Air quality forecasting using neural networks

Chen Zhao

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.5.2016

Thesis supervisor:

Prof. Juha Karhunen

Thesis advisor:

D.Sc. Mark van Heeswijk

Aalto University
School of Science

AALTO UNIVERSITY SCHOOL OF SCIENCE	ABSTRACT OF THE MASTER'S THESIS

Author: Chen Zhao

Title: Air quality forecasting using neural networks

Date: 30.5.2016 Language: English Number of pages: 6+48

Department of Computer Science

Supervisor: Prof. Juha Karhunen

Advisor: D.Sc. Mark van Heeswijk

In this thesis project, a special type of neural network: Extreme Learning Machine (ELM) is implemented to predict the air quality based on the air quality time series itself and the external meteorological records. A regularized version of ELM with linear components is chosen to be the main model for prediction. To take full advantage of this model, its hyper-parameters are studied and optimized. Then a set of variables is selected (or constructed) to maximize the performance of ELM, where two different variable selection methods (i.e. wrapper and filtering methods) are evaluated. The wrapper method ELM-based forward selection is chosen for the variable selection. Meanwhile, a feature extraction method (Principal Component Analysis) is implemented in the hope of reducing the candidate meteorological variables for feature selection, which proves to be helpful. At last, with all the parameters being properly optimized, ELM is used for the prediction and generates satisfying results.

Keywords: ELM, neural network, time series prediction, meteorological data period analysis

Preface

This master thesis is made in (and also supported by) the Applications of Machine Learning (AML) research group under the Department of Computer Science at the Aalto University School of Science.

It is a great honor for me to have the opportunity working as a research assistant in the AML group. I was always admiring those senior students with an office in the department. Becoming one of them used to be my dream, even since the first "machine learning: basic principles" class five years ago, when I was just a bachelor exchange student. I cannot tell the excitement as I am now typing the preface of the master thesis, right in the office A344 with a good view of the building and the closest distance to the coffee room.

I would express my sincere gratitude to Professor Juha Karhunen, who provided me the chance to live through the life as a researcher. He took care of everything so my concentration could stay on the thesis. His passion towards research and professional comments on the work will be my lifetime treasure.

I would appreciate my instructor and friend Mark van Heeswijk. I cannot finish (or even start) the thesis without his delicate guidance. Every minute we spent together in front of the white boards, monitors and draft papers, is the best time in the age of 26.

I would thank my roommate Alexander Grigorievskiy for his advices and the beneficial talks and Emil Eirola for his kindly suggestions on dealing with missing data. Many thanks to Luiza Sayfullina and the other colleagues in the group and the department for the great time we spent together.

Special appreciation to Prof. Matti Hämäläinen who organized my exchange affairs and also to all the teachers who taught and helped me during my study at Aalto University.

Love and thanks to my parents for their taking care during of my growing up and the extend thanks to all my family relatives and friends for their support in the past 26 years.

Finally, I am glad to be able to "master" something. I wish I can serve the world with such little knowledge and make it a better place.

It must be a better place: Summer is Coming.

Otaniemi, 30.5.2016

Chen Zhao

Contents

Symbols and abbreviations

Symbols

x or x_i	one sample in vector form
y or y_i	one target value
d	dimensionality of the samples
n	amount of samples
sub	a subset of the original variables
β	hidden layer weights
b	bias
k	the number of hidden neurons in ELM
θ	parameters of a model
var or var_i	a variable of x
x_{-i},y_{-i}	a subset of sample with the (x_i,y_i) excluded
σ_x	the variance of the x

Operators

$arg\,min$	the arguments making the following expression minimized
$E[x]$	the expectation of x
$cov(x, y)$	the covariance of x, y
$O(x)$	the big O notation

Abbreviations

ELM	Extreme Learning Machine
NN	Neural Network
MLP	Multilayer Perceptron
PCA	Principal Components Analysis
PCs	Principal Components
LARS	Least Angle Regression
LOO	Leave-One-Out

1 Introduction

1.1 Air condition prediction

It is a truth universally acknowledged, that a man in possession of a good fortune must want healthy living environment. Air quality is one of the most important parts defining the environment and has a close relationship with human health [54, 5, 43, 7]. In practice, many of the studies have been conducted to discover the relationship between the pollutant variables and health problems, especially for people living in the urban area [12, 44, 51]. Such variables include the nitrogen oxide (NO_x), sulfur dioxide (SO_2), ozone (O_3) and small particulates (PM).

For instance, Burnett found that NO_2 is a major factor for hospital admissions in terms of gaseous pollutants [5]. For these cardiovascular causes of death, a $10-\mu g/m^3$ elevation in the concentration of particulate matter (PM) was associated with 8% to 18% increase in mortality risk [43]. A two-year study shows that short-term O_3 exposure could bring about acute coronary events [45].

The solution for the improvement of air conditions involves great and long time effort from the whole society, however, some immediate methods have been suggested [56] to help the public to prevent themselves from getting harmed by the bad air conditions during the certain periods of a day.

Such suggestions, however, introduced a new issue: to predict the air condition so the public could arrange their outdoor activities accordingly, in respect of both time and location. It has been referred as deterministic approaches [18], where the trajectory of air mass or chemical materials would be calculated for the predictions. Many of them are taking into consideration both the evolution of the air concentration time series itself and more important the movement of the atmosphere. Just as summarized by Zhang [64], one trend of the forecasting is to take advantage of computational fluid dynamic models on a smaller scale (1 km or less) with the help of powerful computers. Several models have been proposed for this air quality forecasting purpose, such as the PREV'AIR [25] from France, BOLCHEM [41] for Ozone concentration in Italy and EURAD-IM [17] for NO concentration in mid and west Europe.

On the other hand, due to the complexity of air dynamic simulations, it might take huge resources for the calculation, making the real-time forecasting difficult and hard for the public to access. At the same time, instead of deterministic methods, researchers also tried the neural networks for the prediction. In most cases, meteorological data is considered as it could improve the accuracy for the air quality forecasting [9]. Gardner [19] uses the multilayer perceptron network (MLP) to model the NO_x in London based on hourly meteorological data and found that it could capture the complex patterns of source emissions without any external guidance. Kolehmainen, based on experiment on hourly time series of NO_2 in the city of Stockholm [37], states that using MLP directly on the original NO_2 records outperforms the models using periodic regression methods. Corani [8] compared pruned neural networks (PNNs) and lazy learning (LL) on daily Ozone and PM_{10} forecasting in Milan. He pointed out the PNN is better on prediction for the exceedances of the air quality standards. Voukantsis [61] applied neural networks to predict the particulate matter in both Helsinki and Thessaloniki, where the accuracy of the model is identical regardless of the quite different weather conditions of the two sites. He also got the models optimized by adding a variable selection process coupled with principal component analysis (PCA), and found only using the top principal components can explain the air condition records nicely. Zheng [65] uses the neural network to extract spatial features and the linear chain conditional random field (CRF) to model the temporal characteristic of air quality. The model shows advantage to the decisions tree and CRF/ANN alone.

Despite the well-developed traditional MLP neural networks, there is another kind of feedforward neural network showing its success in the past decade, the Extreme Learning Machine (ELM) [30, 31]. It is a single hidden layer feedforward neural network with the weights of the hidden layer being randomly generated. It could be trained much faster than the neural network using traditional learning algorithms, such as backpropagation (BP). Unlike the well-developed MLP neural networks, there are very few papers mentioning the application of Extreme Learning machine (ELM) in meteorology. Considering the similar time series prediction problems, van Heeswijk [58] proposed the adaptive ensemble models of ELM, which outperforms the support vector machine (SVM) model. Yan [63] used ELM ensemble to predict the short term load of electricity in Australia, where the model beats the backpropagation neural network and radial basis function the neural network in both prediction accuracy and training efficiency. Deo [11] has applied this algorithm for the prediction of the Effective Drought Index in Australia, using meteorological records as inputs, and found ELM performs better than the backpropagation neural network. As one of the very few applications of ELM on air quality prediction, Vong [60] carried out an experiment on predicting the daily average level (3 classes) of PM_{10} in Macau, where ELM is superior over SVM in this case.

In this project, a version of the Regularized Extreme Learning Machine [10] is implemented to predict the hourly concentrations of various air pollutants, on the basis of the time series of the hourly records of air pollutants and some basic type of meteorological data, from multiple stations within a relatively small area. Compared with the experiments mentioned above, the use of both spatial and temporal time series records with such shorter interval (hourly) makes it unique among related research.

1.2 Data set description

1.2.1 Air quality records

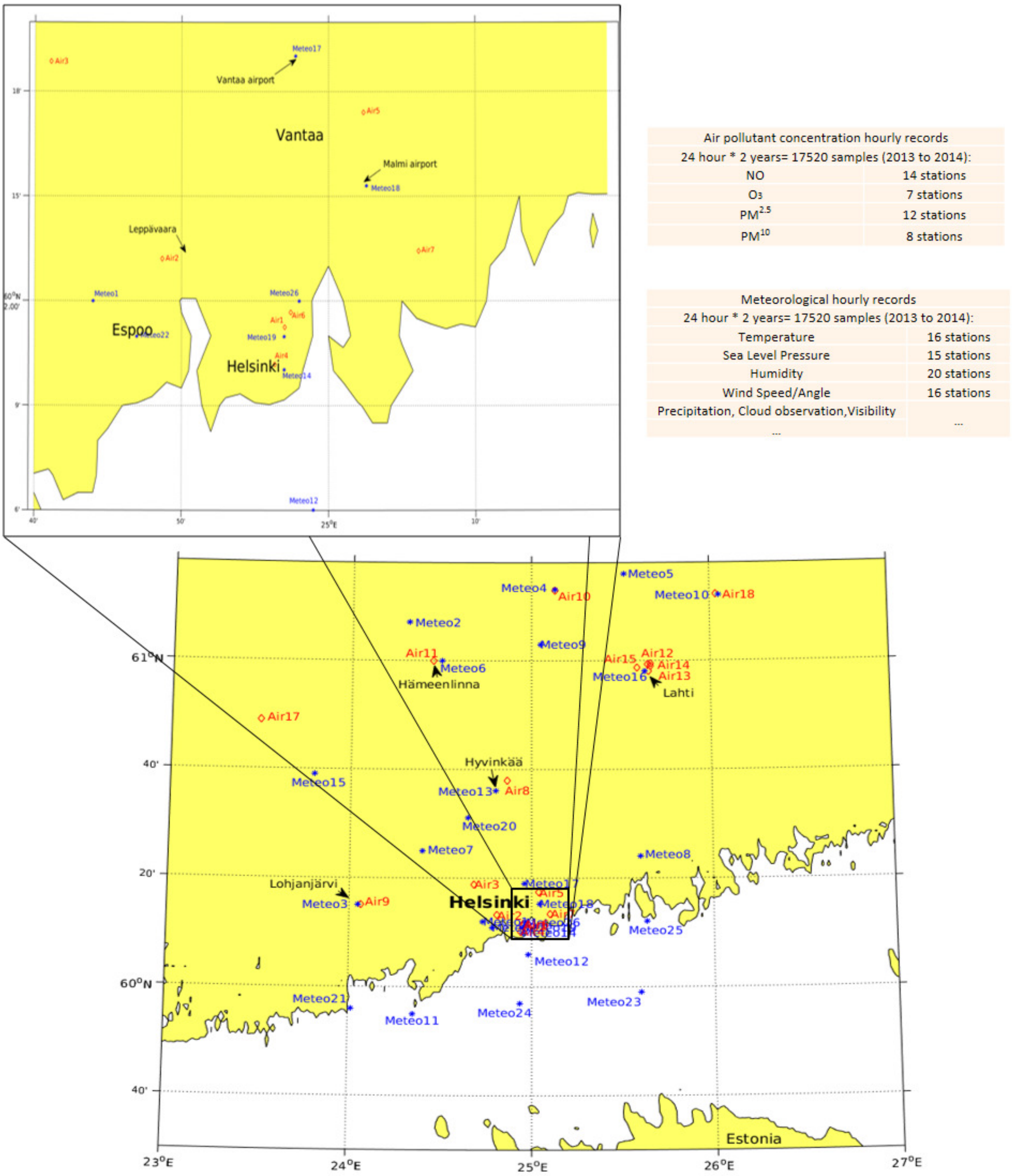
Thanks to the Finnish Meteorological Institute, there are data records about air pollution components published on-line [13]. The data set used in this study contain the hourly records of NO, O_3, PM_{10} and $PM_{2.5}$, from 01.01.2013 to 30.12.2014. There are altogether 25 stations available around the Helsinki Metropolitan Area[1] recording the related data, however, there are very few stations with all the types of pollutants recorded. After the data preprocessing step, which will be explained in detail later, 14, 12, 8 and 7 stations have been finally taken into consideration for pollutant types NO, PM_{10}, $PM_{2.5}$ and O_3 respectively.[2]

1.2.2 Meteorological records

In addition to the air component concentrations, certain types of meteorological records are also involved in this project. The data set contains some basic weather records, such as wind, visibility, temperature, pressure and relative humidity from the monitor stations around Helsinki. They are hourly data downloaded from NOAA's National Centers for Environmental Information (NCEI) [42]. After removing some stations with a huge proportion of missing values, the sample size introduced in this project is listed in Figure 1.

[1] A region with urban kernel (including Vantaa, Espoo, Kauniainen) and commuter towns (Hyvinkää, Kirkkonummi and etc. surrounding Helsinki)

[2] For sulfur dioxide and carbon monoxide CO concentrations only a few stations (less than 4) have sufficient records. As one of the targets of the project is to reveal the potential spatial and temporal patterns among a sufficient amount of stations, SO_2 and CO are excluded from this study.



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 ...	...

Figure 1: Distribution of meteorological stations and air quality records stations.

2 Neural networks

History of neural networks

The study of the brain of human beings has never stopped. About one century ago, William James published his “The Principles of Psychology” [33], meanwhile the researchers worked hard to reveal the secret of the brain. Thanks to the concept of the McCulloch-Pitts model, where the two authors state that the activity of neurons could be represented by means of propositional logic, it provided the possibility of creation of artificial neural networks with electronic machines [38]. A few years later, Donala O. Hebb proposed the idea that the connections between neurons could be reinforced, thus provide the neural network have the ability to learn from stimulus under a given structure [23]. Frank Rosenblatt, brought about the concept of “Perceptron” in 1958, where he proposed how the information is sensed, remembered and retained in the brain. It greatly enlightened the forthcoming researchers. Two years later in 1960, Bernard Widrow and Marcian Hoff proposed a training algorithm taking advantage of minimizing the least mean square error, which is one of the foundations of the current artificial neural network development [62].

Then the book “Perceptrons” was published, where Minsky and Papert pointed out that a neural network without a hidden layer could only solve the linear problem [40]. Such conclusion, along with the shifting of focus to the development of Von Neumann structure computers, slowed down the research in the neural networks field for many years. During that time, “Adaptive Resonance Theory” was proposed, suggesting that if one neuron is activated, the nearby neurons would be suppressed at the same time [6]. Then Teuvo Kohonen brought about a model for associative memory and developed it into the self-organizing map, a type of unsupervised neural network that capable of classification [36].

With the development of the Hopfield neural network, which illustrated the convergence of the neural network under an energy function, the focus has been directed to this field again [26]. The Boltzmann machine was proposed by Hinton and Sejnowski to avoid stepping into a local minima of Hopfield network [1]. Then the error backpropagation algorithm, which is able to adjust the weights of the neurons from a network with multiple layers [46], provided the researchers with a powerful tool for their studies. During and after this time, various related theories and new types of neural networks have been proposed, such as the radial basis function neural network, the support vector machine and ELM and so on.

