Artificial neural networks to forecast air pollution

Chapter Number

Artificial Neural Networks to Forecast Air Pollution

Eros Pasero, Luca Mesin
Dipartimento di Elettronica, Politecnico di Torino
Italy

1. Introduction

European laws concerning urban and suburban air pollution requires the analysis and implementation of automatic operating procedures in order to prevent the risk for the principal air pollutants to be above alarm thresholds (e.g. the Directive 2002/3/EC for ozone or the Directive 99/30/CE for the particulate matter with an aerodynamic diameter of up to 10 μm called PM_{10}). As an example of European initiative to support the investigation of air pollution forecast, the COST Action ES0602 (Towards a European Network on Chemical Weather Forecasting and Information Systems) provides a forum for standardizing and benchmarking approaches in data exchange and multi-model capabilities for air quality forecast and (near) real-time information systems in Europe, allowing information exchange between meteorological services, environmental agencies, and international initiatives. Similar efforts are also proposed by the National Oceanic and Atmospheric Administration (NOAA) in partnership with the United States Environmental Protection Agency (EPA), which are developing an operational, nationwide Air Quality Forecasting (AQF) system.
Critical air pollution events frequently occur where the geographical and meteorological conditions do not permit an easy circulation of air and a large part of the population moves frequently between distant places of a city. These events require drastic measures such as the closing of the schools and factories and the restriction of vehicular traffic. Indeed, many epidemiological studies have consistently shown an association between particulate air pollution and cardiovascular (Brook et al., 2007) and respiratory (Pope et al., 1991) diseases. The forecasting of such phenomena with up to two days in advance would allow taking more efficient countermeasures to safeguard citizens' health.
Air pollution is highly correlated with meteorological variables (Cogliani, 2001). Indeed, pollutants are usually entrapped into the planetary boundary layer (PBL), which is the lowest part of the atmosphere and has behaviour directly influenced by its contact with the ground. It responds to surface forcing in a timescale of an hour or less. In this layer, physical quantities such as flow velocity, temperature, moisture and pollutants display rapid fluctuations (turbulence) and vertical mixing is strong.
Different automatic procedures have been developed to forecast the time evolution of the concentration of air pollutant, using also meteorological data. Mathematical models of the

advection (the transport due to the wind) and the pollutant reactions have been proposed. For example, the European Monitoring and Evaluation Programme (EMEP) model was devoted to the assessment of the formation of ground level ozone, persistent organic pollutants, heavy metals and particulate matters; the European Air Pollution Dispersion (EURAD) model simulates the physical, chemical and dynamical processes which control emission, production, transport and deposition of atmospheric trace species, providing concentrations of these trace species in the troposphere over Europe and their removal from the atmosphere by wet and dry deposition (Hass et al., 1995; Memmesheimer et al., 1997); the Long-Term Ozone Simulation (LOTOS) model simulates the 3D chemistry transport of air pollution in the lower troposphere, and was used for the investigation of different air pollutions, e.g. total PM_{10} (Manders et al. 2009) and trace metals (Denier van der Gon et al., 2008). Forecasting the diffusion of the cloud of ash caused by the eruption of a volcano in Iceland on April 14th 2010 is finding great attention recently. Airports have been blocked and disruptions to flight from and towards destinations affected by the cloud have already been experienced. Moreover, a threatening effect on European economy is expected.

Real time and low cost local forecasting can be performed on the basis of the analysis of a few time series recorded by sensors measuring meteorological data and air pollution concentrations. In this chapter, we are concerned with specific methods to perform this kind of local prediction methods, which are generally based on the following steps.

a) Information detection through specific sensors and sampled at a sufficient high frequency (above Nyquist limit).
b) Pre-processing of raw time series data (e.g. noise reduction), event detection, extraction of optimal features for subsequent analysis.
c) Selection of a model representing the dynamics of the process under investigation.
d) Choice of optimal parameters of the model in order to minimize a cost function measuring the error in forecasting the data of interest.
e) Validation of the prediction, which guides the selection of the model.

Steps c)-e) are usually iterated in order to optimize the modelling representation of the process under study. Possibly, also feature selection, i.e. step b), may require an iterative optimization in light of the validation step e).

Important data for air pollution forecast are the concentration of the principal air pollutants (Sulphur Dioxide SO_2, Nitrogen Dioxide NO_2, Nitrogen Oxides NO_x, Carbon Monoxide CO, Ozone O_3 and Particulate Matter PM_{10}) and meteorological parameters (air temperature, relative humidity, wind velocity and direction, atmospheric pressure, solar radiation and rain). We provide an example of application based on data measured every hour by a station located in the urban area of the city of Goteborg, Sweden (Goteborgs Stad Miljo). The aim of the analysis is the medium-term forecasting of the air pollutants mean and maximum values by means of meteorological actual and forecasted data. In all the cases in which we can assume that the air pollutants emission and dispersion processes are stationary, it is possible to solve this problem by means of statistical learning algorithms that do not require the use of an explicit prediction model. The definition of a prognostic dispersion model is necessary when the stationarity conditions are not verified. It may happen for example when it is needed to forecast the evolution of air pollutant concentration due to a large variation of the emission of a source or to the presence of a new source, or when it is needed to evaluate a prediction in an area where there are no measurement points.

The best subset of features that are going to be used as the input to the forecasting tool should be selected. The potential benefits of the features selection process are many: facilitating data visualization and understanding, reducing the measurement and storage requirements, reducing training and utilization times, defying the curse of dimensionality to improve prediction or classification performance. It is important to stress that the selection of the best subset of features useful for the design of a good predictor is not equivalent to the problem of ranking all the potentially relevant features. In fact the problem of features ranking is sub-optimum with respect to features selection especially if some features are redundant or unnecessary. On the contrary a subset of variables useful for the prediction can count out a certain number of relevant features because they are redundant (Guyon and Elisseeff, 2003). Depending on the way the searching phase is combined with the prediction, there are three main classes of feature selection algorithms.

1. Filters are defined as feature selection algorithms using a performance metric based entirely on the training data, without reference to the prediction algorithm for which the features are to be selected. In the application discussed in this chapter, features selection was performed using a filter. More precisely a selection algorithm with backward eliminations was used. The criterion used to eliminate the features is based on the notion of relative entropy (also known as the Kullback-Leibler divergence), inferred by the information theory.
2. Wrapper algorithms include the prediction algorithm in the performance metric. The name is derived from the notion that the feature selection algorithm is inextricable from the end prediction system, and is wrapped around it.
3. Embedded methods perform the selection of the features during the training procedure and are specific of the particular learning algorithm.

The Artificial Neural Networks (ANN) and the Support Vector Machines (SVM) have been often used as a prognostic tool for air pollution (Benvenuto and Marani, 2000; Perez et al., 2000; Božnar et al., 2004; Cecchetti et al., 2004; Slini et al., 2006).

ANNs are interesting for classification and regression purposes due to their universal approximation property and their fast training (if sequential training based on backpropagation is adopted). The performances of different network architectures in air quality forecasting were compared in (Kolehmainen et al., 2001). Self-organizing maps (implementing a form of competitive learning in which a neural network learns the structure of the data) were compared to Multi-layer perceptrons (MLP, dealt with in the following), investigating the effect of removing periodic components of the time series. The best forecast estimates were achieved by directly applying a MLP network to the original data, indicating that a combination of a periodic regression and the neural algorithms does not give any advantage over a direct application of neural algorithms. Prediction of concentration of PM_{10} in Thessaloniki was investigated in (Slini et al., 2006) comparing linear regression, Classification And Regression Trees (CART) analysis (i.e., a binary recursive partitioning technique splitting the data into two groups, resulting in a binary tree, whose terminal nodes represent distinct classes or categories of data), principal component analysis (introduced in Section 2) and the more sophisticated ANNs approach. Ozone forecasting in Athens was performed in (Karatzas et al., 2008), again using ANNs. Another approach in forecasting air pollutant was proposed in (Marra et al., 2003), by the use of a combination of the theories of ANN and time delay embedding of a chaotic dynamical system (Kantz & Schreiber, 1997).

