Performance Evaluation of Short Term Wind Speed Prediction Techniques

1. K. SREELAKSHMI and 2. Dr. P. RAMAKANTH KUMAR

Assistant Professor, RV College of Engineering
Bangalore – 560059 Karnataka, INDIA

Professor, RV College of Engineering
Bangalore – 560059 Karnataka, INDIA

Summary – Wind speed prediction from past observations has applications in many diverse fields such as Target tracking, Missile guidance, Satellite launch, Air traffic control, Weather forecasting, Ship navigation, Electrical power generation using wind energy. The wind speed is determined by many other atmospheric variables, such as pressure, moisture content, humidity, rainfall etc. Literature reports a number of models for wind speed prediction. In this paper a comparison is carried out between Neural network models and Support Vector Machine models built for predicting wind speed in short term. Analysis shows that SVM models compute faster & give better accuracies than the Neural Network models.

Keywords — Short term wind speed prediction, Neural networks, Back propagation, Support Vector Machine [SVM], forecasting, hyper plane, kernels, classification.

1 Introduction

Development and analysis of Wind energy models helps in energy forecasting, planning, research and policy making. Wind speed prediction is needed as wind energy is discontinuous and it is also one of the main sources of alternative energy to the exhaustible petroleum resources [1]. Neural networks using back propagation algorithm and Support Vector machines are used in this work to model wind speed predictions.

2 Neural Networks

A feed-forward neural network has layers of processing elements, which makes independent computations on data that it receives and passes the results to another layer and finally, a subgroup of one or more processing elements determine the output from the network. Each processing element makes its computation based upon a weighted sum of its inputs. The first layer is always the input layer and the last layer is always the output layer. The layers placed between the first and the last layers are the hidden layers. The processing elements are seen as units that are similar to the neurons in a human brain, and hence, they are referred to as cells or artificial neurons.

![Basic Structure of a Neuron](image)

The output of any neuron is the result of thresholding, if any, of its internal activation, which, in turn, is the weighted sum of the neuron’s inputs. The Activation function denoted by \( \phi(v) \), defines the output of a neuron in terms of the induced local field \( v \). Here, Sigmoid function is used as the activation function. It is defined by the equation,

\[
\phi(v) = \frac{1}{1+\exp(-av)}
\]

(1)

Where ‘a’ is the slope parameter of the sigmoid function.
3 Support vector machines

Support vector machines (SVM) are a set of related supervised learning methods used for classification and regression. Their common factor is the use of a technique known as the "kernel trick" to apply linear classification techniques to non-linear classification problems. SVMs are based on the concept of decision planes that define decision boundaries. A decision plane is a boundary between a set of objects having different class memberships.

3.1 Linear SVM

This method helps in classifying some data points into two classes by a hyper-plane. The hyper plane is so chosen as to separate the data points "neatly", with maximum distance to the closest data point from both classes; this distance is called the margin. If such a hyper-plane exists, it is known as the maximum-margin hyper-plane or the optimal hyper-plane, as are the vectors that are closest to this hyper-plane, which are called the support vectors.

3.2 Non-linear SVM [18]

The original feature space can always be mapped to some higher-dimensional feature space where the training set is separable using Regression. The basic idea behind support vector machines (SVM) for regression is to map the data x into a high dimensional feature space through a nonlinear mapping. Once mapping is done then SVMs perform a linear regression in this feature space.

A Support Vector Machine (SVM) is an algorithm that uses a nonlinear mapping to transform the original training data into a higher dimension. This is a promising new method for the classification of both linear and non linear data. Within this new dimension, it searches for the linear optimal separating hyper-plane that is, a “decision boundary” separating the tuples of one class from another.

The advantages of the SVM method are as follows:

- Highly accurate
- Ability to model complex nonlinear decision boundaries
- Less prone to over fitting than other models
- Provides a compact description of the learned model
- Can be used for prediction as well as classification.

Owing to the innumerable plus points of the SVM method, it has a wide range of applications. SVM have been applied to a number of areas, including handwritten digit recognition, object recognition, speaker identification and also, benchmark time series prediction tests.

4 Literature review

The literature reports a number of Statistical, Intelligent systems, Time series, Fuzzy logic, neural network methods for wind speed estimation from meteorological data as inputs, topological data as inputs, wind turbine data as inputs. They are usually suited for long term predictions, need more computational time and lack in accuracy. Also adaptation to fast changes is not inbuilt into these models. They are based on non-statistical approaches and depend on the experience of a meteorologist.