In 2006, Hinton proposed a fast greedy algorithm to train the network with many hidden layers, which contributed to a second Neural Network Renaissance [24, 48]. The applications of deep neural networks started to offer more promising results in the areas such as computer vision, artistic image processing [20] , natural language processing and artificial intelligence for challenging games [49].

2.1 Linear model

Before moving to discuss neural networks, the concept of the linear model is introduced as it is used in the latter part of the project.

A linear model is described as:

$$y = \beta x + b$$

where y is the target value for the prediction, x as the input variables from one sample, β as the weights for each variable in x and b the bias. In a time series prediction problem, x could be a combination of records in the past at $T - k_1$, $T - k_2$, ... , $T - k_t$ moment, represented by x_{T-k_1}, x_{T-k_2}, ... , x_{T-k_t}, The goal of the model is to find a suitable set of parameters β^* and b^* to make the predicted value at moment T: $\hat{y}_T = \beta^* x + b^*$ as close as possible to the real data y_T. The error is defined as the difference between the predicted value and real value:

$$\begin{aligned}\epsilon &= \hat{y} - y \\ &= (\beta x + b) - y\end{aligned}$$

There are many ways to evaluate the error, and one of the most common measurement is the mean squared error (MSE), defined as

$$\begin{aligned}\epsilon_{MSE} &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \\ &= \sum_{i=1}^{n} ((\beta x_i + b) - y_i)^2\end{aligned}$$

Thus, the solution of a linear model with can be defined as

$$\arg\min_{\beta,b}\sum_{i=1}^{n}((\beta x_i + b) - y_i)^2$$

In most cases, there are multiple samples available to derive the parameters. Denote $X = (x_1, x_2, ..., x_n)^T$ and $Y = (y_1, y_2, ..., y_n)^T$, where n is the number of samples. For convenience, the β and b are rewritten as β', where X is extended with an additional column of ones. The above solution could be presented in matrix form as

$$\arg\min_{\beta'=[\beta,b]}||X\beta' - Y||^2$$

There is a closed-form expression to solve the above problem, called least mean square error approximation. It is defined as follows [3]:

$$\begin{aligned} X\beta' &= Y \\ X^T X\beta' &= X^T Y \\ (X^T X)^{-1} X^T X\beta' &= (X^T X)^{-1} X^T Y \\ \beta' &= (X^T X)^{-1} X^T Y \end{aligned}$$

The part $(X^T X)^{-1} X^T$ in the above equation is called Moore–Penrose pseudoinverse of X, represented by X^+. The expression $X^+ = (X^T X)^{-1} X^T$ holds if $X^T X$ is invertible, when the rank of $X^T X$ equals its dimension, i.e. the number of sample n is larger than the number of variables d. There is another form $X^+ = X^T (X X^T)^{-1}$ for the case $n < d$. Since in this project $n \gg d$, only the former situation is considered.

The HAT matrix is introduced here, defined as $HAT = XX^+ = X(X^T X)^{-1} X^T$. The interpretation of the HAT matrix would be, it transforms the real value of Y into its predicted version $\hat{Y}$:

$$\hat{Y} = X\beta = X(X^T X)^{-1} X^T Y = HAT \cdot Y$$

In this project, the advantage of HAT matrix is taken to accelerate the training process significantly in the later section.

2.2 Feedforward neural network

Let n be the total number of training samples (instances) where each sample is represented by pair (x_j, y_j) where x_j is the jth sample with dimensionality d and y_j the observed output. Assume the weights for the hidden neurons are in vector form $W = (w_1, w_2, ..., w_k)$ and bias as $b = (b_1, b_2, ..., b_k)$. The network with k hidden neurons will have $\beta = (\beta_1, \beta_2, ..., \beta_k)$ as weights for output layer and $f(.)$ is the activation function. The mathematical expression of SLFN can be written as follows:

$$\hat{y}_j = \sum_{i=1}^{k} \beta_i f(w_i x_j + b_i) \;\; j \in [1, n]$$

In the case with only one hidden neuron ($k = 1$) and the activation function $f(.)$ is a linear function, for instance $f(x) = x$, the SLFN would be in a special form:

$$\hat{y}_j = w x_j + b + \epsilon \;\; j \in [1, n]$$

It is actually the linear regression between x and y, with ϵ as the error. The above equation could be solved with least mean square error approximation, where the solution $w = (X^T X)^{-1} X^T y$. In the following discussion, this linear model would be used as a baseline for performance comparison purpose.

When the activation function is non-linear, for instance to use a binary step function or a sigmoid function instead, the SLFN with at most k hidden neurons can learn k distinct observations with zero error [30]. In such case, the both weights w and β need to be adjusted carefully with various methods to ensure its approximation capability [32].

2.3 Extreme Learning Machine (ELM)

In general, ELM has been developed from single layer feedforward networks, where the input weights of the hidden neurons are randomly generated. With the same expression of SLFN, the ELM model to describe a certain set of observations (x, y) could be presented as:

$$\hat{y}_j = \sum_{i=1}^{k} \beta_i f(w_i x_j + b_i) \;\; j \in [1, n]$$

where $\hat{y}_j$ is the predicted value of y, and w is generated randomly and remain fixed all the time. Denote

$$H = \begin{bmatrix} f(w_1x_1+b_1) & f(w_2x_1+b_2) & ... & f(w_kx_1+b_k) \\ f(w_1x_2+b_1) & f(w_2x_2+b_2) & ... & f(w_kx_2+b_k) \\ ... & ... & ... & ... \\ f(w_1x_n+b_1) & f(w_2x_n+b_k) & ... & f(w_kx_n+b_k) \end{bmatrix}_{n*k}$$

and

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_k \end{bmatrix}_{k*1}$$

Then the above ELM model which interpreting the observation would be represented by:

$$\hat{Y} = H\beta$$

The only unknown variable in the above equation is the output layer weights β. The above formula with SLFN structure can be used to interpret the data set if β is properly calculated. Huang and etc. have proved the theorem on the universal approximation capabilities using SLFNs with random nodes [31, 29, 27, 28]. It indicates that if the number of hidden neurons is large enough, with β being derived from the ordinary least square algorithm to minimize $||H\beta - Y||$, the error of such single layer neural network with random fixed weights converges to 0.

Thus, combined with the content related to the least square error in Section 2.1, the algorithm for standard ELM is listed in Algorithm 1.

Algorithm 1 The basic ELM algorithm

1. Design an SLFN with a proper number of hidden neurons k and a suitable activation function;
2. Initialize the SLFN with random weights w and bias b for hidden neurons;
3. Compute the weights for the output layer β with ordinary least square algorithm with the given training samples.

Figure 2 shows the structure of a typical ELM.

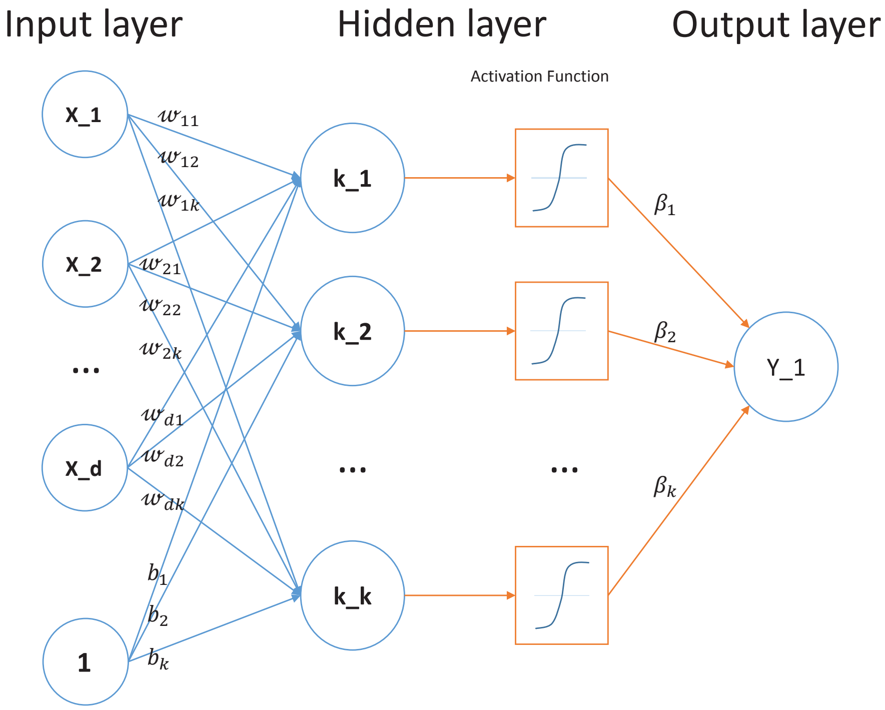


Figure 2: Structure of a typical ELM

2.4 Regularized ELM with linear components

Deriving the weights of the hidden layer involves the ordinary least square solution. However, it brought about two issues: 1) the solutions (weights) are calculated based on the training data only. It can get overfitted, where the model performs much worse on the test data than in the training. 2) the weights might be too sparse, with a relative large amount of neurons being assigned with small weights, making it difficult for model interpretation [39].

One solution proposed by Deng [10] introduces a regularization parameter, γ, which applies directly to the weights for the output layer:

$$min \sum_{j=1}^{n} (\sum_{i=1}^{k} \beta_i f(w_i x_j + b_i) - y_j)^2 + \gamma \sum_{i=1}^{k} \beta_i^2$$

The above method is also called L_2 or Tikhonov regularization. $\gamma \sum_{i=1}^{k} \beta_i^2$ acts as a penalty for the complexity of the model. With a given γ could be solved as well by the ordinary least square solution, where $\beta = (\gamma I + H^T H)^{-1} H^T Y$. By adjusting the value of γ, the regularized version of ELM can prevent itself from overfitting [39, 57].

Furthermore, adding a linear component in the ELM might be helpful in some cases, making the model as:

$$min \sum_{j=1}^{n}[(\sum_{i=1}^{k}\beta_i f(w_i x_j + b_i) + \sum_{i=1}^{d}\beta_i^{'} x_{ji}) - y_j]^2 + \gamma(\sum_{i=1}^{k}\beta_i^2 + \sum_{i=1}^{d}\beta_i^{'2})$$

where $\beta_i^{'}$ represents the output layer weights for the linear component and x_{ji} is the i^{th} variable in the j^{th} sample. It can help to overcome two issues: (1) the nonlinear activation function in the hidden layer makes it difficult to approximate the linear relationships in the data; (2) the performance of ELM influenced by the randomized weights for the hidden neuron layer. By adding the linear component in the model, it makes sure that the ELM always contains a linear approximation to the problem regardless of the initialization of the network and the approximation by the nonlinear part in ELM. Its structure is shown in Figure 3.

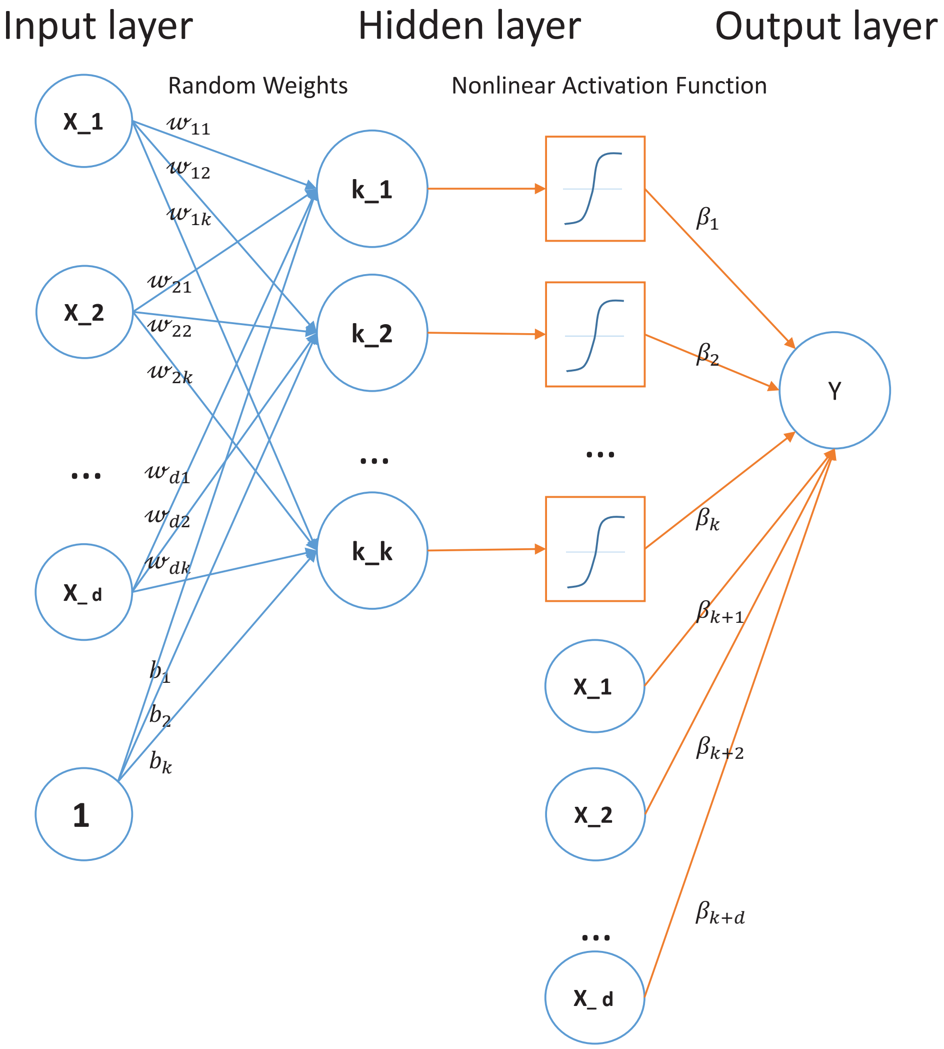


Figure 3: The structure of the ELM used in this project

3 Model structure selection

As the ELM is decided as the model in this project, it comes to its specifics. For instance, the suitable set of input variables (denoted by Sub, which is a subset of the original variables of x) and parameters (denoted by θ) for a better prediction. Let the true target value be y, the predicted value be $\hat{y} = f(x, \theta, Sub)$ where $f(.)$ denotes the ELM model, the training criteria of the model can be presented as:

$$\underset{\theta, Sub}{\arg\min} \sum_{i=1}^{n} (f(x_i, \theta, Sub) - y_i)^2$$

where n is the number of samples. It is the model parameter set θ and variable selection Sub that need to be derived in the training stage.

3.1 Cross-validation

As illustrated by the theorem in Section 2.3, once the model is complex enough, the error ϵ could be close to 0. However, another aspect to measure the performance of a model is its ability for generalization, which is the performance of the model on data different from the one used during the training stage. The idea is that, if the model only relies on its performance on the training data set, it might become over-fitted, where its lead to a good performance in the training data set but poor in other data set. Thus, a separate data set is used for the measurement of performance of a trained model, which is called the validation data set [3]. The mathematical description is:

$$min\, \epsilon = \frac{1}{n^{'}} \sum_{i=1}^{n^{'}} (f(x_i^*, \theta^*, Sub^*) - y_i^*)^2,\ (x^*, y^*) \in validation\, data\, set$$

$$\underset{\theta, Sub}{\arg\min} \sum_{i=1}^{n} (f(x_i, \theta, Sub) - y_i)^2,\ (x, y) \in training\, data\, set$$

where θ^* and Sub^* is derived from $\underset{\theta, Sub}{\arg\min} \sum_{i=1}^{n} (f(x_i, \theta, Sub) - y_i)^2$. The goal of the training process is to find the parameter θ^* and variable subset Sub^* that could minimize the error ϵ on the validation data set.

