Neural Comput & Applic (2009) 18:193–205
DOI 10.1007/s00521-007-0171-1

ORIGINAL ARTICLE

The behaviour of the multi-layer perceptron and the support vector regression learning methods in the prediction of NO and NO_2 concentrations in Szeged, Hungary

István Juhos · László Makra · Balázs Tóth

Received: 18 January 2007 / Accepted: 20 December 2007 / Published online: 9 January 2008

Abstract The main aim of this paper is to predict NO and NO_2 concentrations 4 days in advance by comparing two artificial intelligence learning methods, namely, multi-layer perceptron and support vector machines, on two kinds of spatial embedding of the temporal time series. Hourly values of NO and NO_2 concentrations, as well as meteorological variables were recorded in a cross-road monitoring station with heavy traffic in Szeged, in order to build a model for predicting NO and NO_2 concentrations several hours in advance. The prediction of NO and NO_2 concentrations was performed partly on the basis of their past values, and partly on the basis of temperature, humidity and wind speed data. Since NO can be predicted more accurately, its values were considered primarily when forecasting NO_2. Time series prediction can be interpreted in a way that is suitable for artificial intelligence learning. Two effective learning methods, namely, multi-layer perceptron and support vector regression are used to provide efficient non-linear models for NO and NO_2 time series predictions. Multi-layer perceptron is widely used to predict these time series, but support vector regression has not yet been applied for predicting NO and NO_2 concentrations. Three commonly used linear algorithms were considered as references: 1-day persistence, average of several day persistence and linear regression. Based on the good results of the average of several day persistence, a prediction scheme was introduced, which forms weighted averages instead of simple ones. The optimization of these weights was performed with linear regression in linear case and with the learning methods mentioned in non-linear case. Concerning the NO predictions, the non-linear learning methods give significantly better predictions than the reference linear methods. In the case of NO_2, the improvement of the prediction is considerable, however, it is less notable than for NO.

Keywords Artificial neural networks · Multi-layer perceptron · Support vector machines · Support vector regression · Forecast

1 Introduction

Nitric oxide (NO), as one of the nitrogen oxides (NO_x), is a highly reactive gas. Human activity has drastically increased the production of nitric oxide by traffic. It is produced by the chemical union of O_2 and N_2 in the cylinders of internal combustion engines. (However, the catalytic converter in automobile exhaust systems reduces air pollution by oxidizing hydrocarbons to CO_2 and H_2O and, to a lesser extent, converting nitrogen oxides to N_2 and O_2.) Nitric oxide plays a major role in the photochemical reactions, which lead, among other things, to the formation of nitrogen dioxide (NO_2) and photochemical smog. Since NO_2 absorbs in the visible wavelength region, creating brown cloud over megacities (e.g., Mexico City and Beijing), can be photolysed and yield oxygen atoms that can react with molecular oxygen to create ozone. NO_2 and the NO/NO_2 ratio are important in tropospheric chemistry. Nitrogen dioxide is formed primarily from burning fuel in motor vehicles, power plants, and other industrial,

I. Juhos · B. Tóth
Department of Computer Algorithms and Artificial Intelligence,
University of Szeged, P.O. Box 652, 6701 Szeged, Hungary
e-mail: juhos@inf.u-szeged.hu

L. Makra (✉)
Department of Climatology and Landscape Ecology,
University of Szeged, P.O. Box 653, 6701 Szeged, Hungary
e-mail: makra@geo.u-szeged.hu

commercial, and residential sources that burn fossil fuels. Nitrogen oxides, reacting with other substances in the air, form acid rain that accelerate the corrosion of buildings and monuments, and reduce visibility.

Exposure to nitrogen dioxide can irritate the lungs and may lower resistance to respiratory infections. Sensitivity increases for people with asthma and bronchitis. NO_2 is also a major source of fine particulate pollution, which is a significant health concern.

Due to the harmful effects of these pollutants on human health, it is important to have reliable methods enabling the prediction of their concentrations several hours in advance, so that the public authorities could avoid the harmful consequences of severe air pollution episodes.

The more accurate prediction of future values of a time series will improve performance in each field of everyday life. The classical statistical procedures, as well as neural network have already been applied for short-term prediction of air pollutants by several authors.

Artificial neural networks appear as useful alternatives to traditional statistical modelling techniques in many scientific disciplines. They are composed of a large number of possible non-linear functions (neurons) each with several parameters that are fitted to data through a computationally intensive training process. Some statisticians and forecasters, who prefer the statistical approach to forecasting may disregard the performance of neural networks because of their lack of rigorous statistical foundation. However, neural networks do fit comfortably with the heterogeneous background of alternative forecasting techniques.

Gardner and Dorling [1] present wide fields of recent applications of the multi-layer perceptron as one type of artificial neural network in the atmospheric sciences. They applied MLP neural networks to model hourly NO_x and NO_2 concentrations in Central London from basic hourly meteorological data [2]. The results of the models perform well when compared to those received by using regression based models. They also demonstrated that MLP neural networks offer several advantages over traditional multivariate linear regression models. Jorquera et al. [3] developed an accurate forecast of ozone episodic days for downtown Santiago, Chile. The simple model structure included a combination of persistence and daily maximum air temperature as input variables. The model was validated by comparing the outcome of three different modelling schemes: linear model, fuzzy models and neural networks. The three forecasts developed present significant improvement of successful forecasts compared with pure persistence. Predictions of $PM_{2.5}$ [4], as well as NO and NO_2 [5], plus SO_2 concentrations [6] were compared, and produced by three different methods: persistence, linear regression and multi-layer perceptron neural networks. Furthermore, Perez and Reyes [7] improved $PM_{2.5}$ predictions several hours in advance with a type of neural network which was equivalent to a linear regression. The effect of meteorological conditions was included by using real values of temperature (T), relative humidity (H) and wind speed (W) at the time of the intended prediction as inputs to the different models. It was revealed that a three-layer neural network gave the best results to predict the concentrations of the pollutants in the atmosphere of downtown Santiago, Chile several hours in advance, when hourly concentrations of the previous day were used as input. A multivariate regression model is also used by [8] for comparing with the results obtained by using the neural network model. Their results indicate that the neural network is able to give better predictions with less residual mean square error than those given by multivariate regression models. Mechaqrane and Zouak [9] compared a linear model with MLP, when predicting indoor temperature of a building. Maqsood et al. [10] used data of temperature, wind speed and relative humidity to train and test seven different models for weather forecasting. With each model, they made 24 h ahead forecasts for all seasons. In comparison, they found the ensemble of neural networks to produce the most accurate forecasts.

Agirre-Basurko et al. [11] developed two multi-layer perceptron based models and one multiple linear regression based model. The models utilized traffic variables, meteorological variables and O_3 and NO_2 hourly levels as input data. The performances of these three models were compared with persistence of levels and the observed values. The results indicated improved performance for the multi-layer perceptron-based models over the multiple linear regression models if they considered predictions for more than 3 h in advance. Hansen et al. [12], by using neural network techniques, achieved impressive increases in forecasting accuracy. According to their results, genetic-algorithms-guided selections of neural network architectures displayed distinct improvements over statistical refinements and other heuristic architectures. Additionally, they concluded that neural networks could improve forecasting performance dramatically and found structure in data, which remained hidden to other techniques. According to the investigations of Small and Tse [13], an artificial neural network is particularly well suited to modelling chaotic time series data. Castillo and Melin [14] also use neural networks for simulation and forecasting economic time series. The performance of the neural networks (MLPs) was compared with classical regression models. Time series prediction gave the best result when neural networks were used, compared to that of the other regression models. Kukkonen et al. [15] evaluated five neural network models (MLPs), a linear statistical model and a deterministic modelling system for the prediction of urban

NO_2 and PM_{10} concentrations. They found that the non-linear neural network models performed slightly better in terms of the model performance values than the deterministic model. Furthermore, the results also showed improved performance for most of the neural network models, compared with the linear statistical model, both for predicting NO_2 and PM_{10} concentrations. Ordieres et al. [16] compared three different topologies of neural networks to two classical models: a persistence model and a linear regression. The results clearly demonstrated that the neural approach not only outperformed the classical models but also showed fairly similar values among different topologies.

