Available online at www.sciencedirect.com

SCIENCE DIRECT®

Ecological Modelling 185 (2005) 513–529

ECOLOGICAL MODELLING

www.elsevier.com/locate/ecolmodel

Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning

Giorgio Corani*

Dipartimento di Elettronica, Politecnico di Milano, Via Ponzio, 34/5-20133 Milano, Italy

Received 25 May 2004; received in revised form 21 December 2004; accepted 3 January 2005

Abstract

Ozone and PM_{10} constitute the major concern for air quality of Milan. This paper addresses the problem of the prediction of such two pollutants, using to this end several statistical approaches. In particular, feed-forward neural networks (FFNNs), currently recognized as state-of-the-art approach for statistical prediction of air quality, are compared with two alternative approaches derived from machine learning: pruned neural networks (PNNs) and lazy learning (LL). PNNs constitute a parameter-parsimonious approach, based on the removal of redundant parameters from fully connected neural networks; LL, on the other hand, is a local linear prediction algorithm, which performs a local learning procedure each time a prediction is required. All the three approaches are tested in the prediction of ozone and PM_{10}; predictors are trained to return at 9 a.m. the concentration estimated for the current day.

No strong differences are found between the forecast accuracies of the different models; nevertheless, LL provides the best performances on indicators related to average goodness of the prediction (correlation, mean absolute error, etc.), while PNNs are superior to the other approaches in detecting of the exceedances of alarm and attention thresholds. In some cases, data-deseasonalization is found to improve the prediction accuracy of the models.

Finally, some striking features of lazy learning deserve consideration: the LL predictor can be quickly designed, and, thanks to the simplicity of the local linear regressors, it both gets rid of overfitting problems and can be readily interpreted; moreover, it can be also easily kept up-to-date.

Keywords: Feed forward neural networks; Pruned neural networks; Lazy learning; Time series prediction; Atmospheric pollution

1. Introduction

The Milan urban area, located at the center of the Po Valley, is the most industrialized and populated district in Italy. According to the State of the Environment Report (Agenzia Milanese Mobilità Ambiente, 2003), the yearly average of pollutants such as SO_2, NO_x, CO, TSP has decreased of about 90%, 50%, 65%, 60% during the last decade, respectively; no exceedances of alarm and attention thresholds have been observed since 1997 for SO_2 and TSP, and since 1999 for CO.

* Tel.: +39 02 2399 3562; fax: +39 02 2399 3412.
E-mail address: corani@elet.polimi.it (G. Corani).

0304-3800/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ecolmodel.2005.01.008

The situation is slightly worse for NO_x which, though a clear decreasing trend, showed on the average 8 yearly exceedances of the attention threshold (none of the alarm) over the period 1997–2001 (Agenzia Milanese Mobilità Ambiente, 2003).

Also the yearly averages of micro-pollutants such as benzene and lead are largely under the thresholds established for human health protection (Agenzia Milanese Mobilità Ambiente, 2003). Such significant results are due to different interventions, such as the improved formulation of fossil oils for industrial activities, the large adoption of methane for residential heatings, and the introduction of catalytic converters and unleaded petrol, which resulted in considerably lower vehicle emission factors.

A severe health issue is, on the other hand, constituted by high levels of both PM_{10} and ozone; these pollutants, associated in the epidemiological literature with an increase in the mortality and cardiorespiratory hospitalizations (Ostro et al., 1999a,b), constitute the major concern regarding the air quality of the city.

The yearly average of PM_{10} has been substantially stable (about 45 $\mu g/m^3$) since the beginning of monitoring in 1998; in the Milan area, suspended PM_{10} is mainly emitted from vehicular traffic (83%) and residential heating (16%) (Agenzia Milanese Mobilità Ambiente, 2003); moreover, a further significant part (*secondary*) of PM_{10} is produced in the atmosphere because of particular chemical and physical mechanisms.

The very last regional law [Regional Decree 13858 29/7/2003] does not introduce any attention or alarm threshold for PM_{10}; rather, it preventively decrees the blockage, in some periods during the winter, of the pre-Euro vehicles (i.e., first-registered before the introduction of Euro I emissions standards). Nevertheless, the previous law [Regional Decree 6501 19/10/2001] fixed the attention and alarm thresholds to 50 $\mu g/m^3$ and 75 $\mu g/m^3$, respectively; the "*attention state*" was declared if the attention threshold was exceeded for 5 consecutive days. On average, in the last decade, PM_{10} exceeded the attention threshold for about 100 days/year, and about 20 "*attention state days*" have been declared yearly (Agenzia Milanese Mobilità Ambiente, 2003).

Ozone is a secondary pollutant, produced in the atmosphere in presence of high solar radiation and primary pollutants (NO_x and VOC). The regional law [Regional Decree VII/6501 19-10-2001] sets the attention and alarm levels to 180 $\mu g/m^3$ and 360 $\mu g/m^3$, respectively, for the maximum hourly concentration, while the health target on the 8-h moving average is fixed to 110 $\mu g/m^3$. Ozone began to rise in the Milan area in the early 90s, partly as a consequence of the reduced SO_2 and CO concentrations, which caused a more oxidant atmosphere. Since ozone levels strongly depend on meteorology, great variations are observed among different years; however, in the period 1997–2002 about 10 yearly exceedances of the attention threshold have been recorded, and none of the alarm one. In the same period, about 50 yearly exceedances of the health threshold have been observed.

A system able to predict ozone and PM_{10} concentrations with sufficient anticipation, can provide Public Authorities the time required to manage the emergency, for instance, by planning an increase in the public transports in the case of an incoming traffic block, or by issuing early warnings.

Over the last years, black box approaches have been recognized to constitute a viable alternative to conceptual models for input–output simulation and forecasting, and also to allow to shorten the time required for the model development. In particular, neural networks collected a general consensus in predicting different pollutants time series, as shown by the review of Gardner and Dorling (1998).

However, the specific literature about PM_{10} forecasting is quite recent and, as such, still limited; for example, the review of Gardner and Dorling (1998) did not include any PM_{10} application. An attempt in predicting the time series of PM2.5 (Perez et al., 2000) and PM_{10} (Perez and Reyes, 2002) in Santiago, Chile showed better performances of neural networks with respect to linear regressors and persistent predictors.[1] The EU-funded project APPETISE (http://www.uea.ac.uk/env/appetise) carried out an inter-comparison of different air pollution forecasting approaches; the application to PM_{10} prediction in Helsinki (Kukkonen et al., 2003) showed a better forecast accuracy for neural networks than for other approaches, such as linear regressors and also deterministic physically-based models. A further study related to project APPETISE and regarding Helsinki (Zickus et al., 2002) applied four different machine learning

[1] The persistent predictor issues a simple forecast, hypothesizing that the value observed at time $(t+1)$ will equal the measure taken at time (t).

approaches to the prediction of the PM_{10} exceedances of the 50 μg/m^3 threshold. Also in this case, the authors found FFNNs to be one of the best performing approaches; however, they concluded that, since small differences in performances are not important in practical applications, a model which can be more readily interpreted than FFNNs, such as logistic regression, can be chosen as alternative.

The neural network approach proved to be viable also for ozone forecasting, outperforming alternative techniques in different case studies (Prybutok et al., 2000; Balaguer Ballester et al., 2002; Nunnari et al., 1998; Viotti et al., 2002). A remarkable intercomparison of statistical approaches – involving 15 different statistical techniques, 10 different sites across Europe and different forecast horizons – has been carried out (Schlink et al., 2003) in the context of the APPETISE project. The general conclusions drawn by the authors indicated neural networks and generalized additive models as the favored forecasting approaches.

According to this literature review, neural networks represent the state-of-the-art for air pollution statistical prediction; among them, *feed-forward neural networks* have been shown to outperform other types of neural networks, such as neuro-fuzzy networks (Nunnari et al., 1998) and Kohonen self-organizing maps (Kolehmainen et al., 2001) and are therefore often adopted. A recognized drawback of neural networks is however their tendency to overfitting, because of their high parameterization; moreover, their “very black-box” nature makes difficult the assessment of the relevance of the different input variables within the model.

