Air Quality Forecasting using Neural Networks

Chen Zhao, Mark van Heeswijk, and Juha Karhunen
Aalto University School of Science, FI-00076, Finland
{chen.zhao, mark.van.heeswijk, juha.karhunen}@aalto.fi

***Abstract*—In this paper, a neural network approach is proposed for air quality forecasting based on the air quality time series itself as well as external meteorological records. A regularized version of the Extreme Learning Machine is used as the main model for the forecasts and feature selection is performed to select the most relevant model inputs. The proposed method is evaluated under different approaches for performing spatial data fusion. Experiments show that accuracy is increased by considering meteorological data; that it matters greatly for the model how the spatial aspect of the problem is taken into account; and finally, that the model is generally able to select relevant inputs and provide accurate air quality forecasts.**

I. Introduction

Air quality is one of the most important parts defining the environment and has a close relationship with human health. For example, a $10 - \mu g/m^3$ elevation in the concentration of particulate matter (PM) has been associated with 8% to 18% increase in mortality risk [11]; and short-term O_3 exposure could bring about acute coronary events [12]. Therefore, it is important to predict the air quality so that people can arrange their activities accordingly, both in time and location.

There are two main approaches to air quality prediction: deterministic approaches (e.g. modeling the process by which pollutants spread, and predict air quality given information on pollutant sources and meteorological conditions), and statistical approaches (e.g. relying on extensive historical air quality records) [3]. However, in the deterministic approach, real-time forecasting can be difficult due to the complexity of the simulations, and the accuracy depends on the available knowledge on the pollutant sources, which might be incomplete. Statistical approaches, however, rely on extensive historical measurements at spatially distributed monitoring stations.

An example of the statistical approach is the work by Gardner et al. [4], which investigate neural networks for air quality prediction. The research shows the capability of neural networks of learning complex patterns, even with a simple multilayer perceptron (MLP) with 20 neurons and 2 layers [4]. Further work by Kolehmainen et al. shows that a 16-neuron MLP outperforms the regression method with periodic components extracted from NO_2 and meteorological time series data [8]. Finally, Voukantsis et al. [16] uses principal component analysis (PCA) to summarize the information across 9 meteorological variables into 4 principal components (PC), and shows that reasonable accuracy can be achieved, while reducing the number of variables.

However, due to limited computational capability, the number of neurons in the above MLP networks is no larger than 20. The amount of neural network input variables is also relatively small in the above cases: Gardner et al. [4] uses meteorological records from only one station, while Kolehmainen et al. [8] uses the average of records from four stations. And while Voukantsis et al. [16] reduces the number of variables using PCA, only daily records are used as input for the MLP when hourly records are available.

Training of a large-scale MLP network can be slow, and tricky due to local minima and other problems such as long plateaus in error minimization. Extreme Learning Machine (ELM) [7] is a simpler neural network which is fast to train and does not suffer from local minima. Yet it has a good representative power, and it provides good results, both in classification and regression problems. It has been used successfully in a variety of applications [6], including air quality prediction [15], where ELM was used to predict the daily average level of PM_{10} in Macau and shown to provide better results than a Support Vector Machine.

This paper proposes a method based on ELM to predict the air quality using hourly records of air condition and meteorological data from multiple stations within a relatively small area around Helsinki. Unlike above research projects, the training data size is much larger, namely, two years of hourly air quality data from multiple locations. Furthermore, there are 5 types of meteorological hourly records available from more than 20 stations. Because of the large amount of data and possible inputs for the model, variable selection and dimensionality reduction are used to identify and generate variables important for accurate predictions. The proposed approach is tested using different methods for incorporating the spatial aspect of the problem, and is shown to be able to identify relevant variables for the prediction and provide accurate air quality forecasts.

II. Theory / Methods

A. Extreme Learning Machine (ELM)

1) Basic version: Let N be the total number of training samples where each sample is represented by a pair $(\mathbf{x}_j, y_j)$ where $\mathbf{x}_j$ is the j^{th} sample with dimensionality d and y_j is the observed scalar output. Assume the weights for the hidden neurons are in vector form $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_L)$ and bias as $\mathbf{b} = (b_1, b_2, ..., b_L)$, where $\mathbf{w_i}$ is the d-dimensional weight vector containing the weights between the components of the input vectors and the i^{th} neuron in the hidden layer, and b_i is the bias term of the i^{th} hidden neuron. The network with L hidden neurons will have $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_L)$ as weights for output layer and $f(.)$ is the activation function. The mathematical expression of the network can be written as

$$\hat{y}_j = \sum_{i=1}^{L} \beta_i f(\mathbf{w}_i^T \mathbf{x}_j + b_i) \;\; j \in [1, N],$$

where $\hat{y}_j$ is the predicted value of y_j. The key idea in the Extreme Learning Machine is that the weights $\mathbf{W}$ and biases $\mathbf{b}$ of the hidden layer are generated randomly and remain fixed. Denote

