

Sandy Mahfouz, Farah Mourad-Chehade, Paul Honeine, Joumana Farah, and Hichem Snoussi

Abstract—This paper introduces an original clusterized framework for the detection and estimation of the parameters of multiple gas sources in WSNs. The proposed method consists of defining a kernel-based detector that can detect gas releases within the network’s clusters using concentration measures collected regularly from the network. Then, we define two kernel-based models that accurately estimate the gas release parameters, such as the sources locations and their release rates, using the collected concentrations.

Index Terms—gas diffusion, machine learning, one-class classification, ridge regression, source parameter estimation

I. INTRODUCTION

Gas releases might occur either accidentally, such as for the Union Carbide release in Bhopal, India in 1984 [1], or deliberately, such as for the Sarin gas attack on Tokyo, Japan in 1995 [2]. Such explosions are potent threats to the environment and to human society, wherever they occur. Considering the potentially catastrophic consequences, it is important to detect the gas explosion and estimate its parameters, including the source’s location and the release rate. These computations require the collection of gas concentration measurements from the area of explosion, which necessitates specially trained people with appropriate protective equipments. Alternatively, wireless sensor networks (WSNs) have proven to be very useful in such scenarios, where human intervention is risky and expensive [3]. Typically, sensors are deployed in the area to be monitored and are regularly and continuously collecting measurements from the area. The collected information is reported back to a fusion center, where it is processed in order to estimate the source’s parameters.

Several methods have been proposed in the literature for the estimation of the source’s parameters. For instance, the authors of [4] developed an inverse model for inferring the parameters of an instantaneous point source from gas concentration measurements. The method solves a non-linear least squares estimation problem. In [3], the authors determined the source’s parameters using mobile robots that collect concentration measurements. The study was focused on the selection of the sequence of locations where each robot should be moved in order to obtain accurate real-time

estimates. Another method was proposed in [5], where the problem of pollutant source localization and flow estimation is addressed in a one-dimensional context, using a single remote sensor. The pollutant is assumed to be generated by one out of several possible sources, and the task is viewed as a conditional deconvolution which requires a priori knowledge. In the end, a joint estimation decision is derived in a Bayesian framework.

In terms of computational concept, algorithms in WSNs can be divided into centralized, distributed, and clusterized algorithms [6]. The centralized algorithms require the transmission of all measurements to a fusion center (e.g., a sink node) for processing. Such strategies often result in prohibitive wasteful energy and bandwidth consumptions and can thereby reduce the lifetime and utility of the network. In this paper, we propose a new clusterized framework for the detection and estimation of multiple gas sources in wireless sensor networks. Sensors are uniformly or randomly deployed in the region of interest and collect concentration measurements at regular and short sampling intervals. The region is divided into clusters, each having its own cluster head. Consequently, information processing is done locally. Compared to a centralized strategy, this clusterized approach is more robust to failures, since several cluster heads are engaged in the detection phase. It is also less energy consuming and thus more adapted to WSNs limitations.

The proposed framework consists of two phases: the detection phase and the estimation phase. In the first one, a kernel-based detector is defined using the Support Vector Data Description (SVDD) [7]. The detector is used by each cluster head in order to identify any gas release in the specified cluster. When an anomaly is spotted, the concentration vector that first triggered the alert is treated to estimate the gas release parameters, such as the source’s location and the release rate. Note that not all the concentrations will be processed in the estimation phase, since some are irrelevant for parameters estimation. Details about the selection of the useful information will be given in the sequel. A first estimate of the parameters is obtained using a non linear model, which is also defined within the framework of kernel methods, proved to be successful for solving nonlinear regression problems [8], [9], [10]. Then, part of the estimated parameters are processed as internal feedback, along with the measured concentrations, in order to provide a more accurate estimate of the source location. Simulations show that the proposed estimation method yields accurate results in the case of a single source, as well as in the case of multiple sources.

S. Mahfouz, F. Mourad-Chehade, P. Honeine and H. Snoussi are with the Institut Charles Delaunay (CNRS), Université de Technologie de Troyes, France, e-mail: {sandy.mahfouz, farah.mourad, paul.honeine, hichem.snoussi}@utt.fr

J. Farah is with the Faculty of Engineering of the Lebanese University, Roumieh, Lebanon, e-mail: joumanafarah@ul.edu.lb

This work is supported by the Champagne-Ardenne Region in France, grant WiDiD: Wireless Diffusion Detection.

The rest of the paper is organized as follows. Section II introduces the considered advection-diffusion model. Section III provides a thorough description of the detection phase in the proposed framework. In Section IV, the estimation phase is described, along with details about the definition of the two kernel-based regression models. Next, in Section VI, the effectiveness of the method is discussed for different scenarios, and a comparison to other methods is provided. Finally, Section VII concludes the paper.

II. THE ADVECTION-DIFFUSION MODEL

This section describes the advection-diffusion model considered in this paper for the generation of the concentrations. Note that this model is a generalized version of the one developed in [4]. Now consider an instantaneous gas release of Q kg, assumed to occur at time t_0 at location (x_0, y_0, z_0) . A wind with mean velocity $\mathbf{U} = (u_x, u_y, 0)$ spreads the gas particle in the region. The mass concentration C of the released agent, at an arbitrary location (x, y, z) and at time t , is governed by the equation of mass conservation given by the following:

$$\frac{\partial C}{\partial t} = -\nabla \mathbf{q}, \quad (1)$$

where ∇ is the gradient operator, and \mathbf{q} is the pollutant mass flux per unit area. The flux \mathbf{q} is given by:

$$\mathbf{q} = C \mathbf{U} - \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} \otimes \nabla C, \quad (2)$$

where $C \mathbf{U}$ is the mean mass advection by the wind, \otimes is the tensor product, and K_x, K_y, K_z are eddy diffusivities in the X, Y and Z directions respectively. Equation (2) can then be written as follows:

$$\mathbf{q} = \left(C u_x - K_x \frac{\partial C}{\partial X}, C u_y - K_y \frac{\partial C}{\partial Y}, -K_z \frac{\partial C}{\partial Y} \right). \quad (3)$$

By substituting (3) into (1), we get an equation that can be solved subject to two boundary conditions [3], [4]. The first condition results from the fact that the concentration is zero at infinity in all spatial directions, and the second condition is that the gas is not absorbed by the ground. To simplify the model, the velocity of the wind \mathbf{U} , as well as the eddy diffusivities K_x, K_y, K_z , are assumed to be constant. Following these assumptions, we get the following solution:

$$\begin{aligned} C(x, y, z, t) = & \frac{Q}{8 \pi^{\frac{3}{2}} (K_x K_y K_z)^{\frac{1}{2}} \Delta t^{\frac{3}{2}}} \\ & \times \exp \left(-\frac{(\Delta x - u_x \Delta t)^2}{4K_x \Delta t} - \frac{(\Delta y - u_y \Delta t)^2}{4K_y \Delta t} \right) \\ & \times \left(\exp \left(-\frac{\Delta z^2}{4K_z \Delta t} \right) + \exp \left(-\frac{\Delta z'^2}{4K_z \Delta t} \right) \right), \end{aligned} \quad (4)$$

where $\Delta x = x - x_0$, $\Delta y = y - y_0$, $\Delta z = z - z_0$, $\Delta z' = z + z_0$ and $\Delta t = t - t_0$.

