Spatial regression analysis of NO\textsubscript{x} and O\textsubscript{3} concentrations in Madrid urban area using Radial Basis Function networks

S. Salcedo-Sanz \textsuperscript{a,\*}, J.A. Portilla-Figuera\textsuperscript{a}, E.G. Ortiz-Garcia \textsuperscript{a}, A.M. Perez-Bellido \textsuperscript{a}, R. Garcia-Herrera \textsuperscript{b}, J.I. Elorrieta \textsuperscript{c}

\textsuperscript{a} Department of Signal Theory and Communications, Universidad de Alcalá, Spain
\textsuperscript{b} Department of Physics of the Earth, Astronomy and Astrophysics II, Universidad Complutense de Madrid, Spain
\textsuperscript{c} Cementos Portland, Sustainability and Environment Department, Madrid, Spain

\textbf{ARTICLE INFO}

Article history:
Received 1 April 2009
Received in revised form 23 July 2009
Accepted 25 July 2009
Available online 6 August 2009

Keywords:
Spatial regression
RBF networks
Concentration of pollutants
NO\textsubscript{x}, O\textsubscript{3}
LUR models
Geographic information system

\textbf{ABSTRACT}

This paper discusses the performance of Radial Basis Function networks (RBF) in a problem of spatial regression of pollutants in Madrid. Specifically, the spatial regression of NO\textsubscript{x} and O\textsubscript{3} is considered, in such a way that, starting from a set of measuring points provided by the air quality monitoring network of Madrid, the complete surface of the pollutants in the city is obtained. This pollutant surface can be used as an initial step for modeling intra-urban pollution using land-use regression techniques for example. Also, different works have used a pollutant surface to study the patterns of pollution in different cities in the world and also to establish their air monitoring networks under mathematical criteria. The paper is focussed in analyzing the performance of RBF networks to obtain this first pollutant surface, so different RBF training algorithms are tested in this paper. Specifically, evolutionary-based RBF training algorithms are described, and compared with classical training algorithms for RBF networks with Gaussian kernels. The inclusion of meteorological variables in the RBF networks are also discussed in the paper. The experimental part of the article studies real results of the application of RBF networks to obtain a first pollutant surface of NO\textsubscript{x} and O\textsubscript{3}, using the data of the air pollution monitoring network of Madrid and the meteorological network of the city.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

As in many other cities, oxides of nitrogen (NO\textsubscript{x}) and ozone (O\textsubscript{3}) are the most relevant air pollutants in Madrid, Spain. It is well known that ozone is a secondary pollutant, since it is not directly emitted into the air. On the contrary, tropospheric ozone is produced when the primary pollutants, mainly nitrogen oxides (NO\textsubscript{x}) and Volatile Organic Compounds (VOC) interact under the action of the sunlight [1,2]. The study of the concentrations of these two pollutants is of major interest, since several studies have revealed serious health effects associated with the continuous exposure to high concentration values of these pollutants [3].

Several works on modeling and forecasting of NO\textsubscript{x} and O\textsubscript{3} can be found in the literature [4–7,9,10] in many different cities and with very different methods. One important problem previously tackled is the time forecasting of pollutants, in which several factors such as meteorology [11], past concentrations, traffic or city structure [13] are considered in order to provide a value of concentration in a given point in the future, usually specific measuring points are selected. Thus, in this problem it is not necessary to have a large amount of measuring points, and this approach can be carried out even in cities with a small air quality monitoring network [12].

On the other hand, the spatial modeling of contamination within urban areas is another important problem, in this case focussed on spatial regression of pollutants from different measuring points [1,3,13,20,41–43]. Different approaches can be found in the literature to tackle the problem of assessing intra-urban air pollution, many of them focussed on these spatial regressions of contaminants. Specifically, the works on this topic can be classified by the type of algorithm proposed: first, some problems consider dispersion models of gas-contaminants. In this works, the methodology consists of locating the possible source of contamination, such as roads, factories etc., taking into account the specific structure of the city and wind rose, and applying dispersion models of contaminants verifying the results on the measuring stations. Examples of this methodology are [19,20]. In this latter work, the authors analyze the ozone response to variations in road traffic and total anthropogenic emissions, in two different summer ozone episodes occurred in Madrid. The sensitivity of the dispersion model used to changes in emissions of NO\textsubscript{x} and volatile organic compounds is also studied.

A second methodology which has obtained very good results in the analysis of spatial structure of contaminants in intra-urban areas are...
the direct spatial interpolation methods such as kriging, TIN-contouring, trend surface analysis, triangulation, splines etc. Recent significant examples of the application of these techniques can be found [13,1]. In [1], the authors comment that the best results are obtained using thin plate spline functions, and use this method in the study of NO$_2$ in Paris metropolitan area. The authors also comment in this paper that thin plate splines used can be seen as a kind of Radial Basis Function, so this may be considered the first work applying RBFs to the modeling of pollutants in urban areas. In a more recent work, Kanaroglou et al. [3] proposed the modeling of pollutants using regression methods to obtain a pollutant surface. This surface is then used to locate the measuring stations of the monitoring networking in an optimal way. The authors apply their model to the city of Toronto, using mobile and fixed measuring stations, focused on NO$_2$.