Besides the good non-linear regression abilities of neural networks, they also have some drawbacks. During the neural network optimization process, we have to move on to a surface having many local optima. Neural network's learning/optimizing algorithms cannot avoid from being stuck in a local optimum, which can lead to a sub-optimal solution. Another important deficiency is that the structure of a network is not given in advance; therefore, we have to optimize it as well. It means that we have to decide how many neurons and hidden layers would be necessary, and what kind of activation function or functions would be appropriate, and how to connect neurons with each other to form a network. Fortunately, an MLP with two hidden layers can approximate an arbitrary function [17–19], which negates some drawbacks mentioned above, however, the others remain unsolved.

Support vector regression addresses these limitations and gives promising results [20]. The basic idea, which is behind the SVR technique, is to start with linear regression, which has no parameters or only a few, and able to control the possible hypotheses (capacity) by considering the width of the margin of the regression plane. It would be useful to extend this technique in order to hold the almost parameterless property and the capacity control of the possible hypotheses, as well. An MLP, which is also an extension of the linear regression technique, chooses an explicit way for a non-linear description. It takes several linear regression methods and non-linearizes them by an activation function to get building blocks of the model. Then, it connects these blocks together to make a comprehensive non-linear model. This approach makes it hard to control the good properties of the regression mentioned above. Applying implicit mapping via a kernel function, an SVR redefines the dot product in the linear regression method in order to get a linear regression-like method. Due to the implicit non-linear mapping, this regression becomes non-linear. Thus, the simple and good properties of the linear regression are inherited. The implicit mapping provides an implicit description instead of the neural network's explicit one, which expresses the model required with an explicitly defined composite function. While SVR solves several drawbacks of the MLP (optimizing many parameters, choosing topology; being stuck in a local optimum [21]), it brings a new problem; namely, choosing an appropriate kernel function [22]. Nevertheless, this problem can be solved easily by choosing a proper kernel function from a small set. We have to maintain some additional parameters; namely, capacity control and parameters of the chosen kernel function [23]. Several papers suggest that SVR performs well in many time series prediction problems [24–29].

Both MLPs and SVRs have several successful applications in the field of prediction (see above). Therefore, they are readily chosen for predicting NO and NO_2 series. When a Gaussian kernel is used in an SVR, it corresponds to a radial basis MLP with Gaussian radial basis functions ('rbf') and one hidden layer. While the size of the hidden layer is unknown in the MLP approaches, the SVR automatically sets it. The size, i.e., the number of hidden neurons, is obtained as a result of the SVR optimization procedure. Hidden neurons and support vectors correspond to each other. Thus, the centre problem of the radial basis network is solved since the support vectors serve the centres of the basis functions. Considering this fact, radial basis network is not used in the paper. Instead of the radial basis function, a sigmoid activation function was used in our MLPs. The use of sigmoid-like functions is popular in the practice for both MLP and SVR [30, 31]. The application of the sigmoid function is even more justified, since training algorithms of these neural networks do not require positive definite property of the activation function, as opposed to an SVR, which needs positive definiteness for its kernel function [21]. In case of SVR, the parameters of the sigmoid function were set so that the function has positive definite property [32]. It allows us to compare the results of the application of the sigmoid-like function by two learning techniques.

In spite of the fact that neural networks and SVR are different learning techniques, the learnt hypotheses can be the same [33]. This is also a reason for comparing these methods.

The aim of the paper is to predict hourly averages of NO and NO_2 concentrations in a traffic junction in Szeged downtown, which has heavy traffic especially in rush hours. Furthermore, the methods mentioned above are compared with the reference ones. Reference methods do not have any parameters; nevertheless, our MLP and SVR methods have some. They were set by preliminary tests on the historical data by grid search in the parameter space of the algorithms, considering the suggestions of the program libraries used. Generally, these suggestions lead to good enough results, which are proved by our experiments as well. Nevertheless, parameter tuning by grid search leads to further considerable improvements.

2 Geographical, topographical, climatological and air quality characteristics of Szeged

2.1 The geographical position and topographical characteristics of Szeged

Szeged, as the largest town in SE Hungary (20°06E; 46°15N), is located at the confluence of the Tisza and Maros Rivers characterized by a landscape of extensive flats and an elevation of 79 m a.s.l. (Fig. 1). The built-up area covers a region of about 46 km^2 with about 165,000 inhabitants.

Szeged and its surroundings are not only characterized by extensive lowlands, but this city also has the lowest elevation value not only in Hungary but in the Carpathian Basin as well, rendering it a so-called "basin in the basin" or "double basin" situation. This special situation favours the development of stronger anticyclonic activity, enabling higher concentrations of pollutants in the air.

2.2 The climatic conditions of Szeged

The climate of Szeged is characterised by hot summers and moderately cold winters. The distribution of rainfall is fairly uniform during the year, with a share of 29 and 19% for the summer (JJA) and the winter (DJF) seasons, respectively. Mean daily summer temperatures are around 22.4°C, while the mean daily winter temperatures are 2.3°C. The irradiance values also exhibit large-scale variances with an average of 20.2 and 4.2 MJ m^{-2} in summer and winter days, respectively. The most frequent winds blow along the NNW–SSE axis, with prevailing air currents arriving from NNW (42.3%) and SSW (24%) in the summer and from SSE (32.6%) and NNW (30.8%) during the winter. Due to its unique geographical position, Szeged is characterised by relatively low wind speeds with mean daily summer and winter values of 2.8 and 3.5 m s^{-1}, respectively. The highest hourly wind speeds have been recorded during the spring with a rate of 5 m s^{-1} [34].

2.3 The air quality conditions of Szeged

Urban air quality largely depends on the actual measured values of meteorological parameters. The recorded averages of these variables for the city of Szeged are the following: annual mean temperature: 11.2°C; mean January and July temperatures: −1.2°C and 22.4°C, respectively, relative humidity: 71%; mean annual precipitation total: 573 mm; mean annual sunshine duration: 2,102 h; and mean annual wind speed: 3.2 m s^{-1}.

The city structure is very simple, characterized by an intertwined network of boulevards, avenues and streets sectioned by the River Tisza (Fig. 1). However, this simplicity largely contributes to the concentration of traffic, as well as air pollution within the urban areas.