The aim of this work is to assess the suitability of two statistical non-linear modelling approaches, i.e., pruned neural networks and lazy learning, and to compare them with feed-forward neural networks in predicting ozone and PM_{10} in Milan.

Pruned neural networks constitute an important research field in the machine learning area (see Reed, 1993 for a review); nevertheless, they are still not widely used for applications. The basic idea of pruning algorithms is to remove the redundant parameters from a fully connected neural network; pruned networks can contain one order of magnitude less parameters than fully connected ones and, as such, are no longer prone to overfitting, thus providing a parameter parsimonious neural networks approach. Pruned networks have been used in fields such as chemical processes identification (Henrique et al., 2000), predictive microbiology (García-Gimeno et al., 2002), near infrared spectroscopy (Poppi and Massart, 1998) and water resources (Castelli et al., 2003); in Cecchetti et al. (2004), they have been applied to the prediction of air quality and have been shown to constitute a viable alternative to fully connected neural networks.

Lazy learning is a local linear modelling approach, whose main peculiarity is that the regressor estimate is deferred until a specific prediction request is received: then, a subset of the training data containing samples “close enough” to the current one is selected, and the local model identified on such a subset returns the prediction. The local predictor is hence discarded and a new local model will be set up once a novel prediction will be required. Lazy learning has been shown to outperform a number of approaches such as regression trees and feed-forward neural networks in many time series prediction case studies (Bontempi, 1999); moreover, it ranked second (out of 17 participants) at an International Competition for Time Series Prediction (Birattari et al., 1998). In our knowledge, it has been not yet applied to air pollution modelling.

The remaining of the paper is organized as follows: Section 2 describes in detail the different prediction algorithms; Sections 3 and 4 present the results obtained on ozone and PM_{10} studies, respectively and Section 5 carries out the conclusions.

2. Modelling methodologies

Statistical approaches predict the concentration at time $(t+1)$, denoted as $y(t+1)$, starting from a set of n input variables observed at time t $[u_1(t), \ldots, u_n(t)]$. The estimated forecast $\hat{y}(t+1)$ is issued by means of an approximating function $f(u_1(t), \ldots, u_n(t), \theta)$, where θ denotes the set of parameters of f. Such an approximating function is a priori unknown; it is learned from data, both in its structure and in its parameters, by means of suitable algorithms.

An important remark regards the data partitioning approach, shared by all the introduced methods. Before training any model, the dataset is split into an *identification* and a *testing* set; then the identification set is further subdivided into a *training* and a *validation* set. Let us denote in particular as N the cardinality of the training set.

The training set is used to estimate the model parameters, the validation set to choose among a set of different already trained alternative models, and the testing set to run the chosen approximating function on previously unseen data, in order to get an objective measure of its generalization performances.

Such a measure can however this way depend on how data have been partitioned, and on the samples which have been left in the testing set. In order to address such a problem, we adopt a cross-validation approach: subsets are repetitively shuffled, until each subset has been used once as testing. At each iteration, the model is identified ex novo by using the current identification set and evaluated on the current testing set; finally, the average of the indicators computed on all the different testing sets is retained. Cross-validation is a recognized approach in air pollution modelling (Kukkonen et al., 2003; Dorling et al., 2003).

In each experiment of the cross-validation procedure, all the datasets are standardized, using the means and the variances of the different variables on the identification set; indeed, standardization makes the training algorithms numerically robust and leads to faster convergence (Norgaard et al., 2000).

In the following sub-sections, we describe the different black-box prediction approaches.

2.1. Feed-forward neural networks (FFNNs)

The structure of a feed-forward neural network can be represented as in Fig. 1. Input values are first collected in the *input layer*, and then sent to different processing units (*neurons*), which constitute the *hidden layer* of the network. Let us denote as *m* the number of neurons in the hidden layer.

Each neuron in the hidden layer computes a weighted sum of the inputs; for instance, in the case of neuron *j*, we have:

$$z_j = \sum_{i=1}^{n} w_{ij} u_i - b_j \qquad (1)$$

where w_{ij} is the weight of input u_i at neuron *j*; b_j is the *bias* of neuron *j*, which can be thought as the weight of an input having constant value 1.

The quantity z_j computed at each neuron becomes then argument of a specific function (*activation function*), which resides in the neuron itself. Different kinds of functions are referenced in the literature, such as linear, sigmoid, hyperbolic tangent, logistic, etc. (Norgaard et al., 2000). In the following, we consider a hyperbolic tangent activation function for the neurons in the hidden layer; hence, the value returned by the activation function of neuron *j* of the hidden layer is:

$$a_j = f(z_j) = 1 - \frac{2}{(\exp(2z_j) + 1)} \qquad (2)$$

The *m* values $(a_1, \dots, a_m)$ are sent to the *output layer*, which contains a unique output neuron (see Fig. 1).

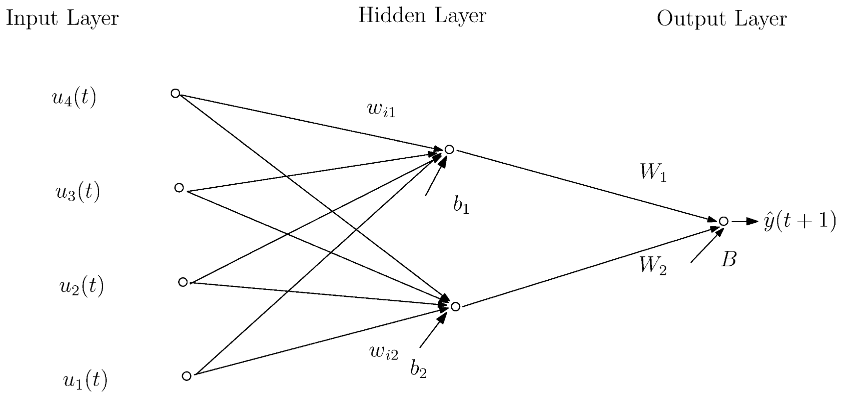

Fig. 1. Sample of a feed forward neural network with four input variables, two neurons in the hidden layer and one output neuron.

It returns the forecast as:

$$\hat{y}(t+1)=\sum_{j=1}^{m} W_j a_j - B \tag{3}$$

where W_j and B denote the weights and the bias of the output neuron, respectively.

It is worthwhile noting that a network architecture having just one hidden layer, and activation functions arranged as described above, constitutes a universal approximator; i.e., given enough hidden neurons, it can *theoretically* approximate any continuous function to any degree of accuracy. In practice, such degree of flexibility is not achievable because parameters must be estimated from sample data, which are both finite and noisy.

Since data flow within the network from a layer to the next one without any return path, such kind of network is defined as *feed-forward*.

2.1.1. FFNNs training

In order to find out the optimal neural network model for a given application, it is necessary to identify an optimal combination between the set of input variables and the number of neurons in the hidden layer. Since there are no a priori guidelines suitable in order to establish an optimal configuration of such characteristics, one has to train of many different neural networks prototypes, and to finally choose the best among them.

The FFNNs training is accomplished by iterative algorithms, which have to be initialized with a parameter guess θ_0 and then evolve, by incrementally updating the parameter estimate θ and decreasing the value of the square error function $J(\theta)$ on the training set, defined as follows:

$$J(\theta)=\frac{1}{2N}\sum(y_i(t)-\hat{y}_i(t,\theta))^2$$

In particular, we use the Levenberg–Marquardt training algorithm (Norgaard et al., 2000), which is well known because both of its speed and robustness.

However, *overfitting* problems can easily occur if the training algorithm runs up to the numerical convergence of the parameters estimate: indeed, reducing more and more the error on the training set causes the network to learn also the noise contained in the data. In order to deal with such an issue, the *early stopping* approach (Bishop, 1995) is suitable: at each iteration of the training algorithm, the square error is evaluated also on the validation set. Validation error normally decreases during the first training iterations, starting then to rise when the network begins to overfit the data. Early stopping provides a termination condition for the training which allows to prevent overfitting, since the parameters estimate corresponding to the minimum validation error is finally retained.

