Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Renewable Energy 29 (2004) 939–947

RENEWABLE ENERGY

www.elsevier.com/locate/renene

Support vector machines for wind speed prediction

M.A. Mohandes [a,*], T.O. Halawani [a], S. Rehman [b], Ahmed A. Hussain [a]

[a] *Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*

[b] *Center for Engineering Research, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*

Received 6 December 2002; accepted 20 November 2003

Abstract

This paper introduces support vector machines (SVM), the latest neural network algorithm, to wind speed prediction and compares their performance with the multilayer perceptron (MLP) neural networks. Mean daily wind speed data from Madina city, Saudi Arabia, is used for building and testing both models. Results indicate that SVM compare favorably with the MLP model based on the root mean square errors between the actual and the predicted data. These results are confirmed for a system with order 1 to system with order 11.

Keywords: Wind speed prediction; Neural networks; Multilayer perceptron; Support vector machines

1. Introduction

Renewable energy resources are increasingly utilized due to global political uncertainty and alarmingly increasing pollution levels in air, water, and soil. Wind energy has become the focal point for energy seekers/developers due to the availability of megawatt size wind machines, accessible management facilities, ease and low cost of maintenance, government subsidies, tax benefits, etc. The power of wind is a clean, inexhaustible, and a free source of energy. This source has served humankind for many centuries by propelling ships and driving wind turbines to

* Corresponding author. Tel.: +966-3-860-4709; fax: +966-3-860-3535.
E-mail address: mohandes@kfupm.edu.sa (M.A. Mohandes).

0960-1481/$ - see front matter
doi:10.1016/j.renene.2003.11.009

grind grains and pumps water [1]. Due to the availability of a cheap and plentiful supply of petroleum (pre-1970s), the high cost and uncertainty of wind placed it at an economic disadvantage. However, after the 1973-oil embargo, it is realized that the world's oil supplies would not last forever and other energy sources have to be developed.

For proper and efficient utilization of wind power, the prediction of wind speed is very important. It is needed for site selection, performance prediction, planning of windmills and the selection of an optimal size of the wind machine for a particular site. Wind speed can be predicted by using traditional autoregression modeling with moving averages [2], and more recently using artificial neural network methods [3,4]. This paper uses the support vector machine method and compares its performance with that of multilayer perceptron (MLP).

2. Support vector machines

The main objective of regression is to approximate a function $g(x)$ from a given noisy set of samples $G = \{(x_i, y_i)\}_{i=1}^{N}$ obtained from the function g. The basic idea of support vector machines (SVM) for regression is to map the data x into a high dimensional feature space via a nonlinear mapping and to perform a linear regression in this feature space [5,6].

$$f(x) = \sum_{i=1}^{D} w_i \phi_i(x) + b \tag{1}$$

where $\{\phi_i(x)\}_{i=1}^{D}$ are called features, b and $\{w_i\}_{i=1}^{D}$ are coefficients that have to be estimated from the data. Thus, a nonlinear regression in the low dimensional input space is transferred to a linear regression in a high dimensional (feature) space. The coefficients $\{w_i\}_{i=1}^{D}$ can be determined from the data by minimizing the function

$$R[w] = \frac{1}{N} \sum_{i=1}^{N} |f(x_i) - y_i|_{\varepsilon} + \lambda \| |w| \|^2 \tag{2}$$

where λ is a regularization constant and the cost function defined by

$$|f(x_i) - y_i|_{\varepsilon} = \begin{cases} |f(x) - y| - \varepsilon & \text{for } |f(x_i) - y_i| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

is called Vapnik's ε-insensitive loss function. It can be shown that the minimizing function has the following form

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) k(x_i, x) + b \tag{4}$$

with $\alpha_i \alpha_i^* = 0, \ \alpha_i, \alpha_i^* \geq 0 \quad i = 1, \ldots, N$ and the kernel function $k(x_i, x)$ describes the inner product in the D-dimensional feature space.