In the following, for the sake of simplicity, we assume that all measurements are taken at ground level. We also

suppose that the gas release occurs at ground level, which means that the gas source is at location $(x_0, y_0, z_0) = (x_0, y_0, 0)$. Therefore, using (4), the concentration measured at a location $(x, y, 0)$ is given by the following:

$$\begin{aligned} C(x, y, 0, t) = & \frac{Q}{4 \pi^{\frac{3}{2}} (K_x K_y K_z)^{\frac{1}{2}} \Delta t^{\frac{3}{2}}} \\ & \times \exp \left(-\frac{(\Delta x - u_x \Delta t)^2}{4K_x \Delta t} - \frac{(\Delta y - u_y \Delta t)^2}{4K_y \Delta t} \right). \end{aligned} \quad (5)$$

Note that the wind velocity \mathbf{U} could be provided by an anemometer, and is therefore treated as a known constant [3]. The other parameters of the model are unknown and need to be estimated when a gas explosion occurs over the area under scrutiny. Hence, having gas concentrations measured over a certain area using a WSN, the aim of this paper is to detect the gas explosion when it occurs. Then, the objective is to estimate the location $(x_0, y_0, 0)$ of the explosion, the gas release mass Q , and the eddy diffusivities K_x, K_y and K_z , with K_x and K_y assumed to be equal [4].

III. THE DETECTION PHASE

As we already stated, the proposed method uses a WSN to continuously measure gas concentrations over a certain area Ω . Then, computations consist of two phases: the detection phase and the estimation of the source's parameters phase. In this section, we first give a brief description of the network's setup and of the detection phase. Then, we define a classifier that is capable of optimally separating the data into normal and abnormal data, where the abnormality denotes the occurrence of a gas diffusion.

A. Description of the detection phase

Consider N sensors deployed in the area Ω to be monitored, at fixed locations $(x_n, y_n, z_n) = (x_n, y_n, 0)$, $n \in \{1, \dots, N\}$. These sensors measure, at each time t , the gas concentrations at their locations. Let $(C(x_1, y_1, z_1, t) \dots C(x_N, y_N, z_N, t))^T$ be the $N \times 1$ vector of the gas concentrations measured at time t by the N sensors. These concentrations are assumed to follow the advection-diffusion model, detailed in Section II. In order to thoroughly monitor the area of interest, the concentrations are measured at regular and short sampling intervals by the deployed sensors.

Assume now that an instantaneous gas release of Q kg occurs at time t_0 at location $(x_0, y_0, z_0) = (x_0, y_0, 0)$, all these parameters being unknown. The gas particles are then spread by a wind with a mean known velocity $\mathbf{U} = (u_x, u_y, 0)$. Since a gas is released in the area, an alarm should be triggered based on the vector of measured concentrations. To this end, we need to develop a detector, capable of determining whether the concentrations are normal or not.

We propose to partition the area into Z distinct clusters, in order to make the computations less complex and the method more robust to transmission impairments and network failures. Each cluster is managed by a cluster head, that is a smart central processing unit (CPU), responsible

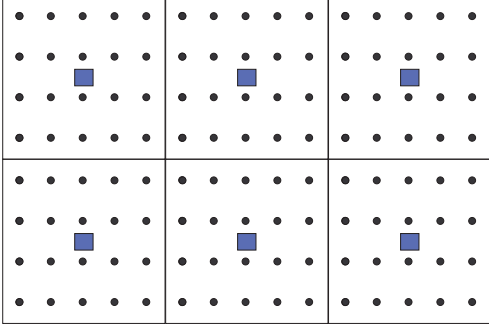


Fig. 1: Region topology- ■ represents the cluster heads and ● represents the deployed sensors.

of handling (gathering and synchronizing) data, performing calculations, and exchanging information with the sensors. Note that this device could also be one of the sensors of the network, and it can be placed anywhere in its specified cluster. In addition, all cluster heads can communicate with each other, and clusters could have any shape or dimension. Without loss of generality, we consider here that clusters are rectangular, with the cluster heads located at their centers, as illustrated in Fig. 1. Now that the network has been configured, each cluster head receives the measured concentrations from the sensors in its cluster at every time t . Let n_z denote the number of sensors in cluster z , $z \in \{1, \dots, Z\}$, and let $\mathbf{C}^{(z)}(t)$ denote the concentration vector of size $n_z \times 1$ collected by the cluster head of cluster z at time t . The received data are then processed instantly to detect whether a gas diffusion has occurred or not.

To achieve this, we aim at finding a detector that can detect an anomaly in the monitored region; in other words, it can detect whether or not there is one or more gas diffusions. A one-class classifier is able to do this task. Indeed, the Support Vector Data Description (SVDD) has been introduced to address the problem of anomaly detection [11], [7].

B. Definition of the detector using SVDD

The objective here is to define the class boundary, given data that are originated from a single class, i.e., the normal data, but are possibly contaminated with a small number of outliers, i.e., the abnormal data. For simplicity, the number of sensors in each cluster is assumed to be the same, so that the same defined detector could be used for all the clusters, as will be explained in the experimental results. As we already explained, a one-class classifier can be used for the detection, such as the Support Vector Data Description (SVDD) [12], [13], [7]. It essentially fits the smallest possible sphere around the given normal data, allowing some samples to be excluded as outliers. Therefore, a spherically shaped decision boundary with minimum radius is computed to enclose most of the training data. Data lying outside this decision boundary are considered as abnormal, i.e., outliers.