SVMs are a statistical learning technique, based on the computational learning theory, which face the problem of minimization of the structural risk (Vapnik, 1995). An online method based on an SVM model was introduced in (Wang et al., 2008) to predict air pollutant levels in a time series of monitored air pollutant in Hong Kong downtown area.
Even if we refer to ANN and SVM approaches as black-box methods, in as much as they are not based on an explicit model, they have generalization capabilities that make possible their application to not-stationary situations.
The combination of the predictions of a set of models to improve the final prediction represents an important research topic, known in the literature as stacking. A general formalism that describes such a technique can be found in (Wolpert, 1992). This approach consists of iterating a procedure that combines measurements data and data which are obtained by means of prediction algorithms, in order to use them all as the input to a new prediction algorithm. This technique was used in (Canu and Rakotomamonjy, 2001), where the prediction of the ozone maximum concentration 24 hours in advance, for the urban area of Lyon (France), was implemented by means of a set of non-linear models identified by different SVMs. The choice of the proper model was based on the meteorological conditions (geopotential label). The forecasting of ozone mean concentration for a specific day was carried out, for each model, taking as input variables the maximum ozone concentration and the maximum value of the air temperature observed on the previous day together with the maximum forecasted value of the air temperature for that specific day.
In this chapter, the theory of time series prediction by ANN and SVM is briefly introduced, providing an example of application to air pollutant concentration. The following sections are devoted to the illustration of methods for the selection of features (Section 2), the introduction of ANNs and SVMs (Section 3), the description of a specific application to air pollution forecast (Section 4) and the discussion of some conclusions (Section 5).

2. Feature Selection

The first step of the analysis was the selection of the most useful features for the prediction of each of the targets relative to the air-pollutants concentrations. The database considered for the specific application discussed in Section 4 was based on meteorological and air pollutant information sampled for the time period 01/04÷10/05. For each air pollutant, the target was chosen to be the mean value over 24 hours, measured every 4 hours (corresponding to 6 daily intervals a day). The complete set of features on which was made the selection, for each of the available parameters (air pollutants, air temperature, relative humidity, atmospheric pressure, solar radiation, rain, wind speed and direction), consisted of the maximum and minimum values and the daily averages of the previous three days to which the measurement hour and the reference to the week day were added. Thus the initial set of features, for each air-pollutant, included 130 features. From this analysis an apposite set of data was excluded; such a set was used as the test set.
Popular methods for feature extraction from a large amount of data usually require the selection of a few features providing different and complementary information. Different techniques have been proposed to individuate the minimum number of features that preserve the maximum amount of variance or of information contained in the data.
Principal Component Analysis (PCA), also known as Karhunen-Loeve or Hotelling transform provides de-correlated features (Haykin, 1999). The components with maximum

energy are usually selected, whereas those with low energy are neglected. A useful property of PCA is that it preserves the power of observations, removes any linear dependencies between the reconstructed signal components and reconstructs the signal components with maximum possible energies (under the constraint of power preservation and de-correlation of the signal components). Thus, PCA is frequently used for a lossless data compression.
PCA determines the amount of redundancy in the data x measured by the cross-correlation between the different and estimates a linear transformation W, which reduces this redundancy to a minimum. The matrix W is further assumed to have a unit norm, so that the total power of the observations x is preserved.
The first principal component is the direction of maximum variance in the data. The other components are obtained iteratively searching for the directions of maximum variance in the space of data orthogonal to the subspace spanned by already reconstructed principle directions

$$\mathbf{w}_1 = \arg\max_{\|w\|=1} E\left[\left(\mathbf{w}^T\mathbf{x}\right)^2\right] \qquad \mathbf{w}_k = \arg\max_{\|w\|=1} E\left[\left(\mathbf{w}^T\left(\mathbf{x} - \sum_{i=1}^{k-1}(\mathbf{w}_i^T\mathbf{x})\mathbf{w}_i\right)\right)^2\right] \tag{1}$$

The algebraic method for the computation of principal components is based on the correlation matrix of data

$$\hat{\mathbf{R}}_{\mathbf{xx}} = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{bmatrix} \tag{2}$$

where r_{ij} is the correlation between the ith and the jth data. Note that $\hat{\mathbf{R}}_{\mathbf{xx}}$ is real, positive, and symmetric. Thus, it has positive eigenvalues and orthogonal eigenvectors. Each eigenvector is a principal component, with energy indicated by the corresponding eigenvalue.
Independent Component Analysis (ICA) determines features which are statistically independent. It works only if data (up possibly to one component) are not distributed as Gaussian variables. ICA preserves the information contained in the data and, at the same time, minimizes the mutual information of estimated features (mutual information is the information that the samples of the data have on each others). Thus, also ICA is useful in data compression, usually allowing higher compression rates than PCA.
ICA, like as PCA, performs a linear transformation between the data and the features to be determined. Central limit theorem guarantees that a linear combination of independent non-Gaussian random variables has a distribution that is "closer" to a Gaussian than the distribution of any individual variable. This implies that the samples of the vector of data x(t) are "more Gaussian" than the samples of the vector of features s(t) that are assumed to be non Gaussian and linearly related to the measured data x(t). Thus, the feature estimation can be based on minimization of Gaussianity of reconstructed features with respect to the possible linear transformation of the measurements x(t). All that we need is a measure of (non)Gaussianity, which is used as an objective function by a given numerical optimization

technique. Many different measures of Gaussianity have been proposed. Some examples are the followings.

1. Kurtosis of a zero-mean random variable v is defined as

$$K(v) = E[v^4] - 3E[v^2]^2 \tag{3}$$

where E[] stands for mathematical expectation, so that it is based on 4th order statistics. Kurtosis of a Gaussian variable is 0. For most non-Gaussian distributions, kurtosis is non-zero (positive for supergaussian variables, which have a spiky distribution, or negative for subgaussian variables, which have a flat distribution).

2. Negentropy is defined as the difference between the entropy of the considered random variable and that of a Gaussian variable with the same covariance matrix. It vanishes for Gaussian distributed variables and is positive for all other distributions. From a theoretical point of view, negentropy is the best estimator of Gaussianity (in the sense of minimal mean square error of the estimators), but has a high computational cost as it is based on estimation of probability density function of unknown random variables. For this reason, it is often approximated by kth order statistics, where k is the order of approximation (Hyvarinen, 1998).
3. Mutual Information between M random variables is defined as

$$I(y_1,...,y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}) \tag{4}$$

where $\mathbf{y} = [y_1,...,y_m]$ is a M-dimensional random vector, and the information entropy is defined as

$$H(\mathbf{y}) = \sum_{i=1}^{m} -\mathrm{P}(\mathbf{y} = \mathbf{a}_i) \log \mathrm{P}(\mathbf{y} = \mathbf{a}_i) \tag{5}$$

Mutual information is always nonnegative, and equals zero only when variables $y_1,...,y_m$ are independent. Maximization of negentropy is equivalent to minimization of mutual information (Hyvarinen & Oja, 2000).