3.1.1 K-fold cross-validation

In many cases, the size of the data set available for training and validation is small. It is infeasible to split the data into entire independent training and validation sets. As a result, the K-fold cross-validation is introduced (only in this subsection K represents the number of parts in K-fold cross-validation), where the data set is divided into K parts. Each time the model uses the $K-1$ subsets for training while the remaining subset is used for validation. It repeats K times until all the parts are used as validation set once. The model takes the average error of the K repeats as the criterion for determining the best θ^* and Sub^*.

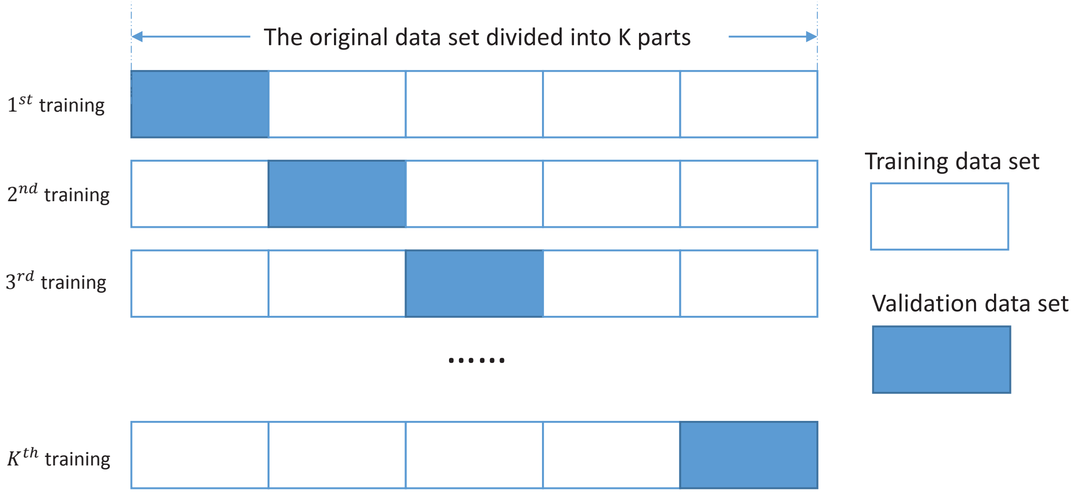


Figure 4: Illustration of data sets in cross-validation

The advantage of this strategy is that the model will go through the whole data set to assess its performance (each part of the data is evaluated separately once). However, it makes the training K times longer than its original version. Combined with the variable selection procedure mentioned in the later section, it will increase the training time to an unacceptable level.

3.1.2 Leave-One-Out (LOO)

If $K = n$ where n is the total amount of samples in the K-fold cross-validation, it forms a special type of cross-validation called Leave-One-Out. Generally speaking, using this technique, the model is trained n times. In each training only one sample is excluded from the training data set and is used for calculating the validation error. Denote (x_{-i}, y_{-i}) as the sample with only the (x_i, y_i) point excluded, (θ, Sub) the selected input variables and parameter and $f(.)$ the model, then the LOO can be described as:

$$\begin{aligned} min\, \epsilon_{LOO} &= \frac{1}{n}\sum_{i=1}^{n}(f(x_i,\theta^*,Sub^*)-y_i)^2 \\ \arg\min_{\theta,Sub} &\sum_{i=1}^{n}(f(x_{-i},\theta,Sub)-y_{-i})^2 \end{aligned}$$

where ϵ_{LOO} is called leave-one-out error. Sharing the same advantage of K-fold cross-validation, it could make the training too computationally intensive. However, for linear models, there is an estimation method called the prediction sum of squares (PRESS) statistics [4] , which could help to calculate the ϵ_{LOO}:

$$\begin{aligned} \epsilon_{LOO} &= \frac{1}{n}\sum(\epsilon_{-i})^2 \\ &= \frac{1}{n}\sum(y_i-\beta^{-i}f(x_i))^2 \\ &= \frac{1}{n}\sum(\frac{\epsilon_i}{1-HAT_{ii}})^2 \end{aligned}$$

where β^{-i} is weights parameter learned from using the data set (x_{-i}, y_{-i}). The HAT_{ii} is the i^{th} element in the diagonal of $HAT = X(X^TX)^{-1}X^T$. Recall that the HAT matrix is already calculated when deriving the solution for ELM, thus, in the calculation for LOO error ϵ_{LOO}, it could be used directly again , adding only very little computational load. Comparing with K-fold cross-validation, it only needs the ELM to be solved once for a given parameter and variable setting, which saves a great deal of training time. Meanwhile, it could provide a more reliable validation error than the traditional cross-validation.

As a result, the LOO error will be used in this project to determine the performance of a model in the training stage.

3.2 Dimensionality reduction

In the previous Section 3.1, the structure of ELM is determined, i.e. ELM with a regularization parameter and linear components. Then it comes to the question about which or what kind of variables should be used as the inputs for the ELM model, namely the features or predictors. In a time series related problem, one could either (1) use a subset of variables from the original inputs, or (2) use another technique to construct a set of new variables as input. Both of them will have a smaller amount of variables. The former is usually called variable selection or feature selection, while the latter is feature extraction. In a time series related problem, the number of potential inputs can be huge, ranging from 1 to the length of the temporal sequence.

For instance, in this project the future values of air pollutant concentrations are predicted according to their past records. However, should the record on hour $T - 1$ being considered as input? Or will the record on hour $T - 24$ be helpful to improve the accuracy? What about using the $T - (24 \times 365)$ hour record or even multiple of them?

The intuition of this issue is that not all the historical values are needed for the prediction. Including too many of them could either significantly increase the computational load for training the ELM, or it might also introduce larger errors if the values are irrelevant. The latter means it actually adds some noise into the model. According to the Huang's theorem in the previous Section 2.3, ELM could approximate any continuous function given sufficient number of neurons. On the other hand, it indicates that with a fixed size of hidden layer, the more complex the problem, the less reliable the approximation. Adding irrelevant variables is just like adding irrelevant patterns to the model, which might decrease the accuracy. Techniques such as feature extraction and variable selection could be used to alleviate such problems.

3.3 Feature extraction

The goal of feature extraction is to replace the old features (or variables) with a set of new features. The amount of new features is often smaller than the amount of old ones, but without a significant loss of original information. The new features are usually derived from the old ones using a certain algorithm or method, such as principal components analysis (PCA), Fourier transform (FT) and Wavelet analysis [22].

Principal Components Analysis (PCA)

PCA is an eigenvalue decomposition process of the covariance matrix of the variables. It aims to transform the original data set into one with lower dimension. Let $X = (var_1^T, var_2^T, ..., var_d^T)$, where var_i is a $1 \times n$ vector, representing the series of a variable in X. One measurement of the similarity between any two variables is the covariance.

$$cov(var_i, var_j) = E[(var_i - E[var_i])(var_j - E[var_j])]$$

It gives a numeric analysis of how much two variables change together. The covariance matrix $\sum$ is generated when it is applied to all the variables of X

$$\sum = cov(X) = \begin{bmatrix} cov(var_1, var_1) & cov(var_1, var_2) & ... & cov(var_1, var_d) \\ cov(var_2, var_1) & cov(var_2, var_2) & ... & cov(var_2, var_d) \\ ... & ... & ... & ... \\ cov(var_d, var_1) & cov(var_d, var_2) & ... & cov(var_d, var_d) \end{bmatrix}$$

where the element $\sum_{ij} = cov(var_i, var_j)$ represents the covariance between the i^{th} and j^{th} variables. A matrix $V = (v_1, v_2, ..., v_d)$, formed by the eigenvectors v_i, that diagonalizes the covariance matrix can be calculated

$$\sum = V \Lambda V^{-1}$$

Λ is a diagonal matrix with the associated eigenvalues on its diagonal. The eigenvalues indicate the amount of variance in the data set that can be explained by the corresponding eigenvector. Then the original data set can be represented as

$$X = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ ... \\ \mathbf{t}_n \end{bmatrix} \cdot \begin{bmatrix} v_1 & v_2 & ... & v_d \end{bmatrix}$$

where $\mathbf{t}_i$ is a row vector called score, which is the representation of X in the principal component space defined by V. In practice, the eigenvectors in the above formula are rearranged in descending order according to their eigenvalues. So that the first new variable in the scores reflects the largest "variance", while the second new variable contains the second largest variance etc. Those new variables are called principal components (PCs).

The solution for T can be derived from eigenvalue decomposition of the covariance matrix or singular value decomposition (SVD) of the data set [34]. In this thesis project, the PCA function from MATLAB is used with its default SVD configuration.

The advantage of PCA is that, if a few top components derived by PCA from a high dimensional data set could explain most of its variance, then only those important components (determined by experiment) are needed for the future consideration. Thus, it remarkably decreases the computational load caused by the high dimensionality.

3.4 Feature (variable) selection

When the algorithm for feature extraction is difficult to design, selections made directly to the raw available features (variables) could be considered. In this project, the target of feature selection is to determine which past records (referred as delays) should be included for the prediction of air quality in next hour. Theoretically, all the records can be relevant, however, due to the limitations of computational resources only historical data within a certain time range regarding to the current time will be considered. These historical features for the selection are called candidate features. In the scope of this thesis, the recent 2000 hours (125 days) of historical data is considered due to the limitation of computational capability. The assumption is the air pollutant concentration in next hour should be related to some of its past records within 2000 hours. In the previous studies, only the latest several days are included as variable and good predictions are generated [47, 55]. 2000 hours should be sufficient as candidate features.

3.4.1 Feature selection path (FSP)

The feature selection path is a useful tool for visualizing how the features are selected during the feature selection process [2]. It is a two-dimensional graph. The x-axis is the current number of features being selected. The y-axis represents the candidate features, which is nominal. If a feature i is selected at step j, meaning there are total j features being selected in this step, the grid would be painted with yellow.

The above FSP shows that, when only 1 feature should be included, then feature 1 is used. If the model allows two features being used, feature 1 and 3 are considered

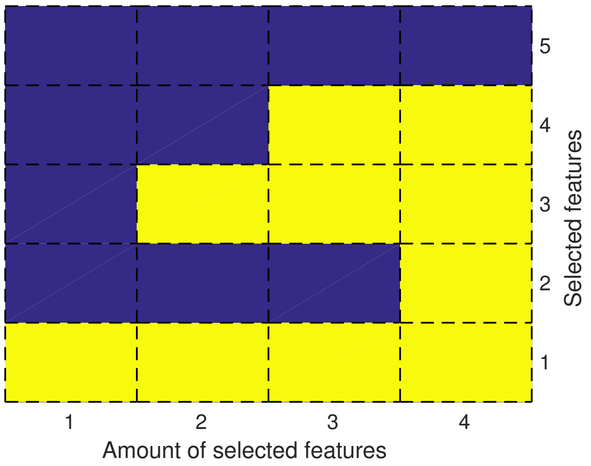


Figure 5: A typical feature selection path

and when 3 features could be used, they would be 1 3 4, etc. Obviously, it is much clearer to understand the development of selected features via FSP than using a vector form such as {1, 1 3, 1 3 4}.

3.4.2 Filtering and wrapper methods

When feature selection is not embedded inside the training algorithm itself, the selection could be classified as filtering and wrapper methods [21].

If the feature selection is carried out without the involvement of the model, which in our case is ELM, but by some other algorithms, the selection is categorized as filtering method. The algorithms mainly work directly to the features themselves, independent of the models in use. Such methods include Least Angle Regression (LARS), the Delta test and mutual information (MI) etc. LARS is discussed in the 3.4.3 and is tested in this thesis project due to its simplicity. Meanwhile, two other methods (the Delta test and mutual information) are introduced here for their reliability, good performance and fast running time [14, 52, 15]. As a future plan, the Delta test and MI could be implemented for detailed tests. Delta test as a variable selection method is introduced by Eirola [16]. It selects the subset of variables based on their performance Delta test, where the differences in the outputs associated with neighboring input points are measured. Mutual information is a quantitative assessment of dependence between the variables, measured according to the Shannon entropy. A subset of features will be selected if it contains the maximum mutual information [14].

The wrapper method, in contrast to the filtering methods, uses the main model itself (in our case it is ELM) as a tool for the variable selection. The criterion is based on the performance of the subsets of variables. Thus, in some cases such methods are quite time consuming. However, the main model used in this thesis project, ELM, is relatively fast in training and testing. It gives the possibility to evaluate the performance of feature selection using the performance of ELM as the criteria. Frenay et al. proposed a similar idea and evaluated it on some classic data sets [2].

In this thesis project, the performance of a filtering method using LARS, and a wrapper method using ELM-based forward selection[3], is studied. The results are shown in Section 4.3.

3.4.3 Least Angle Regression (LARS)

One solution to feature selection is to use the input variable(s) having the largest correlation with the target value. It is referred as Pearson correlation coefficient [22], where the covariance between a certain variable $x_{d'}$ and target value y is measured:

$$\rho_k = \frac{Cov(x_{d'},\, y)}{\sigma_{x_{d'}} \cdot \sigma_y}$$

where $\sigma_{x_{d'}}$,σ_y is the variance of the $x_{d'}$,y series. However, such solution might fail to handle the situation when there is nonlinearity inside the data set. Moreover, the hidden information lying between the variables is not considered. An improved version is called least absolute shrinkage and selection operator LASSO [53]:

$$\arg\min_{w} ||y - Xw|| \;\; subject\,to\, ||w||_1 < t$$

The solution of LASSO would be the weight vector w, where most of its values (weights) for less informative variables would be 0, given a properly set hyper parameter t. As the method takes all the variables into evaluations at a time, it overcomes the issue of the Pearson correlation coefficient where the interactions between the variables are ignored. It could be regarded as a feature selection method, where only the variables with larger weights would be used in the model [53].

LARS is an efficient implementation of LASSO, involving the philosophy of forward selection. It is described in Algorithm 2.

[3] Forward selection will be discussed in detail in Section 3.4.4

Algorithm 2 LARS

```
1. Start the LASSO with all the components in $w$ as 0;
2. Select the most informative variable $\mathbf{x}_j$;
3. Increase the weight $w_j$ for $\mathbf{x}_j$, until another predictor $\mathbf{x}_{j'}$ has as much correlation with the current residual;
4. Increase the weights $w_j$, $w_{j'}$ of $\mathbf{x}_j$ and $\mathbf{x}_{j'}$ (in addition to the previous $w_j$ in step 3), until a third predictor $\mathbf{x}_{j''}$ appears and repeat step 4
5. The procedure will be stopped once certain criteria are reached and the current $w$ would be the solution of LARS
```

In general, LARS calculates the weights of features using a forward selection framework. Illustrated in Figure 6, for instance, x_1 and x_2 are used to predict y. LARS first find the variable with the largest correlation with y, which is x_1 in this case. It moves the current estimation along the direction of x_1, where the residual is represented by $\overrightarrow{r}$, until the angle between $\overrightarrow{r}$ and $x_1(\alpha_1)$ and the angle between $\overrightarrow{r}$ and $x_2(\alpha_2)$ is the same. Let the current direction $\overrightarrow{r}$ be v, then the estimation starts to move in the direction of v and so on. This shows where the name least angle regression is coming from.