The industrial area is mainly restricted to the north-western part of the city. Thus, the prevailing westerly and

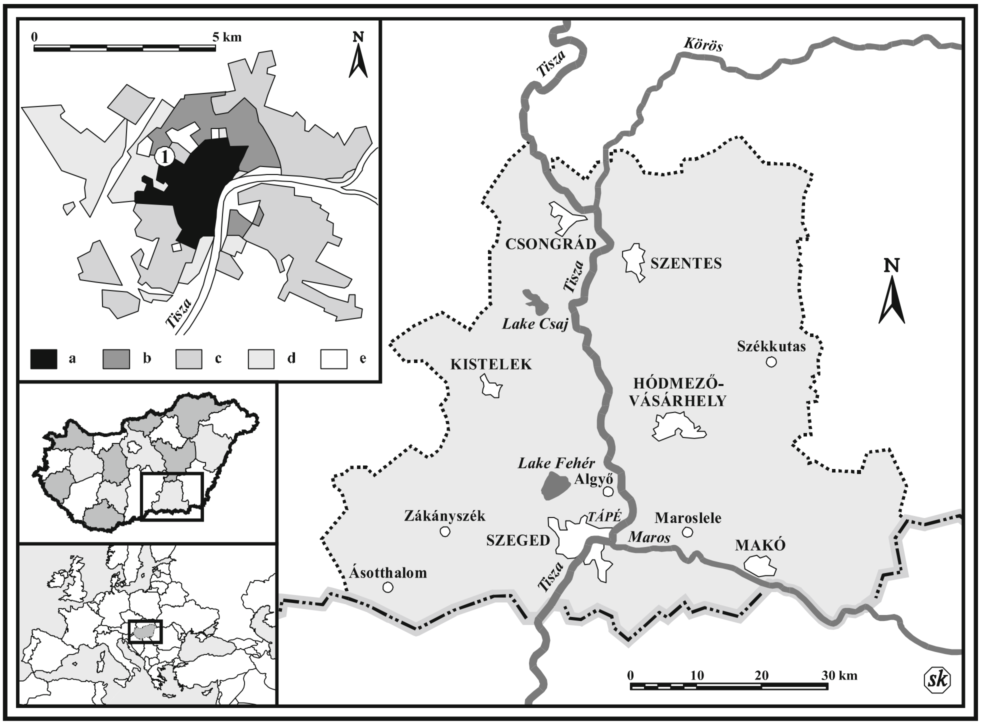

Fig. 1 The geographical position of Szeged, Hungary and built-up types of the city (*left, up*) **a** city centre (2–4-storey buildings), **b** housing estates with prefabricated concrete slabs (5–10-storey buildings), **c** detached houses (1–2-storey buildings), **d** industrial areas, **e** green areas, (1): monitoring station

northerly winds tend to carry the pollutants deriving from this area towards the centre of the city.

3 Data basis

3.1 Local meteorological and air pollution data basis

The data come from the monitoring station, which is located in Szeged downtown in a crossroad with heavy traffic (Fig. 1). The station is operated by the ATIKÖFE (Environmental protection inspectorate of Lower-Tisza Region, Branch of the Ministry of Environment). The training database comprises the period between 1 September 2000 and 12 March 2001, including autumn 2000, as well as the winter of the years 2000/2001. Hourly average mass concentrations of NO and NO_2 (in $\mu g\ m^{-3}$), as well as hourly means of temperature (°C), relative humidity (%) and wind speed ($m\ s^{-1}$) are considered for the period indicated. On the other hand, prediction occurred for the time span of 13–16 March, 2001.

4 Time series prediction

In the time series prediction, the aim is to predict the value of a variable that varies in time using previous values and/or other variables. Typically, the variable is continuous, so time series prediction is usually a specialized form of regression. Several authors [35] transformed the temporal dimension of a time series into a spatial vector of the l dimension embedding space by taking a moving window over the last l elements of the series. We can define two kinds of forecasts: (a) when no other information is used apart from the time series being examined (i.e., predicting without external variables); and (b) when other information is also available, (i.e., predicting with external variables).

4.1 Prediction without external variables

Suppose we have a real-valued time series $\{y_t\}_{t=1}^{n}$, i.e., the historical values of the series. When other time series, which can affect the y series are not given, the task is to predict y_{n+k} values with $k > 0$ based on the historical values. In other words:

$$y_{n+k} = h(y_n, y_{n-1}, \ldots, y_{n-(l-1)}), \quad k, l > 0 \tag{1}$$

The usual way to make the prediction is to find an appropriate l and a function h, which describes the relation between l consecutive elements and the next element of the series. Here, l denotes the historical window size and k represents the horizon of the future.

Our first aim is to use the above prediction schemes to predict NO and NO_2 time series. Secondly, external influences (external variables) are considered to improve the results obtained if possible.

4.2 Prediction with external variables

In some cases other time series are also known, which can influence the y-series under examination. They are called the external variables. If several factors are available, we can represent them as a vector series consisting of more than one scalar time series; namely, $\{\mathbf{z}_t\}_{t=1}^{n} = \{z_{t1}, \ldots, z_{tm}\}_{t=1}^{n}, \quad m > 0$. We can include this information in Eq. (1):

$$y_{n+k} = h(\hat{z}_{n+k}, y_n, y_{n-1}, \ldots, y_{n-(l-1)}), \quad k, l > 0 \tag{2}$$

Now suppose the (n+k)th value of the z-series or its estimate is known from some source, which will be denoted by $\widehat{z}_n$. Unfortunately, the problem of obtaining the information about $\widehat{z}_n$ is similar to that of Eq. 1. Hence, the **z**-series prediction should be made before y_n prediction. When the aim is to predict only the future value of y_{n+k}, the prediction is called a one-step-ahead prediction. However, if we intend to estimate values beyond y_{n+k}, we have to use the previously predicted values in the h function, and we call this a many-step-ahead prediction, especially with N-step-ahead predictions where the intention is to forecast the next N values.

5 Inductive learning

The inductive learning of a concept requires recognizing a hypothesis for this concept after presenting training instances, which is supervised by a defined classification. The instances are generally given in the following format during the learning/training process:

$$\underbrace{\underbrace{x_{i1}, x_{i2}, \ldots, x_{il}}_{\text{attributes}}, \underbrace{y_i}_{\text{class}}}_{i\text{th instance}}, \quad i \in N \tag{3}$$

In order to seek a relation (the hypothesis) between the attributes and their classes, a h function based on the training instances has to be approximated:

$$y_i = h(x_{i1}, x_{i2}, \ldots, x_{il}) \tag{4}$$

The above formula is suitable for the prediction schemes in Chap. 4 if we make an appropriate replacement in the arguments of the h function. The ith instance means the ith data window, while y_i is the value to be predicted. From here on, this more general notation will be used. The artificial intelligence (AI) learners were applied multi-layer perceptron and support vector regression; both of them have good approximation characteristics [18, 36].

5.1 Multi-layer perceptron (MLP)

The two-layered MLP is capable of approximating arbitrary finite sets of real numbers [18]. Hence, a maximum two hidden-layered MLP was used with a sigmoid activation function:

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{5}$$

where the input and output layers have linear units. When the l attributes of the ith learning instance take the form x_{i1}, … , x_{il}, then the class of this instance produced by the one hidden-layered MLP is shown in Eq. (6), while the result of the one hidden-layered MLP is described in Eq. (7). Equation (6) is a special one hidden-layered MLP called Perceptron, which is the building block of the multi-layer perceptron.

$$o_i^{1s} = \sigma\left(\sum_{t+1}^{l} w_t^{1s} x_{it} + w_{\text{bias}}^{1s}\right) \tag{6}$$

$$\begin{aligned} y_i &= \sum_{r=1}^{l_2} w_r o_i^{1s} \\ y_i &= \sum_{r=1}^{l_2} w_r \sigma\left(\sum_{s=1}^{l_1} w_s^{2r} o_i^{1s} + w_{\text{bias}}^{2r}\right) \end{aligned} \tag{7}$$

o_i^{1s}	output of the sth perceptron for the ith instance
w_i^{1s}, w_t^{2r}, w_r	weights in the first and second layers and output unit
w_{bias}^{1s}, w_{bias}^{2r}	biases in the first and second layers
l_1,l_2	number of perceptrons in the first and second layers
σ	sigmoid activation function

Changing the weights is the basis of the learning process. The well-known back-propagation method with momentum is used for adjusting the weights during the training process. An implementation was provided by the Weka software library [37, 38].