Finally, in the last few years a new regression technique has gained relevance, the so called Land-Use Regression (LUR). LUR combines monitoring of air pollution at specific locations and several other statistic and stochastic models using predictor variables, many of them obtained through Geographic Information Systems. LUR models, initially called Regression Mapping Models [15], usually involves other variables such meteorological or altitude in addition to strict land use ones [18]. One interesting characteristics of LUR models is that they differ much from one applications to others, so the variables involved and specific methodologies are quite different in different LUR applications. Recent specific examples of LUR application in spatial regression of NO$_2$ and O$_3$ pollutants are [8,13]. In [13] a model of LUR is applied to model traffic-related pollutants in the city of Shizuoka, Japan. As novelty, this work proposes the inclusions as variable in the LUR the distance from the sea, which improves the performance of the RBF results, and how the obtained data can be used within a LUR model. Section 5 closes the paper giving some final conclusions.

1.1. Paper’s objective and structure

All the spatial regression techniques discussed above have been applied to different real problem of pollutant concentration estimation, with notable success. This paper is focussed, however, on spatial interpolation methods using a class of neural network (RBFs), a novelty in this field. The main objective of this paper is to discuss the performance of different RBF training methods in the problem of spatial regression of pollutants. Specifically, in this paper we evaluate the performance of RBF networks with Gaussian kernels for the problem of spatial regression of pollutants concentration (NO$_2$ and O$_3$) in Madrid urban area. RBF networks with Gaussian kernels have been successfully used in different regression problems before [21–24] and in this paper we show their applicability to problems of spatial regression of pollutants, for obtaining contaminant surface, or as initial points for the application of a LUR model. Different training methods for RBF networks will be tested in the paper, focussing on novel methods, mainly evolutionary-type training approaches, that have been recently applied to RBF training [25–29]. The proposed algorithms have been tested in Madrid, with data from the air quality monitoring network of the city. The results obtained in this paper using the RBF networks proposed serve as initial points for LUR models, which will provide accurate punctual estimations for NO$_2$ and O$_3$. Discussion on the improvement of the RBF models using meteorological data are also carried out in the paper.

The structure of the rest of the paper is as follows: next section presents the air pollution monitoring network of Madrid, where this study is carried out. Section 3 conforms to the body of the paper, in this section the main characteristics of Gaussian kernel RBFs are presented, and the main training methods described. Specifically, evolutionary-based training algorithms (Evolutionary Programming and Differential Evolution algorithms) are described. Section 4 presents the main results obtained in the spatial regression of NO$_2$ and O$_3$ pollutants in Madrid, comparing the performance of the different RBF training methods. Also, this section discusses the possibility of using meteorological data to improve the performance of the RBF results, and how the obtained data can be used within a LUR model. Section 5 closes the paper giving some final conclusions.

2. The air pollution monitoring network of Madrid

The air pollution monitoring network of Madrid is the largest in Spain, and one of the largest in Europe. It is currently formed by 27 measuring (fixed) stations spread out in the city. Fig. 1 and Table 1

![Fig. 1. Location of the measuring stations of the air quality monitoring network of Madrid.](Image)