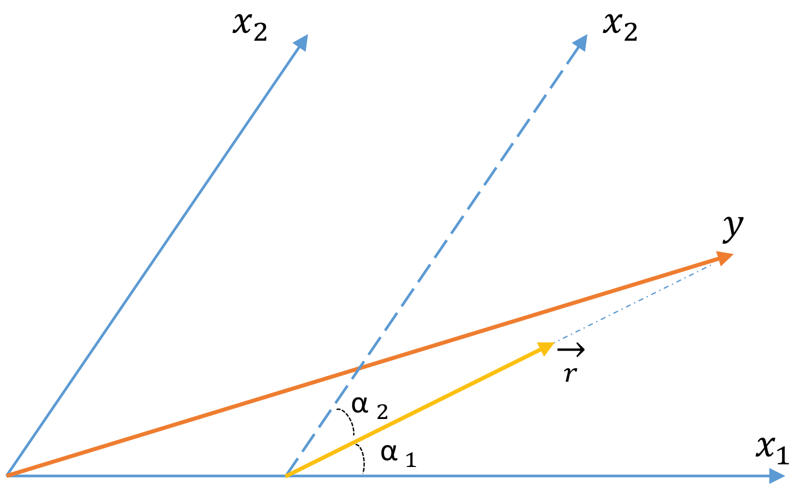


Figure 6: Demonstration of LARS

The linearity and use of forward selection making the process of LARS extremely fast, compared with the other feature selection method such as mutual information

[39] or Delta test [16]. However, the advantage of linearity makes it less reliable when the data set includes non-linear patterns.

In this thesis project, an external tool box for the LARS variable selection written by Timo Similä etc. is used [50].

3.4.4 ELM-based forward selection

In terms of wrapper methods, forward selection is a greedy algorithm. It would find the most suitable variable (i.e. the variable provides the best performance), along with the already selected variable(s), at a time. Then the variables are added into the selected variable set and such process repeats until certain criteria are reached. The steps of ELM-based forward selection is shown in Algorithm 3.

Algorithm 3 ELM-based forward selection

```
1. Initialize the candidate variable set with all the available variables and the
   empty selected variable set.
2. Find a 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 the $x_j$ into the 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.
```

The main advantage of forward selection is fast with a time complexity of $O(n_{Sub} \cdot d)$, where the n_{Sub} is the required 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.

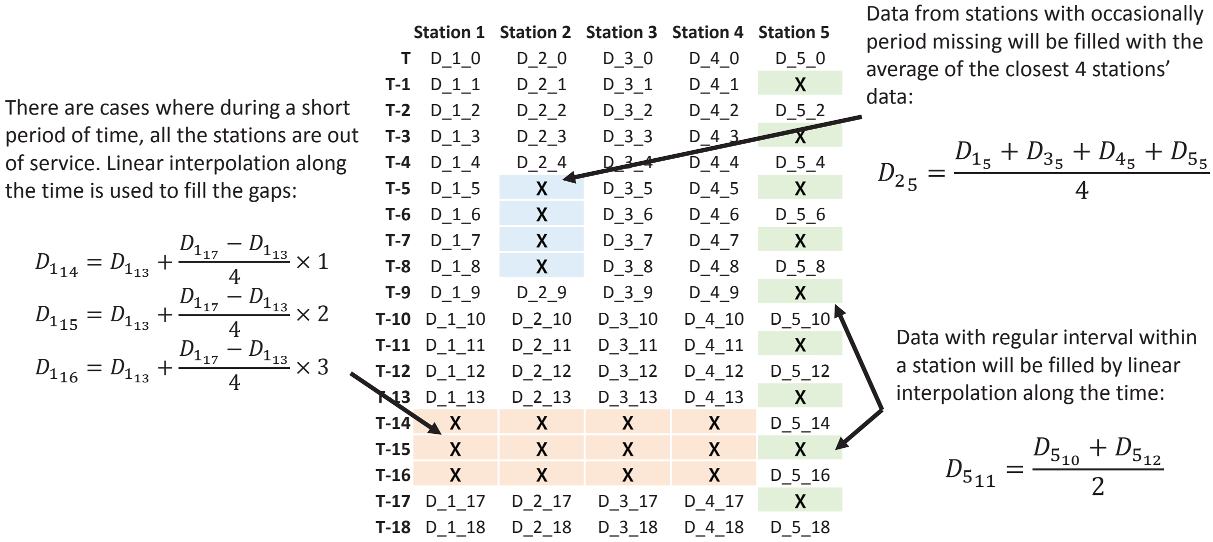
However, it does not guarantee the global optimal solution, as not all the feature combinations in the search space are evaluated. The performance of forward selection would also be influenced by the initialization of the selected variable set. In some cases, certain features are included in the selected variable set at the very beginning, thus they are always used in the model.

There are some other algorithms available in wrapper methods, such as backward or bidirectional elimination [21]. They might be able to provide a better result [35]. However, in this thesis project, the scope is limited to the comparison of a filtering method and a wrapper method. The LARS algorithm will be used as a representative of the filtering method, while forward selection as the wrapper method is implemented.

4 Experiments

4.1 Data preprocessing

Recall that (Section 1.2) the training data includes the two-year hourly records of air pollutant concentration from 25 stations and four types of meteorological records from multiple stations with the same length. These data cannot be used directly as the records are not well formatted. For instance, both air and meteorological data sets contain some missing values. Samples between 1.1.2013 and 30.12.2014 are considered and stations with more than 40% of data points missing are ignored. 5% to 20% of total data points are still missing at the remaining stations. Figure 7 illustrates how those missing values are processed.



	Station 1	Station 2	Station 3	Station 4	Station 5
T	D_1_0	D_2_0	D_3_0	D_4_0	D_5_0
T-1	D_1_1	D_2_1	D_3_1	D_4_1	X
T-2	D_1_2	D_2_2	D_3_2	D_4_2	D_5_2
T-3	D_1_3	D_2_3	D_3_3	D_4_3	X
T-4	D_1_4	D_2_4	D_3_4	D_4_4	D_5_4
T-5	D_1_5	X	D_3_5	D_4_5	X
T-6	D_1_6	X	D_3_6	D_4_6	D_5_6
T-7	D_1_7	X	D_3_7	D_4_7	X
T-8	D_1_8	X	D_3_8	D_4_8	D_5_8
T-9	D_1_9	D_2_9	D_3_9	D_4_9	X
T-10	D_1_10	D_2_10	D_3_10	D_4_10	D_5_10
T-11	D_1_11	D_2_11	D_3_11	D_4_11	X
T-12	D_1_12	D_2_12	D_3_12	D_4_12	D_5_12
T-13	D_1_13	D_2_13	D_3_13	D_4_13	X
T-14	X	X	X	X	D_5_14
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:

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

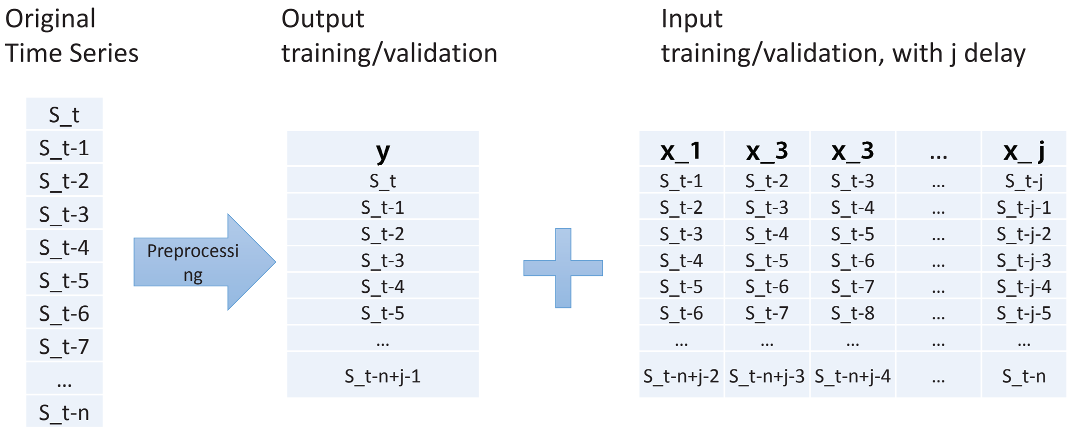
Data with regular interval within a station will be filled by linear interpolation along the time:

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

Figure 7: Treatments towards missing values in various conditions.

For a time series problem, one of the most important issues about the model is the choice of input, i.e. which delay(s) should be used. In some cases, the most recent variable, which is the sample value at the $T-1$ moment, is the best predictor; in some other cases, a selection of samples from $T-24$ to $T-1$ might be able to provide information deeply embedded in the time series, which will then improve the accuracy of the prediction. Such an issue is the core of the feature selection procedure, even before the training of models for the prediction.

There are various methods which could be applied at the feature selection stage: correlation analysis, subset selection and feature dimensionality reduction, etc. [21] Here is an example illustrating how the sample(s) are constructed according to certain delays and at the same time being divided into target output value and corresponding input values for training or validation purpose. Assume j is the potential max delay length, m is the number of meteorological data monitoring stations and S_{T-j} means the sample value of station S at $T-j$ moment. After the transform, an $(n-j+1) \times j$ matrix represents all the candidate data and features is used as the model input. The task of feature selection would be to find the most suitable column(s) in the input matrix to predict the target Y value. (See Figure 8)



S_t
S_t-1
S_t-2
S_t-3
S_t-4
S_t-5
S_t-6
S_t-7
...
S_t-n

y
S_t
S_t-1
S_t-2
S_t-3
S_t-4
S_t-5
...
S_t-n+j-1

x_1	x_3	x_3	...	x_ j
S_t-1	S_t-2	S_t-3	...	S_t-j
S_t-2	S_t-3	S_t-4	...	S_t-j-1
S_t-3	S_t-4	S_t-5	...	S_t-j-2
S_t-4	S_t-5	S_t-6	...	S_t-j-3
S_t-5	S_t-6	S_t-7	...	S_t-j-4
S_t-6	S_t-7	S_t-8	...	S_t-j-5
...	...	...	...	...
S_t-n+j-2	S_t-n+j-3	S_t-n+j-4	...	S_t-n

Figure 8: How the time series sequence is transformed into a matrix with the given delay

For each dimension of the original data set, the values are normalized with zero mean and unit variance.

Details about how PCA is involved in this project

There are two data sets used in this project: the air quality data set and meteorological records from various stations. For instance, the records of computed relative humidity are coming from 21 stations in each hour. After applying PCA on the 21 records at every time stamp, the top five components could explain 93.9% of the variance in the original data set. For some other meteorological type of data, even less components are needed. Figure 9 illustrated the percentage of variance explained by every component for each type of meteorological data.

For all of them, the 1st components represent their largest variance. It shows that the records from different stations are highly correlated for all types of meteorological data. In the later part where feature selection will be introduced, the time complexity

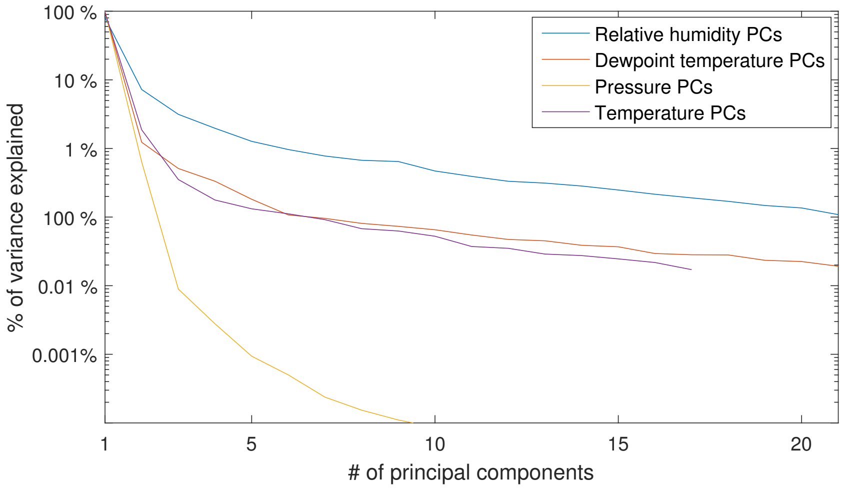


Figure 9: The percentage of variance explained by each principal component

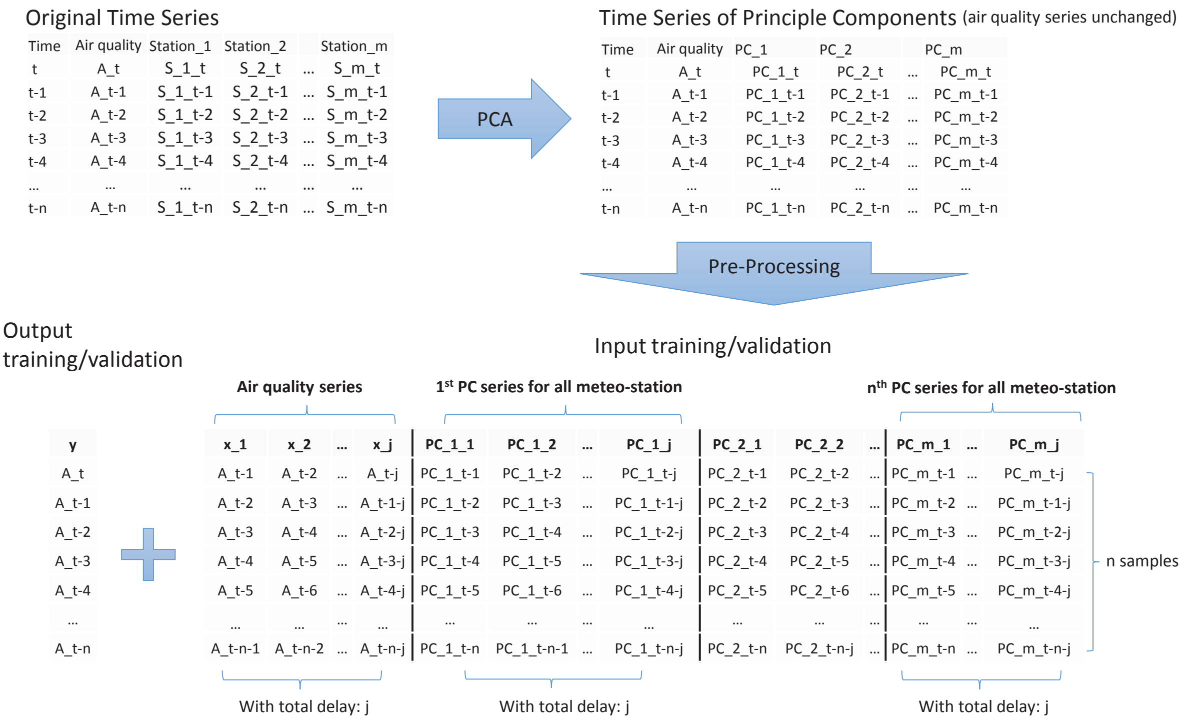
for the selection process is $O(k^2)$ where k is the number of potential features being evaluated. If the top 5 principal components (PCs) instead of the 21 raw records are used as candidate features, the training process would only take $(5/21)^2 \approx 5\%$ of the original training time.

However, the limitation of PCA is obvious. Since it is only a linear transform of the original data set, it could neither be able to extract the nonlinear features of the variables nor interpret the information between the samples across the time. For instance, some air pollutants could diffuse or move through an area with time passing by, where the process is non-linear. When it comes to the PCA, the top components would be most likely representing the average level of the pollutant, instead of the evolution of the pollutant along the time and space.

In this thesis, the results from ELM-based on both the raw meteorological data from stations and the ELM-based on the principal components of those data are implemented and compared.