5.2 Support vector regression (SVR)

There are two commonly used support vector machines for regression; namely, the ε-SVR algorithm and its extension the ν-SVR algorithm [25]. We chose the ν-SVR because it has an advantage compared to ε-SVR. Namely, it is able to adjust automatically the width of the ε-tube around the function being approximated. An SVR maps the $x_i = x_{i1},\ldots,x_{il}$ attributes to a generally higher dimension space, called the feature space via a $\phi : R^l \to R^L$, $L \geq l$ map function. Then it makes a linear fit to certain accuracy by optimizing the weights $w = w_1,\ldots,w_L$ and w_{bias}:

$$y_i \approx \sum_{j=1}^{L} w_j \phi(x_{ij}) + w_{\text{bias}} \quad (= \langle w, \phi(x_i)\rangle + w_{\text{bias}}) \tag{8}$$

We can reformulate Eq. 8 by expanding the weight vector as a linear combination of the instance vectors $\left(w = \sum_{t=1}^{n} (\alpha_t^* - \alpha_t)\phi(x_t), \quad \alpha_t^*, \alpha_t \geq 0 :\right)$

$$\begin{aligned} &\sum_{t=1}^{n} (\alpha_t^* - \alpha_t)\langle \phi(x_t), \phi(x_i)\rangle + w_{\text{bias}} \\ &\quad \left(= \sum_{t=1}^{n} (\alpha_t^* - \alpha_t)\kappa(x_t, x_i) + w_{\text{bias}}\right) \end{aligned} \tag{9}$$

where κ is a kernel function belonging to the ϕ mapping. To obtain $\alpha_t{}^*$, α_t; ν-SVR maximizes the following quadratic problem for C, $\nu > 0$:

$$-\frac{1}{2}\sum_{i,j=1}^{n} (\alpha_i^* - \alpha_i)\kappa(x_i, x_j)(\alpha_j^* - \alpha_j) + \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) y_i \tag{10}$$

subject to the constraints

$$\sum_{i=1}^{n} (\alpha_i^* + \alpha_i) = 0$$

$$\sum_{i=1}^{n} (\alpha_i^* + \alpha_i) \leq Cn\nu$$

$$0 \leq \alpha_i^* + \alpha_i \leq C$$

Three kinds of well-known kernel functions were employed: (a) a radial basis function $\kappa(x_i, x_j) = \mathrm{e}^{-\gamma\|x_i - x_j\|^2}$, (b) a polynomial one $\kappa(x_i, x_j) = -\gamma\langle x_i, x_j\rangle^d$ and (c) a sigmoid-like function, namely the hyperbolic tangent $\kappa(x_i, x_j) = \tanh(\gamma\langle x_i, x_j\rangle)$, where $\tanh(x) = (\mathrm{e}^x - \mathrm{e}^{-x})/(\mathrm{e}^x + \mathrm{e}^{-x})$. A ν-SVR implementation was provided by the LibSVM software library [39, 40].

5.3 Model selection by grid search (GS)

Reference model, i.e., linear regression and persistence, have no parameters to tune, therefore, they are a good choice for making a basis of these experiments. However, MLP and ν-SVR are sophisticated techniques, they suffer from the parameter selection problem, which has a large influence on the results. Both models have several parameters which have default values provided by the libraries used, but for the sake of good prediction these parameter values need modification according to the underlying distribution of the historical data. Among the several parameter estimation techniques the grid search is the most reliable because it makes exhaustive search in the parameter space. Of course, only a subspace of the whole can be discovered due to the huge amount of computational efforts. LibSVM and Weka libraries have built in model selections using grid search technique hence, we relied on them. We compared prediction performance with and without model selection, i.e., using standard library settings of parameters and setting parameters by grid search on the "subspace", more exactly on a grid of the possible parameter values.

In case of the MLP, for calculating the optimal number of neurons in the layers, we used a grid with interval [1, 24] of integers for one coordinate and interval [0, 24] of integers for the other, respectively, to the number of neurons in the first and second layer of the MLP, where back propagation algorithm was used as training. The grid which was used in the optimization of the parameters of the ν-SVR was the following: ν parameter from 0.01 to 0.99 with initial steps of 0.05 and the γ parameter from 0.1 to 8.0 with initial steps of 1.0, where further steps were taken by the usual two factor exponential enlargements.

6 Experiments

In our experiments real meteorological data were used, which came from a monitoring station located in the city of Szeged in Hungary. The time series examined were 1-h averages of nitric oxide and nitrogen dioxide, which consisted of a 6-month data set in the period 1 September 2000–16 March 2001.

The diurnal cycles of NO and NO_2 have the shape of a double wave (Fig. 2), with bigger amplitudes for NO than for NO_2. Due to the traffic density, the concentration of NO is relatively higher on weekdays, than on weekends (Fig. 4). This effect can also be observed for the secondary substance NO_2 (Fig. 4). The average diurnal variations on weekdays are higher for NO than for NO_2, because NO_2 has a longer lifespan than the more reactive NO (Fig. 2). Generally, NO concentrations are higher in the morning, than in the evening (Fig. 2). This can be explained by the fact that in the morning the rush hour is shorter, and the atmosphere near the surface is more stable than in the evening. The low NO concentrations early in the afternoon result mainly from the reduction of O_3 by NO [41]. NO_2 concentration depends on that of NO; hence, concentration of the latter pollutant is very useful to predict NO_2 levels (Fig. 2).

Our experiments showed that concentrations of NO are more precisely predictable than NO_2. Therefore, the NO series were also used as an external variable to predict NO_2 [5, 2]. Temperature (T), relative humidity (H) and wind velocity (W) as external variables were employed by several authors [11, 2, 15], [5, 42]. Thus, values of these meteorological variables might be included as inputs to an algorithm in order to improve the forecast of NO and NO_2 concentrations.

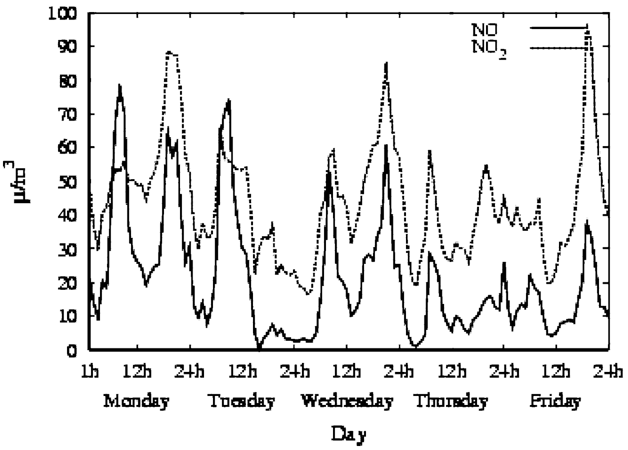

Fig. 2 Real values of the NO and NO_2 series at the forecasting (test) term

One-step-ahead prediction was applied for a 4-day forecast (13–16 March) for the period 1 September–12 March (Fig. 2). Weekend data, because of the less traffic, were omitted from the database. Similar assumptions can be found in the following papers: [11, 2, 15, 5, 42]. The time series have received natural 24-h periods, confirmed by their autocorrelation diagram (Fig. 3).

When external variables were applied in the experiments, their real values were used in order to avoid cumulative errors. However, it is important to examine relations between the time series considered and the external variables mentioned.