<table>
<thead>
<tr>
<th>Number</th>
<th>District</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Altitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Centro</td>
<td>40° 25'36&quot;N</td>
<td>3° 41'31&quot;W</td>
<td>648 m</td>
</tr>
<tr>
<td>2</td>
<td>Retiro</td>
<td>40° 24'33.35&quot;N</td>
<td>3° 41'29.51&quot;W</td>
<td>629 m</td>
</tr>
<tr>
<td>3</td>
<td>Centro</td>
<td>40° 25'09.15&quot;N</td>
<td>3° 42'1.14&quot;W</td>
<td>657 m</td>
</tr>
<tr>
<td>4</td>
<td>Moncloa</td>
<td>40° 25'26.37&quot;N</td>
<td>3° 42'44.40&quot;W</td>
<td>637 m</td>
</tr>
<tr>
<td>5</td>
<td>Puencarral</td>
<td>40° 28'41.62&quot;N</td>
<td>3° 42'41.59&quot;W</td>
<td>673 m</td>
</tr>
<tr>
<td>6</td>
<td>Chamberii</td>
<td>40° 26'15.39&quot;N</td>
<td>3° 41'27.00&quot;W</td>
<td>660 m</td>
</tr>
<tr>
<td>7</td>
<td>Salamanca</td>
<td>40° 25'47.81&quot;N</td>
<td>3° 40'49.19&quot;W</td>
<td>679 m</td>
</tr>
<tr>
<td>8</td>
<td>Salamanca</td>
<td>40° 25'17.63&quot;N</td>
<td>3° 40'56.35&quot;W</td>
<td>672 m</td>
</tr>
<tr>
<td>9</td>
<td>Arganzuela</td>
<td>40° 24'07.68&quot;N</td>
<td>3° 31'36.35&quot;W</td>
<td>679 m</td>
</tr>
<tr>
<td>10</td>
<td>Chamberii</td>
<td>40° 26'43.95&quot;N</td>
<td>3° 42'25.66&quot;W</td>
<td>699 m</td>
</tr>
<tr>
<td>11</td>
<td>Chamartin</td>
<td>40° 27'05.03&quot;N</td>
<td>3° 40'38.47&quot;W</td>
<td>708 m</td>
</tr>
<tr>
<td>12</td>
<td>Salamanca</td>
<td>40° 25'43.70&quot;N</td>
<td>3° 40'06.78&quot;W</td>
<td>788 m</td>
</tr>
<tr>
<td>13</td>
<td>Puente Vallecass</td>
<td>40° 23'17.34&quot;N</td>
<td>3° 39'05.48&quot;W</td>
<td>677 m</td>
</tr>
<tr>
<td>14</td>
<td>Usera</td>
<td>40° 23'06.28&quot;N</td>
<td>3° 42'59.71&quot;W</td>
<td>605 m</td>
</tr>
<tr>
<td>15</td>
<td>Tetuan</td>
<td>40° 28'50.73&quot;N</td>
<td>3° 41'19.29&quot;W</td>
<td>729 m</td>
</tr>
<tr>
<td>16</td>
<td>Ciudad Lineal</td>
<td>40° 26'24.17&quot;N</td>
<td>3° 38'21.24&quot;W</td>
<td>608 m</td>
</tr>
<tr>
<td>17</td>
<td>Villaverde</td>
<td>40° 20'54.25&quot;N</td>
<td>3° 42'41.42&quot;W</td>
<td>593 m</td>
</tr>
<tr>
<td>18</td>
<td>Carabanchel</td>
<td>40° 23'41.20&quot;N</td>
<td>3° 43'54.60&quot;W</td>
<td>625 m</td>
</tr>
<tr>
<td>19</td>
<td>Latina</td>
<td>40° 24'28.29&quot;N</td>
<td>3° 44'30.83&quot;W</td>
<td>632 m</td>
</tr>
<tr>
<td>20</td>
<td>Moratalaz</td>
<td>40° 24'28.64&quot;N</td>
<td>3° 38'43.08&quot;W</td>
<td>671 m</td>
</tr>
<tr>
<td>21</td>
<td>Moncloa</td>
<td>40° 26'27.51&quot;N</td>
<td>3° 43'04.54&quot;W</td>
<td>672 m</td>
</tr>
<tr>
<td>22</td>
<td>Arganzuela</td>
<td>40° 24'22.95&quot;N</td>
<td>3° 42'46.56&quot;W</td>
<td>622 m</td>
</tr>
<tr>
<td>23</td>
<td>San Blas</td>
<td>40° 26'55.44&quot;N</td>
<td>3° 36'34.62&quot;W</td>
<td>637 m</td>
</tr>
<tr>
<td>24</td>
<td>Moncloa</td>
<td>40° 25'09.68&quot;N</td>
<td>3° 44'50.44&quot;W</td>
<td>645 m</td>
</tr>
<tr>
<td>25</td>
<td>Villa de Vallecass</td>
<td>40° 22'44.48&quot;N</td>
<td>3° 36'09.18&quot;W</td>
<td>652 m</td>
</tr>
<tr>
<td>26</td>
<td>Barajas</td>
<td>40° 27'33.56&quot;N</td>
<td>3° 34'48.42&quot;W</td>
<td>620 m</td>
</tr>
<tr>
<td>27</td>
<td>Barajas</td>
<td>40° 28'36.94&quot;N</td>
<td>3° 34'48.10&quot;W</td>
<td>631 m</td>
</tr>
</tbody>
</table>
show the location and other characteristics of the air pollution measuring stations. The automatic air pollution monitoring network of Madrid started in 1978. At the beginning the network was formed by 16 measuring stations, connected by the telephonic network with a center of data control, depending on the department of air quality of Madrid City Council. In 1989 the network was completely renewed, new “intelligent” stations were acquired. At this point the systematic measurement of NOx and O3 started. The monitoring network in its current form was finished in 2001, when the last 2 stations were added to the network, and several other stations were moved from their original location due to technical reasons.

The studies on spatial distribution of contaminants in cities consider two types of monitoring networks to obtain data: routine networks and purpose-design networks [14]. Both networks have advantages and disadvantages, depending on the specific problem to be tackled. Following [14], routine networks are mainly used for studies on long-term concentration of contaminants, since purpose-design networks are usually deployed during a small period of time, with a very specific measuring objective. The air pollution monitoring network of Madrid used in this study is a routine network, since we are interested in long-term concentration of pollutants (quarterly or yearly concentrations). Several other studies have been used data from routine monitoring networks, such as [16] or [17]. The main disadvantages of routine networks versus purpose-design ones is that sometimes routine networks are not dense enough, which may cause problems with the models.

3. Methods: Radial Basis Function networks for regression

In this section the different algorithms to train RBF networks with Gaussian kernels are revised. First, the main characteristics of RBF networks are described. In another subsection, several recent evolutionary-based algorithms for RBF are described.

3.1. Radial Basis Function networks with Gaussian kernels

One of the most promising and used methods for function approximations are the so-called Radial Basis Function Network (RBFs), which have been profusely studied in the last few years. The problem of function approximation can be defined as obtaining a model of a function \( f(x) \), from a set of \( p \) known points (training set) \( T = \{x_i, y_i\}_{i=1}^p \), where \( x_i \) is a vector of dimension \( n \), and \( y_i \) is an scalar. RBFs arise as a modification of the general concept of neural network, where a kernel function is introduced as an activation function for each neuron of the network. In the most general case, a RBF network is formed by two layers (Fig. 2): The first layer is formed by the activation function in form of a kernel. The classical kernel used in RBF networks are Gaussian functions, with the form:

\[
h_j(x) = \exp\left(-\frac{(x-c_j)^T(x-c_j)}{r^2}\right)
\]

where \( c_j \) stands for the center of the \( j \)th Gaussian and \( r \) is a parameter related to the width of each Gaussian. Note that \( r^2 \) is the variance of each Gaussian. Note that this model can be generalized to consider different widths in each direction of the Gaussian:

\[
h_j(x) = \exp(- (x-c_j)^T Q (x-c_j))
\]

where \( Q \) is a matrix related to the covariances of the Gaussian.