4.2 Optimization of hyper-parameters

The experiments on how the influence of hyper-parameters on the performance of ELM are carried out. To simplify the experiment, the nitrogen oxide concentration time series data set is used throughout this subsection, where the predictors are concentration records from past T hours. A combination of records from all the 14 available stations is used to form a more comprehensive test data instead of



Original Time Series
PCA
Time Series of Principle Components (air quality series unchanged)
Pre-Processing
Output training/validation
Input training/validation
Air quality series
1st PC series for all meteo-station
nth PC series for all meteo-station
n samples
With total delay: j
With total delay: j
With total delay: j

Time	Air quality	Station_1	Station_2		Station_m
t	A_t	S_1_t	S_2_t	...	S_m_t
t-1	A_t-1	S_1_t-1	S_2_t-1	...	S_m_t-1
t-2	A_t-2	S_1_t-2	S_2_t-2	...	S_m_t-2
t-3	A_t-3	S_1_t-3	S_2_t-3	...	S_m_t-3
t-4	A_t-4	S_1_t-4	S_2_t-4	...	S_m_t-4
...	...	...	...	...	...
t-n	A_t-n	S_1_t-n	S_2_t-n	...	S_m_t-n

Time	Air quality	PC_1	PC_2		PC_m
t	A_t	PC_1_t	PC_2_t	...	PC_m_t
t-1	A_t-1	PC_1_t-1	PC_2_t-1	...	PC_m_t-1
t-2	A_t-2	PC_1_t-2	PC_2_t-2	...	PC_m_t-2
t-3	A_t-3	PC_1_t-3	PC_2_t-3	...	PC_m_t-3
t-4	A_t-4	PC_1_t-4	PC_2_t-4	...	PC_m_t-4
...	...	...	...	...	...
t-n	A_t-n	PC_1_t-n	PC_2_t-n	...	PC_m_t-n

y
A_t
A_t-1
A_t-2
A_t-3
A_t-4
...
A_t-n

x_1	x_2	...	x_j	PC_1_1	PC_1_2	...	PC_1_j	PC_2_1	PC_2_2	...	PC_m_1	...	PC_m_j
A_t-1	A_t-2	...	A_t-j	PC_1_t-1	PC_1_t-2	...	PC_1_t-j	PC_2_t-1	PC_2_t-2	...	PC_m_t-1	...	PC_m_t-j
A_t-2	A_t-3	...	A_t-1-j	PC_1_t-2	PC_1_t-3	...	PC_1_t-1-j	PC_2_t-2	PC_2_t-3	...	PC_m_t-2	...	PC_m_t-1-j
A_t-3	A_t-4	...	A_t-2-j	PC_1_t-3	PC_1_t-4	...	PC_1_t-2-j	PC_2_t-3	PC_2_t-4	...	PC_m_t-3	...	PC_m_t-2-j
A_t-4	A_t-5	...	A_t-3-j	PC_1_t-4	PC_1_t-5	...	PC_1_t-3-j	PC_2_t-4	PC_2_t-5	...	PC_m_t-4	...	PC_m_t-3-j
A_t-5	A_t-6	...	A_t-4-j	PC_1_t-5	PC_1_t-6	...	PC_1_t-4-j	PC_2_t-5	PC_2_t-6	...	PC_m_t-5	...	PC_m_t-4-j
...	...	...	...	...	...	...	...	...	...	...	...	...	...
A_t-n-1	A_t-n-2	...	A_t-n-j	PC_1_t-n	PC_1_t-n-1	...	PC_1_t-n-j	PC_2_t-n	PC_2_t-n-j	...	PC_m_t-n	...	PC_m_t-n-j

Figure 10: How the input matrices for training are generated from the original data. Here the original meteorological records are replaced by its component

using data from any single station. Please refer to Figure 7 which shows how the combination of records is constructed.

In the rest of the thesis, such data set will be referred as all-station data set, while the term single-station data set indicates that the samples used to train a certain ELM are coming from the same station. To produce comparable results, an all-station data set is generated for experiments in this section, with candidate predictors as x_{T-i} $i \in [1, 2, ..., 100]$ and x_T the target output. The total number of training samples is 3475 and the size of testing samples is 876.

All the seeds for random generators also remain the same. It is because the performance of ELM is affected by the initialization of hidden layer weights. The results can fluctuate if the seed is not fixed. The results will also change along with the alterations of hyper-parameters, such as the number of hidden neurons and the number of input variables. It is these hyper-parameters that need to be optimized, with certain optimization function according to the MSE on the test data set. Thus, such fluctuation in results might affect the convergence of the optimization algorithm.

4.2.1 Optimization for regularization parameter λ

First, two kinds of ELMs are compared, with and without the regularization parameter λ. The purpose is to determine if adding a regularization parameter in ELM could help to avoid overfitting on our data set. The number of input variables increases from 1 to 100 in both cases, where x_{T-1} is used as input variable, then $\{x_{T-1}, x_{T-2}\}$ and go on until $\{x_{T-1}, x_{T-2}, ..., x_{T-100}\}$ become the input. The number of hidden neurons is fixed at 200. The regularization parameter λ is optimized via the MATLAB function "fminsearch" for each input size. The results of mean squared error (MSE) on a separate test data are presented in Figure 11.

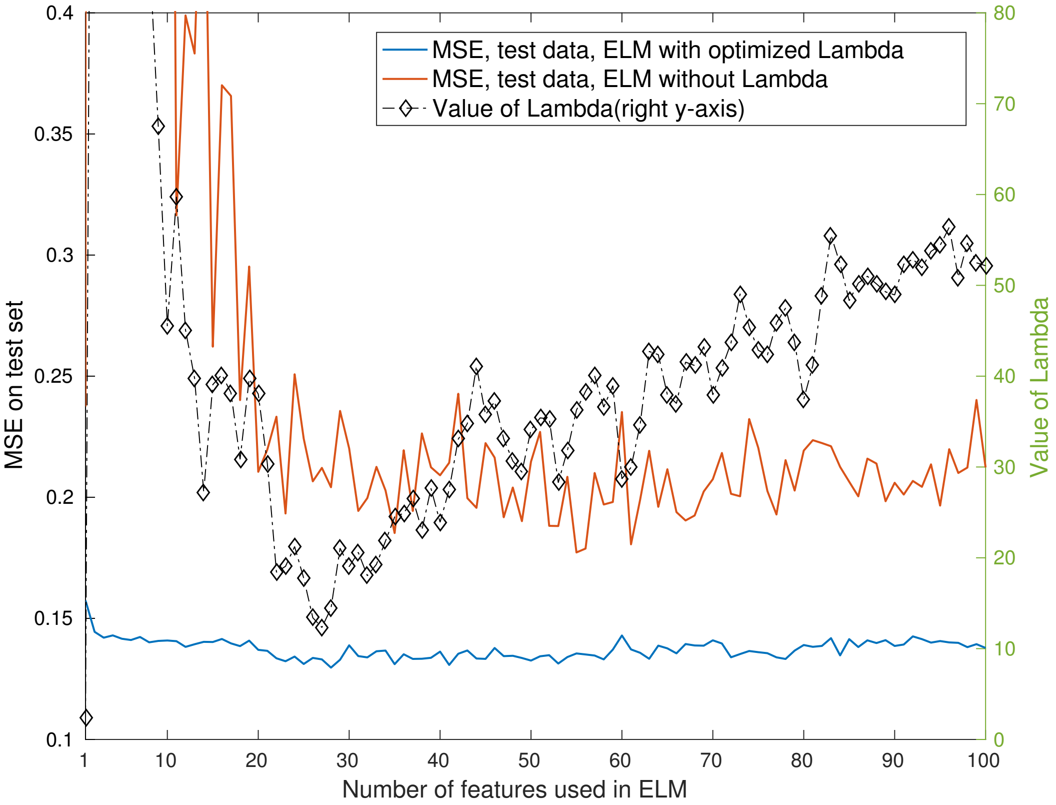


Figure 11: MSE on the test data set, with different number of input variables

The results indicate that when the regularization parameter λ is set properly, ELM could achieve a remarkably lower error than its non-regularized version. Meanwhile, with an increase in model complexity (with more input variables added), the optimized value of λ increases accordingly, keeping the mean squared test error a stable level. Comparing to the slight increasing of MSE on the test data set in the non-regularized ELM version, it is evident that λ is preventing the model from overfitting during the training.

Since the optimal value of λ highly depends on the number of input features, λ is optimized for each number of input features during the model selection or training process. It will cost an additional 20 to 30 runs of ELM for a better λ. In the forward selection, once a λ is optimized, it will remain unchanged with a fixed amount of input features. To evaluate those features, there can be hundreds of times of running the ELM. Thus, the optimization of λ will be less than 10% of the running time and it is acceptable.

4.2.2 Optimization for the number of hidden neurons

The number of hidden neurons, k, is the key parameter in ELM, as it determines how accurate an approximation of ELM can achieve. On the other hand, the model complexity along with the computational load increases significantly with a larger hidden neuron number.

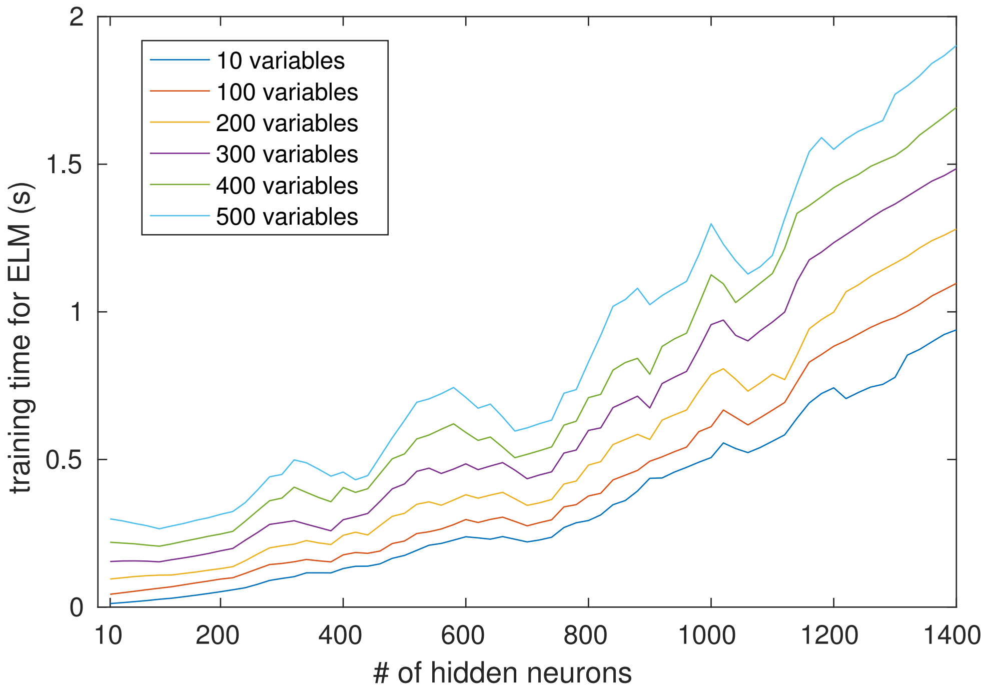


Figure 12: Training time with different hidden layer size and different number of inputs

Trade-off between accuracy and computational time must be made. Thus, investigation on the relationship among the accuracy, the hidden neuron layer size k and the regularization parameter λ is carried out. In the experiment the value of k

changes from 50 to 1050, while the value of λ changes from 1 to 150 (preliminary estimation for range of λ is provided by the results in Section 4.2.1) and the number of variables is fixed at 50. The LOO error on the training data set and the MSE on the testing data set are shown in the Figure 13.

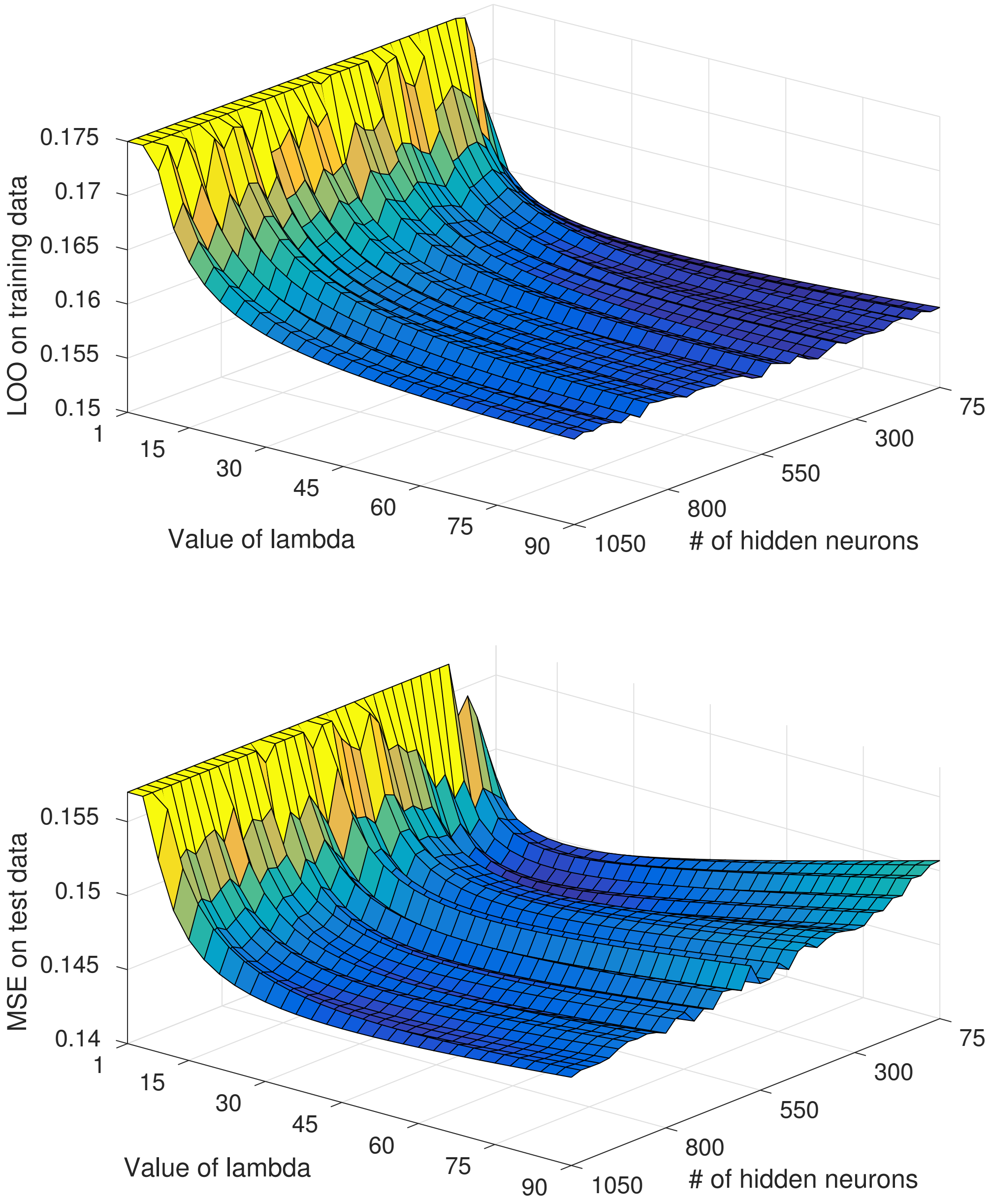


Figure 13: LOO error and MSE with different the numbers of hidden neurons and different values of λ

According to the results, the LOO error keeps decreasing with a larger hidden neuron layer. However, the MSE on the test data converges after the number of neurons reaches a certain level. According to the results, the estimation of the level is around 400. Thus the amount of hidden neurons of the more complex ELM used for the prediction is set to be 400. Moreover, the difference of MSE between $k = 100$ and $k = 500$ is relatively small (5%), which means using a smaller amount of hidden neurons in the later feature selection stage should not affect the results too much. It could, on the other hand, significantly reduce the time for the selection process.

4.2.3 Optimization for other hyper-parameters

The influence of various values for the scales of bias and weights is checked in the same way. The results are shown in Figure 14 and 15.

