Local Short-Term Prediction of Wind Speed: A Neural Network Analysis

C. Pérez-Llera[a], **M.C. Fernández-Baizán**[b], **J.L. Feito**[c] **and V. González del Valle**[a]

[a]*Dpt. Informática, Universidad de Oviedo, Campus Viesques, E-33271 Gijón, Spain*
cpllera@etsiig.uniovi.es

[b]*Dpt. Lenguajes y Sistemas Informáticos e Ingeniería del Software, Universidad Politécnica de Madrid, Campus Montegancedo, E-28660 Boadilla del Monte, Spain*

[c]*Dpt. Dirección y Economía de la Empresa, Universidad de León, Campus Vegazana, E-24071 León, Spain*

Abstract: Predicting short-term wind speed is essential in order to model a system of prevention of environmental contamination produced by the effects of strong winds acting on goods (mainly crushed coal) discharged at a dock. The wind speed in a near future depends on the values of other meteorological variables in previous times. The values are obtained from a meteorological station with several sensors: wind speed, temperature, humidity, pressure ... We have used the SNNS simulator to obtain a neural network able to predict the wind speed 20 min. in advance, with the minimum possible error. The network inputs are basically historical values of the predicted variable as well as a number of other support variables. A feed-forward model has been elected with the aim of carrying out the treatment of the data. The algorithm used for the training phase has been back-propagation.

Keywords: Short-term wind speed prediction; neural network; time-series; weather forecasting; environmental contamination

1. INTRODUCTION

Few studies there are in the literature on very short-range local prediction of wind speed, based in non-statistical paradigms. In particular, concerning to the meteorological variables that influence in a more decisive way. The forecasting methods in the literature have a number of inconveniences: 1) They rely on the knowledge and experience of a meteorologist. 2) They are carried out by interpolating measured data from various sources for extensive areas and long terms of time; instead of belting the problem to a source, a geographical point and an immediate period.

The terminal in Port of Gijón-Spain, European Bulk Handling Installation S.A. (EBHISA), offers facilities for unloading, storage of bulk cargoes. EBHISA takes special care to ensure that the operations do not contaminate the atmosphere. In order to avoid the environmental contamination produced during the loading/unloading operations of goods (mainly coal), accurate forecast of wind speed is crucial. In this work, neural network methodology is applied to construct a meteorological forecasting tool which predicts the local short-term wind speed. Due to the numeric character of the data, a statistical and a neural networks analysis of time series are carried out. The advantages of neural networks method over other prediction methods such as ARIMA, Kalman filters, exponential smoothing or systems of learning based on rules, are: 1) it is an entirely numeric method, as opposed to the implicit symbolism in the systems of rules, 2) It does not demands a previous knowledge of the system that we wish to predict and it is characterised by their robustness and tolerance to noise.

The well-known disadvantages of neural networks method are: 1) it does not contributes knowledge about the background of the problem, while rules based systems are more easily comprehensible; 2) in the majority of cases it does not permits an incremental training.

2. THE INFORMATION SYSTEM

A meteorological station sends the values of 12 meteorological variables to a Personal Computer each 2 min (Table 1).

Channel	Meteorological Variable	Unit
1	Wind Speed	m/s
2	Gust Wind	m/s
3	Wind Direction	Deg. M
4	Air Temperature	Deg. C
5	Relative Humidity	%RH
6	Air Pressure	mb
7	Visibility	Km.
8	Sunshine Duration	Min.
9	Net Atmospheric Radiation	W/m^2
10	Rainfall	mm
11	Solar Radiation	W/m^2
12	Water Temperature	Deg. C

Table 1. Meteorological variables

3. DATA PRE-PROCESSING

In order to carry out the study of the time series, we used a neural networks tool: SNNS (Sttugart Neural Network Simulator), due to its ease of use and wide availability. The simulator needs to receive the training and validation patterns in text format, for this reason we implemented a program that allows: 1) Data exportation present from the Paradox table to an ASCII file. 2) Fusion of a certain record set of the table into a single pattern. Since the number of variables is high, a statistical study of the variables is carried out as a previous step to the neuronal analysis of the time series, specifically: test of lineal correlation, taking the variables two by two, and test of the Spearman rank correlation.