The performance of the mentioned AI methods was compared with three commonly used reference algorithms. The first reference algorithm is called linear regression (LR), where the past values of the data were weighted to result prediction. The next reference algorithm is the persistence, which models the persistence of the values of the days. A prediction value of a future hour has the same value as that for the same hour of the previous day. This simple technique works well on this problem because the series has a 24-h periodicity (Fig. 4). Average values of several past days can lead to a better persistence method; namely, to the averages of several-day persistence (persistence average). Consequently, the persistence and the persistence average are not able to handle external variables. Agirre-Basurko et al. [11], Gardner and Dorling [2], Kukkonen et al. [15] and Perez and Trier [5] showed that

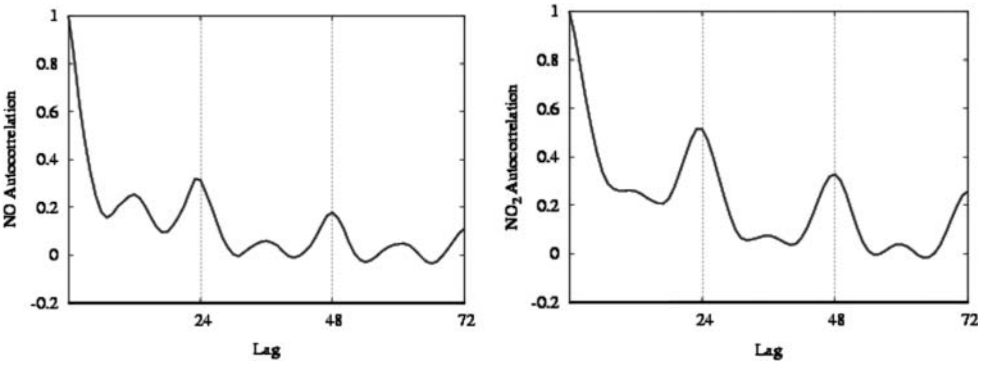

Fig. 3 Autocorrelation of the NO (*left*) and NO_2 series (right)

all the reference methods mentioned work well on NO and NO_2 predictions. They also revealed that these reference methods can be as good as an MLP or, in certain cases, even better.

Our experiments showed that the persistence average performed well better than the persistence itself. Based on the good results of persistence average, a prediction scheme was introduced, which forms not equally weighted averages instead of simple persistence average. Optimization of these weights was performed by linear regression in linear case and by the learning methods mentioned in non-linear case. Another difference between the persistence average and the schemes introduced (Scheme 3–4 in Table 1 and Scheme 7–8 in Table 2) is that 2–10-day looking back periods were employed instead of the whole historical days averages. Example, in case of a 3-day looking back period, it is NO($t+k$) = h(NO($t-23$), NO($t-47$), NO($t-71$)) = h(NO[$t-23$, $t-47$, $t-71$]), k = 1, ..., 24. These schemes can be applied to the NO and NO_2 series predictions. However, the weights were adjusted by optimization learning weights.

Different kinds of MLP learners were used according to the number of neurons in the hidden layers. They are denoted as MLP_{l_1,l_2} where l_1 and l_2 mean the number of neurons in the first and second hidden layers, respectively. Beside the MLPs, different kinds of ν-SVRs were trained by the mentioned kernel types for each hour in a day. They provide learner-specific hypotheses, which will be denoted by MLP_{l_1,l_2} and $\nu - SVR_{\{rbf,poly2,poly3,sigmoid\}}$ (degree is shown in the subscripts in case of the polynomial kernel, e.g. degree is 2 when *poly2* appears). Linear regression, persistence and persistence average have no parameters but MLP and ν-SVR have some, which need to be set properly. Standard library settings (LS) were chosen for each in order to analyse their standard behaviour on the data. In the case of MLP, these settings were as follows: learning rate was 0.3, momentum was 0.2, and the number of training epochs was 500. Weka software library [37] suggests $MLP_{1,0}$, $MLP_{a,1}$, $MLP_{a,a/2}$ to use, where "a" is the number of attributes. A scheme was chosen for the best model of these three different models, which was denoted by MLP (LS). Due to the different library suggestions, MLP (LS) can be varied from scheme to scheme. This is the case for the ν-SVR (LS) as well, where the four kinds of kernels (rbf, poly2, poly3, sigmoid) gave alternatives, however, in fact, rbf and poly2 performed good enough on the problems. For other parameters of the ν-SVR learner the following settings were chosen: γ is $1/a$, ν is 0.5, and ε is

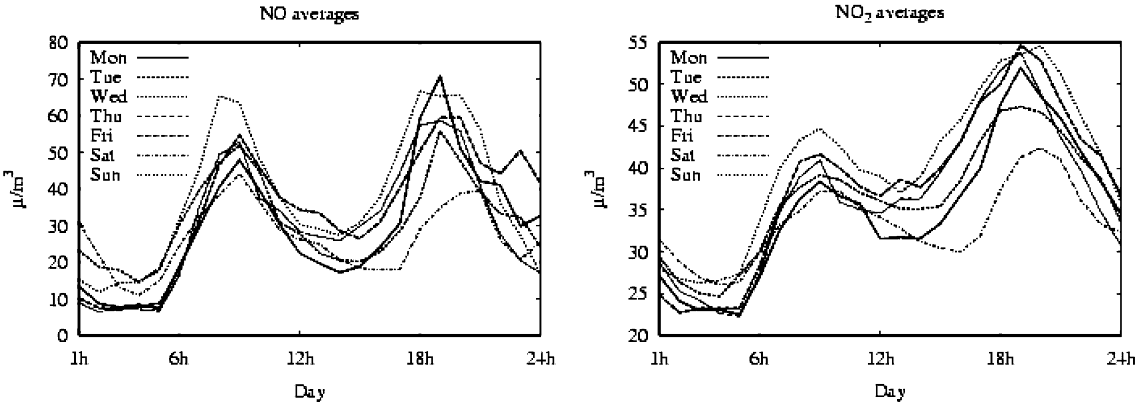

Fig. 4 Hourly average values of NO (*left*) and NO_2 (*right*) in the historical period

Table 1 NRMSE of the one-step ahead predictions of the NO series, according to different prediction schemes, using reference model (italic letters), and learning models with selected library settings (LS) and optimized settings by grid search (GS)

NO	Scheme 1 (S1) NO[t, t–1,…, t–23] each hour of the previous day NRMSE	Scheme 2 (S2) NO[t, t–1,…, t–23], $W(t+k)$ each hour of the previous day and a known factor for the prediction time NRMSE	Scheme 3 (S3) NO[t–24+k, t–48+k] 2-day step back for the same hour NRMSE	Scheme 4 (S4) NO[t–24+k, t–48+k], $W(t+k)$ 2-day step back for the same hour and a known factor for the prediction time NRMSE
Linear regression	*0.369*	*0.381*	*0.437*	*0.408*
Persistence	*0.408*	*0.408*	*0.408*	*0.408*
Persistence average	*0.386*	*0.386*	*0.386*	*0.386*
N-SVR (LS)	0.291	0.284	0.314	0.279
MLP (LS)	0.351	0.365	0.583	0.583
N-SVR (GS)	0.224	0.212	0.276	0.206
MLP(GS)	0.259	0.298	0.511	0.484

The following settings were found the best and applied for the models to the (S1, S2, S3 and S4) schemes, separately: kernel types of ν-SVR (LS) were (poly2, poly2, rbf, rbf); kernel types of ν-SVR (GS) were (rbf, poly2, poly2, rbf); number of nodes in the hidden layers of MLP (LS) were ([24, 12], [24, 1], [1, 0], [1, 0])

