Artificial neural networks wind forecasts for safety at sea and yacht racing tactics

ARTIFICIAL NEURAL NETWORKS WIND FORECASTS FOR SAFETY AT SEA AND YACHT RACING TACTICS

Francesca TAGLIAFERRI[1,2], Ignazio Maria VIOLA[1,3], Robert J DOW[1,4]

[1] Yacht and Superyacht Research Group, School of Marine Science and Technology, Newcastle University, UK
[2] f.tagliaferri@ncl.ac.uk
[3] ignazio.viola@ncl.ac.uk (corresponding author)
[4] bob.dow@ncl.ac.uk

ABSTRACT

Producing accurate and reliable wind forecasts can be very helpful during offshore navigation, both for on board safety and for taking tactical decisions during sailing yacht races. In this work different models based on artificial neural networks are used for short-term wind forecasts. A time-series approach is used, and the forecast is based on past values of wind velocity. The wind velocities measured at previous instants are input of the proposed algorithm, which predicts the velocities for future instants. The peculiarity of this method is that no other physical values are needed to obtain the forecast. A computer program was implemented and tested on different time series: daily-averages, ten-minutes and three-second measurements. Using daily-averaged data, the algorithm was able to accurately forecast when the wind was going to increase or decrease for the following day. Ten-minute data allowed forecasting up to 4 steps ahead with good accuracy, while three-second data allowed accurately forecast up to 20 steps ahead, subjected to an adequate training of the network. Also, updating the network with real-time velocity measurements, an iterative algorithm was achieved allowing a continuous forecast.

1. INTRODUCTION

The decisions that are taken during navigation are highly influenced by currently experienced and anticipated weather and sea conditions. In some cases, the main factor influencing those decisions is the wind. A key example is given by sailing competitions, where anticipating a gust can determine or influence the decision for making a sudden tack, and the strategy during a competition is based on the anticipated changes in wind direction and speed. Even for non-competitive sailing, wind is the most influencing factor on routing and can be extremely important when facing extreme weather conditions that may require emergency procedures. In recent years a large number of studies on wind forecasting have been carried out, most of them driven by the growing interest in renewable energy and, in particular, wind energy production (see for inst. Masson, 2000; Kavasseri and Seetharaman, 2009; Foley *et al*, 2012). The aim of those studies was mainly to forecast the level of production of energy based on historically recorded values for wind speed. Studies aiming at forecasting the wind speed instead of the actual energy production, can be divided into those based on several velocity measurements taken at a single location, and those based on a spatial velocity distribution. The latter uses physical considerations about the coherent structures within the atmospheric boundary layer. Conversely, the former studies do not take into account any physical consideration about the nature of the flow field. Some of these methods were based on least squares optimisation of polynomial functions (Zhang *et al*, 2011), on non-linear methods based on Kalman filters (Louka *et al*, 2008) and on artificial neural networks (More and Deo, 2003; Khatib and Al-Sadi, 2011). Artificial neural networks (ANN) have been successfully used to attack a wide range of problems – such as speech recognition (Morgan *et al*, 1995), image classification (Lawrence *et al*, 1997), function approximation (Poggio and Girosi, 1990) and financial forecasting (Kaastra and Boyd, 1996). The present study uses ANN to predict the future wind velocity based on several successive sets of velocity measurements taken at a single location. Differently from earlier studies (e. g. More and Deo, 2003), where temperatures, humidity and pressures were also input to the model, our model uses only the recent past velocities which can be easily recorded on-board of a sailing yacht. Its simplicity allows the ANN model to run in real time, progressively adapting itself to the wind data that is also being measured in real time. ANN are mathematical structures that can emulate the process of learning from previous experience. For instance, expert sailors are able to recognise situations that have already occurred in the past and can predict similar evolutions of the current weather scenario. This human approach can be emulated using ANN. By using a time series approach, the future wind velocities are obtained as function of past and current values of measured wind speed data. This function is not chosen *a priori*, but it is approximated by the ANN with an algorithm that allows the network to gradually adapt itself to the character of past data. In this study we present several numerical models based on daily averages, and on ten and one minute wind velocity predictions. The use of the three different time scales for the measured wind speed allows for the investigation of different potential applications, depending on the quality and the scope of the information available and on the forecasts that are needed in order to support decisions made during navigation.