For the specific application provided below, the algorithm proposed in (Koller and Sahami, 1996) was used to select an optimal subset of features. The mutual information of the features is minimized, in line with ICA approach. Indicate the set of structural features as $F = (F_1, F_2,..., F_N)$; the set of the chosen targets is $Q = (Q_1, Q_2,..., Q_M)$. For each assignment of values $f = (f_1, f_2,..., f_N)$ to F, we have a probability distribution P(Q | F = f) on the different possible classes, Q. We want to select an optimal subset G of F which fully determines the appropriate classification. We can use a probability distribution to model the classification function. More precisely, for each assignment of values $g = (g_1, g_2,..., g_P)$ to G we have a probability distribution P(Q | G = g) on the different possible classes, Q. Given an instance f=(f_1, f_2,..., f_N) of F, let f_G be the projection of f onto the variables in G. The goal of

the Koller-Sahami algorithm is to select G so that the probability distribution P(Q | F = f) is as close as possible to the probability distribution P(Q | G = f_G).
To select G, the algorithm uses a backward elimination procedure, where at each state the feature F_i which has the best Markov blanket approximation M_i is eliminated (Pearl, 1988). A subset M_i of F which does not contain F_i is a Markov blanket for F_i if it contains all the information provided by F_i. This means that F_i is a feature that can be excluded if the Markov blanket M_i is already available, as F_i does not provide any additional information with respect to what included in M_i

$$P(Q \mid M_i, F_i) = P(Q \mid M_i). \tag{6}$$

In order to measure how close M_i is to being a Markov blanket for F_i, the Kullback-Leibler divergence (Hyvarinen, 1999) was considered. The Kullback-Leibler divergence can be seen as a measure of a distance between probability density functions, as it is nonnegative and vanishes if and only if the two probability densities under study are equal. In the specific case under consideration, we have

$$\delta_G(\boldsymbol{F_i} \mid \boldsymbol{M_i}) = \sum_{f_{M_i}, f_i} P(\boldsymbol{M_i} = f_{M_i}, \boldsymbol{F_i} = f_i) \cdot \sum_{\boldsymbol{Q_i} \in \boldsymbol{Q}} P(\boldsymbol{Q_i} \mid \boldsymbol{M_i} = f_{M_i}, \boldsymbol{F_i} = f_i) \cdot log \frac{P(\boldsymbol{Q_i} \mid \boldsymbol{M_i} = f_{M_i}, \boldsymbol{F_i} = f_i)}{P(\boldsymbol{Q_i} \mid \boldsymbol{M_i} = f_{M_i})}. \tag{7}$$

The computational complexity of this algorithm is exponential only in the size of the Markov blanket, which is small. For the above reason we could quickly estimate the probability distributions $P(\boldsymbol{Q_i} \mid \boldsymbol{M_i} = f_{M_i}, \boldsymbol{F_i} = f_i)$ and $P(\boldsymbol{Q_i} \mid \boldsymbol{M_i} = f_{M_i})$ for each assignment of values f_{M_i} and f_i to M_i and F_i, respectively.
A final problem in computing Eq. (7) is the estimation of the probability density functions from the data. Different methods have been proposed to estimate an unobservable underlying probability density function, based on observed data. The density function to be estimated is the distribution of a large population, whereas the data can be considered as a random sample from that population. Parametric methods are based on a model of density function which is fit to the data by selecting optimal values of its parameters. Other methods are based on a rescaled histogram. For our specific application, the estimate of the probability density was made by using the kernel density estimation or Parzen method (Parzen, 1962; Costa et al., 2003). It is a non-parametric way of estimating the probability density function extrapolating the data to the entire population. If $x_1, x_2, ..., x_n \sim f$ is an independent and identically distributed sample of a random variable, then the kernel density approximation of its probability density function is

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{8}$$

where the kernel *K* was assumed Gaussian with standard deviation *h*. The result is a sort of smoothed histogram for which, rather than summing the number of observations found within bins, small "bumps" (determined by the kernel function) are placed at each observation.

Koller-Sahami algorithm was applied to the selection of the best subset of features useful for the prediction of the average daily concentration of PM_{10} in the city of Goteborg. In fact from the data it was observed that this concentration was often above the limit value for the safeguard of human health (50 μg/m3). The best subset of 16 features turned out to be the followings.

1. Average concentration of PM_{10} on the previous day.
2. Maximum hourly value of the ozone concentration one, two and three days in advance.
3. Maximum hourly value of the air temperature one, two and three days in advance.
4. Maximum hourly value of the solar radiation one, two and three days in advance.
5. Minimum hourly value of SO_2 one and two days in advance.
6. Average concentration of the relative humidity on the previous day.
7. Maximum and minimum hourly value of the relative humidity on the previous day.
8. Average value of the air temperature three days in advance.

The results can be explained considering that PM_{10} is partly primary, directly emitted in the atmosphere, and partly secondary, that is produced by chemical/physical transformations that involve different substances as SO_x, NO_x, COVs, NH_3 at specific meteorological conditions (see the "Quaderno Tecnico ARPA" quoted in the Reference section).

3. Introduction to Artificial Neural Networks and Support Vector Machines

3.1 Artificial Neural Networks (ANN)

ANNs are biologically inspired models consisting of a complex network of interconnections between basic computational units, called neurons. They found applications in complex tasks like patterns recognition and regression of non linear functions. A single neuron processes multiple inputs applying an activation function on a linear combination of the inputs

$$y_i = \varphi_i\left(\sum_{j=1}^{N} w_{ij} x_j + b_i\right) \tag{8}$$

where $\{x_j\}$ is the set of inputs, w_{ij} is the synaptic weight connecting the jth input to the ith neuron, b_i is a bias, $\varphi_i(\cdot)$ is the activation function, and y_i is the output of the ith neuron considered. Fig. 1A shows a neuron. The activation function is usually non linear, with a sigmoid shape (e.g., logistic or hyperbolic tangent function).

A simple network having the universal approximation property (i.e., the capability of approximating a non linear map as precisely as needed, by increasing the number of parameters) is the feedforward ANN with a single hidden layer, shown in Fig. 1B (for the case of single output, in which we are interested).

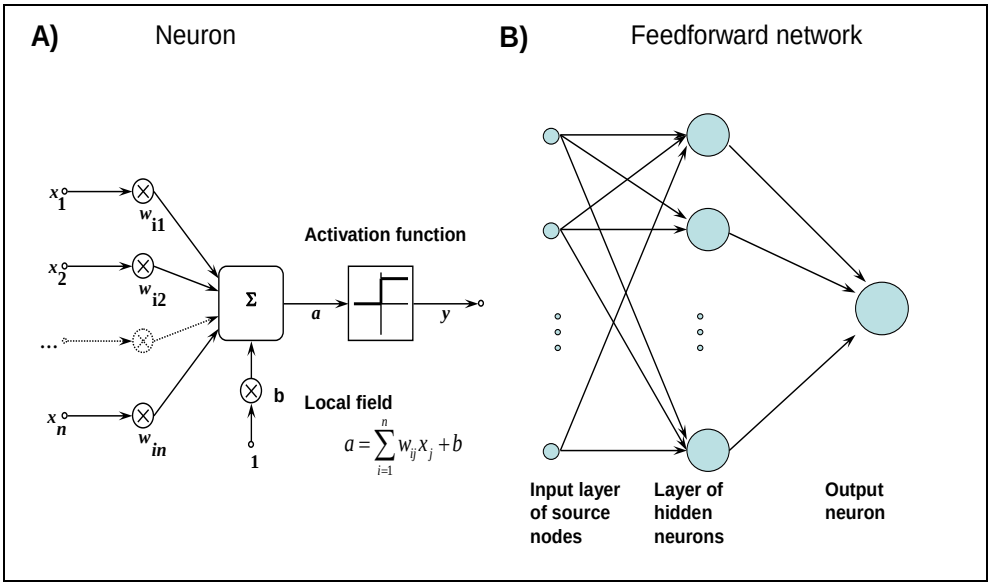

Fig. 1. A) Sketchy representation of an artificial neuron. B) Example of feedforward neural network, with a single hidden layer and a single output neuron.