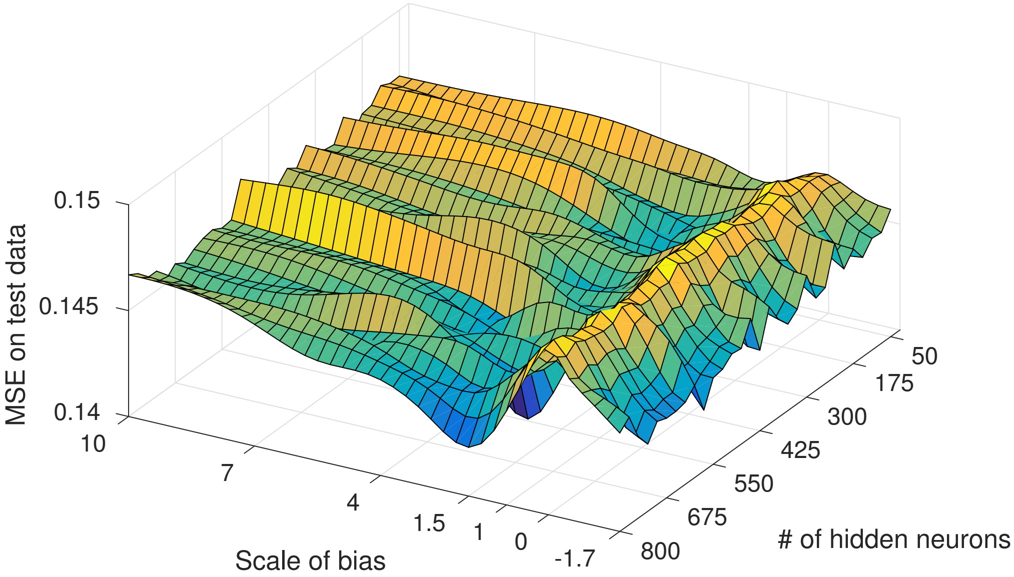


Figure 14: MSE with give the number of hidden neurons and scale of bias

As shown in the results, the optimal scales for bias and weight are independent with the size of the hidden neuron layer. The difference on the MSE caused by the changes of both bias and weight is relatively small (less than 10%). According to the experiments, 1.5 and 1 are selected as the scales for bias and weight. They will remain constant in the rest of the project.

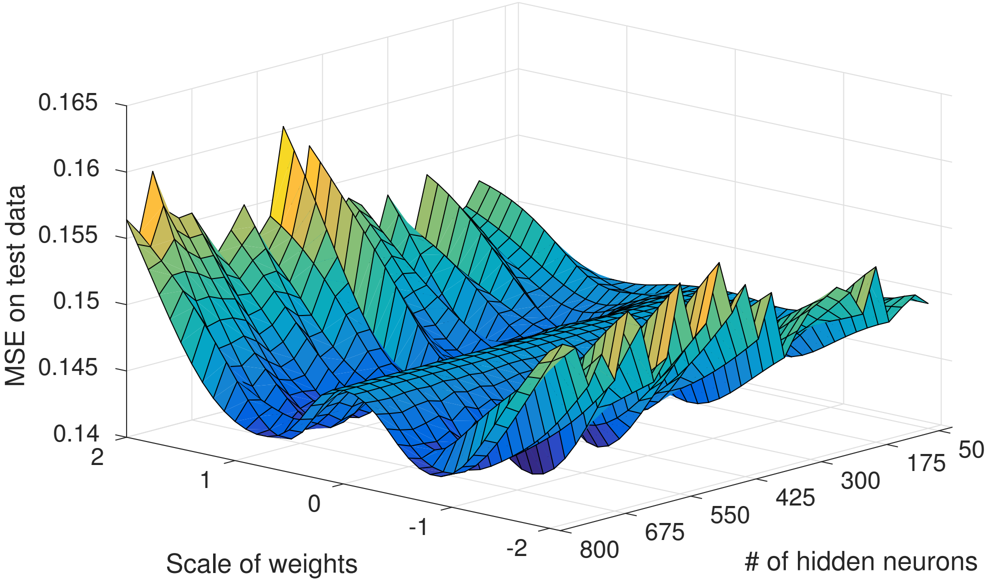


Figure 15: MSE with give the number of hidden neurons and scale of weights

4.3 Determining feature selection method

After the hyper-parameters of ELM are settled, the input features (variables) for ELM are decided via feature selection. First the reliability of the two feature selection methods is assessed. The one with better performance is used to execute the feature selection. The performance of a filtering method (Least Angle Regression) and a wrapper method (ELM-based forward selection) is compared.

When the selections are verified by the ELM, the same set of hyper-parameters ($scale_{bias} = 1.5$, $scale_{weights} = 1$, the number of neurons $k = 50$) is used and the seed for the random generator remains the same. Meanwhile, an all-station data set with 2000 candidate variables, i.e. $\mathbf{x}_{T-i}, i \in [1, 2, ..., 2000]$ is generated with the same number of samples for this evaluation. The goal is to select the top 200 variables from among the 2000 candidates, which should maximize the performance of ELM in the prediction. For a better understanding about the performance, a linear model based forward selection is implemented as a baseline for the analysis. The results are shown in Figure 16.

	ELM-based FS	LARS
minimum error achieved on test set	0.1082	0.112
Time used for the selection	126 min	5.088s

Table 1: Performance comparison between the two feature selection methods

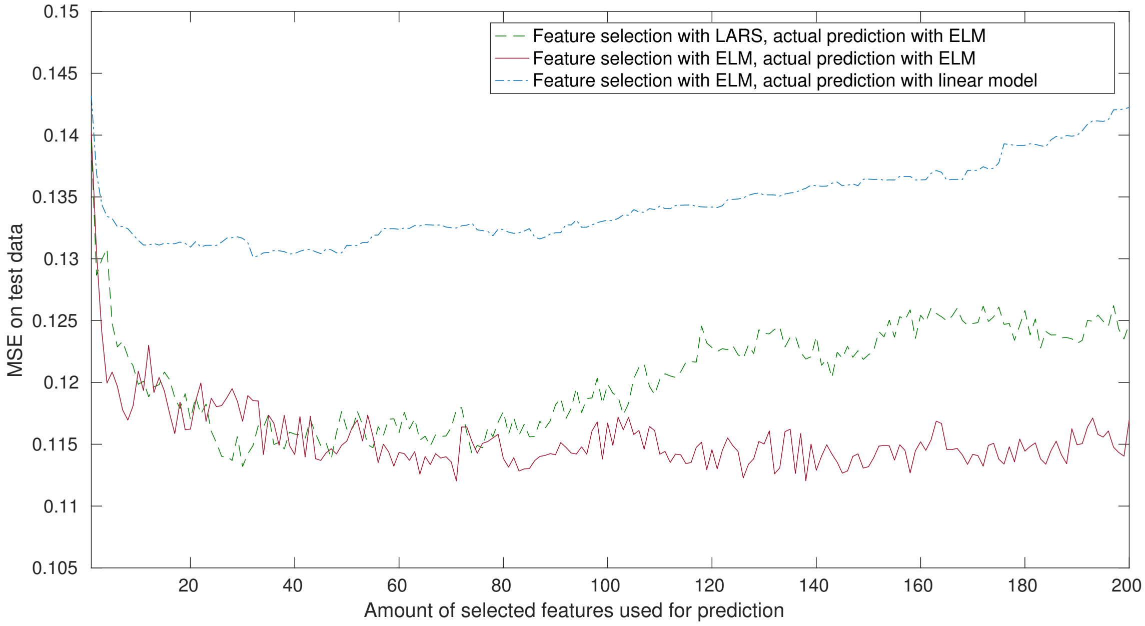


Figure 16: Performance comparison between the two feature selection methods and the linear model

The results show that the filtering method LARS runs much faster than the wrapper methods using ELM. However, the selection of variables provided by the ELM-based forward selection outperforms the other two models in the prediction stage, which is around 20% better than the linear model based forward selection and 5% better than using features selected by LARS.

At last, different hyper-parameters are tested for the ELM-based forward selection, to check if the features selected by the methods stay constant. Figure 17 shows the results for different sizes of hidden neuron layers. It indicates that the positions of the majority of the top selected features in each ELM configuration remain stable regardless of the hidden neuron amount. In the future, some other quantitative methods might be used for further investigation on this consistency issue.

Taking into account of the performance, ELM-based forward selection is used as the variable selection method in the following experiment.

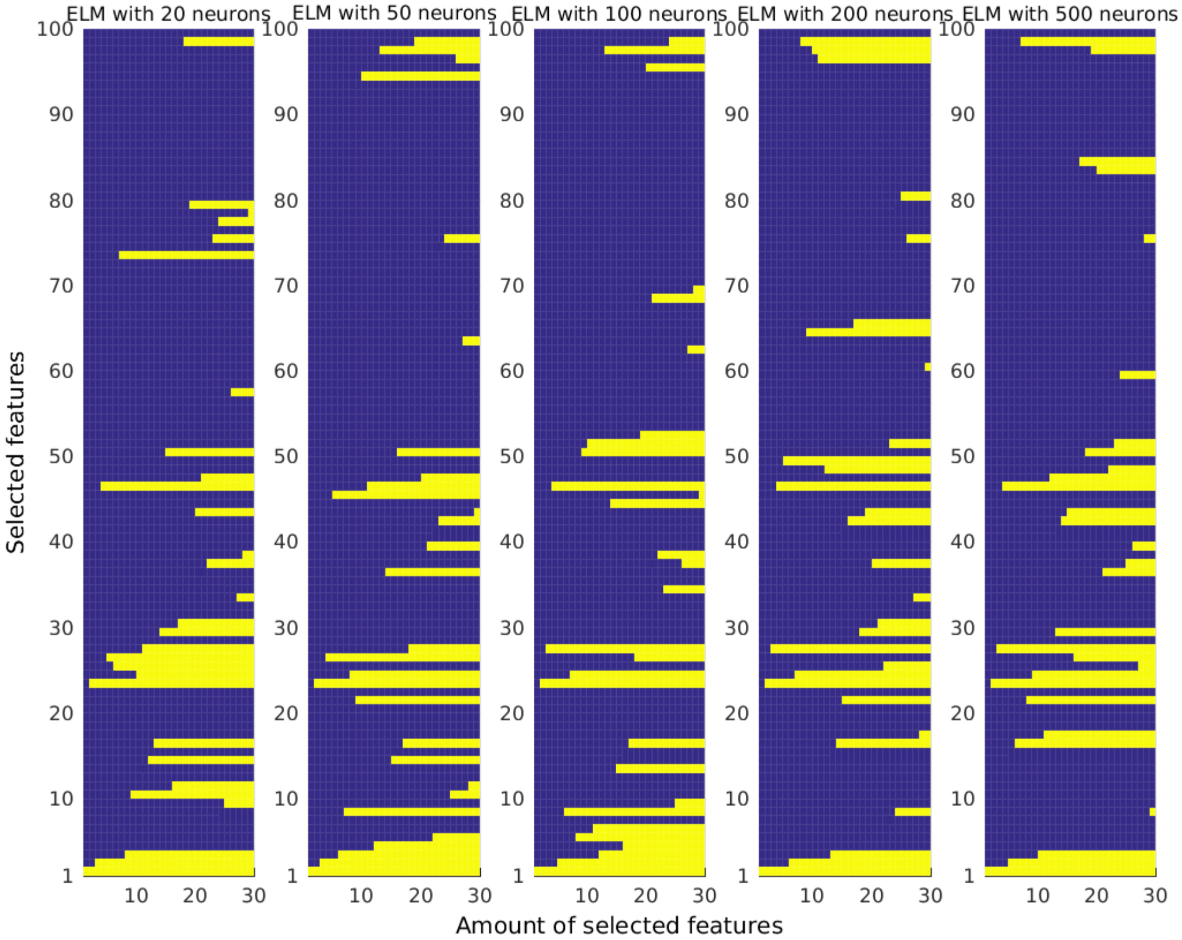


Figure 17: The feature selection paths from ELM-based forward selection with increasing neuron numbers

4.4 ELM-based forward selection

4.4.1 Candidate variable sets

Feature selection needs to be done on both air concentration time series data and meteorological records. The number of candidate variables (delays) for air concentration time series is set as 2000, while the candidate delay for meteorological records is set as 30 (hours) for each station per data type. Predictions for the air concentration for next hour are based on the historical records of the concentration and the recent records of all the meteorological types.

Such settings are coming from the situation that (21 relative humidity + 21 dew point temperature + 15 pressure + 17 temperature)=74 meteorological records are available each hour. The small increase in the number of candidate delays would produce a great number of total candidate features. Even only to consider the meteorological records from recent 7 days will generate a quite large amount of potential variables (74×24×7=12432) for the feature selection. Here 30 (hours) is used as it should be sufficient to provide information covering the periodicity within one whole day, while the computational time is still acceptable.

To decrease the time required by the feature selection process, a two-stage ELM-based forward selection is implemented. At the first stage, the top 50 variables in each type of data, represented by $\mathbf{x}_{air}$ for air concentration, $\mathbf{x}_h$ for relative humidity, $\mathbf{x}_{dt}$ for dew point temperature, $\mathbf{x}_p$ for pressure and $\mathbf{x}_t$ for temperature, are selected using a relatively simple ELM (hidden layer size is 50). Then at the second stage, feature selection is done separately on candidates feature formed by $[\mathbf{x}_{air}]$, $[\mathbf{x}_{air}, \mathbf{x}_h]$, $[\mathbf{x}_{air}, \mathbf{x}_{dt}]$, $[\mathbf{x}_{air}, \mathbf{x}_p]$, $[\mathbf{x}_{air}, \mathbf{x}_t]$[4] using a more complex ELM (hidden layer size is 400). The ELM should be able to reveal if a certain type of meteorological data can improve the air quality prediction.

The above selections are carried out separately on both all-station data set and the single-station data sets. At the same time, two types of data preprocessing method are implemented. In one case, the raw meteorological data is used (normalized with zero mean and unit variance). In the other case the principal components of meteorological data are used. The other settings such as hyper-parameters of the ELM remain the same. Table 2 shows the combination of candidate feature space, where the feature selections are carried out on each of them separately.

	Data from all stations or a single station	
Preprocessed with PCA or not	all-station data + meteo raw	single-station data + meteo raw
	all-station data + meteo PCs	single-station data+ meteo PCs

Table 2: Four different data sets being evaluated in the experiment

4.4.2 Analysis of feature selection results

The results using the all-station data set are all shown here. Meanwhile, only some of the results for single-station data sets are shown due to the length of the thesis. Only one type of air pollutant concentration (NO) is predicted here for simplicity. Figure 18 shows how the features are selected using ELM-based forward selection.

[4] [x,y] here means to form a new set of input that include x and y together.