The second layer performs a linear approximation of the output of the Gaussians, in order to obtain the final approximation function. Following Fig. 2, function \( f(x) \) is constructed using a linear model including the \( m \) kernels considered in the network:

\[
f(x) = \sum_{j=1}^{m} w_j h_j(x).
\]

Note that the training process of a RBF network requires to obtain the values of \( w, c \) and \( r \), in such a way that a measure of error between the output of the network and the correct output for the corresponding input pattern is minimized. The following error function has been proposed in the literature [24], as a robust error function to guide the training process of RBF networks:

\[
C = \sum_{i=1}^{p} (y_i - f(x_i))^2 + \lambda \sum_{j=1}^{m} w_j
\]

where \( \lambda \) is the so-called regularization parameter. This final term is included to avoid the overfitting of the model, looking for a balance between error minimization and generalization of the final model.

The linear formulation of the RBF networks allows the calculation of the optimal set of weights \( w \), given the values of \( c \) and \( r \) (proof by least squares in [24]):

\[
w = A^{-1} H^T y
\]

where \( w \) stands for the optimal weights of the RBF network (amplitudes of the Gaussian functions of the network), \( H \) is called design matrix of the network (\( T \) stands for the transpose of the matrix), and is defined as:

\[
H = \begin{bmatrix}
h_1(x_1) & h_2(x_1) & \cdots & h_m(x_1) \\
h_1(x_2) & h_2(x_2) & \cdots & h_m(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
h_1(x_p) & h_2(x_p) & \cdots & h_m(x_p)
\end{bmatrix}
\]

and \( A^{-1} \) is known as the variance matrix, defined as:

\[
A^{-1} = (HH^T + \lambda I_m)^{-1}
\]

with \( I_m \) the \( m \times m \) identity matrix.

Note that the procedure shown above allows the exact calculation of the Gaussian amplitudes once the parameters \( c \) and \( r \) are calculated. However, obtaining these parameters is not a straightforward process, and there are many methods that have been proposed in the literature to carry out the search for optimal values of \( c \) and \( r \).

3.2. Classical RBF training algorithms

In this paper we have considered four different algorithms of center and variance selection of the Gaussians. Each algorithm can be considered as a different training algorithm. The calculation of the
weights of the network (amplitudes of the Gaussians) have been carried out using Eq. (5) in all cases.

- **Ridge Regression.** It is one of the most straightforward procedures to train RBF networks. In ridge regression, all the patterns are considered as centers of the Gaussians, optimizing Eq. (4) with regularization parameter not equal to 0. The variances of the Gaussians (given by parameter \( r \) in Eq. (1)) is fixed to be equal to the range of the data \( x \) considered. This is also used in the rest of the considered classical training algorithms explained below.

- **Forward Selection.** The forward selection consists of an heuristics for obtaining the optimal number of Gaussians of the network. The heuristic works by increasing or decreasing the number of the Gaussians in the network depending on how the approximation error of the network varies. Basically, the heuristic is a branching and pruning algorithm, which starts with 1 Gaussian, and increases the number of radial function in the system until the error starts growing. Then, a pruning heuristics is applied in order to stabilize the number of Gaussians in an optimal value. Details on this process can be seen in [23].

- **Tree Decision RBF 1.** In the last years several alternative algorithms have been proposed to train RBFs. One of the most interesting is the use of decision trees to train RBF networks [30]. This method approximates the training set by means of a decision tree, in such a way that it divides the search space in hyper-rectangles, i.e., each node of the tree has associated a Gaussian (center and variance). The algorithm selects then the Gaussian functions of the network, selecting first the largest hyper-rectangles (largest Gaussian function), and going down through the tree adding subnodes (new functions), until reaching the tree roots (see [30] for reference).

- **Tree Decision RBF 2.** The decision tree algorithm can be implemented with several variations. We consider a second algorithm based on decision trees, in which the selection of the Gaussian parameters is carried out using the same decision tree as the Tree Decision RBF 1, but in this case the Forward Selection is used in order to select the nodes of the tree [30].

### 3.3. Evolutionary-based RBF training

The evolutionary-based training of RBF networks is a promising field which has been exploited in different areas [25–29]. It consists of using evolutionary-based approaches to obtain optimal parameters of the RBF network kernel functions. In this paper, we consider Gaussian kernels in the network, so we need to obtain the optimal centers and variances of the Gaussians in the RBF. Specifically, we apply different evolutionary algorithms to obtain the optimal vector \( \mathbf{c} \) and matrix \( \mathbf{Q} \) of each Gaussian (Eq. (2)) in the RBF network. We compare two different evolutionary algorithms which have provided very good results in other problems of continuous optimization: the Evolutionary Programming (EP) and a modification of the Differential Evolution (DE) algorithm.