Let ϕ denote a function that maps the data from the

input space \mathbb{R}^{n_z} , where the concentrations lie, into a higher dimensional feature space \mathcal{H} . Consider a reproducing kernel $\kappa : \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \mapsto \mathbb{R}$, with \mathcal{H} its reproducing kernel Hilbert space (RKHS) with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. We then have $\kappa(\mathbf{C}_i, \mathbf{C}_j) = \langle \phi(\mathbf{C}_i), \phi(\mathbf{C}_j) \rangle$. Now consider a training set \mathbf{C}_i , $i \in \{1, \dots, N_{\text{det}}\}$, N_{det} being the size of the training set for the detection phase. The SVDD consists of estimating the hypersphere with minimum radius that encloses all data $\phi(\mathbf{C}_i)$ in the feature space \mathcal{H} . Let \mathbf{a} be the center of the hypersphere, and $R > 0$ its radius. To allow a better description of data, we allow the presence of outliers in the training set by introducing the slack variables $\xi_i \geq 0$. Consequently, abnormal concentration vectors are also used in the training set. Mathematically, the values for the slack variables are obtained by minimizing a cost function that balances the volume of the hypersphere against the penalty associated with outliers. Note that minimizing the hypersphere's volume is equivalent to minimizing R^2 . Therefore, we get the following constrained optimization problem:

$$\min_{\mathbf{a}, R, \xi_i} R^2 + \frac{1}{\nu N_{\text{det}}} \sum_{i=1}^{N_{\text{det}}} \xi_i \quad (6)$$

subject to

$$\|\phi(\mathbf{C}_i) - \mathbf{a}\|_{\mathcal{H}}^2 \leq R^2 + \xi_i \quad \text{and} \quad \xi_i \geq 0 \quad \forall i = 1, \dots, N_{\text{det}}. \quad (7)$$

The quantity ν is a predefined parameter that regulates the trade-off between the volume of the hypersphere and the number of outliers.

Let L denote the Lagrangian of the above constrained optimization problem. By taking the partial derivatives of L with respect to R , \mathbf{a} and ξ_i , we get the following relations:

$$\sum_{i=1}^{N_{\text{det}}} \alpha_i = 1, \quad \mathbf{a} = \sum_{i=1}^{N_{\text{det}}} \alpha_i \phi(\mathbf{x}_i), \quad \text{and} \quad 0 \leq \alpha_i \leq \frac{1}{\nu N_{\text{det}}},$$

where α_i are the Lagrangian multipliers. Incorporating these relations into the Lagrangian L gives us the following objective functional to be maximized with respect to α_i :

$$L = \sum_{i=1}^{N_{\text{det}}} \alpha_i \kappa(\mathbf{C}_i, \mathbf{C}_i) - \sum_{i=1}^{N_{\text{det}}} \sum_{j=1}^{N_{\text{det}}} \alpha_i \alpha_j \kappa(\mathbf{C}_i, \mathbf{C}_j), \quad (8)$$

subject to $0 \leq \alpha_i \leq \frac{1}{\nu N_{\text{det}}}$. This is a quadratic programming problem, whose solution is found using an off-the-shelf optimization technique. For instance, one can use the Matlab function *quadprog* to solve such a problem, and consequently compute the α_i .

As for the radius of the optimal hypersphere, it is, by definition, the distance from the center \mathbf{a} to any sample $\phi(\mathbf{C}_k)$ on the boundary in the feature space \mathcal{H} . Therefore, the radius is given by the following:

$$R^2 = \kappa(\mathbf{C}_k, \mathbf{C}_k) - 2 \sum_{i=1}^{N_{\text{det}}} \alpha_i \kappa(\mathbf{C}_k, \mathbf{C}_i) + \sum_{i=1}^{N_{\text{det}}} \sum_{j=1}^{N_{\text{det}}} \alpha_i \alpha_j \kappa(\mathbf{C}_i, \mathbf{C}_j).$$

Sensors record the concentrations at regular short intervals T . Therefore, each T seconds, a new concentration vector $\mathbf{C}^{(z)}(t_d)$ is taken from each cluster z , with $t_d = d \times T$, $d \in \mathbb{N}$. The decision rule for each new concentration vector $\mathbf{C}^{(z)}(t_d)$ is obtained by evaluating the distance between the center \mathbf{a} and the mapping $\phi(\mathbf{C}^{(z)}(t_d))$ in the feature space, given by $\|\phi(\mathbf{C}^{(z)}(t_d)) - \mathbf{a}\|_{\mathcal{H}}^2$. Note that to compute this distance, one does not need the exact expressions of ϕ and \mathbf{a} . Instead, by developing the distance expression, one gets scalar products of ϕ functions, that could be replaced by kernels. The new concentration $\mathbf{C}^{(z)}(t_d)$ is considered as normal if the distance calculated is smaller than the radius, i.e., $\|\phi(\mathbf{C}^{(z)}(t_d)) - \mathbf{a}\|_{\mathcal{H}}^2 \leq R^2$. Otherwise, an alert is triggered in the considered cluster. According to [14, Section 7.1], we can provide an upperbound on the false alert rate, that is the probability that a normal concentration vector is mistakenly classified as abnormal.

The proposed algorithm can detect and estimate more than one source. Therefore, more than one detector might detect a gas release at the same time. However, for simplicity, all the diffusions are assumed not to overlap. It is worth noting here that by dividing the whole area into several clusters, the computations are processed in parallel at all the cluster heads. Moreover, the size of the considered data in the classifier for training and detection is limited to n_z , instead of taking all the sensors' concentrations at once. This way, the network could be as large as needed without increasing the computations' complexity.

IV. THE ESTIMATION PHASE

Now that our detector is defined, we are able to detect a gas diffusion in a specified cluster. Once an alert is triggered in a cluster z , we proceed to the processing of the concentration vector $\mathbf{C}^{(z)}(t)$ recorded by the sensors in the specified cluster. In this section, we give a brief description of the estimation phase, and we emphasize on the choice of the concentrations to be used in the source's parameters estimation. The idea here consists of grouping the clusters in alert in a way to discern all the sources of explosion. One can take all the clusters in alert together; however, if the network is too large, with many sources of explosion, one gets high dimension data to process. An alternative way consists of grouping adjacent clusters in alert together; by adjacent, we mean the clusters that share a common boundary. Each cluster in alert but not adjacent to others will form a group on its own. To illustrate this, see Fig. 2, where six explosions are assumed to occur in the region of interest.

At the end of this step, several groups with different sizes are generated, each covering one or more explosion sources. The concentration vectors recorded by the sensors of a group are then communicated to a cluster head of the group, e.g., the one having the highest computation capabilities, and the concentrations are concatenated in a single group vector. Now, in order to detect the multiple gas releases per group, the proposed method uses the concentration group

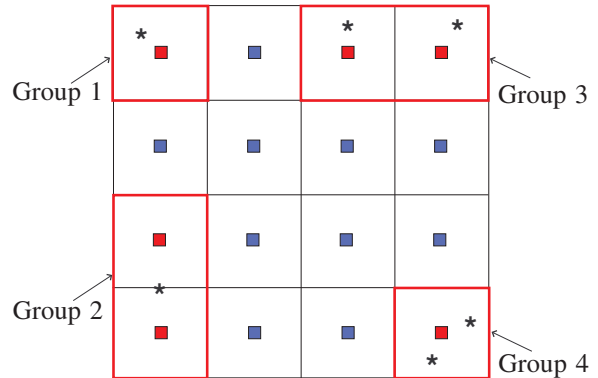


Fig. 2: Region on interest in the case of multiple gas releases- \blacksquare represents the cluster heads when there is no alert, \blacksquare when there is an alert, $*$ represents the locations of explosions, and the red rectangle represents the defined groups. Grouping the clusters in alert according to their adjacencies leads here to four groups.