$$\mathbf{H} = \begin{bmatrix} f(\mathbf{w}_1^T \mathbf{x}_1 + b_1) & \cdots & f(\mathbf{w}_L^T \mathbf{x}_1 + b_L) \\ f(\mathbf{w}_1^T \mathbf{x}_2 + b_1) & \cdots & f(\mathbf{w}_L^T \mathbf{x}_2 + b_L) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1^T \mathbf{x}_N + b_1) & \cdots & f(\mathbf{w}_L^T \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1},$$

where β_i is the scalar weight between the single output neuron and the i^{th} neuron in the hidden layer. The output of the ELM, modeling the observations can now be written as

$$\hat{\mathbf{Y}} = \mathbf{H}\boldsymbol{\beta}.$$

Because the output layer in the ELM is linear, in our case consisting of a single linear neuron, the unknown weights $\boldsymbol{\beta}$ of the output layer can be solved from linear equations as follows:

$$\begin{aligned} \mathbf{H}\boldsymbol{\beta} &= \mathbf{Y} \\ \mathbf{H}^T\mathbf{H}\boldsymbol{\beta} &= \mathbf{H}^T\mathbf{Y} \\ (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{H}\boldsymbol{\beta} &= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y} \\ \boldsymbol{\beta} &= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y} \end{aligned}$$

where $(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is known as the Moore-Penrose pseudo-inverse of $\mathbf{H}$, represented by $\mathbf{H}^\dagger$. This is the least-squares solution for the realistic case of an over-determined linear system (i.e. in which the number of training samples N is larger than the number of hidden neurons L). There is another form $H^+ = H^T(HH^T)^{-1}$ for the case $N < L$. However, since in this work $N \gg L$, only the former is considered. Note that although the output layer is linear, the nonlinearities $f(.)$ in the hidden layer still give the ELM a strong representative power. For more details, see [7]. The whole algorithm is summarized in Algorithm 1.

Algorithm 1 The basic ELM algorithm

```
1: Design a network with suitable number of hidden neurons
   and activation functions;
2: Initialize the hidden layer weights $\mathbf{W}$ and biases $\mathbf{b}$ randomly;
3: Compute the output weights $\boldsymbol{\beta}$ given training samples,
   using the pseudo-inverse.
```

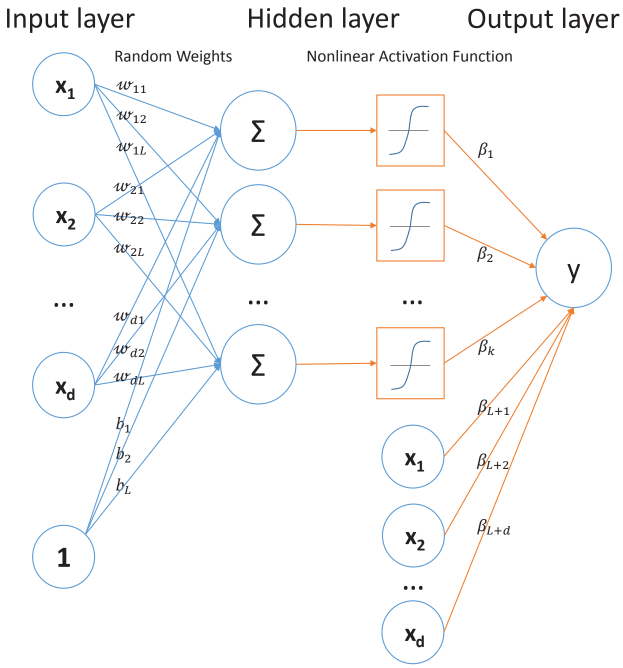


Fig. 1. The structure of the ELM used in this work

2) Adding regularization: Since the weights of the hidden layer are calculated based on the training data only, there is a risk that the model will overfit and performs worse on the test data than on the training data. To prevent this, so-called L_2 or Tikhonov regularization can be used, where a regularization parameter λ penalizes the weights in the output layer of the ELM [2]. The criterion to be minimized is now:

$$C = \sum_{j=1}^{N} (\sum_{i=1}^{L} \beta_i f(\mathbf{w}_i \mathbf{x}_j + b_i) - y_j)^2 + \lambda \sum_{i=1}^{L} \beta_i^2$$

For given λ, the least-squares solution is now given by $\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{Y}$. By properly adjusting λ, overfitting can be prevented [13].

3) Adding linear components: In addition to regularization, adding a linear component to the ELM can be helpful to model any linear relation between $\mathbf{X}$ and $\mathbf{Y}$ [13]. The final structure of the ELM used in this work is shown in Figure 1.