A further approach suitable to avoid overfitting is *regularization*. A regularized function adds a term proportional to the norm $\|\theta\|$ of the parameters to the square error function as:

$$J_{\mathrm{REG}}(\theta)=J(\theta)+\frac{1}{2N}D\|\theta\| \tag{4}$$

The coefficient D which rules the relative relevance of the square error and of the term $\|\theta\|$ is defined as *weight decay*. Regularized cost functions are recognized to decrease the number of local minima and to improve the model generalization (Norgaard et al., 2000); unfortunately however, there are no guidelines suitable to a priori establish a convenient value of D for a given neural network model. Therefore, also coefficient D has to be optimized by trial and error; we let it vary between 0.0001 and 1. The extension of the Levenberg–Marquardt algorithm to case of regularized training function is given in Norgaard et al. (2000).

In order to prevent overfitting as much as possible, we use at the same time both regularization and early stopping for training FFNNs.

2.1.2. FFNNs architecture selection

In order to avoid local minima estimate of the parameters, each network has to be trained several times (we use 25 repetitions) using different random initializations of the parameters.

Among all the many networks generated by such a trial and error procedure, we finally choose the network showing the lowest square error on the validation set. Indeed, selecting the architecture on the base of the validation error is the usually recommended approach (Norgaard et al., 2000) for choosing a model with optimal generalization.

The overall neural network identification and selection procedure has been implemented in Matlab, starting from the powerful functions already provided in the Neural Network for System Identification and Control Toolbox for Matlab (Norgaard, 2000), available as open source software from the website http://www.iau.dtu.dk/research/control/nnsysid.html. Our procedure requires generally a few hours of computation times on a standard PC.

2.2. Pruned neural networks (PNNs)

Undoubtedly, FFNNs are not parameter-parsimonious: for example, a network with an input set of 10 variables may easily contain also a few hundreds parameters. Although some of the many weights that translate the relations between inputs and outputs inside the network are likely to be of little relevance, trial and error approaches can take into account only fully connected neural network architectures, since testing also partially connected models would lead to a combinatorial explosion of the number of trials needed.

Pruning algorithms constitute an interesting way to address such an issue. The basic idea of pruning algorithms is to start from a FFNN model, considered large enough to capture the desired input–output relationship; at each iteration, the pruning algorithm computes a measure of the contribution of each parameter to the network efficiency, and consequently removes the less influential one from the network architecture. Hence, a novel model is generated, containing one parameter less than the previous one; the procedure continues this way until just a unique parameter remains in the network. It is finally up to the modeler the choice of the optimal architecture among the many ones generated during the pruning session. A graphical sample of a pruned architecture is given in Fig. 2.

The criterion which determines the parameter to be eliminated from the network plays clearly a key role within such kind of algorithms. The simple *magnitude-based* pruning (Hertz et al., 1991) removes, for instance, the smallest weight from the network, assuming that it is irrelevant. The optimal brain damage (OBD) pruning algorithm (Le Cun et al., 1990), on the other hand, estimates the increase on the training set resulting from the removal of each parameter from the network architecture (*parameter saliency*), and then removes the parameter showing the lowest saliency.

In the following we describe the optimal brain surgeon (OBS) algorithm; it is again a saliency-based approach, which has been shown (Hassibi and Stork, 1993) to be more affordable than both OBD and magnitude-based pruning. For the sake of simplicity it is described in the case of a square error training function. However, the description of the OBS algorithm for a regularized error function can be found in Norgaard et al. (2000).

Assuming (a) to approximate the square error on the training set $J(\theta)$ at the second order around a point $\bar{\theta}$ in the parameters space and that (b) $\bar{\theta}$ represents a local minimum for $J(\theta)$ (i.e., $\bar{\theta}$ is the parameters estimate after the training), the increase in the training error due to the removal of parameter $\bar{\theta}_i$ can be estimated as:

$$J\,(\text{removed}\,\theta_i) - J(\bar{\theta}) = \frac{1}{2}\frac{\bar{\theta}_i^{2}}{[(H_{\bar{\theta}})^{-1}]_{ii}} \qquad (5)$$

where $[(H_{\bar{\theta}})^{-1}]_{ii}$ denotes the (i, i) element of the inverse of the Hessian of $J(\theta)$, evaluated in $\bar{\theta}$. The right-hand side of Eq. (5) is actually the OBS estimate of the *saliency* of $\bar{\theta}_i$. The parameter showing the lowest saliency is pruned from the network architecture.

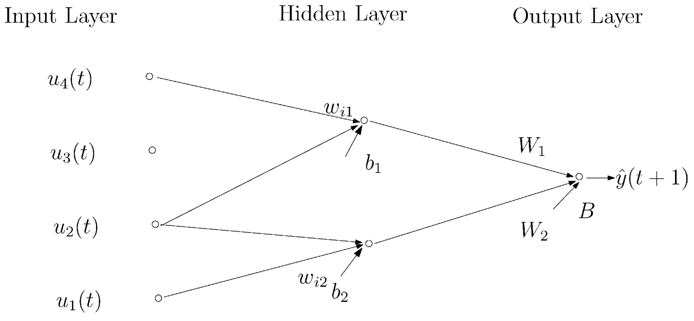

Fig. 2. Sample of a pruned architecture. In this case, input u_3 is no longer connected to the hidden layer and u_2 is the only fully connected input variable.

OBS provides also an update rule in order to adjust the parameters of the novel architecture, minimizing the (second-order approximated) training error $J(\theta)$. Such an update rule is as follows:

$$(\theta - \bar{\theta}) = -\frac{\bar{\theta_i}}{[(H_{\bar{\theta}})^{-1}]_{ii}}(H_{\bar{\theta}})^{-1}e_i \qquad (6)$$

where e_i is a unit vector in the parameters space, parallel to axis θ_i.

An OBS pruning session can be finally schematized as follows:

(1) training (without early stopping) of the initial fully connected architecture, which has to be "large enough" to capture the input–output relationship;
(2) ranking of the network parameters on the base of their saliences, computed as in Eq. (5);
(3) elimination of the parameter with the lowest saliency and generation of a new architecture;
(4) adjustment of the remaining parameters according to the update rule (6). As a small part of the parameters (3–5%) has been eliminated, it is however recommended to retrain ex novo the network (Norgaard et al., 2000), in order to avoid using repetitively an update rule based on an approximation of the error function;
(5) evaluation of mean squared error of the new network on training and validation sets;
(6) back to step 2, until there are parameters left;
(7) choice of the pruned architecture showing the lowest squared error on the validation set, consistently to the selection criterion used for FFNNs. Such an architecture contains usually 75–95% less parameters than the initial network, and is therefore much less prone to overfitting.

Fig. 3 shows a sample of the error functions behavior during an OBS session; the algorithm starts from the fully connected network at the very right of the figure and moves to the left eliminating the parameters one at a time. The training error J_{TR} shows, from right to left, a monotonically increasing behavior, while the validation error J_{VAL} shows a roughly convex behavior.

The parameter parsimony of the pruned model can be conveniently exploited by retraining the model on the whole identification set (i.e., the merging of training and validation), letting the training algorithm run until convergence without no longer using early stopping. This way, a greater amount of data can be used to improving the parameters estimate with reference to FFNNs.

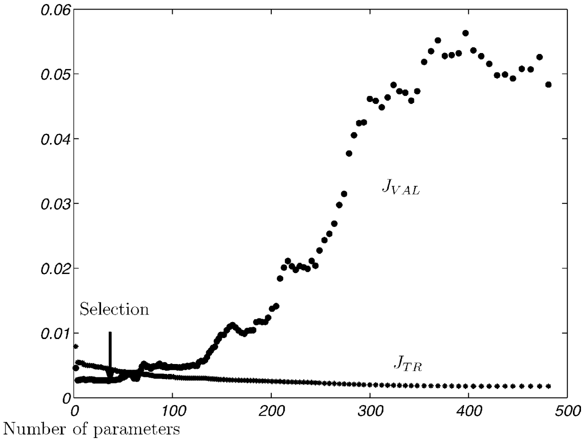

Fig. 3. Values of training and validation square error as a function of the number of parameters during a pruning sessions.