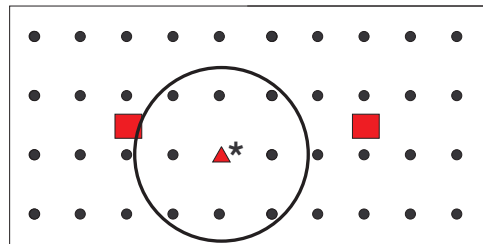


Fig. 3: Group topology- \blacksquare represents the cluster heads, \bullet represents the deployed sensors, $*$ represents the location of the explosion, and \blacktriangle represents a local maximum.

vectors to compute local maxima, leading ideally to the concentrations of the closest sensors to the gas sources. Let N_s be the total number of local maxima detected at all groups. This means that N_s sources have exploded at time t , or have exploded at different times, but are being detected and estimated simultaneously, since the maximal concentrations would be at the release source. Consider now the example of Fig. 3, where the source of explosion is near the boundary of two clusters. The alert is then triggered in both clusters, the gas being spread all around the source. Since the clusters are adjacent, they are grouped together in order to form one group. Then, the local maximum is estimated, leading to only one sensor, instead of two if the clusters were considered separately. This means that by grouping the clusters, one would identify the exact number of exploded sources, even if the gas is spread over several clusters.

Having detected the local concentrations maxima, we now need to process the information in each group in order to estimate all N_s sources' parameters. It is important to mention that not all concentration information have the same importance in the algorithm. Consequently, only the information collected around the local maxima will be treated. This solution relies on the fact that the highest con-

concentrations are collected by the sensors around the source of explosion, and these concentration measures are the most relevant for parameters' estimation. If we reconsider Fig. 3, only the concentration information from the sensors around the red triangle denoting the maximum will be considered. Note that the size of the clusters and the number of sensors in the figure are only chosen for illustrative purposes.

In this paragraph, we propose to choose the concentrations to be treated based on the advection-diffusion model of Section II. In other words, we need to estimate the size of the small local zone around the local maximum as illustrated in Fig. 3. Let C_{thres} be a threshold under which the concentrations are considered really small, which means that they are measured by sensors far from the source. Depending on the considered application and the type of monitoring needed, one can determine an approximation of the maximum release mass Q_{max} . Using C_{thres} and Q_{max} , we can find the maximal distance between a source and a sensor. Then, this distance is used to determine the boundaries of the local zone, whose center is the local maximum and the estimated maximal distance is the distance separating the local maximum from the boundary of the zone. By taking the logarithm of (5), we get the following:

$$\log C_{\text{thres}} = \log \left(\frac{Q_{\text{max}}}{4 \pi^{\frac{3}{2}} (K_x K_y K_z)^{\frac{1}{2}}} \right) - \frac{3}{2} \log \Delta t - \frac{\Delta x^2 + u^2 \Delta t^2 - 2u \Delta t \Delta x}{4K_x \Delta t} + \frac{\Delta y^2}{4K_y \Delta t}$$

For simplicity, assume now that the local zone is circularly shaped, which leads to $\Delta x = \Delta y$. In addition, having $K_x = K_y$ yields the following:

$$\log C_{\text{thres}} = \log \left(\frac{Q_{\text{max}}}{4 \pi^{\frac{3}{2}} (K_x^2 K_z)^{\frac{1}{2}}} \right) - \frac{3}{2} \log \Delta t - \frac{2\Delta x^2}{4K_x \Delta t} - \frac{u^2 \Delta t}{4K_x} + \frac{2u \Delta x}{4K_x}.$$

Finally, a second degree polynomial in terms of Δx is obtained as follows:

$$a \Delta x^2 + b \Delta x + c = 0, \quad (9)$$

where

$$\begin{cases} a &= -\frac{2}{4K_x \Delta t}, \\ b &= \frac{2u}{4K_x}, \\ c &= \log \left(\frac{Q_{\text{max}}}{4 \pi^{\frac{3}{2}} (K_x^2 K_z)^{\frac{1}{2}}} \right) - \frac{3}{2} \log \Delta t - \frac{u^2 \Delta t}{4K_x} - \log C_{\text{thres}}. \end{cases}$$

The local zone dimensions $\Delta x = \Delta y$ are then obtained by resolving the quadratic equation (9), using Q_{max} , C_{thres} , and initial approximative values of K_x and K_z . Then, only the concentrations measured by the sensors in that zone will be considered for the source parameters estimation; in other words, the considered concentrations are the ones measured by the sensors at a maximal distance Δx from the local maximum. For simplicity, we assume that we have the same Q_{max} and C_{thres} for all the groups, thus for all N_s local maxima. Therefore, the same local zone size will be

considered for all N_s explosions. Nevertheless, depending on the type of application, one can define different sizes for the local zones around the N_s maxima.

Now let $\mathbf{c}^{(s)}(t)$, $s \in \{1, \dots, N_s\}$, denote the vector of useful concentrations collected at time t around each local maximum s , i.e., around each source s . For the sake of clarity, we drop the time t in the following, since only the concentrations around the detection time are used. The first objective is to define a model ψ_A that takes as input the concentration vector $\mathbf{c}^{(s)}$ and yields as output the vector of parameters $\boldsymbol{\theta}^{(s)}$ of source s , which includes the gas release mass $Q^{(s)}$, the source location $(x_0^{(s)}, y_0^{(s)}, z_0^{(s)})$ and the diffusivity constants. Then, the next objective is to define a second model ψ_B that provides an enhanced estimation of the source location. Kernel methods [15] provide an elegant framework to define both models, as it will be shown in the following section. Note that estimations are handled by the cluster heads that are the closest to the local maxima sensors.

V. DEFINITION OF THE MODELS ψ_A AND ψ_B

Let V be the number of sensors in the local zones. As we mentioned earlier, only the concentrations around the local maxima will be considered in the estimation phase. Therefore, the vector $\mathbf{c}^{(s)}$ of size $V \times 1$ is the vector that will be processed to find the parameters of source s . In the following paragraph, we define the model ψ_A , that takes as input the concentration vector $\mathbf{c}^{(s)}$ and yields as output the source's parameters $\boldsymbol{\theta}^{(s)}$, that are the gas release mass $Q^{(s)}$, the source location $(x_0^{(s)}, y_0^{(s)}, 0)$ and the diffusivity constants. Then, we define a new model ψ_B that provides an enhanced location estimation of the source, using a part of the estimated source's parameters and the measured concentration vector $\mathbf{c}^{(s)}$.