Table 2 NRMSE of the one-step ahead predictions of the NO2 series, according to different prediction schemes, using reference model (italic letters), and learning models with selected library settings (LS) and optimized settings by grid search (GS)

NO_2	Scheme 5 (S5) NO_2[t, t–1, …, t–23] each hour of the previous day NRMSE	Scheme 6 (S6) NO_2[t–23+k, t–47] 2-day step back for the same hour NRMSE	Scheme 7 (S7) NO_2[t–24+k, t–48+k], $H(t+k)$, $^2T(t+k)$, $W(t+k)$ 2-day step back for the same hour and known factors for the prediction time NRMSE	Scheme 8 (S8) NO_2[t–24+k, t–48+k], $H(t+k)$, $T(t+k)$,$W(t+k)$, NO($t+k$) 2-day step back for the same hour and known factors for the prediction time NRMSE
Linear regression	*2.154*	*0.963*	*0.816*	*0.735*
Persistence	*1.286*	*1.286*	*1.286*	*1.286*
Persistence average	*0.889*	*0.889*	*0.889*	*0.889*
ν-SVR (LS)	0.757	0.871	0.744	0.744
MLP (LS)	1.216	1.073	0.903	0.849
ν-SVR (GS)	0.662	0.672	0.647	0.600
MLP(GS)	1.095	0.895	0.810	0.794

The following core settings were found the best and applied for the models to the (S1, S2, S3 and S4) schemes, respectively: kernel types of ν-SVR (LS) were (poly2, poly2, rbf, rbf); kernel types of ν-SVR (GS) were (rbf, rbf, rbf, rbf); number of nodes in the hidden layers of MLP (LS) were ([1, 0], [1, 0], [1, 0], [1, 0])

0.1. When grid search tuned the parameters of the learners, then abbreviation (GS) was used to define the presence of automatic parameter selection process.

First, an integrated result is shown characterizing the performance. The normalized root mean squared error (see NRMSE in Eq. 11) gives rough but important integrated information about the performance. Normalization of the error is important to compare the predictions of the time series obtained in different places or times.

$$\mathrm{NRMSE} = \frac{\sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}}}{\mathrm{stdev}(y_i)}, \tag{11}$$

where $\hat{y}$ is the estimation of the y_i-series and n is the length of the series, while stdev is the abbreviation of the standard deviation. In order to get a detailed comparison of the predictions, the daily averages of the hourly absolute errors ($|y_i - \hat{y}|$) were also analyzed in the forecast period. Thus, we can define those hours of the day, where a method performs well.

Since the series have 24-h periods, it is expedient to choose the size of the embedding dimension, i.e., the window size for a prediction scheme 24 (e.g. NO($t+k$) = h(NO[t, $t-1,\ldots,t-23$]), k = 1,...,24).

6.1 NO series prediction

We found that the Scheme 4 type predictions gave the best results if we used only 2-day looking back period in the past, where looking back trials were applied from 2 to 10 days. Based on Schemes 1 and 3 (factor-less predictions), investigations were made by extending these schemes with external variables: wind velocity, humidity and temperature; first one of them at a time, then all of them together. Results showed that only wind velocity could improve the accuracy of the predictions (Schemes 2 and 4 in Table 1). These factors seem to have less significance to improve the results of Scheme 1. However, the factorless results are very impressive. It is to be noted that if external variables are used, they also need to be predicted. Thus, the results received might be worse due to the cumulative prediction error.

The SVR predictions with the appropriate kernel function performed significantly better than the others. SVR with Gaussian (rbf) and polynomial kernel seems to be an efficient predicting tool with schemes.

Schemes 3 and 4, which look back more than 1 day in the past, did not generally bring additional improvements without using external variables, and showed slightly better results in some cases when wind velocity was used. It is clear that this factor has some, probably non-linear, influences on the NO series (Table 1; Figs. 5, 6).

The SVR error curves show smoother and more reliable predictions in average than the MLP, which produced several peaks, e.g., around 9 am, (Figs. 5, 6). The external variable, the wind velocity, was able to regularize the MLP results and brought improvements in Fig. 6.

On one hand, reference methods gave good results while, on the other, the applied SVR technique outperformed the results of the reference methods. It is to be noted that, however, MLP produced good results, it was not enough on Schemes 3 and 4 where the MLP performance found to be worse than the references.

Furthermore, SVR and linear regression showed their best results using only the wind velocity factor. They could not improve their results using the three factors together with Scheme 1. However, MLP could improve its result by approximately 10% in an average using all the three factors. These self-improvements of the MLPs remained behind the best results of the SVR. While linear regression follows the persistence curve, the MLP and SVR reduced significantly the prediction error after 12 h (Figs. 5, 6).

6.2 NO_2 series prediction

Perez and Trier [5], and Gardner and Dorling [2] suggested using NO series as an external variable for predicting NO_2 series. Relations of the two series are shown in Fig. 2. Perez and Trier [5] showed that the NO series can be predicted with more accuracy than the NO_2 series. Our results confirmed this statement, since there are worse results in Table 2 than in Table 1.

SVR with Gaussian (rbf) and polynomial kernels gave better results in schemes as it has been displayed for the NO predictions (Table 1).

The extension of Scheme 5 with the variants of the factors of humidity, temperature, wind velocity and NO did not bring any additional improvement. It is probably due to

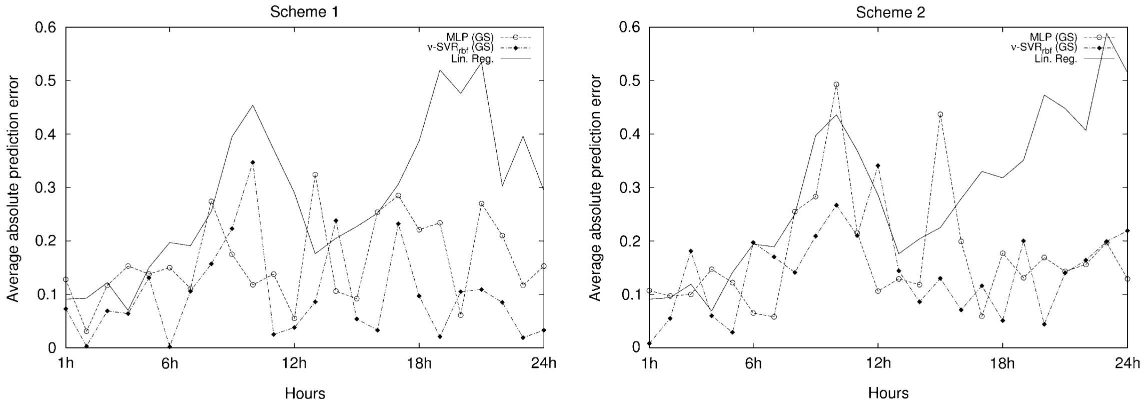

Fig. 5 Average prediction errors of the four prediction days of the NO series. Prediction errors are absolute errors normalized by the standard deviation of the series. The curves are related to the best results in Table 1. The MLP (GS) and ν-SVR (GS) are compared to the best reference result by the scheme 1 (*left*) and scheme 2 (*right*) predictions