Also pruned neural networks have been implemented using the Neural Network Based System Identification Toolbox for Matlab (Norgaard, 2000) (http://www.iau.dtu.dk/research/control/nnsysid.html); the whole procedure of identifying a pruned network took about half a day on a standard PC.

It is finally interesting to remark that, although pruning algorithms address architectural issues in artificial neural networks, they have also a biological plausibility. In particular, according to the "*selectionist*" approach (Edelman, 1987; Changeux et al., 1973), brain development comprises an initial period of over-production of neurons and connections, followed by a more prolonged period of eliminations of redundant ones. Indeed, pruning algorithms can be thought to implement selectionism in artificial neural networks.

2.3. Lazy learning (LL)

Global modelling approaches, such as neural networks, have the main characteristic of providing an approximating function which maps entirely the inputs and the output space. Usually, no learning takes place after the model has been trained and the model is kept unchanged on any prediction. In local modelling, on the contrary, one renounces to a complete description of the input–output relationship, focusing on approximating the system only in the neighborhood of the point to be predicted. Lazy learning (Birattari et al., 1999a, 2001) is a local linear modelling approach which performs a complex learning procedure, resulting in the identification of a local model, as a prediction is required. Such a local predictor is then discarded after having returned the forecast.

Probably, the most relevant contribution to LL development and diffusion in recent years has been done by the research group working at IRIDIA (http://www.iridia.ulb.ac.be/~lazy), that continuously works over LL algorithmic enhancements and applications, and that also releases the LL implementation as open-source code.

A core idea of local modelling is the *query-point*, defined as the vector $\boldsymbol{q}(t) = [q_1(t), q_2(t), \ldots, q_n(t)] = [u_1(t), u_2(t), \ldots, u_n(t)]$ which collects the input values at time (t) in correspondence of which the prediction is required. Moreover, we define as *neighbor* a generic input vector $\boldsymbol{u}(\tilde{t}) = [u_1(\tilde{t}), u_2(\tilde{t}), \ldots, u_n(\tilde{t})]$ available in the identification dataset, whose corresponding output $y(\tilde{t}+1)$ is hence already known.

The first step of the LL algorithm is the evaluation of the distance between the current query-point and all the neighbors available in the identification set. This is accomplished via the Manhattan metric:

$$d(\boldsymbol{q}(t), \boldsymbol{u}(\tilde{t})) = \sum_{i=1}^{n} |\boldsymbol{q}_i(t) - \boldsymbol{u}_i(\tilde{t})| \tag{7}$$

We remark that such a distance makes sense just on standardized data, because of the different measurement units of the variables.

Neighbors are then ranked according to their distance from the current query-point; in particular, the collection of the k neighbors having the minimum distance from the current query-point is defined as k *nearest-neighbor set*.

The neighbors distances have to be evaluated ex novo each time a query-point is provided; hence, the entire identification dataset has to be always kept in memory. LL is therefore defined as a *memory-based* approach, and this constitutes a difference with respect to neural networks, which on the contrary discard the identification set after having been trained.

Different models are identified on nearest neighbors set of different size ($k_{\min} \leq k \leq k_{\max}$); the mathematical structure of such local models is:

$$\hat{y}(t+1) = \theta_1 u_1(t) + \theta_2 u_2(t) + \cdots + \theta_n u_n(t) \tag{8}$$

Deciding the optimal number k of nearest neighbors (*bandwidth)* to be used for the identification of the model is of paramount relevance for successfully local modelling. LL proceeds identifying different local candidates, letting the bandwidth vary among $k_{\min}$ and $k_{\max}$.

In order to choose the predictor among the several local candidates, the generalization of each model is assessed by means of leave-one-out cross-validation. The identification dataset of cardinality k is split into a training subset of cardinality $(k-1)$ and in a validation subset, containing just the remaining sample. Parameters are estimated on the training set, while the generalization is assessed by means of the error on the validation set (*leave-one-out error*). k experiments are performed using each time a different validation sample, and thus generating a vector of k leave-one-out errors. The local model showing the lowest average on the k leave-one-out error is finally chosen, and the prediction is computed through formula (8).

The local learning procedure involves a significant computational burden. Assuming, for instance, to set $k_{\min} = 50$ and $k_{\max} = 300$, 250 models have to be identified and evaluated via leave-one-out cross-validation each time a prediction is required. Such an heavy computational task is, however, greatly speeded up, in the implementation provided by the IRIDIA research group, by exploiting recursive linear techniques (Birattari et al., 1999b); in fact, we report computation times of just a few seconds for simulating time series of about 1000 data.

2.3.1. Noteworthy features of lazy learning

Several characteristics of lazy learning deserve interest. In particular:

- *it gives linearity a chance via locality* (Bontempi, 1999), allowing to reuse a large amount of procedures taken from the linear statistics, such as, for instance, recursive least squares algorithms;

- *it does not make global assumption about the input data distribution $\Pi(\boldsymbol{u})$*. If we consider a non-uniform input distribution, the definition of the learning problem as a global error minimization biases the global approximator towards regions where $\Pi(\boldsymbol{u})$ is higher. This is not true in local modelling, where by definition each prediction problem is independent of the other ones, and therefore negative interference between different regions of the input space are avoided (Bontempi, 1999);
- *it requires to keep the dataset in memory during the whole prediction task*, because of the memory-based nature of the model. On the one hand, this is a drawback in terms of computational requirements; on the other hand, *LL predictor can be easily updated*, by simply adding the new samples to the identification set.

3. Results: ozone

The available dataset comprises 3 years of data (1999–2001); observations refer to a monitoring station located in a residential quarter of the city and, as such, not directly exposed to heavy traffic emissions. In the same location, we have available a set of meteorological variables such as solar radiation, wind speed and direction, rainfall, atmospheric pressure and humidity, etc. Each variable is sampled at hourly time steps.

Because of its dependence on solar radiation and temperature, ozone reaches its maxima between June and August (Fig. 4a), while it is in practice negligible between October and February; we used therefore in our experiments just the data of the period April–September, for a total amount of 540 daily time steps.

Our forecast target is the maximum 8-h ozone moving average; as already outlined, the health protection threshold is fixed on this indicator to 110 μg/m^3. Models have been targeted to predict at 9 a.m. of day *t* the concentration for the current day *t*; farther prediction horizons (for example, 2 days in advance), requiring, in fact, the availability of meteorological forecasts for satisfactory performances, are not considered in this work.

The raw hourly data taken from the measuring station have to be aggregated in order to constitute useful input variables for the predictors: for instance, a model can have as input variable the average of wind speed between 10:00 and 16:00, or its daily maximum, but certainly not each one of the 24 measures taken during the day. Suitable aggregation operators in order to obtain daily values from the original hourly time series are, for instance, the 24-h average, the 24-h maximum, or the average computed over a specific time period during the day (for instance, 10:00–16:00).

We select the best suited input variables by means of a cross-correlation analysis with the ozone time se-

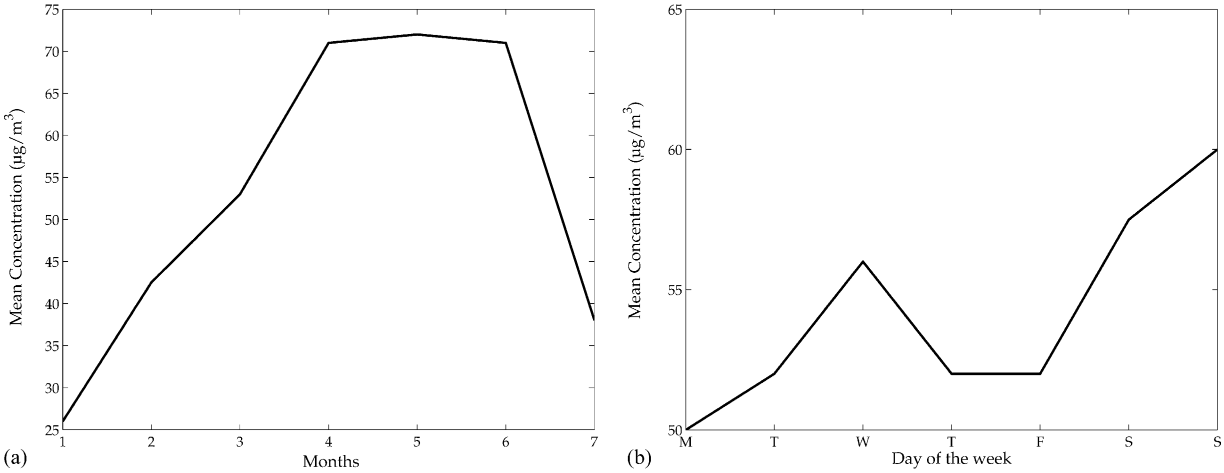

Fig. 4. Average monthly and weekly profiles of ozone time series.