A. First estimation

In this subsection, we define a model $\psi_A: \mathbb{R}^V \mapsto \mathbb{R}^5$, that associates to each concentration vector the corresponding source's parameters. We propose to find the model ψ_A by solving a nonlinear regression problem. The main benefit of such an approach is that no prior knowledge of the model is needed. Kernel methods [15] have been remarkably successful for solving nonlinear regression problems. More specifically, we consider the kernel ridge regression [16] to determine ψ_A , by combining five separate optimization problems, one for each component of the output vector.

Consider the following training set $(\mathbf{c}_\ell, \boldsymbol{\theta}_\ell)$, $\ell \in \{1, \dots, N_{\text{reg}}\}$, where N_{reg} is the size of the training set used for the regression phase and $\boldsymbol{\theta}_\ell = (Q_\ell \ K_x \ K_z \ x_{0\ell} \ y_{0\ell})$. The vector \mathbf{c}_ℓ yields the concentrations recorded by V sensors uniformly distributed in a zone of the same size as the small local zone introduced previously, around the time of a gas release of parameters $\boldsymbol{\theta}_\ell$, with $(x_{0\ell}, y_{0\ell}, 0)$ being the source's location and Q_ℓ the gas release mass. As already mentioned, the eddy diffusivities K_x and K_y , with $K_x = K_y$, are assumed constant, since they depend on the atmospheric conditions and the type of chemical agent,

considered constant as well. Note here that the advantage of using a local small zone to select the relevant concentrations allows us to save computations and reduce the training complexity, since V sensors are now considered instead of N . In the following, let $\Theta = (\theta_1^\top \dots \theta_{N_{\text{reg}}}^\top)^\top$. The matrix Θ is then of size $N_{\text{reg}} \times 5$, having $\Theta_{\ell,i}$ for the (ℓ, i) -th entry, with $\ell \in \{1, \dots, N_{\text{reg}}\}$ and $i \in \{1, \dots, 5\}$. We also denote θ_ℓ by $\Theta_{\ell,*}$, and the i -th column of Θ by $\Theta_{*,i}$.

Let $\psi_A = (\psi_{A1} \dots \psi_{A5})$, where ψ_{Ai} , $i \in \{1, \dots, 5\}$ estimates $\Theta_{\ell,i}$, the i -th component of the vector $\Theta_{\ell,*}$, for an input \mathbf{c}_ℓ . Each function ψ_{Ai} is then determined by minimizing the mean quadratic error between the model's outputs $\psi_{Ai}(\mathbf{c}_\ell)$ and the desired outputs $\Theta_{\ell,i}$:

$$\min_{\psi_{Ai} \in \mathcal{H}_A} \frac{1}{N_{\text{reg}}} \sum_{\ell=1}^{N_{\text{reg}}} (\Theta_{\ell,i} - \psi_{Ai}(\mathbf{c}_\ell))^2 + \eta_A \|\psi_{Ai}\|_{\mathcal{H}_A}^2, \quad (10)$$

where η_A is a regularization parameter that controls the tradeoff between the training error and the complexity of the solution. According to the representer theorem [15], the optimal function ψ_{Ai} can be written as follows:

$$\psi_{Ai}(\cdot) = \sum_{\ell=1}^{N_{\text{reg}}} \beta_{\ell,i} \kappa_A(\mathbf{c}_\ell, \cdot), \quad (11)$$

where “ \cdot ” is the function's input, $\kappa_A : \mathbb{R}^V \times \mathbb{R}^V \mapsto \mathbb{R}$ is a reproducing kernel, and $\beta_{\ell,i}$, $\ell \in \{1, \dots, N_{\text{reg}}\}$, are parameters to be determined. We now denote by β the $N_{\text{reg}} \times 5$ matrix whose (ℓ, i) -th entry is $\beta_{\ell,i}$. The vector $\beta_{*,i}$ denotes then its i -th column, and $\beta_{\ell,*}$ its ℓ -th row. By injecting (11) in (10), the dual optimization problem in terms of $\beta_{*,i}$ is obtained, whose solution is given by taking the derivative of the corresponding cost function with respect to $\beta_{*,i}$ and setting it to zero. We then obtain the following:

$$\beta_{*,i} = (\mathbf{K}_A + \eta_A N_{\text{reg}} \mathbf{I})^{-1} \Theta_{*,i},$$

where \mathbf{I} is the $N_{\text{reg}} \times N_{\text{reg}}$ identity matrix, and \mathbf{K}_A is the $N_{\text{reg}} \times N_{\text{reg}}$ Gram matrix whose (v, w) -th entry is $\kappa_A(\mathbf{c}_v, \mathbf{c}_w)$, for $v, w \in \{1, \dots, N_{\text{reg}}\}$. For an appropriate value of the regularization parameter η_A , the matrix between parenthesis is always non-singular.

Since the same matrix $(\mathbf{K}_A + \eta_A N_{\text{reg}} \mathbf{I})$ needs to be inverted in order to estimate each source's parameter, all five estimations can be collected into a single matrix inversion problem, thus reducing the computational complexity, as follows:

$$\beta = (\mathbf{K}_A + \eta_A N_{\text{reg}} \mathbf{I})^{-1} \Theta. \quad (12)$$

Finally, using equation (11) and the definition of the vector of functions $\psi_A(\cdot)$, we can write ψ_A as follows:

$$\psi_A(\cdot) = \sum_{\ell=1}^{N_{\text{reg}}} \beta_{\ell,*} \kappa_A(\mathbf{C}_\ell, \cdot). \quad (13)$$

Having detected a local maximum in the region, with its relevant concentration vector $\mathbf{c}^{(s)}$, one could then obtain a first estimate of the source s parameters as follows:

$$(\hat{Q} \ \hat{K}_x \ \hat{K}_z \ \hat{x}_0 \ \hat{y}_0) = \psi_A(\mathbf{c}^{(s)}). \quad (14)$$

B. Enhancement of the estimates

In this subsection, we introduce the second model ψ_B , that takes as input the measured concentrations, the estimated mass release and the estimated eddy diffusivities, and gives as output an enhanced estimate of the location of the source. Using the already estimated information as internal feedback is expected to provide an improved accuracy of the first location estimates of the sources.

The definition of the model ψ_B is also done using the ridge regression [16]. However, the training set is defined differently, such that the training input is given by $\mathbf{W}_\ell = (\mathbf{c}_\ell^\top \ Q_\ell \ K_x \ K_z)^\top$ and the training output is the source's location $(x_{0\ell} \ y_{0\ell})$, $\ell \in \{1, \dots, N_{\text{reg}}\}$. Let $\psi_B = (\psi_{B1} \ \psi_{B2})$, where ψ_{B1} and ψ_{B2} estimate $x_{0\ell}$ and $y_{0\ell}$ respectively. Let \mathbf{X}_0 denote the $N_{\text{reg}} \times 1$ vector whose ℓ -th entry is given by $x_{0\ell}$, and \mathbf{Y}_0 denote the $N_{\text{reg}} \times 1$ vector whose ℓ -th entry is given by $y_{0\ell}$.