An ANN may learn a task based on a training set, which is a collection of pairs $\{\vec{x}_k, d_k\}$, where $\vec{x}_k$ is an input vector and d_k is the corresponding desired output. The parameters of the network (synaptic weights and bias) can be chosen optimally in order to minimize a cost function which measures the error in mapping the training input vectors to the desired outputs. Different methods were investigated to avoid to be entrapped in a local minimum. Different cost functions have also been proposed to speed up the convergence of the optimization, to introduce a-priori information on the non linear map to be learned or to lower the computational and memory load. For example, the cost function could be computed for each sample of the training set sequentially for each step of iteration of the optimization algorithm (sequential mode) instead of defining the total cost, based on the whole training set (batch mode). An ANN is usually trained by updating the weights in the direction of the gradient of the cost function. The most popular algorithm is backpropagation, which is a stochastic (i.e., sequential mode) gradient descent algorithm for which the errors (and therefore the learning) propagate backwards from the output nodes to the inner nodes.

The Levenberg-Marquardt algorithm (Marquardt, 1963) was used in this study to predict air pollution dynamics for the application described in Section 4. It is an iterative algorithm to estimate the vector of synaptic weights $\vec{w}$ (a single output neuron is considered) of the model (8), minimising the sum of the squares of the deviation between the predicted and the target values

$$E(\vec{w}) = \sum_{i=1}^{N} \left(d_i - y(\vec{x}_i, \vec{w}) \right)^2 \tag{9}$$

where a batch mode is considered in (9). In each iteration step, the synaptic weights are updated $\vec{w} \to \vec{w} + \vec{\delta}$. In order to estimate the update vector $\vec{\delta}$, the output of the network is approximated by the linearization

$$y(\vec{x}_i, \vec{w} + \vec{\delta}) \simeq y(\vec{x}_i, \vec{w}) + J_i \vec{\delta} \tag{10}$$

where J_i is the gradient

$$J_i = \frac{\partial y(\vec{x}_i, \vec{w})}{\partial \vec{w}} \tag{10}$$

Correspondingly, the square error can be approximated by

$$E(\vec{w} + \vec{\delta}) \simeq \sum_{i=1}^{N} \left(d_i - y(\vec{x}_i, \vec{w}) - J_i \vec{\delta} \right)^2 = \left\| \vec{d} - y(\vec{x}, \vec{w}) - J\vec{\delta} \right\|^2 \tag{11}$$

The choice of update $\vec{\delta}$ minimizing (11) is obtained by pseudoinversion of the matrix J

$$\vec{\delta}_{opt} = (J^T J)^{-1} J^T (\vec{d} - y(\vec{x}, \vec{w})) \tag{12}$$

Levenberg suggested introducing a regularization term (damping factor λ)

$$(J^T J + \lambda I)\vec{\delta}_{opt} = J^T (\vec{d} - y(\vec{x}, \vec{w})) \tag{13}$$

If reduction of the square error E is rapid, a smaller damping can be used, bringing the algorithm closer to the Gauss-Newton algorithm, whereas if the iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the gradient descent direction (indeed the gradient of the error is $-2(J^T(\vec{d} - y(\vec{x}, \vec{w})))^T$). To avoid slow convergence in the direction of small gradients, Marquardt suggested scaling each component of the gradient according to the curvature so that there is larger movement along the directions where the gradient is smaller

$$(J^T J + \lambda diag(J^T J))\vec{\delta}_{opt} = J^T (\vec{d} - y(\vec{x}, \vec{w})) \tag{14}$$

where $J^T J$ was considered as an approximation of the Hessian matrix of the approximating function $y(\vec{x}, \vec{w})$.

For prediction purposes, time is introduced in the structure of the neural network. For one step ahead prediction, the desired output d_n at time step n is a correct prediction of the value attained by the time series at time *n+1*

$$y_n = \hat{x}_{n+1} = \varphi\left(\vec{w} \cdot \vec{x} + b \right) \tag{15}$$

where the vector of regressors $\vec{x}$ includes information available up to the time step n. A number of delayed values of the time series up to time step n can be used together with additional data from other measures (non linear autoregressive with exogenous inputs model, NARX; Sjöberg et al., 1994). Such values may also be filtered (e.g., using a FIR filter). More generally, interesting features extracted from the data using one of the methods described in Section 2 may be used. Moreover, previous outputs of the network (i.e., predicted values of the states/features) may be used (non linear output error model, NOE). This means introducing a recursive path connecting the output of the network to the input. Other recursive topologies have also been proposed, e.g. a connection between the hidden layer and the input (e.g. the simple recurrent networks introduced by Elman, connecting the state of the network defined by the hidden neurons to the input layer; Haykin, 1999).

3.2 Support Vector Machines (SVM)

SVMs constitute a powerful tool for supervized classification. They were first introduced to separate optimally two linearly separable classes. As shown in Fig. 2A, the two sets of points (filled and unfilled points belonging to two different classes), also interpretable as two dimensional vectors, may be separated by a line (in the case of multidimensional vectors, a separation hyperplane is required). Multiple solutions are possible. We consider optimal the solution that maximizes the margin, i.e. the width that the boundary could be increased by before hitting a datapoint, which is also the distance between the two vectors (called support vectors and indicated with x^+ and x^- in Fig. 2B) belonging to each of the two classes placed closest to the separation line.

The problem can be stated as: given the training pairs $\{x_i, y_i = \pm 1\}$ (where the vectors x_i are associated to the class $+1$ or to -1 indicated by the corresponding value of y_i), find the line $\vec{w} \cdot \vec{x} + b = 0$ separating the two classes, which can be obtained by imposing

$$y_i(wx_i + b) \geq 1 \tag{16}$$

where the vector sign was dropped (as in Fig. 2) to simplify notation and we considered that the parameters w and b can be scaled in order that for the support vectors we have $wx^+ + b = 1$ and $wx^- + b = -1$. From these conditions, the margin is given by

$$M = \frac{2}{|w|} \tag{17}$$

so that the following constrained optimization problem can be stated

$$\textit{Minimize} \quad \Phi(w) = \frac{1}{2} w^T w \quad \textit{subject to} \quad y_i(wx_i + b) \geq 1 \tag{17}$$

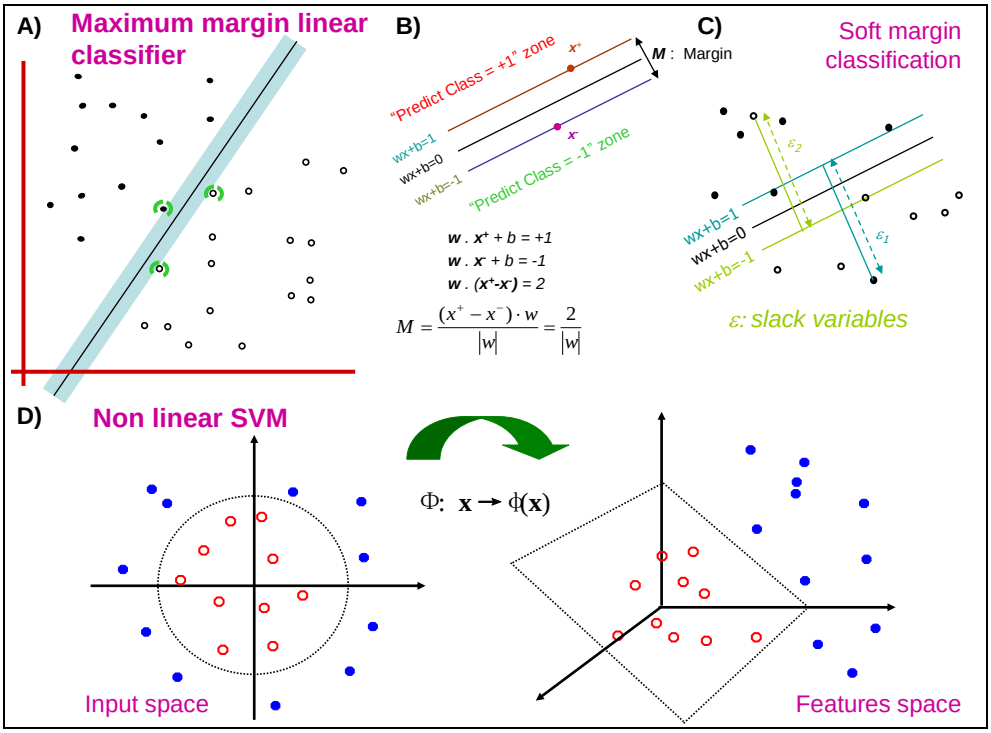

Fig. 2. A) Sketchy representation of support vector machines for solving linear classification problems, with hard or soft margins. B) Support vector machines for non linear classification.