1.1 Artificial Neural Networks

ANN are inspired by the functioning of the biological neural networks in the brains of humans and animals. Brains work as highly-complex, non-linear parallel computers, that also have the ability to adapt their own processing structures according to particular input in order to perform specific computations. The constitutive unit of a neural network is a neuron, which is a singular processing unit that takes several inputs originating from other neurons, and produces an output that is then transmitted to other neurons. The mathematical representation of the structure of a neuron is shown in Figure 1. A neuron itself can be broken down into the following components:

- A set of connecting links, called synapses, where the i-th synapses is characterized by a weight w_i (synaptic weights);
- An adder within the neuron that sums each i-th synapse with weight w_i;
- An activation function, which transforms the sum computed by the adder into the neuron output y. If the activation function is linear, a neuron results in a linear combination of the input values, while non-linear activation functions allow for the modelling of non-linear problems.

Therefore, a neuron can mathematically be described by Equation (1):

$$y = \varphi(\zeta); \;\; \zeta = \sum_{i=1}^{u} w_i x_i \tag{1}$$

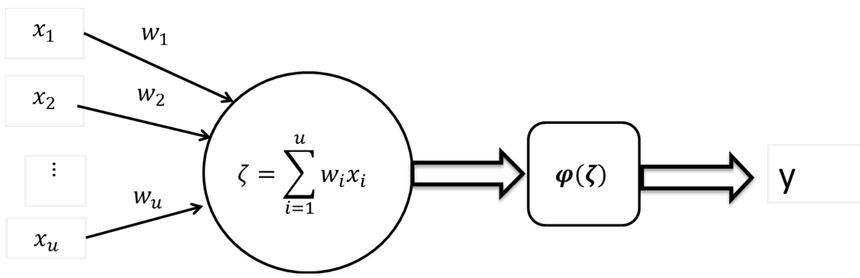


Figure 1. Structure of a generic neuron.

Neurons are assembled together into an integrated structure that depends on the kind of problem that the network has to solve. A structure that has been successfully used in function approximation is the so-called feed-forward multi-layer perceptron. Figure 2 shows a simplified example of a multi-layer feed-forward structure. In a multi-layer structure, neurons are organised in layers, where each neuron does not receive input from any other neuron in the same layer. Indeed, a neuron has a single output, while a layer receives the input from a set of neurons. In a feed-forward structure, the information follows only one direction, i.e. the output of the backward layer is input for the following layer. A first layer, namely the input layer, receives the input vector $(x_1, x_2, \dots, x_u)$ from the environment and transmits it to the adjacent layer, namely the first hidden layer. The following layers are called hidden layers, successively numbered until the last layer is reached, namely the output layer. The ANN illustrated in Figure 2 has an input layer of u neurons, two hidden layers of four and three neurons each, and an output layer of r neurons. It is common practice to use input layers that do not transform the

input vector, which is transmitted to the first hidden layer without being processed by nonlinear functions. Conversely, the output layer combines the results from the last hidden layer and its neurons have identity activation function.
The key feature of an ANN, which makes it widely used in several applications, is in its ability to learn and to adapt itself in order to produce the desired output for given inputs. The learning process involves the continuous modification of the synaptic weights and it is based on the principle of iterative error-correction. The synaptic weights of the various neurons are initialized to random values, then a training set of input and output data is presented to the network. For each input vector $(x_1, x_2, \dots, x_u)$, the initially generated output vector $(y_1, y_2, \dots, y_v)$ is compared with the known true output vector $(y_1^{true}, y_2^{true}, \dots, y_v^{true})$. The synaptic weights of the output layer are then modified by adding a factor that is proportional to the current assessed error and to a learning rate, and those corrections are extended to all of the weights in the network through a back-propagation process. This operation is iterated until successive changes in the synaptic weights are smaller than a given value, or when the errors begin to increase. In particular, when for given number ('validation check') of consecutive adjustments of the synaptic weights, the error consistently increases, then the training process is interrupted ('validation stop'). In fact, when over-trained, an ANN includes the numeric noise in the model and its performance decreases. An excessive number of neurons and layers make the learning process more computational demanding and may lead to over training the data (Principe *et al*, 1999). For further details on training algorithms and validation processes see Haykin (1999).