#### 3.3.1. Evolutionary programming

Evolutionary algorithms (EAs) [31–35], are robust problem’s solving techniques based on natural evolution processes. They are population-based techniques which codify a set of possible solutions to the problem, and evolve it through the application of the so called evolutionary operators [31]. Among EAs, Evolutionary Programming (EP) approaches are usually applied to continuous optimization problems. This algorithm is characterized by only using mutation and selection operators (no crossover is applied). Several versions of the algorithm have been proposed in the literature: The Classical Evolutionary Programming algorithm (CEP) was first described in the work by Bäck and Schwefel in [33], and analyzed later by Yao et al. in [34,36]. It is used to optimize a given function \( f(\mathbf{x}) \) (\( \psi \) or \( \phi \) in our case), i.e. obtaining \( \mathbf{x}_t \) such that \( f(\mathbf{x}_t) < f(\mathbf{x}) \), with \( \mathbf{x} \) \([\text{lim inf}, \text{lim sup}]\). The CEP algorithm performs as follows:

1. Generate an initial population of \( \mu \) individuals (solutions). Let \( t \) be a counter for the number of generations, set it to \( t = 1 \). Each individual is taken as a pair of real-valued vectors \( (\mathbf{x}_i, \mathbf{c}_i) \), \( \mathbf{c}_i \in \{1, \ldots, \mu\} \), where \( \mathbf{x}_i \)'s are objective variables, and \( \mathbf{c}_i \)'s are standard deviations for Gaussian mutations.
2. Evaluate the fitness value for each individual \( (\mathbf{x}_i, \mathbf{c}_i) \) (using the problem’s objective function, \( \psi \) or \( \phi \)).
3. Each parent \( (\mathbf{x}_i, \mathbf{c}_i), (i = 1, \ldots, \mu) \) then creates a single offspring \( (\mathbf{x}_t, \mathbf{c}_t) \) as follows:

\[
\mathbf{x}_t = \mathbf{x}_i + \mathbf{c}_t \cdot \mathbf{N}(0, 1)
\]

\[
\mathbf{c}_t = \mathbf{c}_i \cdot \exp(\tau \cdot \mathbf{N}(0, 1) + \tau \cdot \mathbf{N}(0, 1))
\]

where \( \mathbf{N}(0, 1) \) denotes a normally distributed one-dimensional random number with mean zero and standard deviation one, and \( \mathbf{N}(0, 1) \) and \( \mathbf{N}(0, 1) \) are vectors containing random numbers of mean zero and standard deviation one, generated anew for each value of \( i \). The parameters \( \tau \) and \( \tau' \) are commonly set to \( (\sqrt{2/\pi n})^{-1} \) and \( (\sqrt{2/\pi n})^{-1} \), respectively [36], where \( n \) is the length of the individuals.

4. If \( f(\mathbf{x}_t) > \text{lim sup} \) then \( f(\mathbf{x}_t) = \text{lim sup} \) and if \( f(\mathbf{x}_t) < \text{lim inf} \) then \( f(\mathbf{x}_t) = \text{lim inf} \).
5. Calculate the fitness values associated with each offspring \( (\mathbf{x}_t, \mathbf{c}_t) \), \( \forall i \in \{1, \ldots, \mu\} \).
6. Conduct pairwise comparison over the union of parents and offspring: for each individual, opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is better than the opponent's, it receives a "win".
7. Select the \( \mu \) individuals out of the union of parents and offspring that have the most "wins" to be parents of the next generation.
8. Stop if the halting criterion is satisfied, and if not, set \( t = t + 1 \) and go to Step 3.

A second version of the algorithm is the so called Fast Evolutionary Programming (FEP). The FEP was described and compared with the CEP in [36]. The FEP is similar to the CEP algorithm, but it performs a mutation following a Cauchy probability density function, instead of a Gaussian based mutation. The one-dimensional Cauchy density function centered at the origin is defined by

\[
f_\text{Cauchy}(x) = \frac{1}{\pi \tau^2 + x^2}
\]

where \( \tau > 0 \) is a scale parameter. See [36] for further information about this topic. Using this probability density function, we obtain the FEP algorithm by substituting step 3 of the CEP, by the following equation:

\[
x_t = x_i + \mathbf{c}_i \cdot \mathbf{N}(0, 1)
\]

where \( \mathbf{c}_i \) is a Cauchy random variable vector with the scale parameter set to \( \tau = 1 \).

Finally, in [36] the improved FEP (IFEP) is also proposed, where the best result obtained between the Gaussian mutation and the Cauchy mutation is selected to complete the process.

#### 3.3.2. Differential evolution

Individuals in DE are represented by \( D \)-dimensional vectors \( x_i \), \( i = \{1, \ldots, N\} \), where \( D \) is the number of objective parameters and \( NP \) is the
where $\mathbf{u}_i$ is the offspring of $\mathbf{x}_i$ for the next generation. Several schemes of DE have been described in the literature, based on different mutation strategies [40]:

$$v_i = x_i + F \cdot (x_{i+1} - x_i)$$

(12)

where $i, j, k \in \{1, N\}$ are random, mutually different integers, with different index $i$. Scale factor $F = 0$ is a real constant factor, often set to 0.5.

2. Crossover:

$$u_i = \begin{cases} v_i(j) & \text{if } U_i(0,1) < CR \land j \neq rand(j) \\ v_i(j) & \text{otherwise} \end{cases}$$

(13)

where $U_i(0,1)$ stands for the uniform random number between 0 and 1, and $rand(j)$ is a randomly chosen index to ensure that the trial vector $u_i$ does not duplicate $x_i$. CR (0,1) stands for the crossover rate, which is often set to 0.8.

3. Selection:

$$x'_i = \begin{cases} u_i & \text{if } f(u_i) \leq f(x_i) \\ x_i & \text{otherwise} \end{cases}$$

(14)

where $x'_i$ is the offspring of $x_i$ for the next generation.