3.1. DATACONV: Data conversion and pattern generation

The developed program allows two operation modes: 1) Data conversion, used to export the data in a Paradox table (Table 1) to an ASCII file, 2) generation of patterns for the SNNS, in this mode, the user may modify the following parameters: 1) The number of records per pattern of the Paradox table that will form the *time window* of inputs of the neural network. In prediction problems, the input of the net is usually made up of values of the variables to predict (and other variables) at previous moments. These values are grouped in a *time window*, 2) The distance of the output record from the record that is used to carry out the prediction to the last record of the input *time window*, 3) The list of the decision channels that will be the variables to predict, 4) The list of channels to include in the patterns, in this way, the unnecessary variables in the *time windows* may be discarded, 5) The percentage of validation patterns that will be used for validation after the neural network training. The available patterns are divided into two groups: *training patterns*, that constitute the neural network inputs during the training phase; and *validation patterns*, which are used to verify the generalisation of the neural network after the training phase. There is the possibility of normalising and scaling the data in the obtained patterns. The parameters that the user may select are: 1) The maximum output and minimum output values that could appears in the output generated (in the patterns), 2) The maximum input and minimum input values that could appear under normal conditions per channel. The data is normalised using a lineal scaling function applied to the output range selected by the user. One could work with data without pre-processing (raw data), simply eliminating the activation function of the output neurones. This function identity is supposed to be used exclusively for the output neurones, since in the case of being used with the neurones of the hidden layers, the neural network would become a mere lineal regression and the net would lose the property of non-linearity (the best property of the net).

3.2. Statistical Analysis

3.2.1. Test of lineal correlation

The coefficient is calculated using the following formula:

$$r = \frac{\overline{x.y} - \bar{x}.\bar{y}}{\sqrt{\left(\overline{x^2} - (\bar{x})^2\right)\left(\overline{y^2} - (\bar{y})^2\right)}} = \frac{S_{xy}}{\sqrt{S_{x^2} S_{y^2}}} \qquad (1)$$

The lineal correlation between all the variables was studied taking them two by two, and the result is shown in Table 2. The table is symmetrical with regards to the main diagonal. The shaded cells in the Table 2 represent values that indicate a possible lineal correlation between the variables.

3.2.2. Test of Spearman's correlation

An additional lineal correlation test has been used: Spearman's rank-order coefficient (or non-parametric correlation). Non-parametric statistics work with ordinal variables. In the calculation of Spearman's coefficient the data should be ordered with the purpose of determining the order that corresponds to each value inside the sample. The formula to apply is:

$$r_s = \frac{\sum_{i=1}^{n} \left(u_i - \bar{u}\right)\left(v_i - \bar{v}\right)}{\sqrt{\sum_{i=1}^{n} \left(u_i - \bar{u}\right)^2 \cdot \sum_{i=1}^{n} \left(v_i - \bar{v}\right)^2}} \qquad (2)$$

Where, u_i = Range of the i^{th} element of the variable U and v_i = Range of the i^{th} element of the variable V.

The results obtained are shown in Table 3. We observed: 1) A good correlation between the following pairs of variables: Wind Speed/Wind Gust, Sunshine Duration/Net Atmospheric Radiation, Sunshine Duration/Solar Radiation and Net Atmospheric Radiation/Solar Radiation , 2) The visibility channel hardly presents any variation and therefore it will be eliminated.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	**0.957**	0.319	-0.015	-0.419	-0.146	0.039	0.119	0.182	0.040	0.172	-0.046
2		1	0.413	-0.007	-0.432	-0.172	0.035	0.123	0.199	0.068	0.174	-0.034
3			1	0.008	-0.208	-0.201	-0.034	-0.030	0.056	0.145	-0.011	-0.083
4				1	-0.208	0.017	-0.008	0.177	0.188	-0.092	0.267	-0.184
5					1	0.111	-0.067	-0.292	-0.144	0.216	-0.233	-0.107
6						1	-0.002	0.083	-0.118	-0.079	-0.053	0.381
7							1	0.007	0.007	-0.071	0.011	0.011
8								1	**0.652**	-0.093	**0.692**	0.172
9									1	0.087	**0.900**	0.107
10										1	-0.018	-0.012
11											1	0.089
12												1

Table 3. Spearman's rank-order coefficients. Size of the sample: 5,968 elements.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	**0.895**	0.046	0.004	-0.360	-0.034	0.038	0.114	0.179	0.017	0.153	-0.002
2		1	0.181	0.048	-0.416	-0.054	0.038	0.123	0.187	0.057	0.153	0.015
3			1	-0.059	-0.126	-0.084	-0.030	-0.113	-0.127	0.089	-0.160	-0.042
4				1	-0.271	0.027	-0.006	0.171	0.210	-0.043	0.221	-0.244
5					1	0.049	-0.054	-0.273	-0.219	0.139	-0.246	-0.172
6						1	-0.004	0.045	0.017	-0.020	0.031	0.126
7							1	0.014	0.014	-0.015	0.012	0.015
8								1	**0.741**	-0.060	**0.775**	0.164
9									1	-0.029	**0.987**	0.184
10										1	-0.056	-0.018
11											1	0.196
12												1

Table 2. Coefficients of lineal correlation. Size of the sample: 5,968 elements.