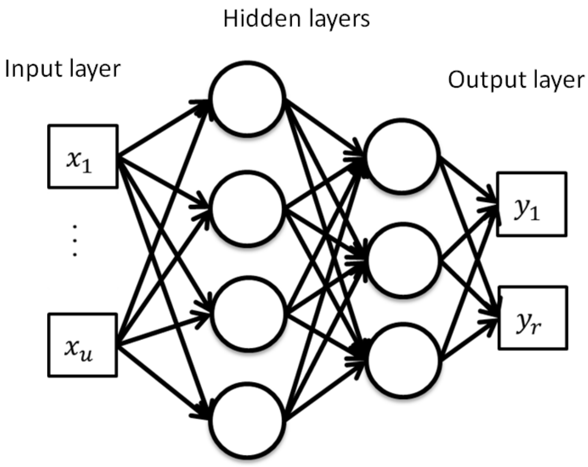


Figure 2: Schematic diagram of a multi-layer feed-forward perceptron.

2 METHOD

In this study past measured and recorded values of speed are used to forecast a time series of future wind velocity and this was undertaken using the scientific software package Matlab. The neural network approach extends the idea of time series linear forecasts. Let $s = s(t)$ be the specific wind speed measured at time $t \in [0, T]$ and consider a sampling of $s(t)$ at times $0 < t_1 < t_2 < \cdots < T$ with increment dt. At any instant t_k, given $s(t_{k-n}), s(t_{k-n+1}), \dots, s(t_k)$, we forecast values of $s(t_{k+1}), s(t_{k+2}), \dots, s(t_{k+m})$.
We write $s_k = s(t_k)$ and we assume that there exists a function f that satisfies Eq. (2).

$$s_{k+1} = f(s_{k-n}, s_{k-n+1}, \dots, s_k) \quad (2)$$

Two different strategies can be used in order to obtain a forecast of the further values immediately ahead in the series. The first strategy is to find a different function g in order to predict the further values from the same input vector. For instance:

$$s_{k+2} = g(s_{k-n}, s_{k-n+1}, \dots, s_k) \quad (3)$$

The second possible strategy is to use the same function f with an input vector shifted ahead of p steps, where $p-1$ steps are computed from the previous iterations. For example, computed s_{k+1} with Eq. (2), the second step ahead can be computed using Eq. (4):

$$s_{k+2} = f(s_{k-n+1}, s_{k-n+2}, \dots, s_{k+1}) \tag{4}$$

2.1 Linear versus non-linear time series forecasts

For time series forecasting, it is common practice to assume a linear function for f in Eq. (2). Therefore the expression for. f becomes:

$$s_{k+1} = a_0 s_k + a_1 s_{k-1} + \cdots + a_n s_{k-n} \tag{5}$$

It can be shown that, if the function s_k is linear combination of n trigonometric functions, then it is possible to obtain an exact forecast as linear combination of $2n$ previous values. When a small disturbance is added to the signal, the function f can only be approximated by means of a least square method. Alternatively, an ANN can be trained, assuming copious amount of suitable data, to model the function f. In this condition, the non-linearity of the activation function may allow the ANN to perform significantly better than the linear model where the coefficients are found using a LSM over more than a period of the lower frequency. For example, we can consider the discrete function $s(t_k)$ as given by Eq. (6):

$$s(t_k) = \sin(\pi t_k) + \frac{1}{2}\cos(5\pi t_k) + \frac{1}{5}\cos(20\pi t_k) + n(t_k) \tag{6}$$

where $n(t_k)$ represents a noise component. In particular, at every instant t_k, s is disturbed by a Gaussian variable $n(t_k)$ with a zero mean and a standard deviation 0.01. Figures 3(a) and 3(b) show a forecast on this discrete function made with a linear model and with the ANN, respectively. The ANN that was used in Figure 3(b) is a multi-layer perceptron with two hidden layers of five neurons each, and having hyperbolic tangent activation functions. Increasing the size of the input vector and the number of neurons on each layer leads to a better forecast but, also, to a slower convergence of the synaptic weights. For example, for the ANN used in Figure 3(b), it was found that more than five neurons per hidden layer led to only marginal improvements in the performance but to a significantly longer training time. If the function $s(t_k)$ did not have a noise component, then a linear model would allow for the exact forecast using six input data values. In fact, $s(t_k)$ is made of the superposition of three trigonometric functions. However, when a noise component is added, the performance of the linear model decays and the ANN approach allows better performance than the linear model. This also occurs when an insufficient number of input values is used. Therefore, in this example, the input vector was made of five consecutive values of the discrete function $s(t)$. The ANN was subsequently trained with 335 values. Figure 3 shows that the ANN provides a reliable forecast for a span that is approximately ten times longer than the linear model does.