$$k(x,y) = \sum_{j=1}^{D} \phi_j(x)\phi_j(y)$$

It is important to note that the features ϕ_j need not be computed; rather what is needed is the kernel function that is very simple and has a known analytical form. The only condition required is that the kernel function has to satisfy Mercer's condition. Some of the mostly used kernels include polynomial, Gaussian, and sigmoidal. Note also that for Vapnik's ε-insensitive loss function, the Lagrange multipliers α_i, α_i^* are sparse, i.e. they result in nonzero values after the optimization (2) only if they are on the boundary, which means that they satisfy the Karush–Kuhn–Tucker conditions. The coefficients α_i, α_i^* are obtained by maximizing the following form

$$R(\alpha^*, \alpha) = -\varepsilon \sum_{i=1}^{N} (\alpha_i^* + \alpha_i) + \sum y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i^* + \alpha_i) \times (\alpha_i^* - \alpha_i) k(x_i, x_j) \quad (5)$$

subject to $\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0, \quad 0 \le \alpha_i, \alpha_i^* \le C$

Only a number of coefficients α_i, α_i^* will be different from zero, and the data points associated to them are called support vectors. Parameters C and ε are free and have to be decided by the user. Computing b requires a more direct use of the Karush–Kuhn–Tucker conditions that lead to the quadratic programming problems stated above. The key idea is to pick those values α_k, α_k^* for a point x_k on the margin, i.e. α_k or α_k^* in the open interval (0, C). One x_k would be sufficient but for stability purposes it is recommended that one take the average over all points on the margin. More detailed description of SVM for regression can be found in Refs. [7–11].

3. Multilayer perceptron

The current interest in artificial neural networks is largely due to their ability to mimic natural intelligence in its learning from experience. They learn from examples by constructing an input–output mapping without explicit derivation of the model equation. They have been used in a broad range of applications including pattern classification, function approximation, optimization, prediction and automatic control and many others. An artificial neural network consists of many interconnected identical neurons. Each neuron computes a weighted sum of its n input signals and pass it to a nonlinear function:

$$y = \varphi\left(\sum_{j=1}^{n} w_i x_j - \theta\right) \quad (6)$$

where x_j, for $j = 1, 2, \ldots, n$, are input signals, w_j is the weight associated with the

jth input, θ is a threshold, and $\varphi(\cdot)$ is a sigmoid activation function defined by

$$\varphi(x) = \frac{1}{1 + \exp(-x)} \tag{7}$$

The weights of the connections between neurons are adjusted during the training process to achieve the desired input/output relation of the network. An MLP network has its neurons organized into layers with no feedback or lateral connections. Layers of neurons other than the output layer are called hidden layers. The input layer consists of a set of sensors that only provide input signals and do not perform any computations. Input signals propagate through the network in a forward direction, on a layer-by-layer basis, where each neuron sums its weighted inputs and then applies the nonlinear activation function to produce an output signal that is used as inputs to the neurons in the proceeding layer, and so on until the output layer.

Learning of MLP is accomplished by adjusting the weights of the connections between neurons. The back-propagation algorithm [12,13] is a supervised iterative training method that uses training data consisting of P input–output pairs of vectors that characterizes the problem. A sample from the training data is randomly chosen and provided to the inputs of the network, which computes the outputs on a layer-by-layer base until the output layer. The difference between the actual output of the network and the correct output that is provided in the training data is used to adjust the weights, so that the next time that same input is provided, the network output will be closer to the correct one. This process is repeated for all other input–output pairs in the training data. Thus, the back-propagation algorithm minimizes an error function defined by the average of the sum square difference between the output of each neuron in the output layer and the desired output.

The iterative process of presenting an input–output pair and updating the weights continues until the error function reaches a pre-specified value or the weights no longer change. In that case, the training phase is done and the network is ready for testing and operation.

4. Results and discussion

The available wind speed data cover a period of 12 years between 1970 and 1982. This data is divided into three parts: training data that is used to build up the models of the above mentioned systems, validation data that is used to select the parameters of the systems that best perform on these data, and the testing data that is neither utilized in building the systems nor on selecting the system parameters. The obtained results on the testing data indicate that the SVM system outperforms the MLP model as indicated by the predication graph and by the root mean square errors.