4. NEURAL NETWORKS RESULTS

Starting from a group of patterns generated by DATACONV, we used the SNNS 4.0 simulator to obtain the neural network able to predict the wind speed 20 min in advance, with the minimum possible error. A *feedforward network* model was used due to its prestige and capacity to solve large amounts of problems. The algorithm used for the training was *backpropagation*. Specifically, three-layer networks were used. The global set of patterns is divided into two randomly selected groups: the training group, corresponding to 90% of the patterns, and the validation group, corresponding to 10% of the patterns; so that the generalisation capacity of the network could be checked after the training phase. In order to check the goodness of the previous training in the validation phase, we used the Mean Squared Error (MSE) as a measure of the error made by the neural network. We used this measure because the SNSS gives us an indication of the evolution of the MSE value of the network for the training patterns. So we were able to appropriately compare the error obtained for the training set and the validation set. The MSE definition that we used is:

$$MSE = \frac{\sum_{i=1}^{n} (o_i - p_i)^2}{n - p} \quad (3)$$

That is, MSE is the sum of the squares of pattern errors divided by the total number of patterns apart from the number of free parameters of the network (i.e. the number of connections between neurones).

4.1. Test 1

In the first test we used almost all the data as input to the network, because the only knowledge that we have about the problem is what we obtained from the prior study phase of the data. Due to the aforementioned reason, only the data relative to the wind direction (in principle, the most chaotic variable), the temperature of the water (during a great part of the period of data capture the plumbing was disconnected or not working properly) and visibility (we have shown in the prior study phase that this is practically invariable) was discarded. From the correlated variables we selected only one representative, and we thus eliminated the historical data relative to wind gust, sunshine duration and solar radiation in the input to the network. In short, the inputs to the net are formed by the values of the variables corresponding to wind speed, air temperature, relative humidity, air pressure, atmospheric radiation and rain. We took values at two instants of time for each of the aforementioned variables. Therefore, the number of inputs to the net in this phase is equal to 12. In order to check the most appropriate parameters for prediction, we carried out a sweeping in the number of neurones of the hidden layer as an initial test. In this way, we trained different networks, varying the number of neurones of the hidden layer between 6 - 40. The results obtained are presented in Table 4.

	Training			**Validation**	
Hidden Neurones	**SSE**	**MSE**	**Time**	**SSE**	**MSE**
6	4.20492	0.00079	621	0.40401	0.00080
7	4.13211	0.00079	711	0.41778	0.00085
8	4.18084	0.00079	800	0.40358	0.00084
9	4.16087	0.00078	878	0.40238	0.00087
10	4.19372	0.00079	968	0.40465	0.00090
11	4.18435	0.00079	1035	0.40323	0.00092
…..	….	…	…	…	…
19	4.16787	0.00078	1761	0.40428	0.00125
20	4.19560	0.00079	1899	0.40043	0.00129
…..					
40	4.17460	0.00078	3708	0.39885	0.01329

Table 4. Results of Test 1

In this test, the weights were initialised for each of the networks with random values within the range [-0.5, 0.5] and the number of iterations that were carried out was 5,000. We used 5,319 training patterns and 591 validation patterns. It can be observed that the successive increase in the number of neurones in the hidden layer hardly diminishes the training error, and also that the validation error increases considerably from 8 or 10 hidden neurones. This phenomenon is known as *overfitting*, i.e. the better the fitting of the error of the training patterns, the worse is the capacity of generalisation. The training time it is approximately lineal and depends on the number of neurones of the hidden layer.