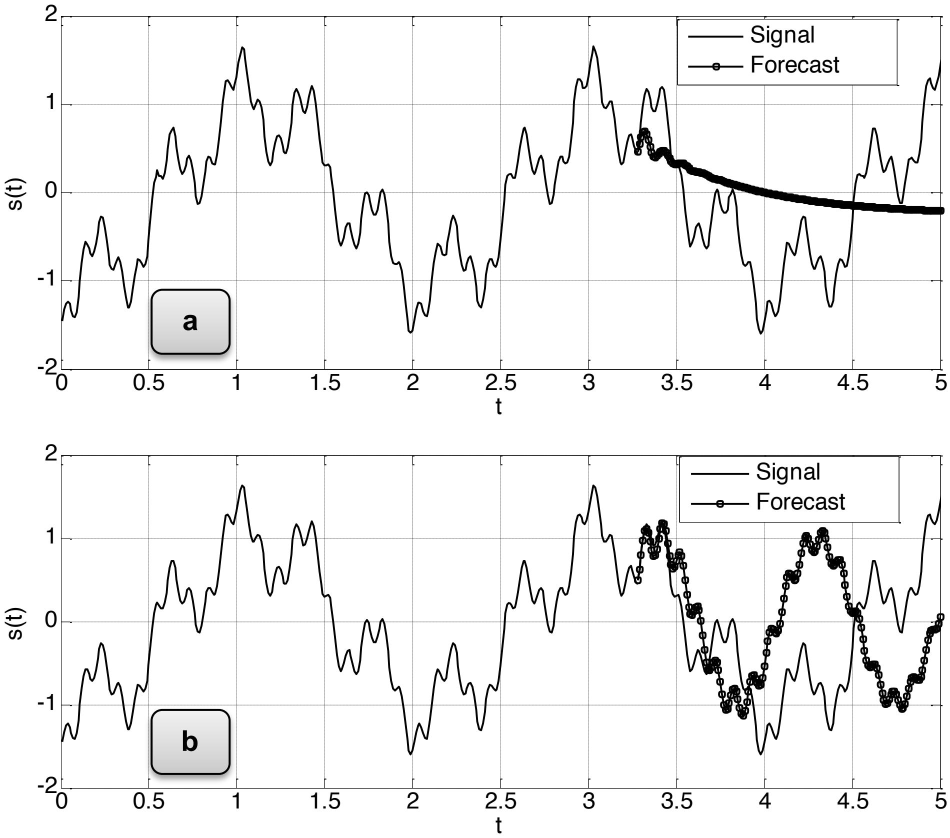


Figure 3. Forecasts with a linear model (a) and with ANN (b)

2.2 Training

When the network is trained, a large set of input vectors is processed and the computed outputs are compared with known data. The size of the training set depends on the problem, though it ranges between hundreds to millions of vectors. For every processed vector, the synaptic weights are updated. Therefore, the training is computationally demanding and the computational time increases with the size of the training set and of the input vector.
As stated above, there is an optimum size of the training set. When not enough data is available, then it is possible to update the training as data becomes available. In Section 3.3, we present an example of a network that is re-trained at regular intervals.

2.3 Correlation indices

The Pearson's correlation coefficient (Eq. 7) was used to determine the correlation between the measured and the forecast values. In particular, the computed vectors of one, two, etc. ($\boldsymbol{s_{k+1}} \boldsymbol{s_{k+2}}, \text{etc.}$) steps ahead were compared with the vectors made of the true values ($\boldsymbol{s_{k+1}}^{true}, \boldsymbol{s_{k+1}}^{true}$, etc.), where bold is used to identify vectors.

$$R_i = \frac{cov(\boldsymbol{s_{k+i}}, \boldsymbol{s_{k+i}}^{true})}{\sigma_{s_{k+i}} \, \sigma_{s_{k+i}}}, \tag{7}$$

where $\sigma_{s_{k+i}}$ and $\sigma_{s_{k+i}}$ are the standard deviations of $\boldsymbol{s_{k+i}}$ and $\boldsymbol{s_{k+i}}^{true}$, respectively; and $R \in [-1,1]$.
Additionally the percentage of uncertainty of the forecast, U, was computed at a 95% confidence level. The differences between the computed values and the measured values for

each temporal step were considered. From this it was then determined the maximum of these differences for the best 95% of them. Thus, the error is smaller than U for 95% of the forecast values. The uncertainty used in the following sections is presented as percentage of the mean values of the whole sets of data.