The performance measure adopted throughout this paper is the mean square error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_{ti})^2 \tag{8}$$

where x_t is the observed value and x is the predicted value. For the best results, the data was normalized between 0 and 1 by dividing all the data by the maximum wind speed value for Medina city. The total available daily wind speed data for Medina city (4228 days) is divided into three parts: the first part (2000 days) is used for training, the second part (1500 days) is used for cross-validation, and the last part (728 days) is used for testing the performance. The training data is used to design the model, while the cross-validation data is used for model selection, where the network with best performance on validation data is adopted. The testing data has never been used on building or selecting the model and is used to estimate the performance of the network on future unseen data. Fig. 1 shows the normalized daily wind speed values for Medina city.

To find the best order, systems of orders from 1 to 11 were studied. The order of the system determines the number of previous wind speed days used as inputs to predict the wind speed of the next day. The first step in utilizing MLP is to determine the structure (number of hidden layers and the number of neurons in each

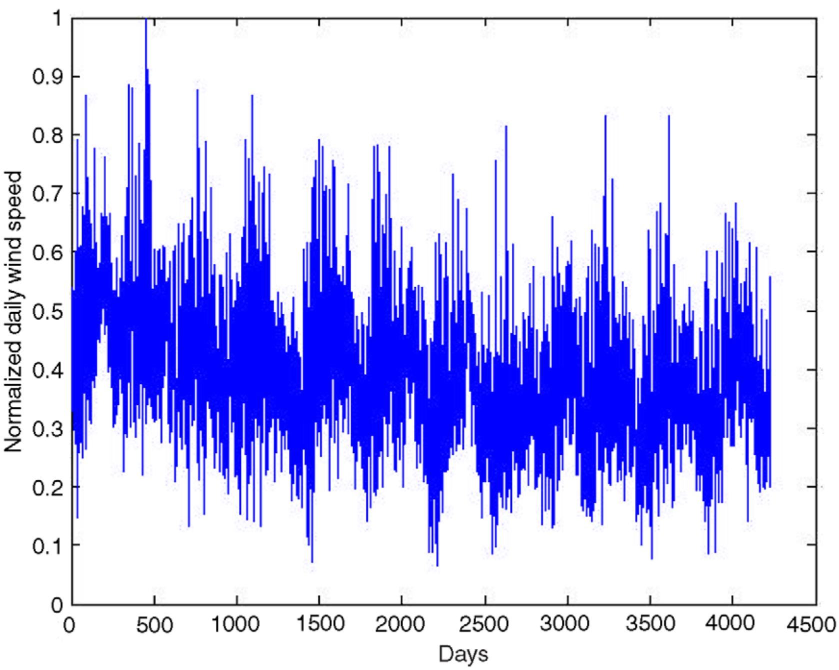

Fig. 1. Normalized mean daily wind speed values for Medina city.

layer) of the network. Therefore, for each order, networks with 2, 4, 6,...100 hidden neurons were considered. Structures that optimized the performance on the validation data were adopted to report the performance on the testing data. The Levenberg-Marquardt optimization method is used in implementing the back-propagation algorithm due to its proven performance. The parameters used are: the activation function of the hidden layer is the tansig function, while the activation function of the output layer is linear function. The number of epochs is 40. This is due to the experiments that showed that after epoch 40, there is hardly any change on the MSE on training data.

The support vector machine used in this paper utilizes the Gaussian kernel. The validation data is used to optimize the parameters C and ε described above. Several trials were used to find reasonably good values of these parameters for the wind speed prediction of Medina city. The performance on testing data for all orders is shown in Fig. 2.

Fig. 2 indicates that the SVM outperforms MLP on all orders. The figure also indicates that as expected, as the order of the system increases, the performance of the system improves. To further compare the performance, we select order 11 for both systems. Fig. 3 shows histograms of the differences between observed and predicted values for SVM and MLP systems with data of order 11. Fig. 4 shows the performance of SVM and MLP on part of the testing data. This figure also indi-