The problem can be solved by determining the saddle point of the Lagrangian

$$J(w,b,\alpha)=\frac{1}{2}w^{T}w-\sum_{i=1}^{N}\alpha_{i}\left[y_{i}(wx_{i}+b)-1\right] \tag{18}$$

by minimizing $J(w,b,\alpha)$ with respect to w, b and maximizing with respect to the non negative Lagrange multipliers α_i (Haykin, 1999). Determining the stationary points of the Lagrangian, only some α_i, *i=1,…,m* are non vanishing and indicate that the corresponding x_i are support vectors. The corresponding constraint is said to be active, which means that the equality sign in the inequality constraint in problem (17) is attained. The following classifying function is obtained

$$f(\vec{x})=\vec{w}\cdot\vec{x}+b \quad \textit{where} \quad \vec{w}=\sum_{i=1}^{m}\alpha_{i}y_{i}\vec{x}_{i},\ b=y_{k}-\vec{w}\cdot\vec{x}_{k} \text{ for an arbitrary support vector } \vec{x}_{k} \tag{19}$$

A generalization is required to apply SVMs to the case of not linearly separable classes. Suppose that two linearly separable classes are corrupted by additive noise that determines the jump of the optimal separation line by a few outliers, as shown in Fig. 2C. A soft margin is introduced to allow for misclassification of a few datapoints. The distance from the misclassified points to the separation line (slack variables) is penalized by adding a regularization term to the cost function to be minimized and weakening the constraint of the optimization problem

$$\textit{Minimize} \quad \Phi'(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{s} \varepsilon_i \qquad \textit{subject to} \;\; y_i (w x_i + b) \geq 1 - \varepsilon_i \,,\; \varepsilon_i \geq 0 \tag{20}$$

where *C* is the regularization parameter to be selected by the user to give the proper weight to the misclassifications and ε_i are the slack variables.

If the two classes are non linearly intermixed, introducing slack variables is not sufficient. An additional method is to map the input space into a feature space in which linear separation is feasible (Fig. 2D)

$$x \in \mathbb{R}^N \rightarrow \varphi(x) \in \mathbb{R}^F \tag{21}$$

Cover's theorem (Haykin, 1999) indicates that the probability of getting linear separability is high if the function mapping the input space into the feature space is non linear and if the feature space has a high dimension (much larger than the input space, $F \gg N$). The linear classification is performed in the feature space as before, obtaining the following classification map which resembles the equivalent expression (19) obtained for the linearly separable classes

$$f(\vec{x}) = \vec{w} \cdot \varphi(\vec{x}) + b \quad \textit{where} \quad \vec{w} = \sum_{i=1}^{m} \alpha_i y_i \varphi(\vec{x}_i),\; b = y_k - \vec{w} \cdot \varphi(\vec{x}_k) \tag{22}$$

where slack variables were not included for simplicity. Comparing the linear and the non linear separation problems, the following inner-product kernel appears

$$K(\vec{x}_i, \vec{x}) = \varphi(\vec{x}_i) \cdot \varphi(\vec{x}) \tag{23}$$

which allows writing the classification map as

$$f(\vec{x}) = \sum_{i=1}^{m} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \tag{24}$$

Different kernels have been applied (Gaussian, polynomial, sigmoidal), with parameters to be chosen in order to optimize the classification performance.

A straightforward generalization to multi-class separation is possible, by solving multiple two-class problems.
Moreover, SVMs may be applied to solve regression problems, which are of interest in the case of air pollution prediction. The following ε - insensitive loss function is introduced to quantify the error in approximating a desired response d using a SVM with output y

$$L_\varepsilon(d,y)=\begin{cases}|d-y|-\varepsilon & if\ |d-y|\geq\varepsilon\\ 0 & otherwise\end{cases} \tag{25}$$

The following non linear regression model

$$d=f(x) \tag{26}$$

is optimized on the basis of a training set $\{\vec{x}_k,d_k\}$. The estimate of d is expressed as the linear combination of a set of non linear basis functions

$$y=\sum_{i=0}^{N}w_i\varphi_i(\vec{x})+b=\vec{w}\cdot\vec{\varphi}(\vec{x})+b \tag{27}$$

The weight vector and the bias are chosen in order to minimize the empirical risk

$$R_{emp}=\frac{1}{N}\sum_{i=1}^{N}L_\varepsilon(d_i,y_i) \tag{25}$$

The problem can be recast in terms of the formulism of SVM, by introducing two sets of non negative slack variables and writing the following constrained optimization problem

$$Minimize\quad \Phi(w)=\frac{1}{2}w^Tw+C\sum_{i=1}^{s}\hat{\varepsilon}_i+\breve{\varepsilon}_i\qquad subject\ to\ \begin{cases}d_i-\vec{w}\cdot\vec{\varphi}(x_i)-b\leq\varepsilon+\hat{\varepsilon}_i\\ d_i-\vec{w}\cdot\vec{\varphi}(x_i)-b\geq\varepsilon+\breve{\varepsilon}_i\end{cases} \tag{20}$$

4. Forecasting when the Concentrations are above the Limit Values for the Protection of Human Health

A set of feedforward neural networks with the same topology was used. Each network had three layers with 1 neuron in the output layer and a certain number of neurons in the hidden layer (varying in a range between 3 and 20). The hyperbolic tangent function was used as activation function. The backpropagation rule (Werbos, 1974) was used to adjust the weights of each network and the Levenberg-Marquardt algorithm (Marquardt, 1963) to proceed smoothly between the extremes of the inverse-Hessian (or Gauss-Newton) method and the steepest descent method. The Matlab Neural Network Toolbox (Demuth and Beale, 2005) was used to implement the neural networks.

An SVM with an ε-insensitive loss function (Vapnik, 1995) was also used. A Gaussian kernel function was used. The principal parameters of the SVM were the regularized constant C determining the trade-off between the training error and model flatness, the width value σ of the Gaussian kernel, and the width ε of the tube around the solution. The SVM performance was optimized choosing the proper values for such parameters. An active set method (Fletcher, 1987) was used as optimization algorithm for the training of the SVM.
The neural networks were trained on a representative subset of the data used for the features selection algorithm. A subset of the first two years of data was used: a measurement sample every three samples after leaving out one sample out of five of the original data. In this way the computational time of the adopted machine-learning algorithms was reduced while obtaining a subset of data as representative as that used for the features selection. In fact such a subset included a sufficient number of all the 6 daily intervals in which the measurement data were divided by our analysis. The test set consisted of the data not used for the features selection algorithm. Since the number of the training samples above the maximum threshold for the PM_{10} concentration was much lower than the number of samples under such threshold, the training of the networks was performed weighting more the kind of samples present a fewer number of times.