B. Efficient LOO Computation and Model Structure Selection

As already mentioned above, in the process of training a model there is a risk of the model overfitting to the training data, resulting in bad performance on unseen test data. A common way to prevent overfitting is to do model structure selection using k-fold cross-validation, where the training data is divided into k folds and the model is tested on each of those k folds, while being trained on the remaining $k-1$ folds. A special case of k-fold cross-validation is leave-one-out (LOO) cross-validation, where k is equal to N, the number of samples in the training set. Hence, in each fold, only one sample is excluded from the training data set and is used for validation. Let ϵ_{-i} denote the validation error on fold i, then the leave-one-out error can be computed as

$$\epsilon_{LOO} = \frac{1}{n}\sum(\epsilon_{-i})^2.$$

This criterion is then used to optimize the structure and hyper-parameters of the model. Generally speaking, a model would need to be trained $k = N$ times. However, for linear models there is a closed-form solution for ϵ_{LOO}, known as the "predicted residual error sum of squares" (PRESS) statistics [9]:

$$\begin{aligned}\epsilon_{LOO} &= \frac{1}{N}\sum(\epsilon_{-i})^2 \\ &= \frac{1}{N}\sum(y_i - \boldsymbol{\beta}^{-i}f(x_i))^2 \\ &= \frac{1}{N}\sum(\frac{\epsilon_i}{1-\mathbf{G}_{ii}})^2\end{aligned}$$

where $\boldsymbol{\beta}^{-i}$ denotes the weights of the model trained on all except the i^{th} sample and $\mathbf{G}_{ii}$ denotes the i^{th} element in the diagonal of $\mathbf{G} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, which is known as the hat matrix [9]. Recall that in the training of ELM, the pseudo-inverse $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is computed. Therefore, computing the diagonal of the $\mathbf{G}$ matrix, and thus ϵ_{LOO}, comes at very little computational overhead. Using this efficient leave-one-out cross-validation, candidate models using e.g. different inputs, numbers of hidden neurons, or regularization parameters can quickly be evaluated and therefore, model structure selection can be efficiently performed. For more details, see [13], [17].

C. Feature Selection and Dimensionality Reduction

In this work, the future values of air quality are predicted based on past hourly records. Since the number of potential inputs can be very large and may contain irrelevant or noisy variables, an important question is which and what kind of variables should be used as the inputs for the model. One possibility would be to perform feature selection, and use a subset of the original variables. Another possibility is to construct a set of new variables, given the original variables.

1) Feature selection: The target of feature selection is to determine which variables should be included in the model. In this work, a wrapper method known as "forward selection" is used [5]. Starting from an empty variable subset, in each iteration it finds and adds that variable which results in the best accuracy along with the variables selected so far. The steps of performing forward selection for a particular ELM are summarized in Algorithm 2. The main advantage of forward selection is that it is fast, with a time complexity of $O(n \cdot D)$, where n is the number of features to be selected and D the total number of candidate features. At the same time, it is easy to understand and implement. However, it does not guarantee the globally optimal solution, as not all the feature combinations in the search space are evaluated.

Algorithm 2 ELM-based forward selection

```
1: Initialize the candidate variable set with all the available variables and the selected variable set as empty.
2: Find the variable $x_j$ in the candidate variable set, combined with all variables in the selected variable set, which leads the smallest LOO error with ELM;
3: Add $x_j$ into selected variable set and remove it from the candidate variable set;
4: Stop when the required number of features are included in the selected variable set, otherwise go to step 2.
```

2) Dimensionality reduction: In dimensionality reduction, the goal is to replace the original variables with a smaller set of new features, without significant loss of the relevant original information. The new features are derived using e.g. principal component analysis (PCA), Fourier transform (FT) or Wavelet analysis [5]. In this work, PCA is used for feature extraction and reducing the dimensionality of the inputs. The focus of this study is on whether or not a set of principle components can be used to replace the station-wise records without loss of accuracy. Thus, the above feature selection procedure is carried out on all principle components, and the number of candidate components is the same as the amount of stations for each type of meteorological records.

III. Experimental Arrangements

A. Data set

1) Air quality records: Thanks to the Finnish Meteorological Institute, there are hourly air quality records published online [1]. The data set used in this study contains hourly records of NO, O_3, PM_{10} and $PM_{2.5}$, from 2013/01/01 to 2014/12/31. There are altogether 25 stations available around the Helsinki Metropolitan Area recording the related data, which can be seen in Figure 2.

2) Meteorological records: In addition to the air quality records, hourly meteorological records are available for the same region from NOAA's National Centers for Environmental Information (NCEI) [10]. This includes basic weather records, such as relative humidity, pressure, temperature, and wind.

3) Missing data imputation and normalization: Since both air quality and meteorological data sets contain missing values, missing value imputation needs to be performed. Stations with more than 40% of data points missing are removed, while for the remaining stations missing values are imputed as summarized in Figure 3. Once the missing values are imputed, the data is normalized to have zero mean and unit variance. The final amount of stations and samples is summarized in Table I. The last 3000 data points are set aside as a test set, while the remainder is used for training and cross-validation.

B. Model Details

Since the performance of ELM is affected by its parameter settings, leave-one-out cross-validation studies were performed to determine the most suitable parameters. These are summarized below.