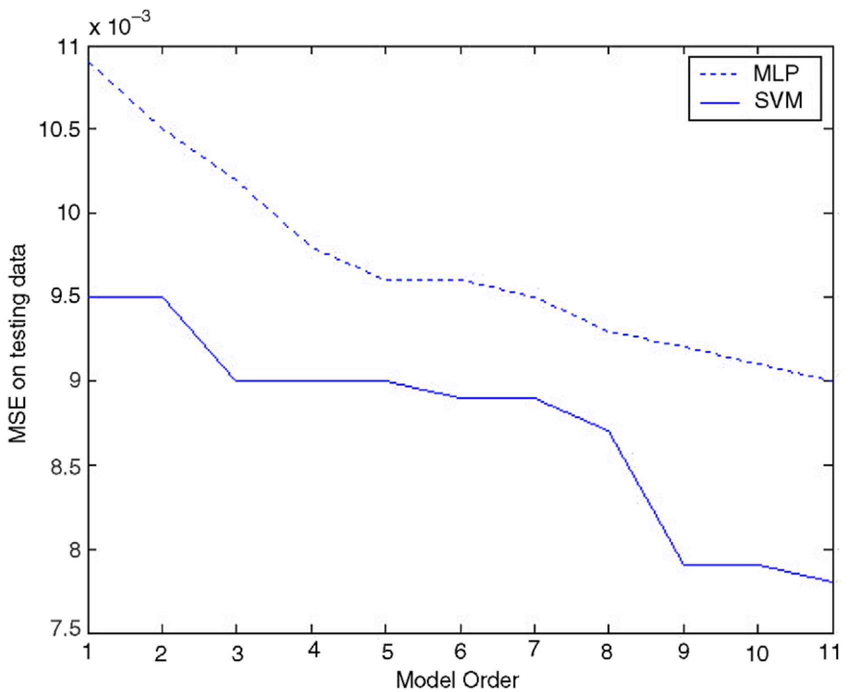

Fig. 2. Comparison between the MSE of SVM and MLP on testing data.

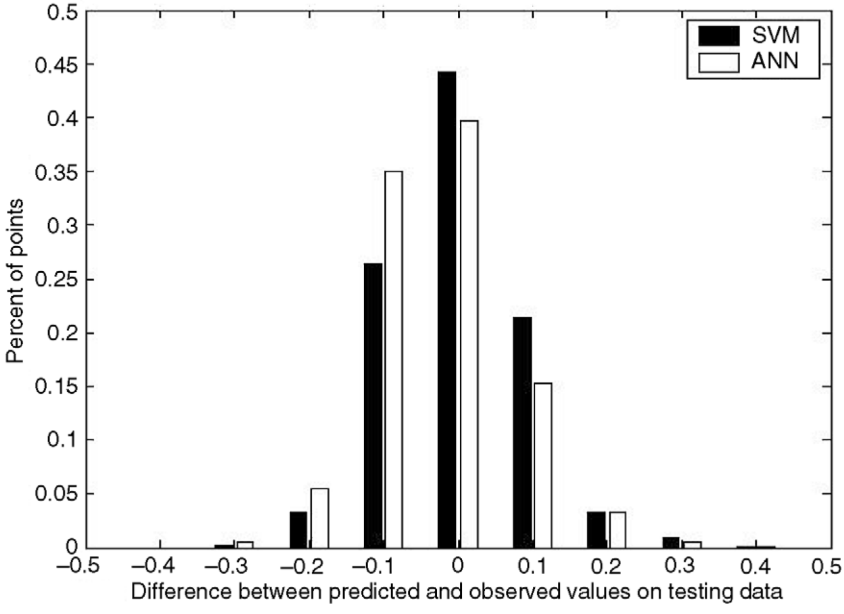

Fig. 3. Comparison between performance of SVM and MLP systems on testing data of order 11.