3.3.3. Self-adaptive DE with neighborhood search

Recently, an improved DE algorithm has been proposed in [40], where a DE variant that utilizes the neighborhood search (NS) strategy in evolutionary programming (EP), and a self-adaptation to consider different mutation schemas in the search has been presented (SaNSDE algorithm).

Specifically, SaNSDE is the same with the classical DE described in the previous section, but, first, scale factor $F$ is replaced by the following equation:

$$F_i = \begin{cases} N_i(0.5, 0.5) & \text{if } U_i(0,1) < 0.5 \\ \delta_i & \text{otherwise} \end{cases}$$

(20)

where $i$ is the index of current trial vector, $U_i(0,1)$ stands for the uniform random number between 0 and 1, $N_i(0.5,0.5)$ denotes a Gaussian random number with mean 0.5 and standard deviation 0.5, and $\delta_i$ denotes a Cauchy random variable with scale parameter $t = 1$. In addition, self-adaptation of the search is included in the algorithm, in order to consider two different mutation schemas into one DE algorithm. This is carried out by introducing a probability $P$ to control which mutation strategy to use, and $P$ is gradually self-adapted according to the learning experience:

$$v_i = \begin{cases} Eq. (15) & \text{if } U_i(0,1) < P \\ Eq. (17) & \text{otherwise} \end{cases}$$

(21)

Further details on the specific implementation of the SaNSDE algorithm can be obtained in [40].

4. Experimental part

4.1. Data available for the study and methodology for the experiments

The available database is formed by hourly measures of NO and O₃ taken in the 27 stations of the air quality monitoring network of Madrid (see Section 2) corresponding to 6 years, from 2002 to 2007.
In order to carry out a significant spatial regression analysis, we only consider quarterly and yearly averages of both pollutants. This eliminates dependencies on specific weather conditions and structure of the city, typically important in the case of hourly measures. In the case of NO\(_x\) the standard mean of the full year period has been used, i.e.,

\[
 y_j = \frac{1}{N} \sum_{i=1}^{N} u_i
\]  

where \(N\) is the total number of hours in each year and \(u_i\) the concentration value in each hour.

In the case of O\(_3\), following the Spanish Regulation [44], the standard mean of the maximums of the moving average with a window of 8 h is used, i.e.,

\[
 y_j = \frac{1}{D} \sum_{i=1}^{D} MV_i
\]  

Fig. 3. Spatial regression maps (yearly mean values) obtained with the RBF trained with the EP algorithm, for the NO\(_x\) data in Madrid; (a) NO\(_x\) levels (µg/m\(^3\)); (b) NO\(_x\) regression map for 2002; (c) NO\(_x\) regression map for 2003; (d) NO\(_x\) regression map for 2004; (e) NO\(_x\) regression map for 2005; (f) NO\(_x\) regression map for 2006; (g) NO\(_x\) regression map for 2007.

Fig. 4. Average NO\(_x\) regression map obtained with the RBF trained with the EP algorithm, corresponding to the 6 years (2002–2007) considered in the study (µg/m\(^3\)).
where $D$ is the total number of days of the year period and

$$MV_i = \max_k \left( \frac{1}{N} \sum_{j=k-7}^{k-1} u_j \right) \quad k = 1, \ldots, 24. \quad (24)$$

note that this moving average takes into account hours of the previous day ($k = 1, \ldots, 6$). For $k = 7$ the moving average considers the interval between the 0 A.M. and 7 A.M. for $k = 24$, the moving average considers the interval between 17 (4 P.M) and 24 (12 P.M.), finally, for $k = 1$ the interval is from 18 (6 P.M.) of the previous day, to 1 A.M. of the current day.

Averaging the data in a quarterly/yearly basis we obtain 24/6 problems of spatial regression for each pollutant. These problems will be solved using RBF networks, trained with the different methods described in Section 3. In order to apply the RBF network to these problems, it is necessary to establish a working methodology: in the general case, a partition of the data to obtain training and test sets is usually carried out. However, in this case we have an extremely small number of samples for each problem (27 at most, corresponding to measures in each station, if all the stations are operative). Therefore, we will use the leave-one-out (LOO) error measure, consisting of evaluating the performance of each sample, using a model trained with the rest of samples. This model has also been used before in similar analysis in other cities such as Paris [1]. Using the LOO, we obtain an estimation for each sample and at the same time we use a training set with almost all samples, which gives a good approximation of the pollution distribution. With all estimations for all the samples, we calculate the root mean square, i.e.,

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \quad (25)$$

where $y_i$ stands for the real value of the output, $\hat{y}_i$ stands for the estimation and $N$ is the number of samples.

The RBF network with Ridge Regression (RR), Forward Selection (FS), Tree Decision 1 (TD1) and Tree Decision 2 (TD2) have been
implemented in Matlab, using the library [37], adapted to the spatial regression problems tackled in this paper. The RBF network with evolutionary training (EP and SaNSDE) has been programmed in C++.

In the case of the evolutionary algorithms, the number of neurons (Gaussians) has been fixed to be equal to the number of samples in each problem, looking for the best set of centers and variances of them. Recall that in the classical algorithm the number of neurons depends on the specific training method.

To represent the results obtained with the RBF we have used the Geographic Information System (GIS) Mapinfo 9.0. The regression technique used to interpolate the points from the RBF to obtain the graphical representation is the Inverse Distance Weighting (IDW) algorithm, that is provided by the used GIS software. The representation of the mean values of the contaminants considered are referred to the maximum values allowed by the legislation (200 µg/m$^3$ in the case of NO$\textsubscript{x}$ and 120 µg/m$^3$ for the O$\textsubscript{3}$). In this way we can evaluate relative concentrations of pollutants, knowing in a sight if the concentrations are near the legal maximums or not.