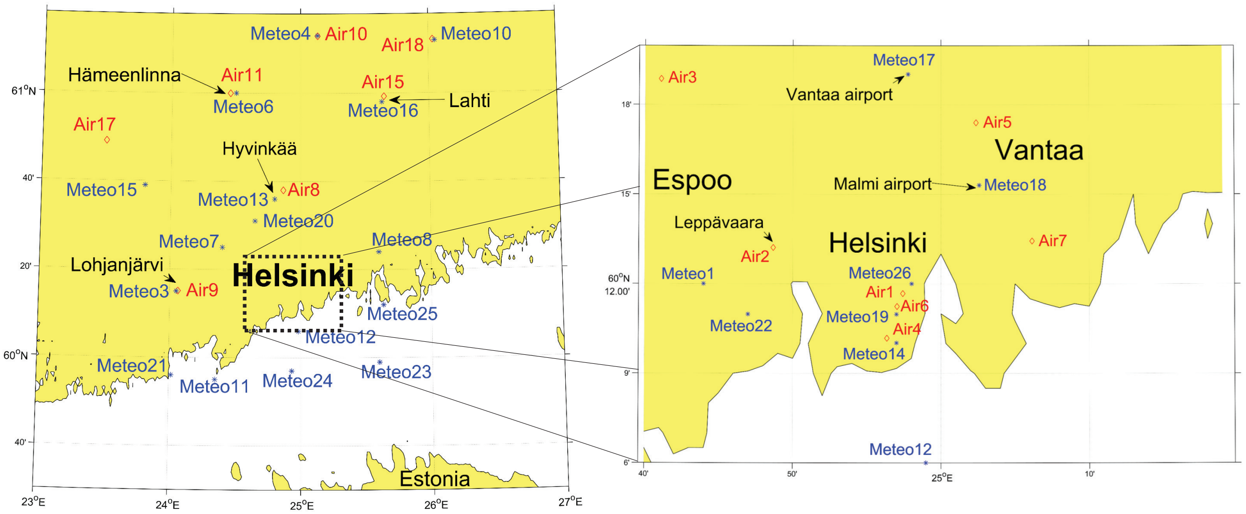
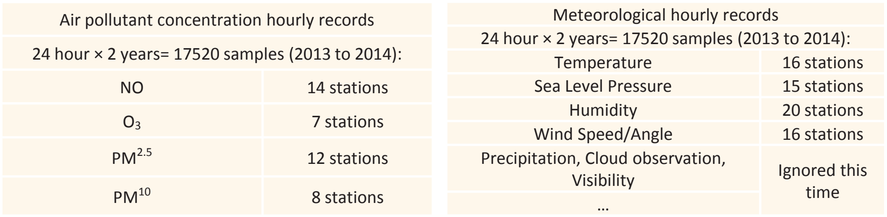


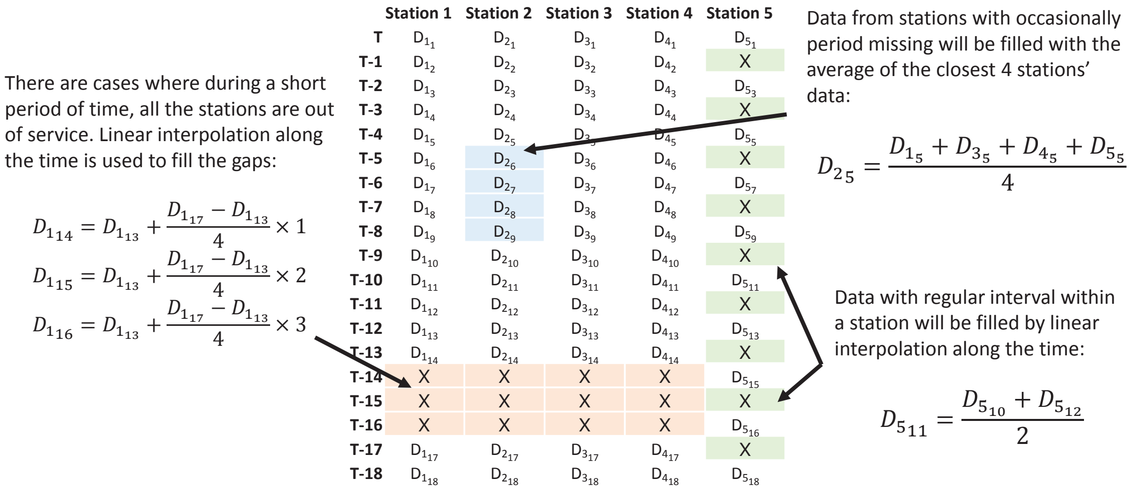
Fig. 2. Distribution of air quality stations and meteorological stations.



Air pollutant concentration hourly records	
24 hour × 2 years= 17520 samples (2013 to 2014):	
NO	14 stations
O_3	7 stations
$PM^{2.5}$	12 stations
PM^{10}	8 stations

Meteorological hourly records	
24 hour × 2 years= 17520 samples (2013 to 2014):	
Temperature	16 stations
Sea Level Pressure	15 stations
Humidity	20 stations
Wind Speed/Angle	16 stations
Precipitation, Cloud observation, Visibility ...	Ignored this time

TABLE I
DISTRIBUTION OF AIR QUALITY STATIONS AND METEOROLOGICAL STATIONS.