Time series models are based on historical wind data and statistical methods, Example: ARMA model, which is also called Persistence model [2]. Time series involves number of equations with many variables and hence require intensive computations.

Fuzzy models are also used to estimate wind speed. They are found to be more efficient than the conventional ARMA models [3].

Regression techniques are found to be less efficient compared to Artificial Neural Network model (ANN) models [4].
Kalman filter models are found to be 10% better than the ARMA. These models are found to be superior to ARMA [5].

Artificial Neural Network models are best suited for wind speed prediction applications as they do not require mathematical models and adapts automatically to changes in the inputs to minimize mean square errors. They have the capability to deal with large data sets. [6].

The contemporary models use Back propagation algorithm [7], Radial basis functions [8], Wavelet techniques and Support vector machines for short term forecasting [9].

SVMs [16] are discriminative classifiers based on Vapnik’s structural risk minimization principle. They can implement flexible decision boundaries in high dimensional feature spaces. The implicit regularization of the classifier’s complexity avoids over fitting and mostly this leads to good generalizations. Some further properties are commonly seen as reasons for the success of SVMs in real-world problems: The optimality of the training result is guaranteed, fast training algorithms exist and little a-priori knowledge is required, i.e. only a labeled training set. Further, SVM models are found to take less computational times compared to ANN models.

Support Vector Machines [SVMs] have achieved excellent recognition results in various pattern recognition applications [13, 14]. Also in off-line optical character recognition (OCR), they have been shown to be comparable or even superior to the standard techniques like Bayesian classifiers or multilayer perceptrons [15].

Support Vector Machines (SVM), the latest neural network algorithms, are introduced to wind speed prediction and their performance is compared with the multilayer perceptron (MLP) neural networks, for modeling mean daily wind speed of Madina city, Saudi Arabia. It is concluded that SVM compare favorably with the MLP model based on the root mean square errors between the actual and the predicted data [19].

Support vector machine models are shown to be more effective than Kalman filter based models for wind speed prediction [20].

5 Methodology

Short term wind speed prediction involves the following steps:

a) Data Acquisition & Pre-processing
b) Data Conversion & Normalization
c) Statistical Analysis
d) Design of Neural Network / Support Vector Machine Model
e) Training
f) Validation
g) Testing

Here, weather station data of 10 years is considered for experiments. Data input is read, filtered and normalized. Patterns are generated and statistical correlation is carried out. A large part of the data is fed into the training network and the remaining part is used for validation and testing.

The parameters that are considered as inputs to the model are listed in Table 1 and a sample input data file is shown in Table 2.

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean temperature</td>
<td>Deg.C</td>
</tr>
<tr>
<td>Humidity</td>
<td>%RH</td>
</tr>
<tr>
<td>Wind gust</td>
<td>m/s</td>
</tr>
<tr>
<td>Wind direction</td>
<td>Deg.M</td>
</tr>
<tr>
<td>Barometric pressure</td>
<td>MB</td>
</tr>
<tr>
<td>Wind speed</td>
<td>m/s</td>
</tr>
</tbody>
</table>

Statistical analysis and correlation is carried out using Spearman’s correlation technique. It is observed that the data is symmetrical with respect to the main diagonal, as shown in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>Correlation</th>
<th>Mean temp</th>
<th>Humidity</th>
<th>Wind gust</th>
<th>Wind dir</th>
<th>Error pre</th>
<th>Wind speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean temp</td>
<td>0.6982</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>-0.9895</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind gust</td>
<td>-0.2580</td>
<td>1</td>
<td>0.8365</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind dir</td>
<td>0.1020</td>
<td>-0.0848</td>
<td>0.8207</td>
<td>-0.0848</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error pre</td>
<td>0.1020</td>
<td>0.8482</td>
<td>0.0949</td>
<td>0.0949</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind speed</td>
<td>0.1020</td>
<td>-0.0848</td>
<td>0.8207</td>
<td>-0.0848</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
5.1 Design of neural network

The design of the neural network involves designing the three fields of neurons: one for input neurons, one for hidden processing elements, and one for the output neurons. The connections are for the feed forward activity. There are connections from every neuron in field 1 to every one in field 2, and in turn, from every neuron in field 2 to every neuron in field 3. Thus, there are two sets of weights, those figuring in the activations of hidden layer neurons, and those that help determine the output neuron activations. Using back propagation algorithm, in each training set, the weights are modified so as to reduce the mean squared error \([\text{MSE}]\) between the network’s prediction and the actual target value. These modifications are made in the reverse direction, from the output layer, through each hidden layer down to the first hidden layer, till the terminating condition is reached.