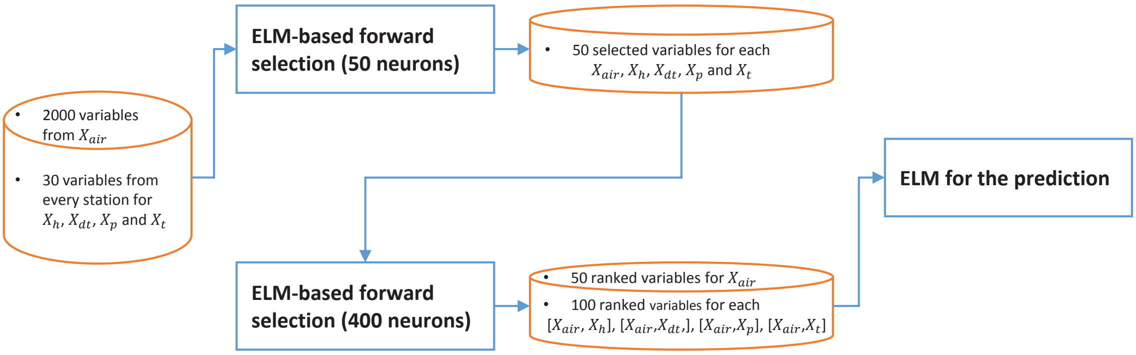

• 2000 variables from X_{air}
• 30 variables from every station for X_h, X_{dt}, X_p and X_t
ELM-based forward selection (50 neurons)
• 50 selected variables for each X_{air}, X_h, X_{dt}, X_p and X_t
ELM-based forward selection (400 neurons)
• 50 ranked variables for X_{air}
• 100 ranked variables for each [X_{air}, X_h], [X_{air},X_{dt},], [X_{air},X_p], [X_{air},X_t]
ELM for the prediction

Figure 18: ELM-based feature selection

All-station data: meteo raw vs. meteo PCs

Figure 19 shows the results base on the all-station data set, where one ELM model is trained to predict the pollutant concentration of all the 14 stations. The errors of predictions using both PCs and raw meteorological data are shown in the same subplots for better comparison.

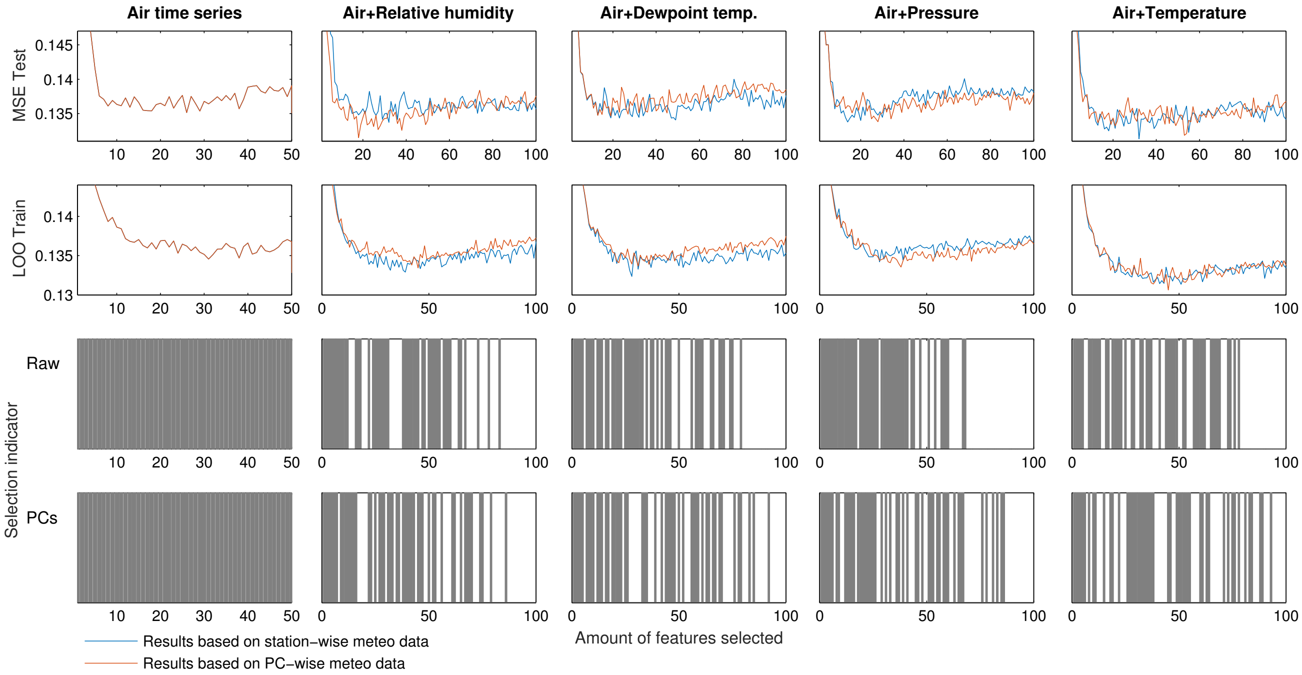


Figure 19: Results comparison between all-station data + raw meteo and all-station data + meteo PCs

The first row shows the MSE of the two predictions, with different feature combinations. The first column shows the prediction based on air pollutant concentration time series only. The latter four plots show the MSE based on features selected with combination [$\mathbf{x}_{air}$, $\mathbf{x}_{humidity}$], [$\mathbf{x}_{air}$, $\mathbf{x}_{dewtemp}$], [$\mathbf{x}_{air}$, $\mathbf{x}_{pressure}$] and [$\mathbf{x}_{air}$, $\mathbf{x}_{temp}$]. The second row holds the same structure while the y-axis is the LOO error during the training. The bar charts in the last two rows indicate which feature, whether an air quality time series variable or a meteorological variable, is chosen. The dark bar represents the selection of an air quality time series variable while the white bar is referring to a meteo variable.

According to the performance on test data set, the feature combination of [$\mathbf{x}_{air}$, $\mathbf{x}_{humidity}$] and [$\mathbf{x}_{air}$, $\mathbf{x}_{temp}$] improved the prediction accuracy by around 5% compared to the model with air quality time series data only. The dew point temperature and pressure variables do not help much in this case.

Then it comes to the comparison between the models using principal components and raw station-wise variables. For the two types meteorological data which reduce the MSE, there is no significant difference in the performance between ELMs using PCs and raw records. However, comparing the bar charts in the last two rows, less meteorological type of variables are selected when PCs are used. It might suggest in this one-model-for-all-station type of prediction, PCs can be used to decrease the candidate variables in the feature selection stage. The features selected from the two data sets are compared in Figure 20.

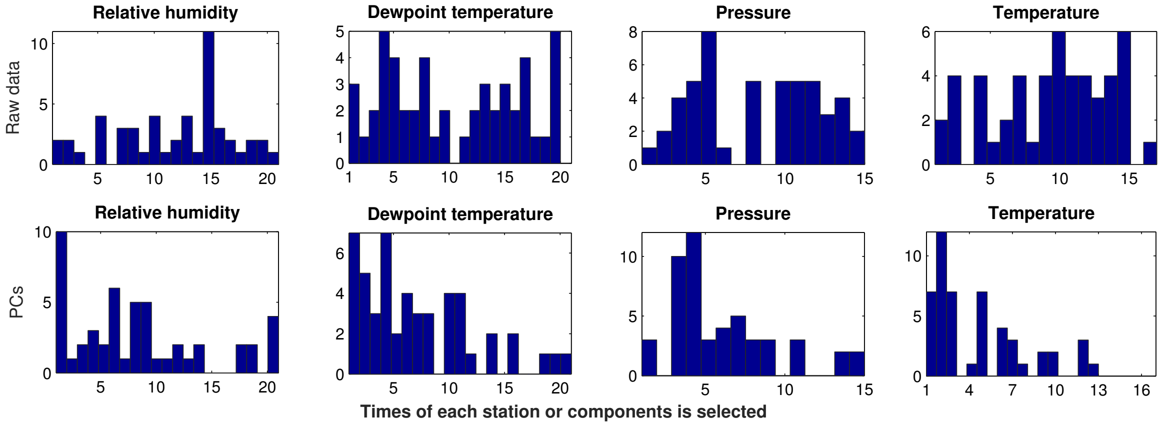


Figure 20: The amount of times a feature (its delays) is selected, all station model

The x-axis of Figure 20 represents the index of a station (or the index of a principal component). The y-axis (value) is the number of features related to the certain station (component), i.e. the delays coming from to the same station (component).

The results show that the selected features from the station-wise raw data set spread evenly among all the stations. When the PCA processed data set is used as the candidate, the top principal components are selected with the highest probability. Since the errors with ELM using the two different data sets are almost the same, it indicates that using the top principal components in the feature selection can reduce the potential candidate features without influencing the model performance.

Single-station data: meteo raw vs. meteo PCs

The similar experiment is carried out, where the air pollutant concentration for a single station is predicted with the station-specific ELM. The ELM uses the historical concentration records only from that station and the same meteorological data in the whole region. Figure 21 shows the results.

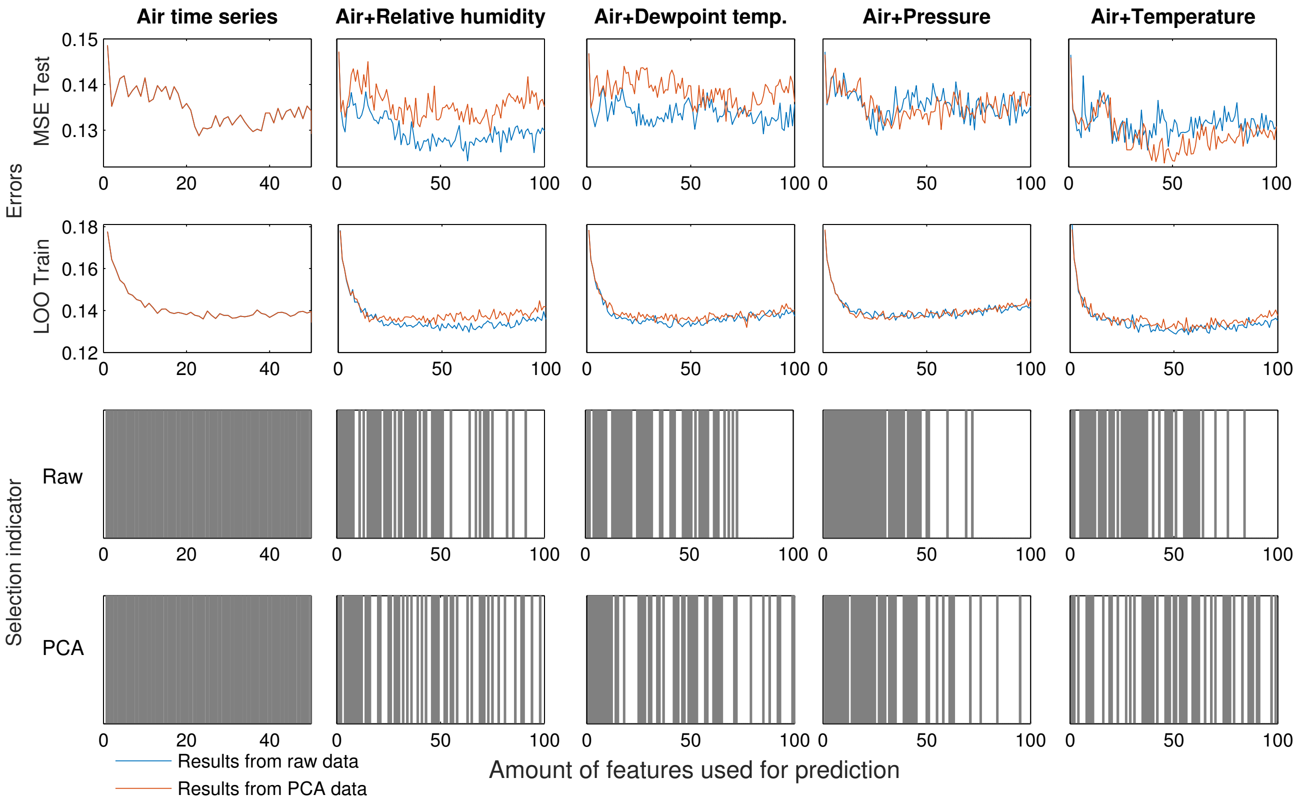


Figure 21: Results comparison between single station data + raw meteo and single station data + meteo PCs

Similar to the results from one-ELM-for-all-station experiment, using air quality records plus the raw relative humidity records and temperature variables (both raw data and PCs) improves the prediction accuracy. Less raw meteorological variables are selected comparing to the amount in previous experiment. An important discovery is that including the PCA processed features do not help with decreasing the error in most cases. The interpretation could be the principal components only contain the spatial information. Such information is independent in between the PCs along the time. On the contrary, the spatial–temporal information is included in the raw meteorological data set. For instance, if station A is upwind from station B and start to record increasing level of a certain pollutant, then most likely in a few hours or even minutes such rising will be detected by station B. When PCA is used on such data, it is possible that such information is lost due to the little variance it can reflect.

Figure 22 illustrates the times of related features (delays) from each station (components) being selected. Top principal components are still favored by the model. However, unlike in the previous experiment with all-station data set where raw features are evenly selected, raw features form some certain stations are selected more frequently. It indicates the strong connection between those stations and the one station whose air quality is predicted.

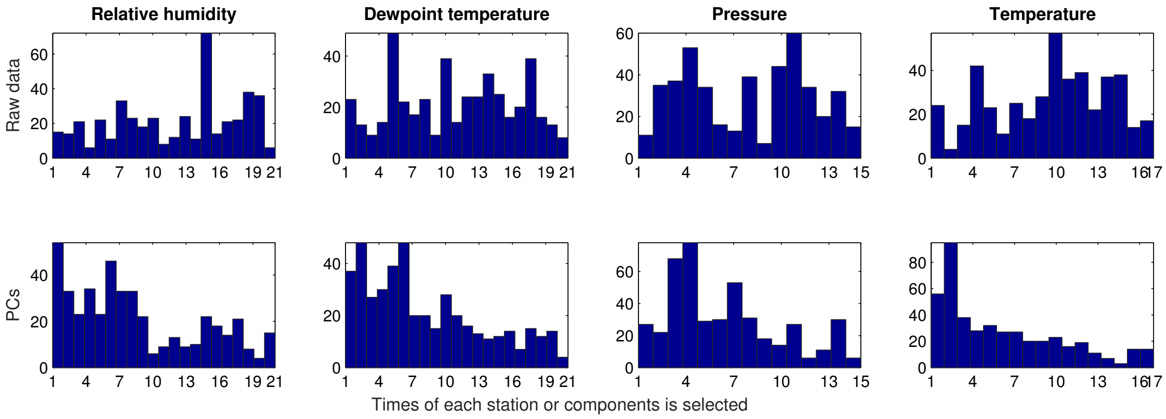


Figure 22: The amount of times a feature (its delays) is selected, single station model

All in all, in most cases, adding relative humidity and temperature improves the prediction accuracy. The pressure and dew point temperature have a little influence on the results. The principal component analysis is an efficient way to reduce the search space for feature selection by replacing the raw meteorological data with its principal components. However, to achieve a lower error, using a station specific

ELM model with raw meteorological data is suggested.

4.5 ELM with selected features and parameters

Using the features and hyper-parameters selected above, the performance of ELM is evaluated against a linear model as a base line. Figure 23 shows 1-step-ahead predictions provided by the two models compared to the real value.