Table 1
Input variables chosen for ozone prediction

Variable	Aggregation	Variable	Aggregation
O_3	$\max[\mu_8(t-1)]$	Temp.	$\mu[6_t - 9_t]$
O_3	9_t	Hum.	$\mu[6_t - 9_t]$
NO	$\mu[6_t - 9_t]$	Global sol. rad.	$\mu[8_t - 9_t]$
NO_2	μ_{t-1}	W. speed	μ_{t-1}
CO	$\mu[6_t - 9_t]$	Rain	μ_{t-1}
Press.	μ_{t-1}	Stab. class.	μ_{t-1}

$\mu(t)$ denotes the average operator.

Table 2
Ozone prediction performances, averaged on the different cross-validation runs

	Normalized data			Deseasonalized data		
	FFNN	LL	PNN	FFNN	LL	PNN
Average goodness indicators						
ρ	0.83	0.84	0.84	0.83	0.86	0.85
MAE	17.02	15.87	17.13	16.87	15.49	16.41
MBE	−0.70	0.76	−1.03	−0.32	−0.86	−2.13
d	0.90	0.91	0.90	0.90	0.92	0.91
Threshold indicators						
TPR	0.72	0.66	0.66	0.67	0.67	0.73
FPR	0.14	0.10	0.11	0.11	0.10	0.12
FA	0.32	0.28	0.30	0.31	0.30	0.31
SI	0.58	0.56	0.55	0.56	0.57	0.61

ries to be predicted; at the same time, we paid attention to avoid input variables having high reciprocal cross-correlations. The selected input variables and their aggregation are presented in Table 1. In principle, it might be surprising to use a linear approach, i.e., cross-correlation analysis, in order to select the inputs of non-linear models; nevertheless, such a procedure leads generally to satisfactory outcomes and constitutes a de facto standard for this kind of studies (see, for instance, Viotti et al., 2002; Nunnari et al., 1998; Ziomas et al., 1995; Balaguer Ballester et al., 2002).

The average goodness of the prediction is firstly evaluated by a series of indicators, such as the true/predicted correlation ρ, the mean absolute error $\mathrm{MAE} = (1/D)(\sum |y(t) - \hat{y}(t)|)$, the mean bias error $\mathrm{MBE} = (1/D)(\sum y(t) - \hat{y}(t))$, and the index of agreement $d = 1 - (\sum(y(t) - \hat{y}(t))^2 / \sum(|\hat{y}(t) - y(t)| - |y(t) - y(t)|))$. In all these formulas, D denotes the total number of days.

We assess then the threshold exceedances forecasts; denoting the correctly predicted exceedances as CP, the predicted exceedances as P, the observed exceedances as O, we compute, as in Schlink et al. (2003), the true predicted rate $\mathrm{TPR} = \mathrm{CP}/O$, the false positive rate $\mathrm{FPR} = (P - \mathrm{CP})/(N - O)$, the false alarm rate $\mathrm{FA} = (P - \mathrm{CP}/P)$, the success index $\mathrm{SI} = \mathrm{TPR} - \mathrm{FPR}$. TPR and SI assume their best value at 1, while the best value for FPR and FA is 0.

Prediction performances are reported in Table 2. All the approaches provide a quite satisfactory accuracy, showing a correlation about 0.85 and an index of agreement about 0.90. Lazy learning shows a slight improvement over both feed-forward and pruned neural networks on the indicators related to the average prediction goodness. The FFNN model appears to be the most effective in capturing the threshold exceedances (a few points of advantage on the TPR indicator) but, correspondingly, also the most prone to false alarms. However, it provides the highest value of SI, which can be thought as an overall score for the threshold exceedances detection.

As further experiment, we try a deseasonalization data pre-processing approach. Indeed, ozone underlies a significant weekly periodicity (Fig. 4b), besides the yearly periodicity caused by the meteorology (Fig. 4a). In particular, ozone concentrations tend to be higher on weekends than on weekdays, despite the lower emissions of ozone precursors on weekends. Higher average ozone concentrations on weekends have been found also in several areas of North America (Vukovich, 2000; Marr and Harley, 2002); the phenomenon is known as the *weekend effect* and, despite the numerous hypotheses proposed, it still lacks a detailed understanding.

There is no general consensus in the literature about either it is convenient to remove periodic components from data before training a non-linear model or not: a series of sample experiments showed (Nelson et al., 1999) that ANN may benefit of data deseasonalization – as statistical methods do – but on the contrary a worsening of the performance has been reported training neural networks to predict NO_2 on the residual of a periodic regressor rather than just on standardized data (Kolehmainen et al., 2001). We try several deseasonalization approaches considering both the yearly and the weekly scale. Finally, we find the most suitable approach to be a weekly-based standardization of the pollutants variables: different means and variances are used to standardize the data, depending on either they refer to a working day or to a weekend day. The cross-correlation analysis on the deseasonalized data leads to slight changes with reference to the input variables set of the previous experiment; both the structure and the parameters of the predictors are identified ex novo accordingly.

The prediction performances on the deseasonalized data are given in Table 2. The FFNN model do not take advantage of such a pre-processing: on the contrary, SI decreases of a couple of points, while the average goodness indicators are substantially unchanged. It is worth noticing that, on the other hand, LL and PNN increase their performances; in particular, the PNN model undergoes an overall improvement of the threshold indicators (+6 points on SI).

Comparing the most effective approaches, i.e., FFNN trained on the standardized data, LL and PNN trained on the deseasonalized data, it can be seen that performances do not vary substantially among the different approaches; however, LL provides the best outcome on the indicators regarding the average prediction goodness, and PNN on the threshold indicators.

The performances of a forecasting system should be evaluated considering also the specification of the threshold on the forecast at which the decision maker will issue the alarm of an expected exceedance. Dealing with ozone, the only possible short-term intervention is issuing the alert to the population, thus trying to minimize exposures to unhealthy air; therefore, the cost of a false alarm (people affected by respiratory pathologies spending time indoor during the warmest hours of the day even if unnecessary) is reasonably much lower than that of an undetected exceedance (increase of the hospitalization). A "conservative" decision maker may therefore issue the notice of an exceedance of the 110 μg/m^3 threshold also in the case of a forecast of just 100 μg/m^3 or 90 μg/m^3. The outcomes of these two hypothetical cases in terms of both TPR and FPR are reported in Table 3; for instance, if a threshold of 100 μg/m^3 is adopted on the forecast, the detection of an exceedance of 110 μg/m^3 has a TPR of about 0.85 and a FPR of about 0.21.

Table 3
Sensitivity of the detection of the 110 μg/m^3 exceedances with respect to the threshold adopted on the forecast for issuing the alarm

Forecast threshold (μg/m^3)	FFNN		LL		PNN	
	TPR	FPR	TPR	FPR	TPR	FPR
90	0.91	0.32	0.95	0.30	0.92	0.32
100	0.86	0.23	0.86	0.19	0.85	0.21

3.1. Comparison with results available in literature

The presented findings can be compared with the results published in Schlink et al. (2003), where 15 statistical techniques are intercompared on 10 case studies of ozone prediction distributed throughout Europe.