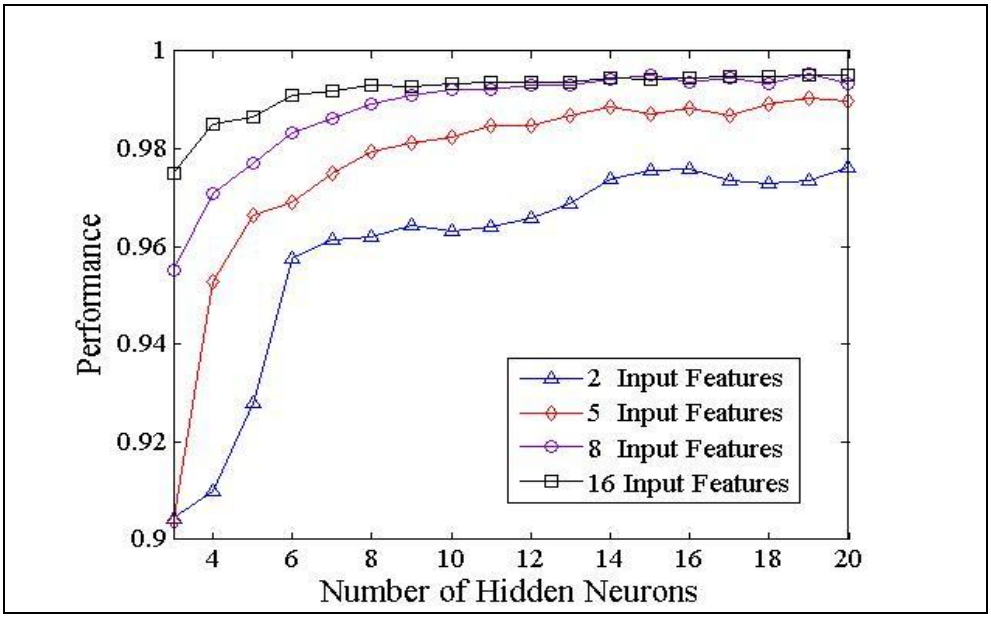

Fig. 3. Performance of the ANN as a function of the number of input Features (samples below the threshold).

As we can see from Fig. 3 and Fig. 4 the ANN performance, both for the samples under the threshold and for the samples above the threshold, increased when the number of input features increased. More precisely the performance increased meaningfully from 2 to 8 input features and tended to flatten when the size of the input vector was greater than 8.
The best subset of 8 features was the following:

1. Average concentration of PM_{10} on the previous day.
2. Maximum hourly value of the ozone concentration one, two and three days in advance.
3. Maximum hourly value of the air temperature on the previous day.
4. Maximum hourly value of the solar radiation one, two and three days in advance.

Selecting as input to the ANN such set of 8 features, the best results could be obtained with a neural network having 18 neurons in the hidden layer. In Table I are displayed the results obtained with 5115 samples of days under the threshold and 61 samples of days above the threshold. It can be noted that the probability to have a false alarm is really low (0.82%) while the capability to forecast when the concentrations are above the threshold is about 80%. Different assignment for SVM parameters ε, σ and C, were tried in order to find the optimum configuration with the highest performance.

Samples	Correct Forecasting	Incorrect Forecasting
Below the threshold	5073	42
Above the threshold	48	13

Table 1. Neural Network performance.

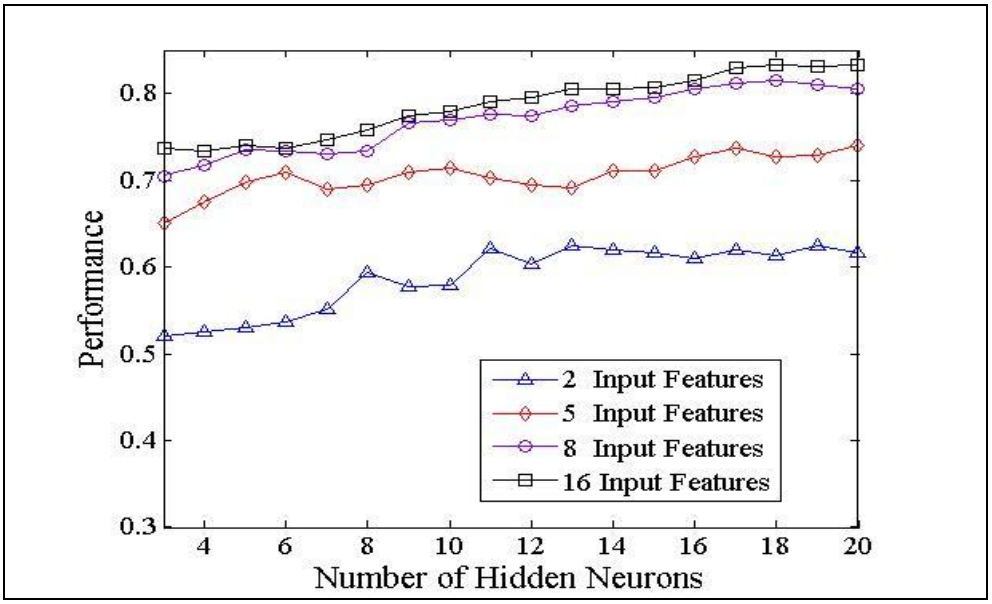

Fig. 4. Performance of the ANN as a function of the number of input Features (samples above the threshold).

As we can see from Fig. 5, when ε and C were kept constant (ε=0.001 and C=1000), the SVM performances referring to samples above the threshold, for a high number of input features, depended on σ and reached a maximum when σ=1, corresponding to an optimum trade-off between SVM generalization capability (large values of σ) and model accuracy with respect to the training data (small values of σ). The value of σ corresponding to this trade-off

decreased to 0.1 for lower values of the input vector size (Fig. 5) and for samples below the threshold (Fig. 6), reflecting the fact that the generalization capability was less important when the training set was more representative.

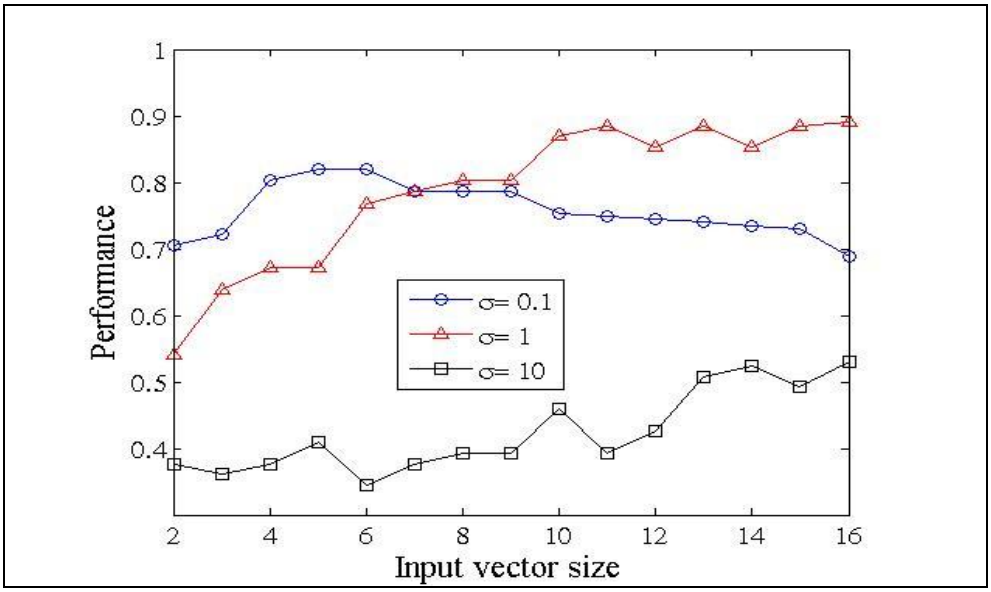

Fig. 5. Performances of the SVM as a function of σ (ε=0.001 and C=1000), samples above the threshold.