4.2. Test 2

In this second test we tried to analyse the effect of the number of iterations of the learning algorithm. We maintained the same structure for the network as in the previous test and checked the results obtained after 1,000 iterations, after 2,000 iterations, and so forth. The obtained results are shown in Table 5:

Number of Iterations	**MSE Training**	**MSE Validation**
1,000	0.00084	0.00097
2,000	0.00081	0.00097
3,000	0.00080	0.00096
4,000	0.00080	0.00097
5,000	0.00079	0.00097
6,000	0.00079	0.00107
7,000	0.00078	0.00096
….	….	….
45,000	0.00064	0.00092
50,000	0.00064	0.00091
60,000	0.00063	0.00093
70,000	0.00063	0.00090
80,000	0.00063	0.00090
90,000	0.00063	0.00091
100,000	0.00062	0.00091

Table 5. Results of Test 2

A network with 12 neurones in the hidden layer was used for this test. We observed that the number of iterations has less influence on the obtained error than the number of neurones in the hidden layer. Since although the training error could be appreciably diminished when increasing the number of iterations, the same does not happen in the validation phase, in which the error remains more stable throughout the experiment than in the case shown in the previous test.

4.3. Test 3

At this point, it begins to be interesting to check the efficacy of the network with a lower number of inputs. Will the network be able to obtain similar results to those obtained until now?. In order to carry out this objective, a new group of patterns is obtained. This time, each pattern will be formed by the values of the variables: wind speed, air temperature and atmospheric pressure, in 3 serial instants. Therefore, the number of inputs to the network decreases from 12 to 9 and the *time window*, on which the prediction is based, also increases. A single training of the network was carried out, in which 5,000 iterations were made and 12 neurones were used in the hidden layer. The MSEs were 0.00082 and 0.00068 for the training and validation phases, respectively. As we do not obtain a greater error on reducing the number of inputs (channels) to the network, we will try in the following tests to reduce this number as far as possible without losing either efficacy or generality.

4.4. Test 4

On our objective towards the reduction of the number of inputs, we began by discarding the values of the air temperature and atmospheric pressure in the first two time instants of the model used in the previous test. The results obtained with the variation of the number of hidden neurones are shown in Table 6.

Hidden Neurones	MSE Training	MSE Validation
6	0.00082	0.00065
8	0.00081	0.00066
10	0.00081	0.00068

Table 6. Results of Test 4

It is proven that with the described inputs, the network has the same power of prediction as the networks explained in the previous tests. Therefore, we could decrease the number of inputs at least to the number that we have indicated in this test.

4.5. Test 5

This test attempts once more to reduce the number of neurones of the hidden layer even more. To do this, we used just two time values of the wind speed variable and the value at the last instant previous to the prediction of the atmospheric pressure variable. The results obtained are shown in Table 7.

Hidden Neurones	MSE training	MSE validation
6	0.00084	0.00056
8	0.00084	0.00057

Table 7. Results of Test 5

A slight degradation in the training error for the proposed network can be observed. However, the validation error is the best one obtained by far regarding all the networks studied during the phase of experimentation.

4.6. Test 6

Due to the final result presented as a conclusion of the previous test and for reasons of greater security, it is interesting to study if an increase in the number of inputs could improve the efficacy of the network. With this purpose in mind, we increased the input layer with an additional variable, corresponding to the air temperature at the last time instant used for the prediction. The input layer presents the following structure:

$$\textit{Wind Speed}_{t-1}, \textit{Wind Speed}_t, \textit{Atmospheric Pressure}_t, \textit{Air Temperature}_t$$

In this case, the error obtained in the training phase is slightly improved. However, the error of the validation phase increases once more.

5. DISCUSSION OF THE RESULTS

We selected, after the previous experiments, the following net model for wind speed prediction: a *feedforward network* with 3 inputs, 6 neurones in the hidden layer and 1 output. We obtained the best results, with regard to capacity of generalisation, with the above topology. Also, due to the limited number of connections, the training times are the shortest of all those tested. For the selected network the study of the errors is:

MSE	0,00056
Maximum error (absolute value)	0.1210
Median error	0.0165
Variance	0.000261

The errors to which this data refers consist of a group of 591 patterns used for validation and that, therefore, were not presented to the network during the training process. The values correspond to the difference between the normalised real value and the normalised predicted value of the wind. After denormalising the data, we obtained:

Maximum Error (absolute value)	6.0505
Median error	0.8262
Variance	0.0130

It can be observed that the maximum error of the prediction for the validation set is quite high (6 m./s. ≈ 22 Km./h.). This phenomenon was repeated for all the studied topologies. On the other hand, the obtained mean error is more than acceptable, being less than 1 m/s. Moreover, such a small variance value seems to indicate that the errors have to a great degree grouped around the mean value. In order to check this, we carried out a study of the percentage of validation patterns whose error (in absolute values) is less than or equal to 1m/s; the resulting percentage being 71.07%. If we increase the margin of error to 2 m/s, the percentage of patterns that fulfil this rises to 92.92%. We present a comparative graph (Figure 1) of the wind speed for 50 validation patterns, which gives the real values versus the predicted values.