Considering the maximum 8-h ozone moving average as forecast target, the average index of agreement *d* found for all the methods over all the case studies is about 0.90. FFNNs provide the best performance among the different techniques: indeed, they provide an index of agreement (averaged over all the case studies) of 0.94. There is no particular dependence of such an outcome on the training algorithm adopted in order to estimate the network parameters. A group of other six statistical techniques provide an average index of agreement higher than 0.90, while the remaining eight shown an average index of agreement lower than 0.90, with a minimum of 0.88. The performances of FFNNs, PNNs, and LL in Milan are therefore comparable with the outcome of the best performing approaches considered in Schlink et al. (2003).

As for threshold exceedances detection, the average success index SI found for all the techniques over all the case studies is about 0.51. Four techniques provide a SI (averaged over all the case studies) lower than 0.50, eight techniques have a SI comprised between 0.50 and 0.60, while three techniques have a SI > 0.60. The average SI for FFNNs ranges about 0.56 and 0.70 depending on the adopted training algorithm.

The outcome on threshold exceedances detection of PNNs (SI $= 0.61$, see Table 2) in Milan is hence comparable to the best performing approaches considered in Schlink et al. (2003), while those of LL and FFNNs are just slightly worse (SI $= 0.58$ and 0.57, see Table 2) .

3.2. Input variables significance

It is of interest to assess the relevance of the different inputs for the prediction. Unfortunately, the "very black-box" nature of feed-forward neural networks ob-

Table 4
Lazy learning regressor average coefficients for meteorological variables

Solar rad.	Temp.	Rain	Press.	Stab. class	Humid.	Wind speed
Low solar radiation						
0.221	0.405	−0.529	0.168	−0.075	−0.079	−0.080
Medium solar radiation						
0.304	0.339	−0.230	0.105	−0.030	−0.013	−0.086
High solar radiation						
0.409	0.174	0.043	0.014	0.036	−0.063	0.005

scures the meaning of the parameters and, despite the several attempts done in the literature, it is not easy to evaluate the importance of the different variables; furthermore, the different methodologies suitable to this end may lead to inconsistent conclusions (Gevrey et al., 2003).

On the other hand, dealing with pruned neural networks, one can easily recognize as redundant those variables which are no longer connected to the hidden layer, the relative parameters having been removed. Unfortunately, we do not find an actual consistence of the pruned variables across the different cross-validation runs. The most frequently removed inputs have been however NO_2 (perhaps because of the VOC-limited chemistry of the area), the atmospheric pressure and the Pasquill stability class (probably somewhat redundant with the solar radiation information).

Variables relevances in lazy learning can be easily assessed by looking at the coefficients of the regressors: since all the inputs are standardized, the coefficients are directly comparable and indicate how much a given variable impacts on the final prediction. However, parameters are estimated differently on each query-point and therefore we have as many different estimates as the testing set is long. In order to analyze these estimates, we divide them in classes according to the value of the solar radiation input, which is indeed recognized to play a major role in ozone formation. The average estimates for the different classes are given in Tables 5 and 4.

When solar radiation is high, it becomes the most important variable of the predictor; as it decreases, temperature and rainfall (which clearly gives a negative contribution) become the key variables between the meteorological data. The coefficients of the two past ozone terms increase with solar radiation; the observation of 9 a.m. has a greater weight than the maximum 8-h moving average of day $(t-1)$, which is actually the autoregressive input. Remarkably, NO_2, atmospheric pressure and stability class are given low coefficients, thus confirming the findings of the pruning session. CO is given a negative coefficient, and in fact, its anti-correlation with ozone in the photochemical mechanisms is well known. Finally, humidity and wind speed appear as inputs of minor relevance; this does not mean that they do not influence the ozone values, but rather than their past values are not useful in predicting the future concentrations.

Definitely, great fluctuations of the parameters of the regressor are observed, especially on solar radiation, temperature, rainfall, past ozone; such variations of the estimates constitute a posteriori confirmation of the need for a non-linear modelling approach to predict ozone concentrations.

Table 5
Lazy learning regressor average coefficients (pollutant variables) as a function of the solar radiation input observed between 8 and 9 a.m.

O_3 max ($\mu_8(t)$)	O_3 (9 a.m.)	CO	NO_2	NO
Low solar radiation				
0.135	0.222	−0.187	0.031	0.461
Medium solar radiation				
0.188	0.242	−0.161	0.064	0.207
High solar radiation				
0.216	0.280	−0.144	0.048	0.042

The first class corresponds to solar radiation lower than 119 mW/cm^2, the second lies between 119 and 421 mW/cm^2, the third corresponds to solar radiation higher than 421 mW/cm^2. Such bounds are given by the (average − standard deviation) and (average + standard deviation) of the time series between April and September.

4. Results: PM_{10}

The available dataset comprises 4 years of data (1999–2002) for a total amount of about 1400 time steps; observations refer to the same monitoring station used for the ozone study. PM_{10} time series underlies a significant periodic behavior: concentrations are about twice during winter than during summer (Fig. 5a), because both of the higher anthropic emissions (see, for instance, building heatings emissions) and the unfavorable dispersion conditions (i.e., lower mixing layer). Concentrations are however not negligible during summer, when some exceedances of the 50 μg/m^3 threshold can be recorded, though not causing the declaration of the attention state. A significant periodicity is detected also at the weekly scale (Fig. 5b): concentrations are, in fact, 25–30% lower on Sunday than on the remaining days of the week.

With reference to the input selection, we use the exhaustive input/output correlation analysis carried out in Corani and Barazzetta (2004) and regarding the same dataset, but aimed at developing a traditional linear ARX model. The analysis grouped all the candidate input variables on all the possible time windows comprised between 0 a.m. of day $t-1$ and 9 a.m. of day t, evaluating accordingly the cross-correlation with the output PM_{10} time series. The selected input variables set is reported in Table 6.

Table 6
Input variables chosen for PM_{10} prediction

Variable	Aggregation
PM_{10}	$\mu[22_{t-1} - 8_t]$
SO_2	$\mu[13_{t-1} - 5_t]$
Temp.	$\mu[4_{t-1} - 8_{t-1}]$
Press.	$\mu[1_{t-1} - 7_t]$

Although the PM_{10} periodicity suggests to try a deseasonalization approach to improve the prediction performances, and despite the variety of attempts performed, we find no significant advantages working on deseasonalized data rather than just on standardized ones. Therefore our results refer, for all the modelling approaches, to the simply standardized data. Performance indicators, obtained as average of four cross-validation runs, are given in (Table 7).

We note that the indicators are significantly better than in the ozone study, i.e., about 5 points higher on correlation and more than 10 points on SI. Such an outcome shows that the underlying dynamic is captured more easily by the models; this can be due to the daily average prediction target, which generates a smoother time series than the maximum of a 8-h moving average, and to the PM_{10} formation process which involves less chemical reactions and can be somewhat easier to predict.

The performances provided by the different approaches are quite similar; nevertheless, we note once

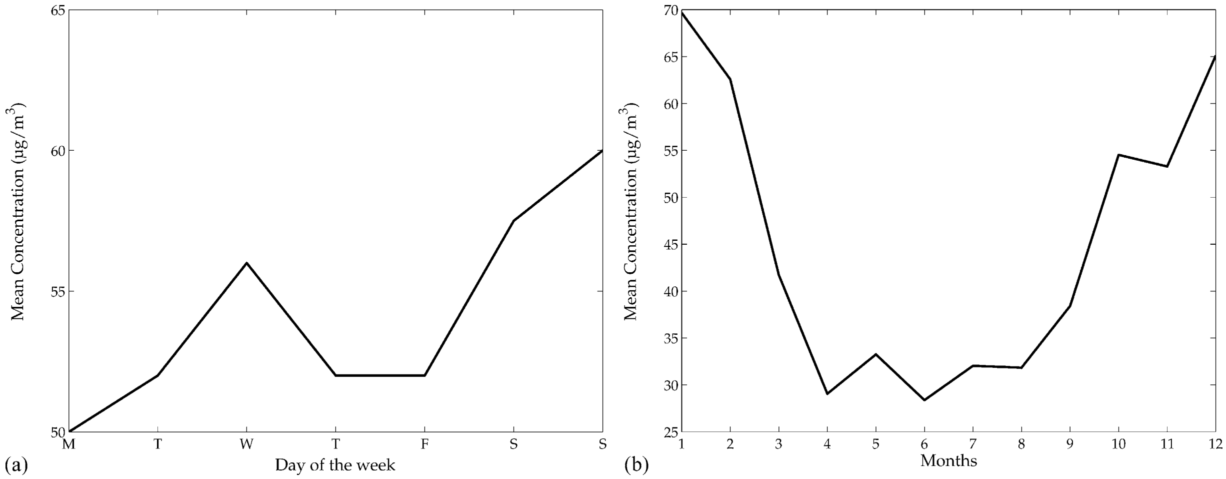

Fig. 5. Average yearly and weekly profiles of PM_{10} time series.