When σ and C were kept constant (Fig. 7: σ=1 and C=1000; Fig. 8: σ=0.1 and C=1000), the best performances were achieved when ε was close to 0 and the allowed training error was minimized.

From this observation, by abductive reasoning we could conclude that the input noise level was low. In accordance with such a behavior the performance of the network improved when the parameter C increased from 1 to 1000. Since the results tended to flatten for values of C greater than 1000, the parameter C was set equal to 1000. The best performance of the SVM corresponding to ε=0.001, σ =0.1 and C=1000 was achieved using as input features the best subset of 8 features previously defined. The probability to have a false alarm was really low (0.13%) while the capability to forecast when the concentrations were above the threshold was about 80%. The best performance of the SVM corresponding to ε=0.001, σ =1 and C=1000 was achieved using as input features the best subset of 11 features.

In this case the probability to have a false alarm was higher than in the previous one (0.96%) but the capability to forecast when the concentrations were above the threshold was nearly 90%.

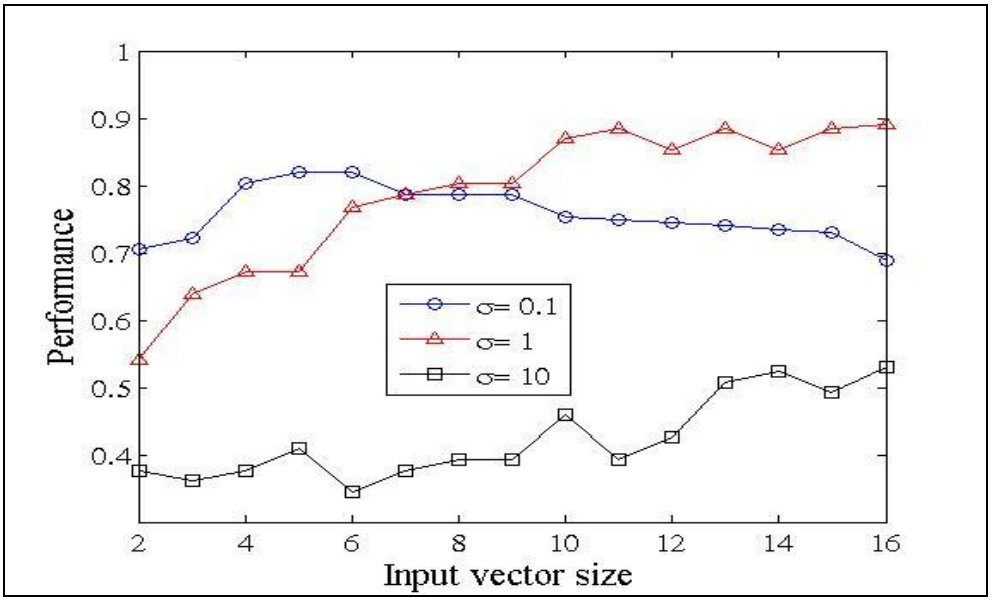

Fig. 6. Performances of the SVM as a function of σ (ε=0.001 and C=1000), samples below the threshold.

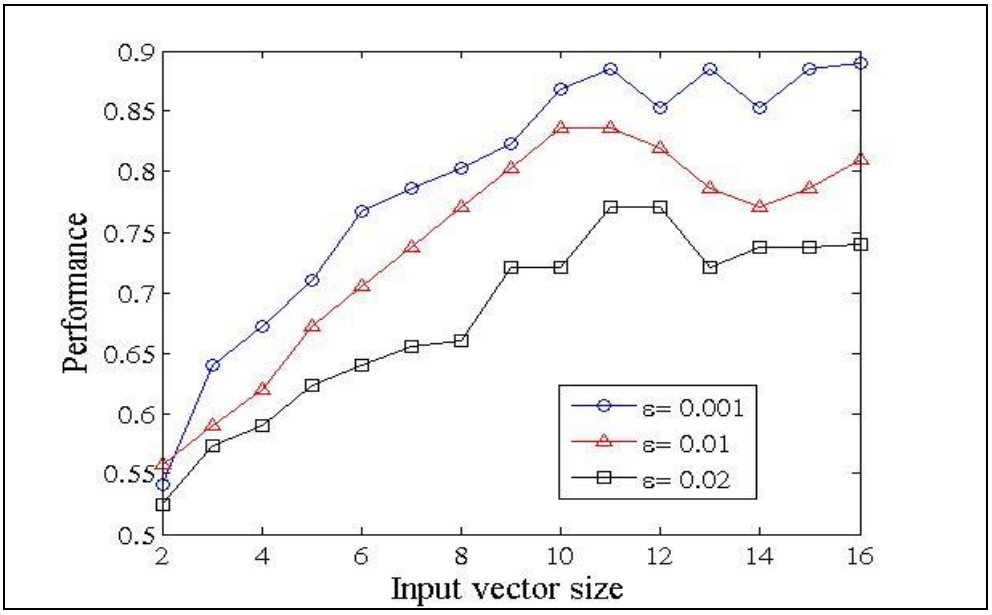

Fig. 7. Performances of the SVM as a function of ε (σ=1 and C=1000), samples above the threshold

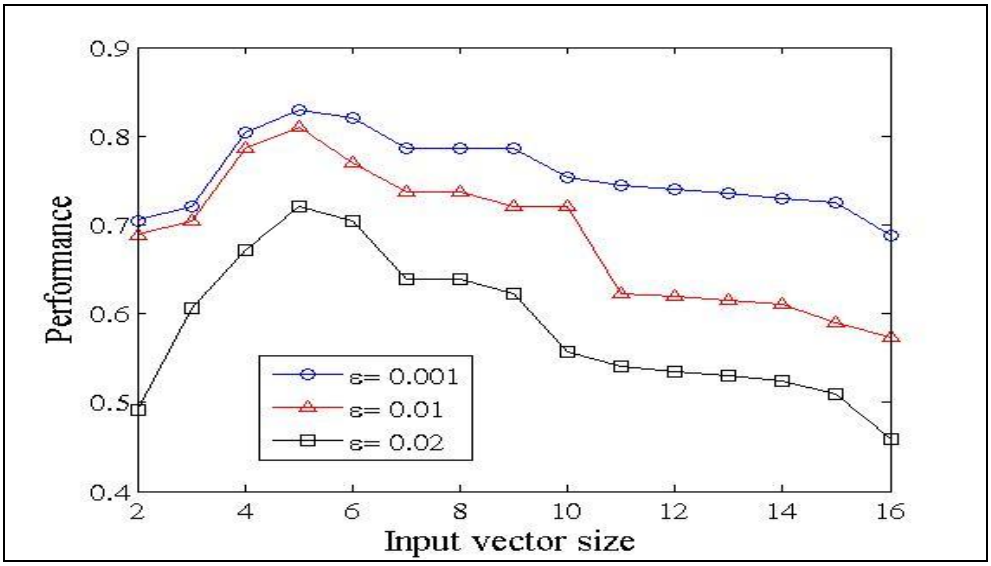

Fig. 8. Performances of the SVM as a function of ε (σ=0.1 and C=1000), samples above the threshold.

5. Discussion and Conclusion

This chapter provides an introduction to non-linear methods for the prediction of the concentration of air pollutants. We focused on the selection of features and the modelling and processing techniques based on the theory of Artificial Neural Networks and Support Vector Machines.

Joint measurements of meteorological data and pollutants concentrations is useful in order to increase the number of parameters to be studied for the construction of mathematical air quality forecasting models and hence to improve forecast performances. Weather variables have a non-linear relationship with air quality, which can be captured by non-linear models such as Artificial Neural Networks and Support Vector Machines.