### 4.2. Results of spatial regression analysis of NO$\textsubscript{x}$ and O$\textsubscript{3}$ with the different RBF networks model considered

The results (in terms of LOO Root Mean Square Error (RMSE)) for the spatial regression carried out considering NO$\textsubscript{x}$ and O$\textsubscript{3}$ are shown in Tables 2 and 3, respectively. The RMSE values displayed in these tables are shown in quarter periods. It is easy to see that the RBF networks are able to obtain a good modeling for the spatial regression of pollutants. It is also immediate that the RBF with evolutionary training improves the results of the classical training algorithms, obtaining a much lower LOO error. This means that the modeling provided by the RBF network with evolutionary training is more general and represents better the pollution structure in the city. Specifically, the EP algorithm obtains better results in training the RBF network than the SaNSDE algorithm. The differences are small for the case of the O$\textsubscript{3}$, and a little more significant for the case of NO$\textsubscript{x}$, where the concentration of pollutants is also larger (Figs. 3 and 4).

The specific RBF model obtained with the EP training can be used to calculate a regular regression grid, really useful to analyze contamination patterns. To do this, we select a grid of about 500 points, regularly spaced forming an square in the metropolitan area of Madrid. We depict the value of pollution given by the RBF model in those points using a Geographic Information System (GIS), obtaining the regression surface of pollution given by the RBF model. Figs. 3 and 5 show the corresponding surfaces for the NO$\textsubscript{x}$ and O$\textsubscript{3}$, respectively, in yearly average (6 surfaces, corresponding to years 2002–2007). Note that we have depicted the measuring stations, with its real pollution level (black points in some surfaces stand for inoperative stations that year). It is easy to see that the surfaces of pollution obtained match quite well with the measured concentration in the stations, both for the NO$\textsubscript{x}$ and the O$\textsubscript{3}$ cases. In the case of the NO$\textsubscript{x}$, a maximum of pollution is found in the center of the city, following the main road from South to North of Madrid (Paseo de la Castellana), where there is heavy traffic all year long. In the case of the O$\textsubscript{3}$ the maximum concentrations are not in the center, but in the east and west zones of the city. Figs. 4 and 6 show the pollution surface in average of the 6 years (complete period considered), for the NO$\textsubscript{x}$ and O$\textsubscript{3}$, respectively.

<table>
<thead>
<tr>
<th>Year/quarter</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RT2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SaNSDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A comparison with the different RBF training algorithms is offered.

![Fig. 6. Average O$_3$ regression map obtained with the RBF trained with the EP algorithm, corresponding to the 6 years (2002–2007) considered in the study (µg/m$^3$).](image-url)
4.3. Including meteorological data in the RBF networks

The spatial regression of NO\textsubscript{x} and O\textsubscript{3} obtained using the RBF networks with the different training methods can be improved by including meteorological data in the RBF network. Specifically, data of wind speed, wind direction, radiation, pressure and temperature measured in the stations of the air quality network of Madrid has been considered to be included in the RBFs. Each training sample is formed in this case by vectors of 7 dimensions (\(x, y\) and meteorological variables). The output of the network is again the concentration of NO\textsubscript{x} or O\textsubscript{3} in a specific point of the grid. A mechanism of LOO is also applied to this case, since again we have 27 samples to train the RBF networks. Tables 4 and 5 show the results (in terms of LOO Root Mean Square Error (RMSE)) for the spatial regression carried out considering NO\textsubscript{x} and O\textsubscript{3}, respectively. Note that the RMSE obtained is improved when the meteorological data are included in the RBF. The better performance is obtained no matter the training algorithm used for the RBF, though again the results obtained using the evolutionary techniques are much better than the alternative standard algorithms (Figs. 7 and 8).

Fig. 7. Spatial regression maps (yearly mean values) obtained with the RBF trained with the EP algorithm, for the NO\textsubscript{x} data in Madrid including meteorological variables; (a) NO\textsubscript{x} levels (\(\mu g/m^3\)); (b) NO\textsubscript{x} regression map for 2002; (c) NO\textsubscript{x} regression map for 2003; (d) NO\textsubscript{x} regression map for 2004; (e) NO\textsubscript{x} regression map for 2005; (f) NO\textsubscript{x} regression map for 2006; (g) NO\textsubscript{x} regression map for 2007.

Fig. 8. Average NO\textsubscript{x} regression map obtained with the RBF trained with the EP algorithm (including meteorological variables), corresponding to the 6 years (2002–2007) considered in the study (\(\mu g/m^3\)).
Figs. 7 and 9 show the surfaces of NO\textsubscript{x} and O\textsubscript{3}, respectively, obtained with the RBF networks taking into account the meteorological variables. The contaminant surfaces obtained including the meteorological variables are similar in structure to the ones obtained without meteorology, however, they are more accurate in terms of the error measure (as can be seen in Tables 4 and 5). This can be also seen the better adjusting of the surfaces to the real measures at the air quality stations. Figs. 8 and 10 show the pollution surface in average of the 6 years (complete period considered), for the NO\textsubscript{x} and O\textsubscript{3}, respectively, including meteorological values. Note that the trend in this case is also similar to the one without meteorology, but more accurate in terms of the error measure.