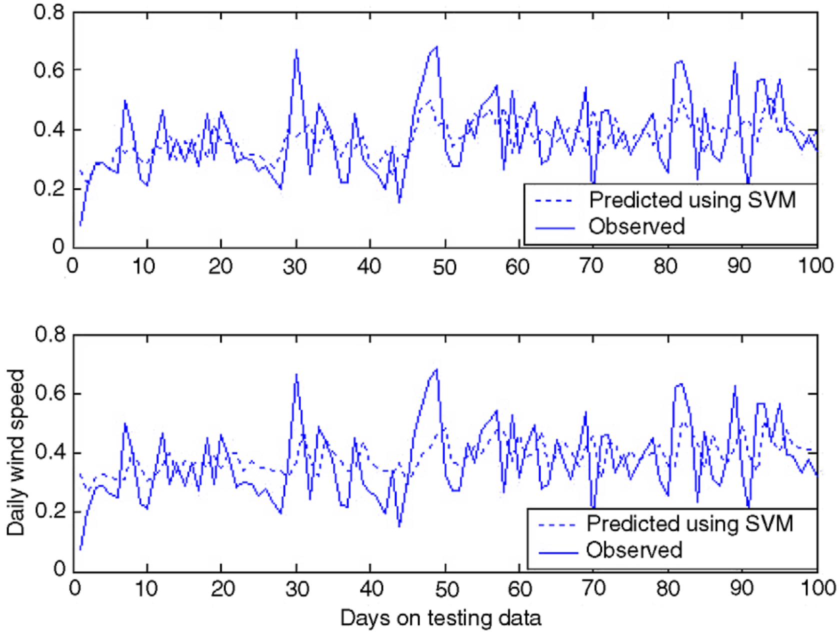

Fig. 4. Comparison between SVM and MLP on part of the testing data.

cates that SVM outperforms MLP in the sense of getting the trend of the data more closely in addition to obtaining a lower MSE. Moreover, in Ref. [3], it was shown that MLP significantly outperforms the classical auto regression model for wind speed prediction. This leads to the conclusion that SVM outperforms the AR model on such applications. Moreover, the computational complexity of neural network is only during training, a process that is done off line. After the training process is completed, the prediction process is comparable with the classical methods.

5. Conclusion

This paper introduces SVM for wind speed prediction. It compares favorably against the MLP for systems with orders 1–11. The parameters for both algorithms were optimized based on the performance on a cross-validation data set. The lowest MSE on testing data for the MLP is 0.0090 while it is 0.0078 for the SVM with data of order 11. In fact, SVM outperforms MLP for all systems with orders ranging between 1 and 11.

Acknowledgements

The authors would like to acknowledge the support of the King Fahd University of Petroleum and Minerals and King Abdulaziz City for Science and Technology.

References

[1] Bakhsh H, Srinivasan R, Bahel V. Colleration between hourly and global radiation for Dhahran, Saudi Arabia. Solar Energy 1985;2(1):59–61.
[2] Rehman S, Halawani TO. Statistical characteristics of wind in Saudi Arabia. Renewable Energy 1994;4(8):949–56.
[3] Mohandes M, Rehman S, Halawani TO. A neural networks approach for wind speed prediction. Renewable Energy 1998;13(3):345–54.
[4] Mohandes M, Rehman S, Halawani TO. Estimation of global solar radiation using artificial neural networks. Renewable Energy 1998;14(1–4):179–84.
[5] Mukherjee S, Osuna E, Girosi F. Nonlinear prediction of chaotic time series using support vector machines. Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Amelia Island, FL, September 1997, p. 511–20.
[6] Müller K-R, Smola A, Rätsch G, Schölkopf B, Kohlmorgen J, Vapnik V. Predicting time series with support vector machines. Advances in kernel methods—support vector learning. Cambridge, MA: MIT Press; 1999, p. 243–54.
[7] Cristiani N, Shaw-Taylor J. Support vector machines and other kernel-based learning methods. Cambridge University Press; 2000.
[8] Vapnik V. Statistical learning theory. Wiley; 1998.
[9] Smola AJ, Schölkopf B. A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
[10] Vapnik V, Golowich SE, Smola A. Support vector method for function approximation, regression estimation, and signal processing. Advances in neural information processing systems 8, San Mateo, CA, 1996.

[11] Collobert R, Bengio S. Support vector machines for large-scale regression problems. IDIAP-RR-00-17, August, 2000.
[12] Rumelhart E, McClelland JLP.D.P. Research Group. Parallel distributed processing. The MIT Press; 1986.
[13] Haykin S. Neural networks: a comprehensive foundation. Macmillan College Publishing; 1994.