	Station 1	Station 2	Station 3	Station 4	Station 5
T	D_{1_1}	D_{2_1}	D_{3_1}	D_{4_1}	D_{5_1}
T-1	D_{1_2}	D_{2_2}	D_{3_2}	D_{4_2}	X
T-2	D_{1_3}	D_{2_3}	D_{3_3}	D_{4_3}	D_{5_3}
T-3	D_{1_4}	D_{2_4}	D_{3_4}	D_{4_4}	X
T-4	D_{1_5}	D_{2_5}	D_{3_5}	D_{4_5}	D_{5_5}
T-5	D_{1_6}	D_{2_6}	D_{3_6}	D_{4_6}	X
T-6	D_{1_7}	D_{2_7}	D_{3_7}	D_{4_7}	D_{5_7}
T-7	D_{1_8}	D_{2_8}	D_{3_8}	D_{4_8}	X
T-8	D_{1_9}	D_{2_9}	D_{3_9}	D_{4_9}	D_{5_9}
T-9	$D_{1_{10}}$	$D_{2_{10}}$	$D_{3_{10}}$	$D_{4_{10}}$	X
T-10	$D_{1_{11}}$	$D_{2_{11}}$	$D_{3_{11}}$	$D_{4_{11}}$	$D_{5_{11}}$
T-11	$D_{1_{12}}$	$D_{2_{12}}$	$D_{3_{12}}$	$D_{4_{12}}$	X
T-12	$D_{1_{13}}$	$D_{2_{13}}$	$D_{3_{13}}$	$D_{4_{13}}$	$D_{5_{13}}$
T-13	$D_{1_{14}}$	$D_{2_{14}}$	$D_{3_{14}}$	$D_{4_{14}}$	X
T-14	X	X	X	X	$D_{5_{15}}$
T-15	X	X	X	X	X
T-16	X	X	X	X	$D_{5_{16}}$
T-17	$D_{1_{17}}$	$D_{2_{17}}$	$D_{3_{17}}$	$D_{4_{17}}$	X
T-18	$D_{1_{18}}$	$D_{2_{18}}$	$D_{3_{18}}$	$D_{4_{18}}$	$D_{5_{18}}$

There are cases where during a short period of time, all the stations are out of service. Linear interpolation along the time is used to fill the gaps:

$$D_{1_{14}} = D_{1_{13}} + \frac{D_{1_{17}} - D_{1_{13}}}{4} \times 1$$

$$D_{1_{15}} = D_{1_{13}} + \frac{D_{1_{17}} - D_{1_{13}}}{4} \times 2$$

$$D_{1_{16}} = D_{1_{13}} + \frac{D_{1_{17}} - D_{1_{13}}}{4} \times 3$$

Data from stations with occasionally period missing will be filled with the average of the closest 4 stations' data:

$$D_{2_5} = \frac{D_{1_5} + D_{3_5} + D_{4_5} + D_{5_5}}{4}$$

Data with regular interval within a station will be filled by linear interpolation along the time:

$$D_{5_{11}} = \frac{D_{5_{10}} + D_{5_{12}}}{2}$$

Fig. 3. Treatment of missing values in various conditions. When data is missing at regular intervals, linear interpolation along time is used based on the assumption that the values change smoothly over time. However, if contiguous records are missing for a single station, linear interpolation based on its neighboring stations is used, in the hope that fluctuations can be recovered.

1) Hidden layer weights and biases: Although in ELM the hidden layer weights and biases are initialized randomly, their scale greatly influences the results. Based on the parameter studies, biases and weights were drawn from a uniform distribution between [-1.5, 1.5] and [-1.0, 1.0], respectively.

2) Number of hidden neurons: With proper regularization, more hidden neurons provide generally better results. However, computational load may increase significantly and there are diminishing returns. As a suitable trade-off between accuracy and speed, the maximum number of hidden neurons is set to be 400, while a smaller amount of 50 hidden neurons is used in the first stage of feature selection. This should not affect the results too much, while significantly reducing time for the feature selection process.

3) Regularization parameter: The results of the initial experiments indicate that when the regularization parameter λ is set properly, ELM achieves remarkably lower error than its non-regularized version. Since the optimal value of λ depends on the number of input features and number of hidden neurons, λ is optimized for each number of input features and number of hidden neurons during the model selection and training.

C. Effect of Augmenting Air Quality with Meteorological Data

1) Candidate variable sets: Predictions for the air quality in the next hour are based on the most recent records of the air quality and meteorological data:

- For the air quality data, the 2000 most recent hourly measurements are considered as candidate variables.
- For meteorological data, the 30 most recent hourly measurements of each station are considered for each type.

As can be seen, the number of potential input variables can quickly get out of hand. Therefore, feature selection needs to be performed on both the air quality time series data and meteorological records in order to reduce the number of input features, and to select the features most relevant for the prediction.