Our analysis carries on the work already developed by the NeMeFo (Neural Meteo Forecasting) research project for meteorological data short-term forecasting (Pasero et al., 2004). The application provided in Section 4 illustrates how the theoretical methods for feature selection (Section 2) and data modelling (Section 3) can be implemented for the solution of a specific problem of air pollution forecast. The principal causes of air pollution are identified and the best subset of features (meteorological data and air pollutants concentrations) for each air pollutant is selected in order to predict its medium-term concentration (in particular for the PM_{10}). The selection of the best subset of features was implemented by means of a backward selection algorithm which is based on the information theory notion of relative entropy. Artificial Neural Networks and Support Vector Machines constitute some of the most wide-spread statistical data-learning techniques to develop data-driven models. Their use is shown for the prediction problem considered.

In conclusion, the final aim of this research is the implementation of a prognostic tool able to reduce the risk for the air pollutants concentrations to be above the alarm thresholds fixed by the law. The detection of meteorological and air pollutant data, the automatic selection of optimal descriptors of such data and the use of Artificial Neural Networks or Support Vector Machines is proposed as an efficient strategy to perform an accurate prediction of the time evolution of air pollutant concentration.

8. References

Benvenuto, F. & Marani A. (2000). Neural networks for environmental problems: data quality control and air pollution nowcasting, *Global NEST: The International Journal* Vol. 2, No. 3, pp. 281-292, Nov. 2000, ISSN.

Božnar, M. Z. ; Mlakar, P. J. & Grašič, B. (2004). Neural Networks Based Ozone Forecasting, *Proceeding of 9th Int. Conf. on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes*, June 1-4, 2004, Garmisch-Partenkirchen, Germany.

Brook, R.D. ; Rajagopalan, S. ; Jerrett, M. ; Burnett, R.T. ; Kaufman, J.D. ; Miller, K. A. & Sheppard, L. (2007). Air Pollution and Cardiovascular Events. *NEJM* Vol. 356, pp. 2104-2106.

Canu, S. & Rakotomamonjy, A. (2001). Ozone peak and pollution forecasting using Support Vectors, *IFAC Workshop on environmental modelling*, Yokohama, 2001.

Cecchetti, M.; Corani, G & Guariso, G. (2004). Artificial Neural Networks Prediction of PM_{10} in the Milan Area, *Proc. of IEMSs 2004*, University of Osnabrück, Germany, 14-17 June 2004.

Cogliani, E. (2001). Air pollution forecast in cities by an air pollution index highly correlated with meteorological variables, *Atm. Env.*, Vol. 35, No. 16, pp. 2871-2877.

Costa, M.; Moniaci, W. & Pasero, E. (2003). INFO: an artificial neural system to forecast ice formation on the road, *Proceedings of IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications*, pp. 216–221, 29-31 July 2003.

Demuth, H. & Beale, M. (2005). Neural Network Toolbox User's Guide, The MathWorks, Inc., 2005.

Denier van der Gon, H.A.C. ; J.H.J. Hulskotte, A.J.H. ; Visschedijk, M. & Schaap, M. (2007). A revised estimate of copper emissions from road transport in UNECE Europe and its impact on predicted copper concentrations, *Atmos. Env.*, Vol. 41, pp. 8697-8710.

Fletcher, R. (1987). Practical Methods of Optimization, John Wiley & Sons, NY, 2nd ed.

Goteborgs Stad Miljo, http://www.miljo.goteborg.se/luftnet/.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection, *The Journal of Machine Learning Research*, Vol. 3, pp. 1157-1182, January 2003.

Hass, H. ; .Jakobs, H.J. & Memmesheimer, M. (1995). Analysis of a regional model (EURAD) near surface gas concentration predictions using observations from networks, *Meteorol. Atmos. Phys.* Vol. 57, pp. 173–200.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, *Prentice Hall*.

Hyvarinen, A. (1998). New approximations of differential entropy for independent component analysis and projection pursuit, *Advances in Neural Information Processing Systems*, Vol. 10, pp. 273-279. MIT press.

Hyvarinen, A. (1999). Survey on Independent Component Analysis, *Neural Computing Surveys*, Vol. 2, pp. 94-128.

Hyvarinen A. & Oja, E. (2000). Independent Component Analysis: Algorithm and applications, *Neural Networks*, Vol. 13(4-5), pp. 411-430.

Kantz, H. & Schreiber, T. (1997). Nonlinear Time Series Analysis, Cambridge University Press.

Karatzas, K.D.; Papadourakis, G. & Kyriakidis, I. (2008). Understanding and forecasting atmospheric quality parameters with the aid of ANNs, *Proceedings of the International Joint Conference on Neural Networks*, IJCNN 2008, Hong Kong, China, June 1-6, 2008, pp. 2580-2587.

Koller, D. & Sahami, M. (1996). Toward optimal feature selection, *Proceedings of 13th International Conference on Machine Learning (ICML)*, , pp. 284-292, July 1996, Bari, Italy.

Manders, A.M.M. ; Schaap, M. & Hoogerbrugge, R. (2009). Testing the capability of the chemistry transport model LOTOS-EUROS to forecast PM_{10} levels in the Netherlands, *Atmos. Env.*, Vol. 43, No. 26, pp. 4050-4059.

Marquardt, D. (1963). An algorithm for least squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, Vol. 11, pp. 431-441, 1963.

Marra, S. ; Morabito, F.C. & Versaci M. (2003). Neural Networks and Cao's Method: a novel approach for air pollutants time series forecasting, *IEEE-INNS International Joint Conference on Neural Networks*, July 20-24, Portland, Oregon.

Memmesheimer, M. ; Ebel, A. & Roemer, M. (1997). Budget calculations for ozone and its precursors: seasonal and episodic features based on model simulations, *J. Atmos. Chem*. Vol. 28, pp. 283–317.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode, *Annals of Math. Statistics*, Vol. 33, pp. 1065-1076, Sept. 1962.

Pasero, E. ; Moniaci, W. ; Meindl, T. & Montuori, A. (2004). NEMEFO: NEural MEteorological Forecast, *Proceedings of SIRWEC 2004*, 12th International Road Weather Conference, Bingen.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Mateo, CA, 1988.

Perez, P. ; Trier, A. & Reyes, J. (2000). Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile, *Atmospheric Environment*, Vol. 34, pp. 1189-1196, ISSN.

Pope, C.A. ; Dockery, D.W., Spengler J.D. & Raizenne, M.E. (1991). Respiratory health and PM_{10} pollution: A daily time series analysis. *Am Rev Respir Dis* Vol. 144, pp. 668–74.

Quaderno Tecnico ARPA (Emilia Romagna) - SMR n°10-2002.

Sjöberg, J. ; Hjalmerson, H. & L. Ljung (1994). Neural Networks in System Identification. Preprints 10th IFAC symposium on SYSID, Copenhagen, Denmark. Vol.2, pp. 49-71.

Slini, T. ; Kaprara, A. ; Karatzas, K. & Moussiopoulos, N. (2006). PM_{10} forecasting for Thessaloniki, Greece, *Environmental Modelling & Software*, Vol. 21, No. 4, pp. 559-565, April 2006.

Vapnik, V. (1995). The Nature of Statistical Learning Theory, New York, Springer-Verlag.

Werbos, P. (1974). Beyond regression: New tools for Prediction and Analysis in the Behavioural Sciences, Ph.D. Dissertation, *Committee on Appl. Math.*, Harvard Univ. Cambridge, MA, Nov. 1974.

Wang, W. ; Men, C. & Lu, W. (2008). Online prediction model based on support vector machine, *Neurocomputing*, Vol. 71, No. 4-6, pp. 550-558.

Wolpert, D. (1992). Stacked generalization, *Neural Networks*, Vol. 5, pp. 241-259, ISSN.