The steps in the algorithm are:
- Initialize the weights
- Propagate the inputs forward
- Back propagate the error
- Terminating condition

5.1.1 Testing

The remaining input values are utilized for testing and validation wherein the wind speed is predicted for the test input and compared with the actual values. The setup giving minimum error is obtained by varying the number of hidden layers and other parameters such as learning parameter, number of hidden layers in the network, number of epochs for test, error tolerance, number of neurons in each layer, etc.

5.2 Results of Back propagation tests

Table 4 Test Specifications for 3 output layers

<table>
<thead>
<tr>
<th>SI No. of test case</th>
<th>Name of test</th>
<th>Error Tolerance</th>
<th>Learning parameter</th>
<th>Number of Epochs</th>
<th>Number of layers</th>
<th>Number of Neurons in each layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Back propagation Test 1</td>
<td>0.001</td>
<td>0.01</td>
<td>50</td>
<td>3</td>
<td>5 3 1</td>
</tr>
</tbody>
</table>

Table 5 Test Specifications for 4 output layers

<table>
<thead>
<tr>
<th>SI No. of test case</th>
<th>Name of test</th>
<th>Error Tolerance</th>
<th>Learning parameter</th>
<th>Number of Epochs</th>
<th>Number of layers</th>
<th>Number of Neurons in each layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Back propagation Test 2</td>
<td>0.001</td>
<td>0.02</td>
<td>100</td>
<td>4</td>
<td>5 5 5 1</td>
</tr>
</tbody>
</table>
### Table 6: Test Specifications for 5 output layers and 50 epochs

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>Error Tolerance</th>
<th>Learning parameter</th>
<th>Number of Epochs</th>
<th>Number of layers</th>
<th>Number of Neurons in each layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Back propagation</td>
<td>0.01</td>
<td>0.05</td>
<td>50</td>
<td>5</td>
<td>5 4 3 2 1</td>
</tr>
</tbody>
</table>

![Wind speed vs Input Patterns](image1)

**Fig. 6 Expected v/s predicted outputs for 5 layers**

### Table 7: Test Specifications for 5 output layers and 100 epochs

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>Error Tolerance</th>
<th>Learning parameter</th>
<th>Number of Epochs</th>
<th>Number of layers</th>
<th>Number of Neurons in each layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Backpropagation</td>
<td>0.01</td>
<td>0.1</td>
<td>100</td>
<td>5</td>
<td>5 4 3 2 1</td>
</tr>
</tbody>
</table>

![Wind speed vs Input Patterns](image2)

**Fig. 7 Expected v/s predicted outputs for 5 layers**

### Table 8: Test Specifications for 5 output layers and 200 epochs

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>Error Tolerance</th>
<th>Learning parameter</th>
<th>Number of Epochs</th>
<th>Number of layers</th>
<th>Number of Neurons in each layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Backpropagation</td>
<td>0.002</td>
<td>0.05</td>
<td>200</td>
<td>5</td>
<td>5 4 3 2 1</td>
</tr>
</tbody>
</table>

![Wind speed vs Input Patterns](image3)

**Fig. 8 Expected v/s predicted outputs for 5 layers**

### 5.3 SVM model methodology

SVM stands for Support Vector Machines. It is an elegant and highly principled learning method for a feed forward network design with a single hidden layer of units that are non linear. Its deviation follows the principle of structural risk minimization which is based on the fact that the error rate of the learning machine on test data is bounded by sum of training error rate. As the name suggests, the design of the machine hinges on the extraction of a subset of the training data that serves as support vectors and hence, represents a stable data characteristic. SVM includes the polynomial learning machine, radial-basis function network and two layer perceptron as special instances. These methods provide different representations of intrinsic statistical regularities contained in the training set.