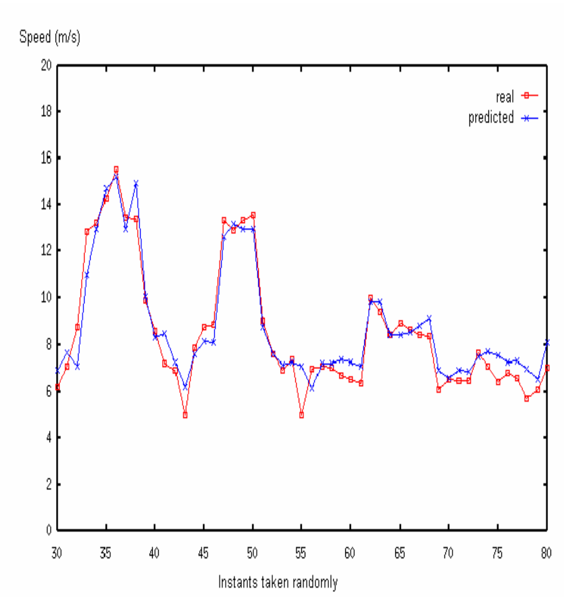

Figure 1. Real values versus predicted values of the wind speed

Likewise, we observe a factor that could negatively affect this good quality: the values of the patterns reside in a not too wide range, between 0 and 20; however, normalisation allows values of the wind speed between 0 and 40.

6. CONCLUSIONS

In these tests, almost all the data was used as input to the network, since the only knowledge that we have concerning the problem is that obtained in the statistical analysis phase. Thus, the only data discarded was that relative to: wind direction (the most chaotic variable), water temperature (the sensor was working incorrectly) and visibility (previous statistical analysis). We selected a representative set among the sets of correlated variables. The inputs of the network that we have left are: Wind Speed, Air Temperature, Relative Humidity, Air Pressure, Net Atmospheric Radiation and Rainfall. We took the values at two instants for each of the variables. The net has a single output, which is the wind speed in a near future. In spite of the small amount of data available up until the moment, a ratio of acceptably low prediction errors was obtained using neural networks. As a result of the tests that were carried out, we have reached the conclusion that in order to make a prediction with an adequate quality, it is sufficient to use the values of the following channels at the previous moment: Wind Speed, Air Temperature, Relative Humidity, Air Pressure, Net Atmospheric Radiation, Rain; and at the current moment: Wind Speed, Air Temperature, Relative Humidity, Air Pressure, Net Atmospheric Radiation and Rainfall. Our preliminary results have clearly indicated the feasibility of our approach.

7. ACKNOWLEGMENTS

Support for this work was funded by University of Oviedo and E.B.H.I. S.A., under the LIFE Environmental Research Project of the European Union 96-E-268.

8. REFERENCES

Brand, S., Englebretson R., Gilmore, R., Mediterranean ports severe weather research at the Naval Research Laboratory, *Meteor. Appl.*, 3 (3), 211-214, 1996.

Deidda, R., Marrocu, M., Speranza, A., Feasibility study of a meteorological prediction model for ESO Observatories in Chile, Universidad de Camerino, Italia, 1997.

Kuciauskas, A. P., Brody, L.R., Hadjimichael, M., Bankert, R.L., Tag, P.M., A fuzzy expert system to assist in the prediction of hazardous wind conditions within the Mediterranean basin, *Meteor. Appl*., 1998.

Kuciauskas, A., Brody, L., Bankert, R., Tag, P., MEDEX: A fuzzy system for forecasting Mediterranean gale force winds, FUZZIEEE 96 Conference on Fuzzy Systems, New Orleans LA, 529-534, 1996.

Kuciauskas, A., Brody, L., Bankert, R., Tag, P., Hadjimichael, M., Automated forecasting of gale force winds in the Mediterranean region, 15th Conference on Weather Analysis and Forecasting, Norfolk VA., *Amer. Meteor. Soc*., 358-361, 1996.

Tag, P.M., Hadjimichael, M., Brody, L.R., Kuciauskas, A.P., Automating the subjective recognition of 50 MB Wind Patterns as Input a meteorological Forecasting System, 15th Conference Weather Analysis and Forecasting, Norfolk VA, Amer. Meteor. Soc., 347-350, 1996.