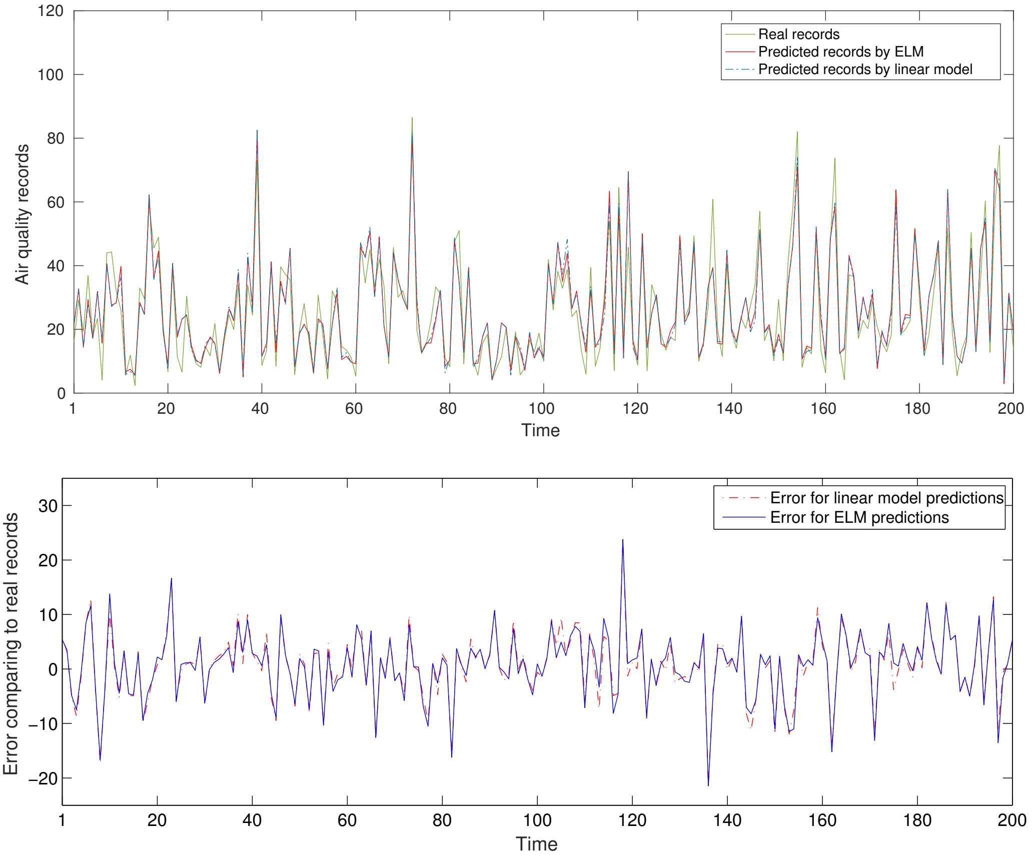


Figure 23: The 1-step-ahead predictions of the ELM model with linear model as a baseline:
true values (above) and errors (bottom)

In this test, the mean squared error for the prediction of ELM is 41.91 while it is 43.68 for the linear model. As the majority of the time is spent on data preprocessing, feature selection and hyper-parameter optimization, the running time of ELM used for the prediction is quite short. From the second figure above, the error made by linear model is visibly larger (red line) comparing with the error from the ELM model (blue line).

5 Summary

5.1 Conclusions

In this thesis project, a special type of neural network: Extreme Learning Machine is implemented to predict the future air quality based on the air quality time series itself and the external meteorological records. A properly preprocessed data set with hourly records in the year 2013 to 2014 is used. A regularized version of ELM with linear components added is chosen as the main model.

First, some experiments are carried out to determine the hyper-parameters: the number of hidden neurons, the regularization parameter and the scales of weights and bias. Results show that there are optimal solutions for both scales of bias and weights, which are independent of other hyper-parameters. The regularization parameter varies in accordance with the size of the hidden layer and the number of input variables. Thus, it is optimized in the model along with the present amounts of hidden neurons and inputs. ELM with a larger number of hidden neurons can generate better predictions. However, the marginal benefits of adding new hidden neurons decrease very quickly after certain numbers are reached. In this particular project, the optimal scales of bias and weights are 1.5 and 1, while the number of neurons is set as 400 based on a trade-off between the ELM performance and training time.

Then the performance of two feature selection methods is compared. The results show that the features selected by ELM-based forward selection method (with a much smaller hidden neuron layer) can generate a lower error than the features selected by Least Angle Regression, however, at a cost of significant longer processing time.

Furthermore, a feature extraction method, i.e. principal component analysis (PCA), is used in the hope of reducing the candidate meteorological variables for feature selection. The experiment shows that if only one ELM model is trained to approximate future air quality for all stations, using the selected principal components can achieve the same level of accuracy comparing to the same ELM using raw meteorological records. In the experiment, the top principal components are selected with the highest probability, which means the PCA can be an effective feature extraction method reducing the number of candidate variables when one "global" ELM is used. However, when the station-specific ELM is developed for the prediction, using the raw meteorological variables leads to better accuracy. It could be that the station-specific ELM is able to use the local information presented in the raw variables which is missing in the principal components.

Finally, the performances of ELMs using different types of meteorological data are compared. It shows that by utilizing certain types of meteorological data, such as the relative humidity and temperature, the model can generate better predictions than the model using historical air concentration records alone.

All in all, with proper parameters and features, Extreme Learning Machine shows its reliability for the prediction of air quality data. In this project the station-specific ELM, with humidity, temperature and historical air quality features selected by the ELM-based forward selection, provides the most accurate prediction.

5.2 Future works and discussion

Although ELM could provide predictions with acceptable accuracy, only one ELM model is used during the prediction. In the future, ensembling multiple ELMs might be able to improve the accuracy, especially for such non-stationary time series prediction problem. Meanwhile, when a more complex model and more variables are introduced, the time for the training process such as feature selection might increase significantly. Thus the parallelization of those algorithms, such as the ELM-based feature selection, might be a promising topic for future implementation [59]. Multi-step head predictions could be studied in the future, which would be able to provide timely forecasts for the public.

ELM is used as the base model in the forward selection at feature optimization stage. Although the constancy of features selected by ELM with different hidden neuron numbers is checked objectively, further study could be addressed to reveal the theories behind it.

Last but not the least, only the air quality time series from the same station is used as predictor in this project. The dispersion of the pollutant is a spatial-temporal process, so the accuracy of prediction might be improved if records of multiple stations are incorporated. This topic looks quite promising, however it also brings challenges in computation and other aspects.

References

[1] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 6

[2] Frénay Benoît, Mark van Heeswijk, Yoan Miche, Michel Verleysen, and Amaury Lendasse. Feature selection for nonlinear models with extreme learning machines. *Neurocomputing*, 102:111–124, 2013. 18, 20

[3] Christopher M Bishop. *Pattern Recognition and Machine Learning*. 2006. 8, 13

[4] Gianluca Bontempi and Souhaib Ben Taieb. *Statistical Foundations of Machine Learning*. OTexts.org. 15

[5] R T Burnett, M Smith-Doiron, D Stieb, S Cakmak, and J R Brook. Effects of Particulate and Gaseous Air Pollution on Cardiorespiratory Hospitalizations. *Archives of environmental health*, 54(2):130–9, 1999. 1

[6] Gail A Carpenter, Stephen Grossberg, and Michael A Arbib. Adaptive Resonance Theory. *Encyclopedia of Machine Learning*, 104(1-2):19–26, 2009. 6

[7] A. J. Chauhan, M. T. Krishna, A. J. Frew, and S. T. Holgate. Exposure to nitrogen dioxide (NO2) and respiratory disease risk. *Reviews on Environmental Health*, 13(1-2):73–90, 1998. 1

[8] Giorgio Corani. Air quality prediction in Milan: Feed-forward neural networks, pruned neural networks and lazy learning. *Ecological Modelling*, 185(2-4):513–529, 2005. 2

[9] Walter F. Dabberdt, Mary Anne Carroll, Darrel Baumgardner, Gregory Carmichael, Ronald Cohen, Tim Dye, James Ellis, Georg Grell, Sue Grimmond, Steven Hanna, John Irwin, Brian Lamb, Sasha Madronich, Jeff McQueen, James Meagher, Talat Odman, Jonathan Pleim, Hans Peter Schmid, and Douglas L. Westphal. Meteorological research needs for improved air quality forecasting. *Bulletin of the American Meteorological Society*, 85(4):563–586, 2004. 2

[10] Wanyu Deng, Qinghua Zheng, and Lin Chen. Regularized Extreme Learning Machine. *2009 IEEE Symposium on Computational Intelligence and Data Mining*, (60825202):389–395, 2009. 2, 11

[11] Ravinesh C Deo and Mehmet Şahin. Application of the extreme learning machine algorithm for the prediction of monthly Effective Drought Index in eastern Australia. *Atmospheric Research*, 153:512–525, 2015. 2

[12] R Derwent. Analysis and interpretation of air quality data from an urban roadside location in Central London over the period from July 1991 to July 1992. *Atmospheric Environment*, 29(8):923–946, 1995. 1

[13] Chris Edsall and Tom Clarkson. The Air Quality Database, http://www.ilmanlaatu.fi/. 4

[14] Emil Eirola. Machine learning methods for incomplete data and variable selection. 2014. 19

[15] Emil Eirola, Elia Liitiäinen, Amaury Lendasse, Francesco Corona, and Michel Verleysen. Using the delta test for variable selection. In *ESANN*, pages 25–30, 2008. 19

[16] Emil Eirola, Elia Liitiäinen, Amaury Lendasse, Francesco Corona, and Michel Verleysen. Using the Delta test for variable selection. In *Proc. of ESANN 2008 European Symposium on Artificial Neural Networks*, pages 25–30, 2008. 19, 22

[17] Hendrik Elbern. EURAD - IM regional forecasting system and performances. pages 1–36, 2012. 1

[18] Xiao Feng, Qi Li, Yajie Zhu, Junxiong Hou, Lingyan Jin, and Jingjie 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. 1

[19] MW Gardner and SR Dorling. Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London. *Atmospheric Environment*, 33(5):709–719, 1999. 2

[20] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 7

[21] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003. 19, 22, 24

[22] Isabelle Guyon and Andre Elisseeff. Feature Extraction, Foundations and Applications: An introduction to feature extraction. *Studies in Fuzziness and Soft Computing*, 207:1–25, 2006. 16, 20

[23] Donald Olding Hebb. *The Organization of Behavior*, volume 911. 1949. 6

[24] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006. 7

[25] Cécile Honoré, Laurence Rouïl, Robert Vautard, Matthias Beekmann, Bertrand Bessagnet, Anne Dufour, Christian Elichegaray, Jean Marie Flaud, Laure Malherbe, Frédérik Meleux, Laurent Menut, Daniel Martin, Aline Peuch, Vincent Henri Peuch, and Nathalie Poisson. Predictability of European air quality: Assessment of 3 years of operational forecasts and analyses by the PREV'AIR system. *Journal of Geophysical Research Atmospheres*, 113(4), 2008. 1

[26] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(April):2554–2558, 1982. 6

[27] Guang-Bin Huang and Lei Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16):3056–3062, 2007. 10

[28] Guang-Bin Huang and Lei Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16):3460–3468, 2008. 10

[29] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *Neural Networks, IEEE Transactions on*, 17(4):879–892, 2006. 10

[30] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, 2:985–990 vol.2, 2004. 2, 9

[31] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006. 2, 10

[32] Yoshifusa Ito. Approximation of continuous functions on Rd by linear combinations of shifted rotations of a sigmoid function with and without scaling. *Neural Networks*, 5(1):105–115, 1992. 9

[33] William James. The Principles of Psychology, 1890. 6

[34] I T Jolliffe. Principal Component Analysis, Second Edition. *Encyclopedia of Statistics in Behavioral Science*, 30(3):487, 2002. 18

[35] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. 22

[36] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982. 6

[37] M. Kolehmainen, H. Martikainen, and J. Ruuskanen. Neural networks and periodic components used in air quality forecasting. *Atmospheric Environment*, 35(5):815–825, 2001. 2

[38] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 6

[39] Yoan Miche, Mark van Heeswijk, Patrick Bas, Olli Simula, and Amaury Lendasse. TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization. *Neurocomputing*, 74(16):2413–2421, 2011. 11, 22

[40] Marvin Minsky and Seymour Papert. *Perceptrons: expanded edition*. 1988. 6

[41] Mihaela Mircea, Massimo D'Isidoro, Alberto Maurizi, Lina Vitali, Fabio Monforti, Gabriele Zanini, and Francesco Tampieri. A comprehensive performance evaluation of the air quality model BOLCHEM to reproduce the ozone concentrations over Italy. *Atmospheric Environment*, 42(5):1169–1185, 2008. 1

[42] NOAA's National Centers for Environmental Information (NCEI). Hourly/Sub-Hourly Observational Data, http://www.ncdc.noaa.gov/. 4

[43] 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. 1

[44] Paul Portney and John Mullahy. Urban Air Quality and Acute Respiratory Illness. *Journal of Urban Economics*, 20(1):21–38, 1986. 1

[45] 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. 1

[46] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 6

[47] S. Salcedo-Sanz, C. Casanova-Mateo, a. Pastor-Sánchez, and M. Sánchez-Girón. Daily global solar radiation prediction based on a hybrid Coral Reefs Optimization – Extreme Learning Machine approach. *Solar Energy*, 105:91–98, 2014. 18

[48] Jurgen Schmidhuber, Dan Cireşan, Ueli Meier, Jonathan Masci, and Alex Graves. On fast deep nets for AGI vision. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6830 LNAI(January):243–246, 2011. 7

[49] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 7

[50] Timo Similä and Jarkko Tikka. Multiresponse sparse regression with application to multidimensional scaling. In *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pages 97–102. Springer, 2005. 22

[51] Brian Jr. Stone. Urban sprawl and air quality in large US cities. *Journal of Environmental Management*, 86(4):688698, 2008. 1

[52] Taiji Suzuki, Masashi Sugiyama, and Toshiyuki Tanaka. Mutual information approximation via maximum likelihood estimation of density ratio. *2009 IEEE International Symposium on Information Theory*, pages 463–467, 2009. 19

[53] Robert Tibshirani. Regression shrinkage and selection via the lasso: A retrospective. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 73(3):273–282, 2011. 20

[54] W. S. Tunnicliffe, P. S. Burge, and J. G. Ayres. Effect of domestic concentrations of nitrogen dioxide on airway responses to inhaled allergen in asthmatic patients. *Lancet*, 344(8939-8940):1733–1736, 1994. 1

[55] Ahmad Zia Ul-Saufie, Ahmad Shukri Yahaya, Nor Azam Ramli, Norrimi Rosaida, and Hazrul Abdul Hamid. Future daily PM10 concentrations prediction by combining regression models and feedforward backpropagation models with principle component analysis (PCA). *Atmospheric Environment*, 77:621–630, 2013. 18

[56] US Environmental Protection Agency. Actions You Can Take to Reduce Air Pollution, https: //www3.epa.gov/region1/airquality/reducepollution.html, 2003. 1

[57] Mark van Heeswijk. *Advances in Extreme Learning Machines*. PhD thesis, 2015. 11

[58] Mark van Heeswijk, Yoan Miche, Tiina Lindh-Knuutila, Peter AJ Hilbers, Timo Honkela, Erkki Oja, and Amaury Lendasse. Adaptive ensemble models of extreme learning machines for time series prediction. In *Artificial Neural Networks–ICANN 2009*, pages 305–314. Springer, 2009. 2

[59] Mark van Heeswijk, Yoan Miche, Erkki Oja, and Amaury Lendasse. GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing*, 74(16):2430–2437, 2011. 42

[60] Chi Man Vong, Weng Fai Ip, Pak Kin Wong, and Chi Chong Chiu. Predicting minority class for suspended particulate matters level by extreme learning machine. *Neurocomputing*, 128:136–144, 2014. 2

[61] Dimitris Voukantsis, Kostas Karatzas, Jaakko Kukkonen, Teemu Räsänen, Ari Karppinen, and Mikko 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. 2

[62] Bernard Widrow and Marcian E Hoff. Adaptive switching circuits., 1960. 6

[63] Yan Xu, Zhao Yang Dong, Ke Meng, Kit Po Wong, and Rui Zhang. Short-term load forecasting of Australian National Electricity Market by an ensemble model

of extreme learning machine. *IET Generation, Transmission & Distribution*, 7(September 2012):391–397, 2013. 2

[64] Yang Zhang, Marc Bocquet, Vivien Mallet, Christian Seigneur, and Alexander Baklanov. Real-time air quality forecasting, part I: History, techniques, and current status. *Atmospheric Environment*, 60:632–655, 2012. 1

[65] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-Air: When Urban Air Quality Inference Meets Big Data. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, page 1436, 2013. 2