Here as well, the models ψ_{B1} and ψ_{B2} are obtained by minimizing the mean quadratic errors between the estimated outputs and the desired ones, as follows:

$$\begin{cases} \min_{\psi_{B1} \in \mathcal{H}_B} \frac{1}{N_{\text{reg}}} \sum_{\ell=1}^{N_{\text{reg}}} (x_{0\ell} - \psi_{B1}(\mathbf{W}_\ell))^2 + \eta_B \|\psi_{B1}\|_{\mathcal{H}_B}^2 \\ \min_{\psi_{B2} \in \mathcal{H}_B} \frac{1}{N_{\text{reg}}} \sum_{\ell=1}^{N_{\text{reg}}} (y_{0\ell} - \psi_{B2}(\mathbf{W}_\ell))^2 + \eta_B \|\psi_{B2}\|_{\mathcal{H}_B}^2 \end{cases}$$

The quantity η_B is also a regularization parameter. In analogy with the previous paragraph, the optimal solutions can be written as follows:

$$\psi_{Bj}(\cdot) = \sum_{\ell=1}^{N_{\text{reg}}} \gamma_{\ell,j} \kappa_B(\mathbf{W}_\ell, \cdot),$$

where $\kappa_B : \mathbb{R}^{V+3} \times \mathbb{R}^{V+3} \mapsto \mathbb{R}$ is a reproducing kernel, and $\gamma_{\ell,j}$, $\ell \in \{1, \dots, N_{\text{reg}}\}$ and $j \in \{1, 2\}$, are the unknown parameters to be determined. Let γ denote the $N_{\text{reg}} \times 2$ matrix whose (ℓ, j) -th entry is $\gamma_{\ell,j}$, and $\gamma_{\ell,*}$ its ℓ -th row. Following the same line of reasoning as in the previous paragraph, one can collect the two estimations into a single one. Then, the unknown parameters $\gamma_{\ell,j}$ are estimated at once, as follows:

$$\gamma = (\mathbf{K}_B + \eta_B N_{\text{reg}} \mathbf{I})^{-1} (\mathbf{X}_0 \ \mathbf{Y}_0), \quad (15)$$

where \mathbf{K}_B is the $N_{\text{reg}} \times N_{\text{reg}}$ matrix whose (v, w) -th entry is $\kappa_B(\mathbf{W}_v, \mathbf{W}_w)$, for $v, w \in \{1, \dots, N_{\text{reg}}\}$. Finally, one is able to write ψ_B as follows:

$$\psi_B(\cdot) = \sum_{\ell=1}^{N_{\text{reg}}} \gamma_{\ell,*} \kappa_B(\mathbf{W}_\ell, \cdot). \quad (16)$$

After determining the first estimates of the parameters of the source s given by equation (14), one can set the $(V+3) \times 1$ vector $\mathbf{W} = (\mathbf{c}^{(s)\top} \ \hat{Q} \ \hat{K}_x \ \hat{K}_z)^\top$. Using the model ψ_B , a new estimate of the source's location is obtained as follows:

$$(\hat{x}_{0\text{enh}} \ \hat{y}_{0\text{enh}}) = \psi_B(\mathbf{W}), \quad (17)$$

where $(\hat{x}_{0\text{enh}} \ \hat{y}_{0\text{enh}})$ is the new enhanced location estimate of the source.

It is important to note that we are only using the first position estimate to compare its accuracy towards

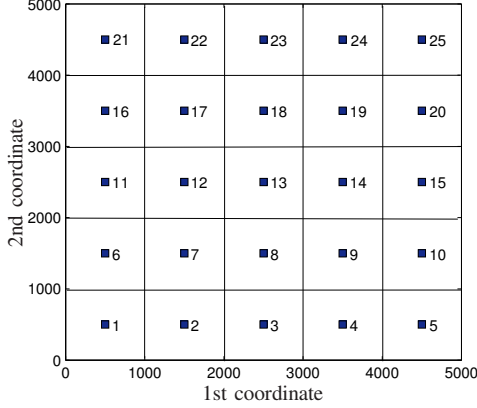


Fig. 4: Region topology- ■ represents the cluster heads.

the enhanced one. When implementing the algorithm in a practical setup, one does not need to find the first position estimate, since it is not used in the enhancement process, which then reduces the complexity of the algorithm in terms of storage and time.

VI. SIMULATIONS AND RESULTS

In this section, we evaluate the performance of the proposed detection and estimation method for different scenarios. We consider a region of study of dimensions $5000 \text{ m} \times 5000 \text{ m}$. Sensors are deployed in a uniformly distributed manner in the area at a rate of one sensor each five meters, which is equivalent to a density of 0.05 sensor/m^2 . The region is divided into $Z = 25$ clusters with same size, thus having the same density of sensors. Fig. 4 shows the region's topology. The concentrations measured by the sensors are generated using the advection-diffusion model of (5). The maximum mass release Q_{\max} is taken equal to 1000 kg . The eddy diffusivity in the Z direction, K_z is taken equal to $0.211 \text{ m}^2/\text{s}$, while the eddy diffusivities K_x and K_y , in the X and Y directions respectively, are taken equal to $12 \text{ m}^2/\text{s}$. As for the mean wind velocity \mathbf{U} , it is taken equal to $(1.8; 0; 0) \text{ m/s}$ for illustrative purposes [4]. Finally, the sampling time T is taken equal to 1 s . The rest of this section is organized as follows. The training parameters are defined in the first subsection. In the second subsection, an evaluation of the method is proposed in the case of a single source with different noise values; then, in the third subsection, an evaluation in the case of multiple gas releases is provided. Finally, in the fourth subsection, a comparison of the proposed framework to the state-of-the-art is provided.

A. Training parameters

First, we start by describing the training phase for the detection. As we already mentioned, we assume that the atmospheric conditions are the same in all the region of interest. Therefore, in order to reduce computations, we only define one detector; then, this detector is communicated to all cluster heads in order to be used in the

clusters. The advantage is that only one training phase is needed now. The size of the training set for the detection N_{det} is taken equal to 330, where 300 are normal data and 30 are outliers. We mean by outliers concentrations measured by sensors in the case of a gas release. In order to generate the outlier data, 30 different gas release scenarios are considered, where the source could be anywhere in the cluster, and the gas release mass Q is also randomly varying between 5 kg and $Q_{\max} = 1000 \text{ kg}$. The Gaussian kernel is considered in our simulations. Consequently, the detection kernel $\kappa : \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \mapsto \mathbb{R}$ is given by the following:

$$\kappa(\mathbf{C}_i, \mathbf{C}_j) = \exp\left(-\frac{\|\mathbf{C}_i - \mathbf{C}_j\|^2}{2\sigma^2}\right),$$

where $i, j \in \{1, \dots, N_{\text{det}}\}$, and σ is the kernel's bandwidth that plays a crucial role in defining the boundary around the training data.