The prerequisites include the three files train.txt, validation.txt and test.txt for training, validation and testing respectively for which the data set is divided into three sets (in the ratio 50%, 25%, 25%). The first set will be used for training. The second set will be used to ascertain the correct kernel and setting to use when
performing prediction. The final set of data will be used in prediction.

5.4 SVM Test Cases and results

The following tables and graphs show the tests and results of experiments carried out in using SVM.

Table 4 Test specifications for polynomial kernel

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>C</th>
<th>Epsilon</th>
<th>Kernel Type</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SVM Test 1</td>
<td>2</td>
<td>0.1</td>
<td>Polynomial</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 9 Outputs for polynomial kernel

Table 5 Test specifications for radial kernel

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>C</th>
<th>Epsilon</th>
<th>Kernel Type</th>
<th>Gamma</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SVM Test 2</td>
<td>1</td>
<td>0.01</td>
<td>Radial</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Fig. 10 Outputs for radial kernel

Table 6 Test specifications for neural kernel

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>C</th>
<th>Epsilon</th>
<th>Kernel Type</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>SVM Test 3</td>
<td>1</td>
<td>0.05</td>
<td>neural</td>
<td>0.02</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Fig. 11 Outputs for Neural kernel

Table 7 Test specifications for Anova kernel

<table>
<thead>
<tr>
<th>Sl No. of test case</th>
<th>Name of test</th>
<th>C</th>
<th>Epsilon</th>
<th>Kernel Type</th>
<th>Gamma</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>SVM Test 4</td>
<td>2</td>
<td>0.02</td>
<td>Anova</td>
<td>0.005</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 12 Outputs for Anova kernel
6 Conclusion

The literatures available for wind speed modeling reveals that majority of the models are being utilized for electrical power demand forecasting. Though many short term models are presented, the accuracy of the models still need to be improved.

In the ANN models using back propagation models, the predicted wind speed is dependent on Error tolerance, learning parameter, No. of epochs, No. of input layers, and No. of neurons in each layer. Wind speed variation predicted is seen to closely follow the expected wind speed. A neural network with 5 input layers, with 5,4,3,2,1 neurons in the layers, with a learning parameter of 0.05 and error tolerance of 0.002 is found to give a minimum error in prediction of 5%, giving a computational accuracy of 95%.

In the wind speed forecasting carried out using Support Vector Machine model, the predicted values are fairly matching the expected values. The results of SVM model predictions are almost following the expected pattern of wind speed variations. Analysis shows that the errors in predictions vary with the type of kernel, values of C, Epsilon and the No. of input patterns. Further the error percentage is lowest for 6 input patterns and highest for 5 input patterns, with all the kernels. The dot kernel with a C=2, epsilon = 0.005 seems to be the optimum as it results in a maximum error of 6.49% and minimum error of 2.94% , for 5 and 6 input patterns respectively. This leads to a computational accuracy of 93.51% at 5 input patterns and 97.06% at 6 input patterns.

Comparing the SVMs with ANNs, we see that they almost give comparable accuracies, as in Fig. 14. This is also supported by the comparison reported in literature [9]. SVM models are found to take less computational times compared to Artificial Neural Network models, using back propagation algorithms.

References


Biographies

The first author is K. Sreelakshmi, Assistant Professor, Dept of Telecommunication Engg., and RV College of engineering Bangalore – 560059. She obtained her BE and ME degrees from Bangalore university, with specializations in Electronics and Communication and is currently pursuing her PhD degree. Her research interests are in the field of Neural network based models and their applications, Electromagnetics and microwaves. She is a member of IETE and ISTE. She has worked in Indian Institute of Science, Bangalore as a JRF. Her work at IISc included characterization of electronic devices at cryogenic temperature, which has resulted in an international paper publication in the Journal, “Cryogenics”.

The second author is Dr.Ramakanthkumar P Bangalore 25/01/1970. He completed his B.E in Computer Science and Engineering from S.J.College of Engineering, Mysore, and M.S in Software Systems from BITS, Pilani. He completed his Ph.D. from Mangalore University in the area of Pattern Recognition. He has authored more than five publications in reputed journals. His areas of interest are Image Processing, Pattern Recognition and Natural Language Processing. He is a member of the Computer Society of India (CSI) and a life member of Indian Society for Technical Education (ISTE). He has completed number research and consultancy projects for DRDO organizations.