Table 7
PM_{10} prediction performances, averaged on the different cross-validation runs

	FFNN	LL	PNN
Average goodness indicators			
ρ	0.88	0.90	0.89
MAE	8.59	8.25	8.55
MBE	−0.12	0.16	0.47
d	0.94	0.94	0.94
Threshold indicators			
TPR	0.82	0.83	0.83
FPR	0.09	0.08	0.07
FA	0.20	0.17	0.16
SI	0.73	0.75	0.76

Table 8
Sensitivity of the detection of the 50 $\mu g/m^3$ exceedances with respect to the threshold adopted on the forecast for issuing the alarm

Forecast threshold ($\mu g/m^3$)	FFNN		LL		PNN	
	TPR	FPR	TPR	FPR	TPR	FPR
40	0.95	0.27	0.95	0.26	0.95	0.24
50	0.82	0.09	0.83	0.08	0.83	0.07
60	0.63	0.03	0.61	0.03	0.63	0.03

more that LL has a slight prevalence over the average prediction goodness indicators, and PNN over the threshold indicators. The traditional linear predictor carried out in Corani and Barazzetta (2004) showed a correlation of about 0.89 and a MAE of about 11 $\mu g/m^3$; therefore, it provided an accuracy just slightly worse than non-linear approaches.

In order to contrast high PM_{10} concentrations, a decision maker may intervene in practice by reducing the vehicular traffic. Significant costs can therefore be expected both for undetected exceedances (health issues due to delays in contrasting the pollution) and false alarms (unnecessary blockage of part the vehicles). In Table 8 we report the outcome (in terms of TPR and FPR) of different sample cases, analyzing several threshold on the forecast which may be adopted by the decision maker in issuing the alarm of an exceedance of the 50 $\mu g/m^3$ threshold. For instance, if costs of undetected exceedances are especially high, he may issue the alarm with a forecast higher than just 40 $\mu g/m^3$, thus obtaining a 0.95 value on TPR and 0.27 on FPR; vice versa, if false alarms are a major concern, he may issue the alarm only if the prediction is higher than 60 $\mu g/m^3$, thus reducing FPR to about 0.03 but penalizing also TPR to about 0.63.

4.1. Comparison with results available in literature

In Zickus et al. (2002), four different machine learning approaches are trained to predict whether the PM_{10} daily average will exceed the limit of the 50 $\mu g/m^3$ or not. Models are trained to return a binary classification (exceedance/not exceedance) rather than to forecast the expected concentration; hence, their performances can be assessed just in terms of exceedances detection. Three techniques (FFNNs, logistic regression and regression splines) are found to provide similar performances, with a SI of about 0.42, while the fourth modelling approach (decision tree) is found to be significantly worse (SI = 0.26). From this data, we can conclude that the performances obtained in the Helsinki case study are worse than those presented in this paper for Milan.

A couple of further papers addresses the problem of PM_{10} prediction in Helsinki (Kukkonen et al., 2003) and Santiago, Chile (Perez and Reyes, 2002), respectively.

Remarkably, while we adopt – as already elucidated – the PM_{10} daily average as forecast target, the study of Kukkonen et al. (2003) tries to forecast the PM_{10} hourly average, and that of Perez and Reyes (2002) the exceedance of the 240 $\mu g/m^3$ limit on PM_{10} daily maximum. Such different forecast targets do not allow a direct comparison of the findings. However, for the sake of completeness, we briefly report the results described in these papers.

In Kukkonen et al. (2003), an average index of agreement of about 0.73 is reported for FFNNs (with small fluctuations depending on the training algorithm, input configuration, etc.), which are shown to be clearly better than linear regressors (index of agreement about 0.60); no evaluation of the performances on threshold exceedances detection is provided.

In Perez and Reyes (2002), the exceedances of the 240 $\mu g/m^3$ limit are detected about 75% of the times by FFNNs. One should be however aware that just 20 exceedances of such a limit are observed every year.

Table 9
Lazy learning average coefficients as a function of the PM_{10} past concentration

Past PM_{10}	SO_2	Temp.	Press.
Low past PM_{10} (<30 μg/m^3)			
0.81	0.11	0.04	0.06
Medium past PM_{10} (30–50 μg/m^3)			
0.75	0.12	0.04	0.01
High past PM_{10} (>50 μg/m^3)			
0.70	0.15	0.11	0.10

4.2. Input relevances

The input set is quite reduced in this case; indeed, pruning trials do not remove any input variable, thus indicating that each of them is valuable in improving the quality of the prediction.

The relative relevance of the variables can be judged by looking at the lazy learning parameters estimate (Table 9); we divide them in classes depending on the past PM_{10} values, which is indeed the far more influential variable. The average estimates for the different classes are given in Table 6. We note that, besides past PM_{10} , SO_2 is the second variable for importance, while temperature and atmospheric pressure appear as of minor relevance. However, when PM_{10} is higher than 50 μg/m^3, all the variables of the regressor are given significant coefficients.

Remarkably, parameters of the local linear regressors fluctuate much less than in the ozone case; we can therefore conclude that the PM_{10} time series underlies weaker non linearities, and such a limited need for changes in the parameters explains the satisfactory performances of the linear regressor cited above.

5. Conclusions

The presented work carries out an intercomparison of different statistical techniques for prediction of ozone and PM_{10} in Milan. In particular, we compare feed-forward neural networks, currently recognized as state-of-the-art approach for statistical prediction of air quality, with two alternative approaches derived from machine learning: pruned neural networks and lazy learning. As far as we know, LL is applied for the first time to the problem of air pollution prediction.

The predictions, issued at 9 a.m. for the current day, show a satisfactory reliability: for instance, the correlation between true and predicted concentrations is about 0.85 and 0.90, respectively, for ozone and PM_{10}, while the success index, related to the correct detection of the threshold exceedances, is about 0.60 and 0.75 in the two cases. There are no strong differences among the performances of the different approaches; however we find lazy learning to be the best performing approach on average goodness indicators (such as mean absolute error and correlation), and pruned neural networks to be the best approach in detecting the threshold exceedances. The better outcome of all the approaches on PM_{10} with respect to ozone can be due to the daily average prediction target, that generates a smoother time series than the maximum 8-h moving average adopted for ozone.

We find that in some cases the forecast accuracy can be improved by training the predictors on previously deseasonalized data; indeed, both air quality and meteorological variables are intrinsically periodic.

Besides the performances assessment, other modelling issues deserve consideration. Well-known drawbacks of FFNNs are, for instance, their tendency to overfitting, the time-consuming procedure required in order to design the network architecture, the difficult interpretation of the meaning of the parameters and of the inputs relevances. Both PNNs and LL can mitigate some of these problems.

In particular, a great feature of PNNs is their parameter-parsimony, which allows to get rid of overfitting problems; indeed, we find PNNs models to contain about 90% less parameters than the corresponding FFNN model. However, designing a PNN model requires to run a complex algorithm, which takes some hours of computation; moreover, in our experiments we did not found an actual coherence among the input variables removed in the different runs of the pruning algorithm, thus not allowing a straightforward identification of redundant inputs.