Next, let us give a brief description of the training phase for the estimation step of the proposed framework. According to Section V, after finding the local maxima, only the concentrations around these maxima are used with the models ψ_A and ψ_B . By solving (9), one can find the size of the zone encapsulating the useful concentrations. For $Q_{\max} = 1000 \text{ kg}$ and $C_{\text{thres}} = 10^{-5} \text{ kg/m}^3$, the solution is then $\Delta x = 26 \text{ m}$. Therefore, we only need to use a zone of $52 \text{ m} \times 52 \text{ m}$ for the training. Let the number of considered scenarios N_{reg} in the training phase be equal to 200, where $x_0 \in [0; 52] \text{ m}$ and $y_0 \in [0; 52] \text{ m}$. Q is again randomly varying between 5 kg and 1000 kg . The Gaussian kernel is used here as well. The first kernel $\kappa_A : \mathbb{R}^V \times \mathbb{R}^V \mapsto \mathbb{R}$ is then given by the following:

$$\kappa_A(\mathbf{c}_v, \mathbf{c}_w) = \exp\left(-\frac{\|\mathbf{c}_v - \mathbf{c}_w\|^2}{2\sigma_A^2}\right),$$

where $v, w \in \{1, \dots, N_{\text{reg}}\}$, and σ_A is the kernel's bandwidth that controls, together with the regularization parameter η_A , the degree of smoothness, noise tolerance, and generalization of the solution.

Due to the difference in the order of magnitude between the elements of the input \mathbf{W} , we consider four different distances and bandwidths for the second kernel $\kappa_B : \mathbb{R}^{V+3} \times \mathbb{R}^{V+3} \mapsto \mathbb{R}$. Then, κ_B is given by the following:

$$\begin{aligned} \kappa_B(\mathbf{W}_v, \mathbf{W}_w) &= \exp\left(-\frac{\|\mathbf{c}_v - \mathbf{c}_w\|^2}{2\sigma_{B1}^2}\right) \times \exp\left(-\frac{(Q_v - Q_w)^2}{2\sigma_{B2}^2}\right) \\ &\times \exp\left(-\frac{(K_{xv} - K_{xw})^2}{2\sigma_{B3}^2}\right) \times \exp\left(-\frac{(K_{zv} - K_{zw})^2}{2\sigma_{B4}^2}\right), \end{aligned}$$

where $v, w \in \{1, \dots, N_{\text{reg}}\}$, and $\sigma_{B1}, \sigma_{B2}, \sigma_{B3}, \sigma_{B4}$ are also the kernel's bandwidths, with $\sigma_{B3} = \sigma_{B4}$, since K_x and K_z have the same physical properties. The regularization parameters and the kernels' bandwidths are chosen in a way to minimize the error on the training set, using the 10-fold cross-validation technique [17]. Simulation results show that taking $\sigma_{B1} = \sigma_{B2} = \sigma_{B3}$ yields very close results to the ones obtained by taking different values for the bandwidths. Therefore, in order to save time and computations in the cross-validation phase, we take $\sigma_{B1} = \sigma_{B2} = \sigma_{B3}$.

TABLE I: Percentage of errors on the estimated source's parameters for different random relative noises.

| Noise | \hat{Q} | \hat{K}_x | \hat{K}_z | \hat{x}_0 | \hat{y}_0 | $\hat{x}_{0\text{enh}}$ | $\hat{y}_{0\text{enh}}$ |
|-------|-----------|-------------|-------------|-------------|-------------|-------------------------|-------------------------|
| 1% | 0.69 | 0.006 | 0.006 | 0.02 | 0.03 | 0.01 | 0.01 |
| 2% | 0.75 | 0.007 | 0.007 | 0.03 | 0.04 | 0.01 | 0.02 |
| 3% | 0.89 | 0.01 | 0.01 | 0.03 | 0.04 | 0.02 | 0.02 |
| 4% | 0.96 | 0.02 | 0.02 | 0.04 | 0.04 | 0.02 | 0.02 |
| 5% | 1.15 | 0.02 | 0.02 | 0.04 | 0.04 | 0.02 | 0.02 |

B. Testing in the case of a single source

Now that we have defined all the parameters for the training of the detector and the two regression models, we can evaluate the performance of the proposed method in the case of a single source. The source is randomly positioned in the region of interest, with $x_0 \in [0; 5000]m$ and $y_0 \in [0; 5000]m$, and $Q \in [5; 1000] \text{ kg}$. All other parameters are set according to the beginning of this section, and as taken in [4]. The testing phase takes place during an interval of 10 s; then, the concentrations are measured each $T = 1 \text{ s}$ during this interval. The gas release can occur at any time in this interval. Note that random relative noises varying from 1% to 5% are added to the simulated concentration vectors, in order to account for transmission impairments or sensing errors. Using the proposed method, the gas release is detected in the affected clusters, and the source parameters are estimated based on the measured concentrations from the clusters. Table I shows the mean percentage of errors on the estimated source's parameters averaged over 50 Monte-Carlo simulations. We have a 100% detection in the affected zones, and one can see that the parameters of the source are accurately estimated. Also, notice the improvement in the source's location after the enhancement phase of Subsection V-B. Moreover, one can see from the small increase in the percentage of errors that the method is robust to noise.

C. Testing in the case of multiple sources

Now let us evaluate the accuracy of the method in the case of multiple gas releases, having different parameters and occurring at different times. To this end, consider four sources, whose parameters are given in Table II. The concentrations are generated using (5), and a random relative noise of 5% is considered. The test phase is run here as well during an interval of 10 seconds. The subplots of Fig. 5 show the concentrations measured by the sensors around Source 2, whose parameters are given in Table II. Notice how the maximal concentration at each time t is moving along with the X-axis, and its value is decreasing with time because of the diffusion phenomenon. The estimated sources' parameters and time of detection of each source are given in Table III. One can see that the estimations are accurate, showing that the proposed method can also be used for the case of multiple sources.