2) ELM-based forward selection: To decrease the time required by the feature selection process, a two-stage ELM-based forward selection is used:

1) At the first stage, the top 50 variables of each type of data are selected using a relatively simple ELM (hidden layer size is 50). These are represented by $\mathbf{x}_{air}$ for air quality, $\mathbf{x}_h$ for relative humidity, $\mathbf{x}_{dt}$ for dew point temperature, $\mathbf{x}_p$ for pressure and $\mathbf{x}_t$ for temperature.
2) At the second stage, feature selection is done using a more complex ELM (hidden layer size is 400) on the combined candidate feature sets formed by $[\mathbf{x}_{air}]$, $[\mathbf{x}_{air}, \mathbf{x}_h]$, $[\mathbf{x}_{air}, \mathbf{x}_{dt}]$, $[\mathbf{x}_{air}, \mathbf{x}_p]$, $[\mathbf{x}_{air}, \mathbf{x}_t]$.

In this paper, all 50 or 100 variables are eventually selected in the second stage and therefore more of a ranking than a selection is achieved. This experiment should reveal if certain types of meteorological data can improve the air quality prediction. In practice, of course, the practitioner can prematurely stop selecting additional variables, once a desired leave-one-out accuracy is achieved on the available training data. The entire procedure is summarized in Figure 4.

	global model or local model	
raw meteo	global + raw	local + raw
pca meteo	global + pca	local + pca

TABLE II
FOUR DIFFERENT APPROACHES FOR SPATIAL DATA FUSION.

D. Effect of Different Approaches to Spatial Data Fusion

Aside from augmenting the data with meteorological measurements and selecting the most relevant variables in time, the spatial aspect of the problem can be incorporated in various ways. Therefore, the proposed model is evaluated using several approaches to spatial data fusion:

- Either a single model is trained for all stations (global model), or each station has its own model (local model). Global models might be able to exploit having larger amount of data, while local models might be able to better model local dependencies.
- Either meteorological data is used as-is, that is, station-wise normalized data, or meteorological data from multiple stations is ”summarized” by considering principal components of multiple meteorological stations instead.

The resulting four different approaches are summarized in Table II. Other settings such as hyper-parameters for the ELM remain the same.

IV. RESULTS

Global Model: raw meteo vs. pca meteo

Figure 5 summarizes the results of the global model approach, where one model is trained to predict the air quality at all stations. The rows show the training and test accuracy respectively, while the columns show the effect of augmenting the air quality data with various meteorological information. From the results, it can be seen that especially relative humidity and temperature can improve the prediction accuracy, while dewpoint temperature and air pressure do not help much. With respect to using station-wise data or PCA-wise data, it can be seen that there is no significant difference between these approaches in terms of accuracy. Therefore, a limited number of principal components could be used as candidate input variables, speeding up the whole approach.

Local Model: raw meteo vs. pca meteo

Figure 6 summarizes the results of the local model approach, where the air quality for a station is predicted with a station-specific ELM, using historical air quality records from only that station and the meteorological data from the whole region. Similar to the results from the global model, prediction accuracy is improved by considering relative humidity and temperature records. An important difference is that for the local model, using PCA-wise data hurts accuracy and it is better to use station-wise data instead. Presumably, by summarizing the meteorological stations using PCA, spatial information useful for local prediction gets lost.