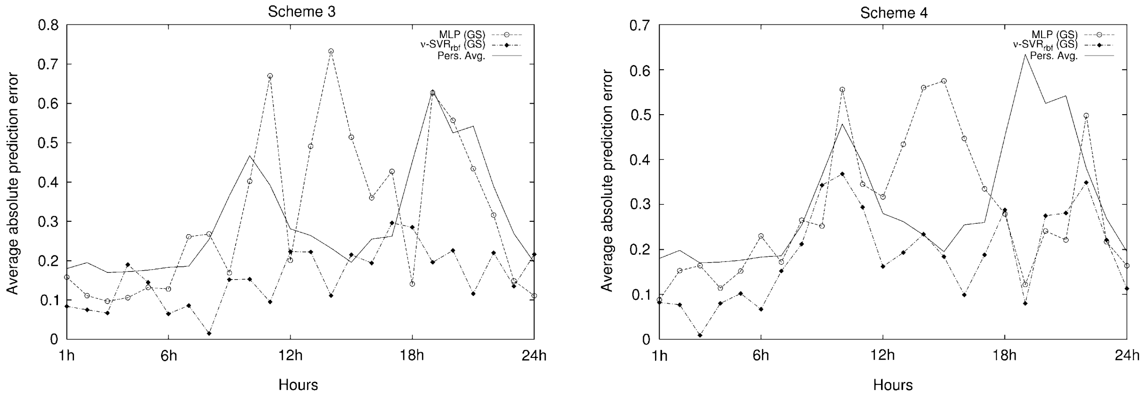

Fig. 6 Average prediction errors of the four prediction days of the NO series. Prediction errors are absolute errors normalized by the standard deviation of the series. The curves are related to the best results in Table 1. The MLP (GS) and *ν*-SVR (GS) are compared to the best reference result by the scheme 3 (*left*) and scheme 4 (*right*) predictions

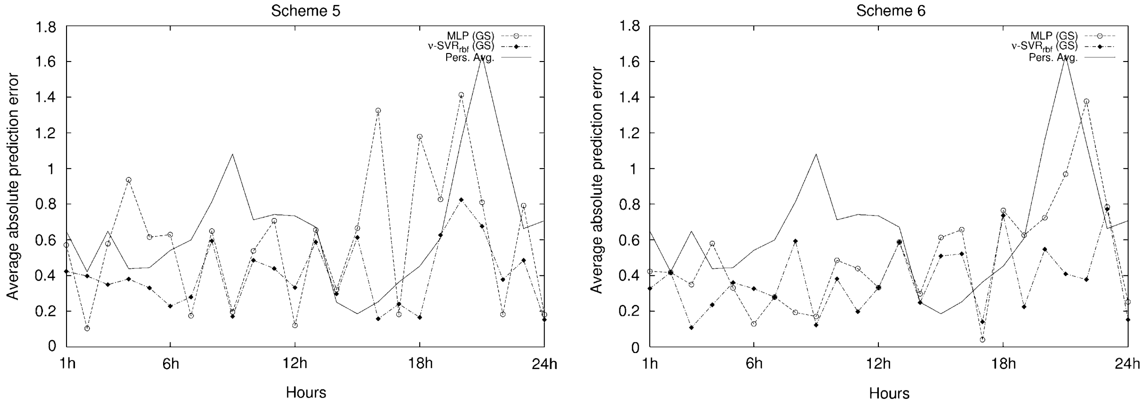

Fig. 7 Average prediction errors of the four prediction days of the NO_2 series. Prediction errors are absolute errors normalized by the standard deviation of the series. The curves are related to the best results in Table 1. The MLP (GS) and *ν*-SVR (GS) are compared to the best reference result by the scheme 5 (*left*) and scheme 6 (*right*) predictions

the dominant number of non-factor variables in this scheme, where 24 non-factor variables are versus maximum 4 factor variables.

The 2-day looking back period scheme (Scheme 6) brought the best results among the 2–10-day looking back schemes. The same phenomenon was experienced for the NO series prediction. In addition, the above mentioned *H*, *W*, *T* and NO external variables brought significant improvements for Scheme 6 (Scheme 7–8 in Table 2). These significant improvements were not experienced for the NO series predictions.

Linear regression and persistence average produce bad predictions in the rush hours (9–11 am and 18–20 pm) compared to the other methods, however, persistence is better considering these terms, but cannot outperform the non-linear MLP and SVR methods in general. Nevertheless, the MLP sometimes produces even worse results than the best reference method. MLP shows its best results in the early morning hours in the NO prediction, which is in agreement with those of Perez and Trier [5]. Figures 7, 8 show that MLP becomes better in the late afternoon hours, when external variables are presented in the forecast of NO_2 series. SVR produces smooth error curves with smallest errors during the whole day, while MLP gives less reliable estimations especially for the hours around 9 am.

7 Conclusions

The experiments clearly showed that the applied forecasting techniques could perform well on the prediction of NO and NO_2 concentrations. Forecasting these air pollutants is difficult because their concentrations fluctuate widely and depends on several factors. In many cases, the three reference algorithms proved to be successful at predicting the future values of the time series examined. These results are in accordance with those of Perez and Trier [5] and Kukkonen et al. [15]. Averages of several-day persistence

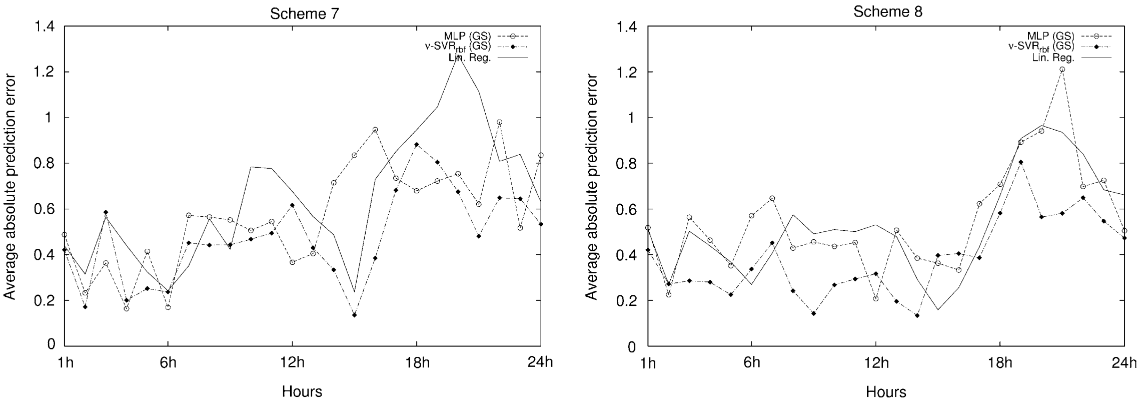

Fig. 8 Average prediction errors of the four prediction days of the NO_2 series. Prediction errors are absolute errors normalized by the standard deviation of the series. The curves are related to the best results in Table 1. The MLP (GS) and ν-SVR (GS) are compared to the best reference result by the scheme 7 (*left*) and scheme 8 (*right*) predictions

performed well. In order to profit from this good performance, schemes based on this kind of persistence were introduced.

MLP (GS) can improve the results of the best reference algorithms by 21–30% in the case of NO prediction using Scheme 1 and Scheme 2, however, it could not improve the other schemes either in case of NO or NO_2 experiments. SVR showed 21–47% improvement for NO prediction and 18–26% for NO_2 prediction. It was found that the applied [t, $t-1$,…,$t-23$] scheme, where the previous day values were considered, was successful for both NO and NO_2 predictions. However, the several-day persistence motivated [$t-24+k$, $t-48+k$,…] scheme (where several previous days were considered) was suitable for the external variables aided predictions, especially, for the NO_2 series. This is probably due to the less number of non-factor variables in this scheme.