TABLE II: Parameters of the four sources.

| | Source 1 | Source 2 | Source 3 | Source 4 |
|---------------------------|----------|----------|----------|----------|
| t_0 (s) | 2 | 3 | 5 | 5 |
| Q (kg) | 500 | 1000 | 20 | 200 |
| x_0 (m) | 100 | 100 | 2500 | 4500 |
| y_0 (m) | 100 | 4000 | 3000 | 1500 |
| K_x (m ² /s) | 12 | 12 | 12 | 12 |
| K_z (m ² /s) | 0.211 | 0.211 | 0.211 | 0.211 |

TABLE III: Estimation of the parameters of the four sources using the proposed framework.

| | Source 1 | Source 2 | Source 3 | Source 4 |
|---------------------------------|----------|----------|----------|----------|
| detection (s) | 3 | 4 | 6 | 6 |
| \hat{Q} (kg) | 503.5 | 993 | 20.3 | 201.1 |
| $\hat{x}_{0\text{enh}}$ (m) | 100.0 | 100.2 | 2499.9 | 4500.3 |
| $\hat{y}_{0\text{enh}}$ (m) | 100.1 | 3999.9 | 2999.2 | 1499.8 |
| \hat{K}_x (m ² /s) | 12.01 | 11.99 | 11.98 | 11.99 |
| \hat{K}_z (m ² /s) | 0.211 | 0.211 | 0.211 | 0.211 |

D. Comparison to the state-of-the-art

In this subsection, we compare the percentage of errors to the ones obtained with the method in [4], for different values of the random relative noise varying from 1% to 5%. In [4], the source's parameters are estimated in an area of around $5500 \text{ m} \times 5500 \text{ m}$, using the measured concentrations and an inverse model obtained after solving a non-linear least squares estimation problem. The source is placed at a fixed location with $x_0 = 5000$ and $y_0 = 100$, and the release rate Q is taken equal to 1000 kg. Table IV shows the percentages of error for both methods, where PM denotes the proposed one and KM the one in [4]. Note that in this table, for the proposed method, only the enhanced estimates of the source's location are taken into account. Also, results over K_z are not shown, since in [4], K_z is assumed to be known. When comparing the errors, one can see that the proposed method greatly outperforms the KM method in terms of accuracy. Moreover, it is important to see here that the proposed method is much more robust to the increase of the noise percentage compared to the KM.

VII. CONCLUSION

This paper introduces a new clusterized method for the detection and estimation of the parameters of multiple gas sources in wireless sensor networks. Both phases of the method are defined within the framework of kernel methods, that proved to be very efficient in modeling nonlinear problems. Indeed, the evaluation of the proposed method on simulated data showed that the method yields accurate estimates, and the accuracy is maintained even with the increase of the noise level and in the case of multiple sources. Future work will handle further improvements of this method, such as to include cases where the eddy diffusivities or the wind's velocity and direction are not constant.

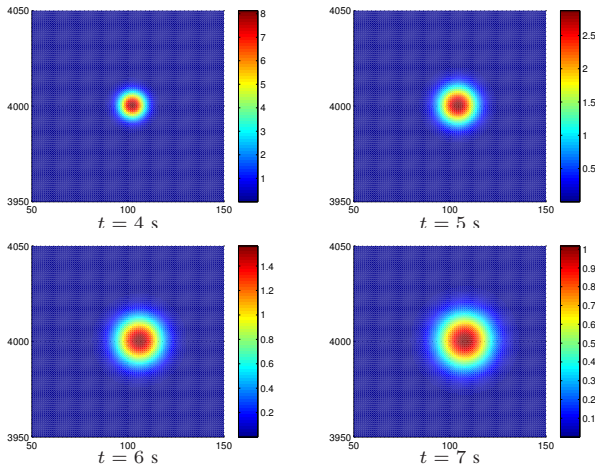


Fig. 5: Concentration distribution in kg/m^3 at different times t around Source 2 ($t_0 = 3$ s).

TABLE IV: Comparison of the percentage of errors obtained by the proposed method (PM) and the method in [4] (KM), for different random relative noises.

| Noise | Method | \hat{Q} | \hat{K}_x | \hat{x}_0 | \hat{y}_0 |
|-------|--------|-----------|-------------|-------------|-------------|
| 1% | PM | 0.32 | 0.01 | 0.01 | 0.02 |
| | KM | 2.7 | 0.7 | 0.8 | 0.4 |
| 2% | PM | 0.41 | 0.01 | 0.01 | 0.01 |
| | KM | 5.8 | 1.4 | 1.5 | 0.7 |
| 3% | PM | 0.52 | 0.01 | 0.01 | 0.02 |
| | KM | 8.0 | 2.2 | 2.4 | 1.1 |
| 4% | PM | 0.82 | 0.02 | 0.01 | 0.02 |
| | KM | 11.0 | 2.7 | 2.9 | 1.6 |
| 5% | PM | 1.05 | 0.02 | 0.01 | 0.03 |
| | KM | 12.3 | 3.7 | 4.0 | 1.9 |

REFERENCES

- [1] A. S. Kalelkar, "Investigation of large-magnitude incidents: Bhopal as a case study," in *Conference On Preventing Major Chemical Accidents, Institute of chemical engineers*, May 1988.
- [2] T. A. T., "Overview of Sarin terrorist attacks in Japan," in *Natural and Selected Synthetic Toxins*, December 1999, pp. 304–317.
- [3] V. Christopoulos and S. Roumeliotis, "Multi robot trajectory generation for single source explosion parameter estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 2803–2809.
- [4] P. Kathirgamanathan, R. Mckibbin, and R. I. McLachlan, "Source term estimation of pollution from an instantaneous point source," *Research Letters in the Information and Mathematical Sciences*, vol. 3, pp. 59–67, 2002.
- [5] G. Delmaire and G. Roussel, "Joint estimation decision methods for source localization and restoration in parametric convolution processes. application to accidental pollutant release," *Digital Signal Processing*, vol. 22, no. 1, pp. 34 – 46, 2012.
- [6] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, August 2007.
- [7] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

- [8] S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi, and J. Farah, "Kernel-based localization using fingerprinting in wireless sensor networks," in *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2013, pp. 744–748.
- [9] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Ridge regression and kalman filtering for target tracking in wireless sensor networks," in *IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, June 2014, pp. 237–240.
- [10] —, "Target tracking using machine learning and kalman filter in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 10, pp. 3715–3725, Oct 2014.
- [11] D. M. J. Tax and R. P. W. Duin, "Data domain description using support vectors," in *Proceedings of the European Symposium on Artificial Neural Networks*, 1999, pp. 251–256.
- [12] B. Schölkopf, C. Burges, and A. Smola, "Introduction to support vector learning," in *Advances in Kernel Methods – Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, 1998, pp. 1–22.
- [13] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*. London, UK: Springer-Verlag, 2001, pp. 416–426.
- [14] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.
- [15] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.
- [16] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 515–521.
- [17] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Royal Stat. Soc.*, vol. 36, no. 2, pp. 111 – 147, 1974.