4.4. Discussion and future lines of research

In this paper we have shown how RBF networks can be used to obtain a contaminants surface (NO\textsubscript{x} and O\textsubscript{3}) starting from measures and meteorological variables in the city of Madrid. Since the study considers long-term measures, we have used the air quality
monitoring network data, which provides data of several years of measures. The method can, of course, be applied to data obtained from purpose-design networks, but usually these networks are not deployed during much time, but they are deployed in specific zones during a few days for specific studies, so it is really difficult to find studies of long-term exposures to contaminants based on purpose-design networks. The use of routing networks allows these long-term studies, but the price paid is on the spatial resolution of the real measurements available for the study. Also, the precision in location of the real sources of contamination (and then the further spatial modeling) can be affected. In this work, these effects can be appreciated in the results obtained, mainly in the areas outside of the city center, where there are very few measuring stations. Thus, the RBF obtains significant results (low LOO error) in points of the city center, well covered by several measuring stations. Note that the results in LOO error reported in all the tables of the paper consider the average of the 500 points grid considered. Note also that using reduced grids, centered in the zones completely covered by measuring stations, we would obtain better values of average LOO error. The possibility of including meteorological variables in the RBF network is also an interesting point to be considered. We have shown that these variables improve the RBF performance, and allows a better estimation of pollutant concentrations, improving the LOO error measure in all cases, independently of the RBF training algorithm used.

The results obtained in this paper have shown that the RBFs (training with evolutionary methods) provide a good way of evaluating the concentration of contaminants in the city, interesting as a first approach (note that good value of root mean quadratic error are obtained with the evolutionary RBF network training). However, the specific obtained values of concentration of contaminants could be improved by hybridizing the RBF result with a LUR method, to improve the resolution of the model. Though it is out of the scope of this paper, and can be included as a future line of research, the hybridization of RBFs networks and LUR methods can be done in the following way:

Usually, LUR models are constructed (trained) from a set of measuring points, using a linear regression method:

\[ y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n. \]

where \( \beta_i \) are the coefficients that must be obtained and characterize the model, \( x_i \) are different land use variables, such as longitude, latitude, distance to road (type of road), population density, distance to sea, even meteorological variables etc., and \( y \) is the measure obtained in a given point. Using all the available measures, the coefficients (\( \beta_i \)) that better describe the LUR can be obtained (training of the system), and then the LUR model is applicable to obtain the concentration of the contaminant in any other required point, using Eq. (26). The idea is that the concentrations obtained with the RBF can be included in the LUR, and modified taking into account LUR parameters such as distance to road, type of road or population density etc. Thus, each estimation of the RBF can be considered as the coefficient \( \beta'_i \) in Eq. (26) (we have a different \( \beta'_i \) for each point), and then the rest of the coefficients are estimated in the standard way.

Another possible line of future research consists of including sources of contaminations in the study. This can be done by considering the transport network as the permanent source of contamination of different intensity. In this respect, existing models of spatial distribution of poisonous substances in case of chemical accident (used for risk assessment) could be useful to model an initial spatial distribution of contaminants. This initial distribution could be included in the RBF network to improve its performance.

Pollution pattern surfaces obtained with RBF (modified by LUR, or including models of gas dispersion to improve them if possible) have different uses: for example re-structuring the air quality monitoring network (see [1]), static analysis of pollutants in the city, and also more specific applications, such as specific actuations on parking regulation following iso-pollution zones, in such a way that to park the car is more expensive in highly NO\(_x\) polluted areas (mainly in the center of the city).

5. Conclusions

In this paper we study the performance of Radial Basis Function (RBF) networks to obtain a first spatial regression of pollutants in urban areas of large cities, and specifically in the city of Madrid. In particular, RBF networks have been used to perform an initial spatial regression of NO\(_x\) and O\(_3\), from the data obtained in the air quality monitoring network and meteorological network of the city. RBF networks with Gaussian kernel have been considered, and different training algorithms have been tested, both classical algorithms and emergent approaches, such as evolutionary-based training. The best quality initial regression for both pollutants in Madrid has been obtained with an evolutionary-based training algorithm, the Evolutionary Programming (EP) approach. The spatial regression maps obtained using the RBF trained with the EP algorithm can be used then as initial points for Land Use Regression methods, in order to improve the resolution of the model proposed. Also, it would be possible to include existing models of spatial distribution of poisonous substances applied in case of chemical accident, and considering the transport network as permanent source of pollution, in order to improve the results of the RBF networks. Several applications of spatial regression such as specific actuation to reduce or control contamination, selection of parking regulation areas, etc., have been discussed in the paper.

Acknowledgement

The authors would like to thank the Air Quality Area of Madrid City Council for their support in this investigation. This work has been partially supported by Comunidad de Madrid, Universidad de Alcalá, through Project CCG07-UAH/AMB-3993. E. G. Ortiz-García is supported by Universidad de Alcalá through a FPI grant. A. M. Pérez-Bellido is supported by a doctoral fellowship by the European Social Fund and Junta de Comunidades de Castilla la Mancha, in the frame of the Operating Programme ESF 2007–2013.

References

[6] E. Aguirre-Basurko, C. Ibáñez-Berastegui, I. Madariaga, Regression and multilayer perceptron-based methods models to forecast hourly \( \mathrm{O}_3 \) and \( \mathrm{NO}_x \) levels in the Bilbao area, Environmental Modelling & Software 21 (2006) 430–446.