Undoubtedly, the application of machine learning techniques mentioned above can be relatively simple and is worth using.

There are several possibilities for future work. We can transform spatial embeddings of the historical values of NO and NO_2 into a lower dimensional space in order to have possibility to exploit better the influence of the external factors and reduce the learning time, making more accurate and faster predictions. Furthermore, we can make a hybrid method using the applied prediction algorithms together. Pre-processing of data (e.g., smoothing or denoising) can lead to more predictable structures. These methods are easily adaptable for forecasting other air pollutants.

Acknowledgments The authors are indebted to Gábor Motika (Environmental protection inspectorate of Lower-Tisza Region, Szeged, Hungary) for handing monitoring data of the meteorological parameters and the air pollutants and Zoltán Sümeghy (Department of climatology and landscape ecology, University of Szeged, Hungary) for digital mapping, as well as Rita Béczi (Department of climatology and landscape ecology, University of Szeged, Hungary) for useful suggestions. This study was supported by the EU-6 Project "QUANTIFY" [No. 003893 (GOCE)] and the high performance computing group of the University of Szeged.

References

1. Gardner MW, Dorling SR (1998) Artificial neural networks (the multi-layer perceptron)—a review of applications in the atmospheric sciences. Atmos Environ 32:2627–2636
2. Gardner MW, Dorling SR (1999) Neural network modelling and prediction of hourly NO_x and NO_2 concentrations in urban air in London. Atmos Environ 33:709–719
3. Jorquera H, Pérez R, Cipriano A, Espejo A, Letelier MV, Acuña G (1998) Forecasting ozone daily maximum levels at Santiago, Chile. Atmos Environ 32:3415–3424
4. Perez P, Trier A, Reyes J, (2000) Prediction of $PM_{2.5}$ concentrations several hours in advance using neural networks in Santiago, Chile. Atmos Environ 34:1189–1196
5. Perez P, Trier A (2001) Prediction of NO and NO_2 concentrations near a street with heavy traffic in Santiago, Chile. Atmos Environ 35:1783–1789
6. Perez P, (2001) Prediction of sulfur dioxide concentrations at a site near downtown Santiago, Chile. Atmos Environ 35:4929–4935
7. Perez P, Reyes J (2001) Prediction of particlulate air pollution using neural techniques. Neural Comput Appl 10(2):165–171
8. Chelani AB, Chalapati RCV, Phadke KM, Hasan MZ (2002) Prediction of sulphur dioxide concentration using artificial neural networks. Environ Modell Softw 17(2):159–166
9. Mechaqrane A, Zouak M (2004) A comparison of linear and neural network ARX models applied to a prediction of the indoor temperature of a building. Neural Comput Appl 13(1):32–37
10. Maqsood I, Riaz Khan M, Ajith Abraham A (2004) An ensemble of neural networks for weather forecasting. Neural Comput Appl 13(2):112–122
11. Agirre-Basurko E, Ibarra-Berastegib G, Madariaga I (2006) Regression and multilayer perceptron-based models to forecast hourly O_3 and NO_2 levels in the Bilbao area. Environ Modell Softw 21(4):430–446
12. Hansen JV, McDonald JB, Nelson RD (1999) Time series prediction with genetic-algorithm designed neural networks: an

empirical comparison with modern statistical models. Comput Intell 15:171–184
13. Small M, Tse CK (2002) Minimum description length neural networks for time series prediction. Phys Rev E 66:066701-1–066701-12
14. Castillo O, Melin P (2002) Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic and fractal theory. IEEE T Neural Netw 13:1395–1408
15. 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 network 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
16. Ordieres JB, Vergara EP, Capuz RS, Salazar RE (2005) Neural network prediction model for fine particulate matter ($PM_{2.5}$) on the US–Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua). Environ Modell Softw 20(5):547–559
17. Blum E, Li L (1991) Approximation theory and feedforward networks. Neural Netw 4:511–515
18. Chester D (1990) Why two hidden layers are better than one. In: Erlbaum L (ed) International Joint Conference on Neural Networks, Proceedings 1, Washington, D.C., pp 265–268
19. Hornik K (1991) Approximation capabilities of multi-layer feedforward networks. Neural Netw 4(2):251–257
20. Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V (1997) Support vector regression machines. In: Mozer M et al (eds.). Advances in neural information processing systems, 9, The MIT Press, Cambridge, pp 155–161
21. Suykens JAK, De Brabanter J, Lukas L, Vandewalle J (2002) Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing 48:85–105
22. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167
23. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press
24. Vapnik V, Golowich S, Smola A (1997) Support vector method for function approximation, regression estimation, and signal processing. In: Mozer M, et al (eds) Advances in neural information processing systems 9. The MIT Press, Cambridge, pp 281–287
25. Schölkopf B, Bartlett P, Smola A, Williamson R (1998) Support vector regression with automatic accuracy control. In: Niklasson L et al (eds) Proceedings of the international conference on artificial neural networks, perspectives in neural computing. Springer, Berlin, pp 111–116
26. Ancona N (1999) Properties of support vector machines for regression. Technical report, Instituto Elaborazione segnali ed immagini, Bari, Italy, pp. 01–99
27. Müller KR, Smola A, Rätsch G, Schölkopf B, Kohlmorgen J, Vapnik V (1999) Using support vector machines for time series prediction. In: Schölkopf B et al (eds) Advances in kernel methods—support vector learning. Proceedings of the NIPS, workshop on support vectors. The MIT Press, Cambridge, pp 1–12
28. Schölkopf B, Burges CJC, Smola AJ (eds) (1999) Advances in kernel methods—support vector learning, proceedings of the NIPS workshop on support vectors. The MIT Press, Cambridge
29. Van Gestel T, Suykens J, Baestaens D, Lambrechts A, Lanckriet G, Vandaele B, De Moor B, Vandewalle J (2001) Financial time series prediction using least squares support vector machines within the evidence framework. IEEE T Neural Network, Spec Issue Neural Network Financ Eng 12(4):809–821
30. Reed RD, Marks RJ (1999) Neural smithing: supervised learning in feedforward artificial neural networks. MIT Press, Cambridge
31. Schölkopf B (1997) Support Vector Learning. PhD Thesis. Oldenbourg R. Verlag, Munich
32. Lin H, Lin C (2003) A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, department of computer science and information engineering, National Taiwan University
33. Kecman V (2001) Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. The MIT Press, Cambridge
34. Péczel, G (1979) Climatology (in Hungarian), Tankönyvkiadó, Budapest, pp. 258–284
35. Weigend AS, Gershenfeld NA (eds) (1994) Time series prediction: forecasting the future and understanding the past. Addison–Wesley, Reading, MA
36. Hammer B, Gersmann K (2003) A note on the universal approximation capability of support vector machines. Neural Process Lett 17:43–53
37. Witten IH, Frank E (2000) Data mining: practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco
38. Weka data mining software library. http://www.cs.waikato.ac.nz/ml/weka. Accessed February 2005
39. Chang CC, Lin CJ (2002) Training nu-support vector regression: theory and algorithms, Neural Comput 14(8):1959–1977
40. LibSVM support vector software library http://www.csie.ntu.edu.tw/~cjlin/libsvm . Accessed February 2005)
41. Makra L, Mayer H, Mika J, Sánta T, Holst J (2008) Variations of traffic related air pollution on different time scales in Szeged, Hungary and Freiburg, Germany. Phys Chem Earth (accepted)
42. Shi P, Harrison RM (1997) Regression modelling of hourly NO_x and NO_2 concentrations in urban air in London. Atmos Environ 31(24):4081–4094