On the other hand, some noteworthy features of LL make it a convenient alternative to global modelling approaches; the LL predictor can be quickly developed, and the simplicity of the local linear regressors allows both to get rid of overfitting problems and to readily interpret the relevances of the different inputs. Moreover, it can be easily kept up-to-date, by simply adding the new samples to its knowledge base.

Acknowledgments

The author thanks G. Guariso, Politecnico di Milano, for his valuable comments and suggestions.

References

Agenzia Milanese Mobilità Ambiente, 2003. Relazione sullo stato dell'ambiente del comune di Milano. Comune di Milano.

Balaguer Ballester, E., Valls, G., Carrasco-Rodriguez, J., Soria Oliva, E., Valle-Tascon, S., 2002. Effective 1-day ahead prediction of hourly surface ozone concentrations in eastern Spain using linear models and neural networks. Ecol. Model. 156, 27–41.

Birattari, M., Bontempi, G., Bersini, H., 1998. Lazy learning for iterated time-series prediction. In: J.A.K. Suykens, J. Vandewalle (Eds.), Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling, pp. 62–68.

Birattari, M., Bontempi, G., Bersini, H., 1999a. Lazy learning for modeling and control design. Int. J. Control 72, 643–658.

Birattari, M., Bontempi, G., Bersini, H., 1999b. Lazy learning meets the recursive least-squares algorithm. In: M.S. Kearns, S.A. Solla, D.A. Cohn (Eds.), Advances in Neural Information Processing System, pp. 375–381.

Birattari, M., Bontempi, G., Bersini, H., 2001. The local paradigm for modeling and control: from neuro-fuzzy to lazy learning. Fuzzy Sets Syst. 121, 59–72.

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press.

Bontempi, G., 1999. Local learning techniques for modeling, prediction and control. Ph.D. thesis, Universite' Libre de Bruxelles, Belgium.

Castelli, S., Corani, G., Guariso, G., 2003. Structural identification of multivariate neural networks for rainfall runoff modeling. In: Proceedings of the 13th IFAC Symposium on System Identification, pp. 1951–1956.

Cecchetti, M., Corani, G., Guariso, G., 2004. Artificial neural networks prediction of PM_{10} in the Milan area. In: Proceedings of the iEMSs 2004 International Congress on Complexity and Integrated Resources Management, Osnabruck.

Changeux, J.-P., Courrége, P., Danchin, A., 1973. A theory of the epigenesis of neuronal networks by slective stabilisation of synapses. Proc. Natl. Acad. Sci. U.S.A., 2974–2978.

Corani, G., Barazzetta, S., 2004. First results in the prediction of particulate matter in the Milan area. In: Proceedings of the 9th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes, Garmisch-Partenkirchen.

Dorling, S., Foxall, R., Mandic, D., Cawley, G., 2003. Maximum likelihood cost functions for neural network models of air quality data. Atmos. Environ. 37, 3435–3443.

Edelman, G., 1987. Neural Darwinism: The Theory of Neuronal Group Selection. Basic Books.

García-Gimeno, R., Hervás-Martínez, C., de Silóniz, M.I., 2002. Improving artificial neural networks with a pruning methodology and genetic algorithms for their application in microbial growth prediction in food. Int. J. Food Microbiol. 72 (1–2), 19–30.

Gardner, M., Dorling, S., 1998. Artificial neural network (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos. Environ. 6 (32), 2627–2636.

Gevrey, M., Dimopoulos, I., Lek, S., 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol. Model. 160, 249–264.

Hassibi, B., Stork, D., 1993. Second order derivatives for network pruning: optimal brain surgeon. In: S.J. Hanson, J.D. Cowan, C. Lee Giles (Eds.), Proceedings of Advances in Neural Information Processing System, pp. 164–171.

Henrique, H.M., Lima, E.L., Seborg, D.E., 2000. Model structure determination in neural network models. Chem. Eng. Sci. 55 (22), 5457–5469.

Hertz, J., Krogh, A., Palmer, R., 1991. Introduction to the Theory of Neural Computation. Addison Wesley.

Kolehmainen, M., Martikainen, H., Ruuskanen, J., 2001. Neural networks and periodic components used in air quality forecasting. Atmos. Environ. 35, 815–825.

Kukkonen, J., Partanen, L., Karppinen, A., Ruuskanen, J., Junninen, H., Kolehmainen, M., Niska, H., Dorling, S., Chatterton, T., Foxall, R., Cawley, G., 2003. Extensive evaluation of neural networks models for the prediction of NO_2 and PM_{10} concentrations, compared with a deterministic modelling system and measurements in central Helsinki. Atmos. Environ. 37, 4539–4550.

Le Cun, Y., Denker, J., Solla, S., 1990. Optimal brain damage. In: Touretzky, D. (Ed.), Advances in Neural Information Processing Systems, vol. 2. Morgan Kaufmann, pp. 598–605.

Marr, L.C., Harley, R., 2002. Spectral analysis of weekday–weekend differences in ambient ozone, nitrogen oxyde, and non methane hydrocarbon time series in California. Atmos. Environ. 36, 2327–2335.

Nelson, M., Hill, T., Remus, W., O'Connor, M., 1999. Time series forecasting using neural networks: should the data be deseasonalized first?. J. Forecasting 5, 359–367.

Norgaard, M., 2000. Neural network based system identification toolbox. Tech. Re 00-E-891, Department of Automation, Technical University of Denmark.

Norgaard, M., Ravn, O., Poulsen, N., Hansen, L., 2000. Neural Networks for Modelling and Control of Dynamic Systems. Springer-Verlag, London.

Nunnari, G., Nucifora, M., Randieri, C., 1998. The application of neural techniques to the modelling of time-series of atmospheric pollution data. Ecol. Model. 111, 187–205.

Ostro, B., Chestnut, L., Vichit-Vadakan, N., Laixuthai, A., 1999. The impact of particulate matter on daily mortality in Bangkok, Thailand. J. Air Waste Manage. Assoc. 49, 100–107.

Ostro, B., Eskeland, G., Sánchez, J., Feyzioglu, T., 1999. Air pollution and health effects—a study of medical visits among children in Santiago, Chile. Environ. Health Perspect. 107, 69–73.

Perez, P., Reyes, J., 2002. Prediction of maximum of 24-h average of PM_{10} concentrations 30 h in advance in Santiago, Chile. Atmos. Environ. 36, 4555–4561.

Perez, P., Trier, A., Reyes, J., 2000. Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile. Atmos. Environ. 34, 1189–1196.

Poppi, R., Massart, D., 1998. The Optimal Brain Surgeon for pruning neural network architecture applied to multivariate calibration. Anal. Chim. Acta 375 (1–2), 187–195.

Prybutok, R., Junsub, Y., Mitchell, D., 2000. Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations. Eur. J. Operational Res. 122, 31–40.

Reed, R., 1993. Pruning algorithms—a survey. IEEE Trans. Neural Networks 4 (5), 740–747.

Schlink, U., Dorling, S., Pelikan, E., Nunnari, G., Cawley, G., Junninen, H., Greig, A., Foxall, R., Eben, K., Chatterton, T., Vondracek, J., Richter, M., Dostal, M., Bertucco, L., Kolehmainen, M., Doyle, M., 2003. A rigorous inter-comparison of ground-level ozone predictions. Atmos. Environ. 37, 3237–3253.

Viotti, P., Liuti, G., Di Genova, P., 2002. Atmospheric urban pollution: application of an artifical neural network to the city of Perugia. Ecol. Model. 148, 27–46.

Vukovich, F., 2000. The spatial variation of the weekday/weekend differences in the Baltimore area. J. Air Waste Manage. Assoc. 50, 2067–2072.

Zickus, M., Greig, A., Niranjan, M., 2002. Comparison of four machine learning methods for predicting PM_{10} concentrations in Helsinki, Finland. Water Air Soil Pollut. 2, 717–729.

Ziomas, I., Melas, D., Zerefos, C., Bais, A., Paliatsos, A., 1995. Forecasting peak pollutant levels from meteorological variables. Atmos. Environ. 29, 3703–3711.