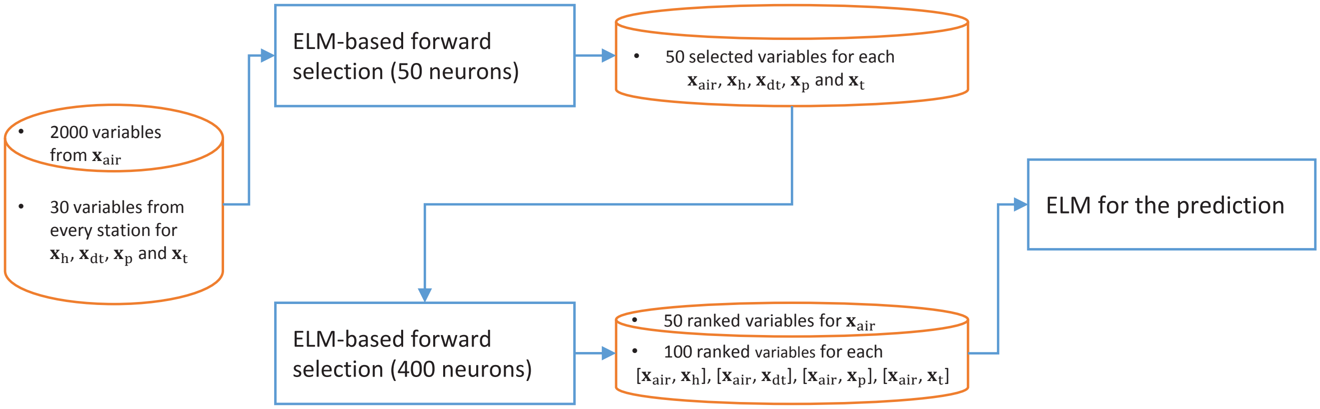

• 2000 variables from $\mathbf{x}_{\mathrm{air}}$
• 30 variables from every station for $\mathbf{x}_{\mathrm{h}}$, $\mathbf{x}_{\mathrm{dt}}$, $\mathbf{x}_{\mathrm{p}}$ and $\mathbf{x}_{\mathrm{t}}$
ELM-based forward selection (50 neurons)
• 50 selected variables for each $\mathbf{x}_{\mathrm{air}}$, $\mathbf{x}_{\mathrm{h}}$, $\mathbf{x}_{\mathrm{dt}}$, $\mathbf{x}_{\mathrm{p}}$ and $\mathbf{x}_{\mathrm{t}}$
ELM-based forward selection (400 neurons)
• 50 ranked variables for $\mathbf{x}_{\mathrm{air}}$
• 100 ranked variables for each $[\mathbf{x}_{\mathrm{air}}, \mathbf{x}_{\mathrm{h}}]$, $[\mathbf{x}_{\mathrm{air}}, \mathbf{x}_{\mathrm{dt}}]$, $[\mathbf{x}_{\mathrm{air}}, \mathbf{x}_{\mathrm{p}}]$, $[\mathbf{x}_{\mathrm{air}}, \mathbf{x}_{\mathrm{t}}]$
ELM for the prediction

Fig. 4. ELM-based feature selection

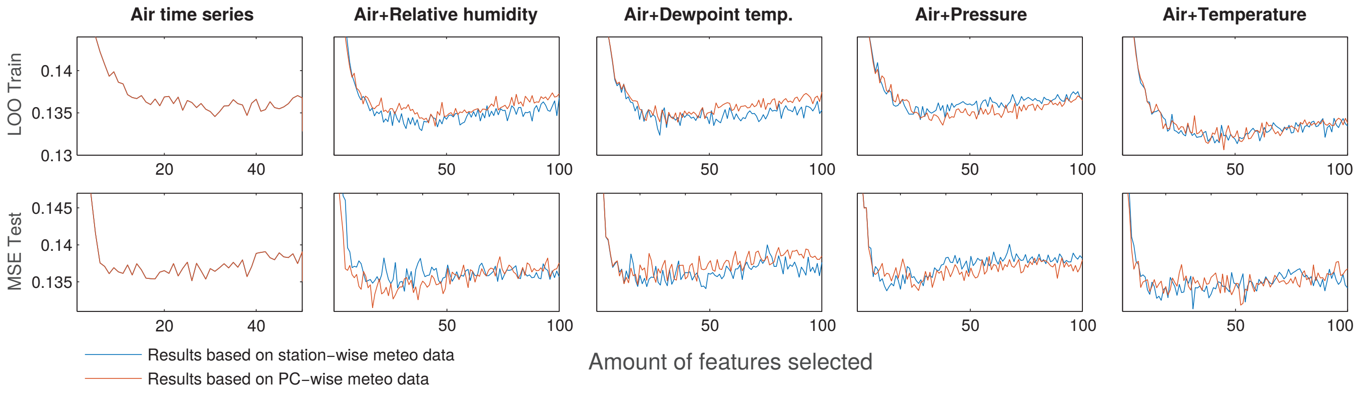


Fig. 5. Results comparison between raw meteo and pca meteo for global model for prediction of NO.

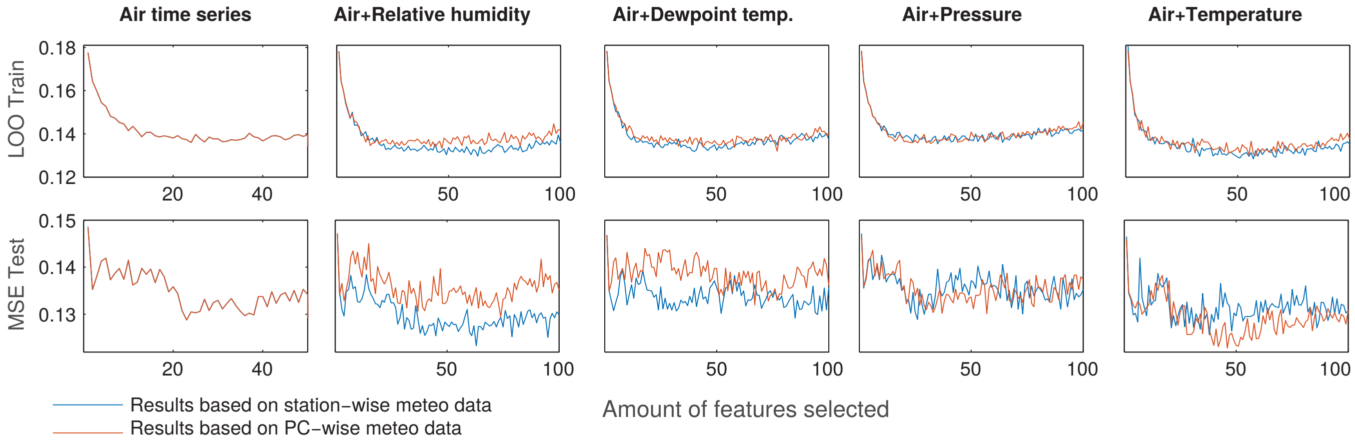


Fig. 6. Results comparison between raw meteo and pca meteo for local model for prediction of NO.

V. Conclusions

In this work, a neural network based approach was proposed for air quality prediction. Special attention was given to the feature selection, and it was investigated whether accuracy can be improved by considering additional sources of information, like meteorological data. The proposed approach was consequently evaluated with different approaches for performing spatial data fusion. Based on the results, a number of conclusions can be drawn:

1) Feature selection is effective: Feature selection is able to identify relevant variables for the prediction and greatly reduces the number of variables used as input in the model, while attaining high accuracy.

2) ELM enables the possibility of real-time processing: With the help of simple structure and fast training algorithm of ELM, combined with feature selection, it is possible to have real-time processing of large meteorological data sets, where they are hundreds or thousands of stations with hourly (or even higher-frequency) measurements.

3) Summarizing station data helps in global models, but not in local models: Depending on whether a global or local model is used, summarizing the meteorological data using PCA can help reduce the search space for feature selection: when training a global model, the station-wise data can be replaced by its top principal components. Therefore, in future studies, a limited amount of principal components can be considered to reduce model complexity. However, when training a local model, this approach discards useful spatial information and using station-wise data results in better accuracy.

In future work, multiple ELMs might be ensembled to further improve accuracy. Furthermore, since the dispersion of the pollutant is a spatial-temporal process, the accuracy of prediction might be improved if also the air quality records of multiple stations are incorporated. However, the increased complexity of the model and number of variables could bring more challenges. Including more stations would increase the size of the input layer of the ELM and would significantly increase the amount of calculations in the feature selection stage. Therefore, to reduce the processing time, the parallelization of those algorithms, such as the ELM-based feature selection would be interesting to explore [14].

In summary, the proposed approach provides a flexible and reliable method for air quality prediction. It provides accurate predictions while being much faster than to train than other types of models, making it suitable for such meteorological problems where massive amount of data is generated every day.

References

[1] Air Quality Database, http://www.ilmanlaatu.fi/.

[2] W. Deng, Q. Zheng, and L. Chen. Regularized Extreme Learning Machine. In *IEEE Symposium on Comp. Intelligence and Data Mining*, pages 389–395, 2009.

[3] X. Feng, Q. Li, Y. Zhu, J. Hou, L. Jin, and J. Wang. Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation. *Atmospheric Environment*, 107:118–128, 2015.

[4] M. W. Gardner and S. R. Dorling. Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London. *Atmospheric Environment*, 33(5):709–719, 1999.

[5] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[6] G. Huang, G.-B. Huang, S. Song, and K. You. Trends in extreme learning machines: A review. *Neural Networks*, 61:32–48, 1 2015.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 12 2006.

[8] M. Kolehmainen, H. Martikainen, and J. Ruuskanen. Neural networks and periodic components used in air quality forecasting. *Atmospheric Environment*, 35(5):815–825, 2001.

[9] R. H. Myers. *Classical and Modern Regression with Applications, 2nd edition*. Duxbury, Pacific Grove, CA, USA, 1990.

[10] NOAA's National Centers for Environmental Information (NCEI). Hourly/Sub-Hourly Observational Data, http://www.ncdc.noaa.gov/.

[11] C. A. Pope, R. T. Burnett, G. D. Thurston, M. J. Thun, E. E. Calle, and D. Krewski. Cardiovascular Mortality and Long-Term Exposure to Particulate Air Pollution: Epidemiological Evidence of General Pathophysiological Pathways of Disease. *Circulation*, 109(1):71–77, 2003.

[12] J. B. Ruidavets, M. Cournot, S. Cassadou, M. Giroux, M. Meybeck, and J. Ferrieres. Ozone air pollution is associated with acute myocardial infarction. *Circulation*, 111(5):563–569, 2005.

[13] M. van Heeswijk. *Advances in Extreme Learning Machines*. PhD thesis, Aalto University, 2015.

[14] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse. GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing*, 74(16):2430–2437, 9 2011.

[15] C.-M. Vong, W.-F. Ip, P.-K. Wong, and C.-C. Chiu. Predicting minority class for suspended particulate matters level by extreme learning machine. *Neurocomputing*, 128:136–144, 3 2014.

[16] D. Voukantsis, K. Karatzas, J. Kukkonen, T. Räsänen, A. Karppinen, and M. Kolehmainen. Intercomparison of air quality data using principal component analysis, and forecasting of PM10 and PM2.5 concentrations using artificial neural networks, in Thessaloniki and Helsinki. *The Science of the total environment*, 409(7):1266–1276, 2011.

[17] C. Zhao. Air quality forecasting using neural networks. M.Sc. thesis, 2016.