3. RESULTS

We present three different examples of the general ANN model that was discussed in Section 2. As shown by the previous exercise, an adequately structured and trained ANN can forecast a time series with a random component, using a small number of previous points as input. We used three different wind speed signals measured experimentally. Being these signals non-analytical, the optimum number of neurons and layers, and the size of the input vector, cannot be determined *a priori*. Therefore, for each of the three wind signals, these were chosen in order to minimise the forecast uncertainty.

In Section 3.1 we use daily averaged wind speeds, and we show an example of a one-step-ahead prediction solving Eq. (2); in Sections 3.2 we use wind speed measurements every 10 minutes and we show how Eq. (3) can be used to forecast up to five steps ahead (i.e. up to 50 minute forecast) with low uncertainty; in Section 3.3 we use the three spatial wind-velocity components, averaged over a minute, for producing a continuous forecast with the method showed by Eq. (4). In this last example, the network was also re-trained every 1.5 minutes.

3.1 Daily data

A set of data from the New Zealand National Climate Database (NIWA, 2012) was employed, which consists of daily averaged wind speed registered in Auckland from 1/1/1990 to 31/12/2008 for a total of 6941 values. Figure 4 shows that the signal is highly oscillating, and presenting opposite trends almost every day. Different networks were trained in order to identify the optimal structure to forecast the wind speed for the next day. It is known that a single layer perceptron can only model linearly separable problems (Haykin 1999). This postulate was tested using a one hidden layer feed-forward structure with a range of neurons of up to 30. It was found that the converged solutions showed a high dependence on the initial values of the synaptic weights. Also, on increasing the number of neurons no significant trends were observed, possibly this being negligible compared to the noise due to the initialisation process. The network performance increased when two hidden layers were used, in agreement with the theory (Haykin 1999). Increasing both the number of neurons per layer and the size of the input vector led the average error to decrease until optimum values were reached, then a further increase in neurons led the performance to decrease again. In particular, the best performance was achieved with two hidden layers with 18 and 10 neurons, respectively. The input vector was made of seven consecutive daily-averaged wind speeds and the single output vector was the predicted wind speed for the eighth day. Either an excessive number of neurons and a too large size of the input vector led the performance to decrease because the number of parameters that are required to be optimised increases and the training becomes inefficient.

The log-sigmoid function and a hyperbolic tangent activation functions were used for the first and second layers, respectively. Having two different non-linear activation functions increases the learnability when dealing with highly non-linear models (Haykin, 1999). The first 75% of the data set was used to train the network (6841 samples), and the ANN was then tested against the remaining 25% of the data set (1841 samples). The training was based on the Levenberg-Marquardt back-propagation algorithm, which is known to be efficient for networks with less than 100 neurons (Hagan and Menjai, 1994). The selected convergence criterion was a maximum weight gradient of 10^{-5} or six consecutive failed validation checks. The latter condition was achieved before the gradient convergence.

Figure 4 shows an example of 100 days, not included in the training set, and hence new for the network, where the model forecast is compared with the measured data. For each day, the average wind speed is forecast using the values measured in the previous seven days. The model is able to correctly predict whether the wind will increase or decrease for the following day. The sign of the velocity gradient is correctly predicted 96% of the time. Conversely, the absolute value of the prediction is computed with a large uncertainty, $U = 45\%$, and a poor correlation factor, $R = 0.6$.

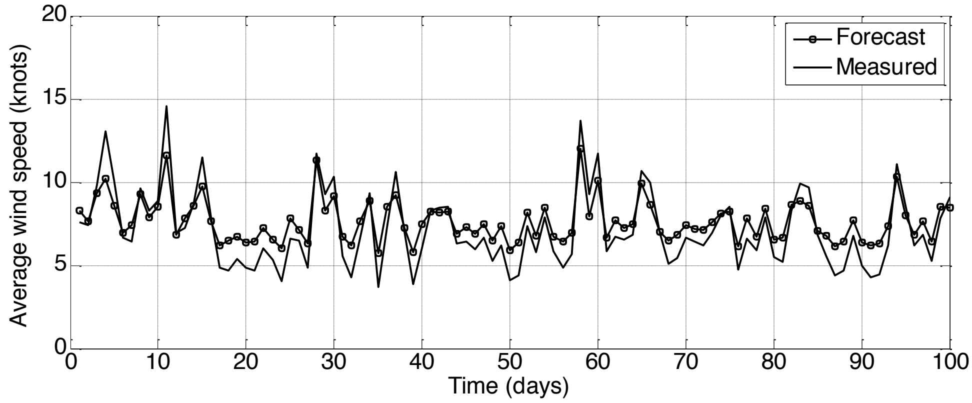


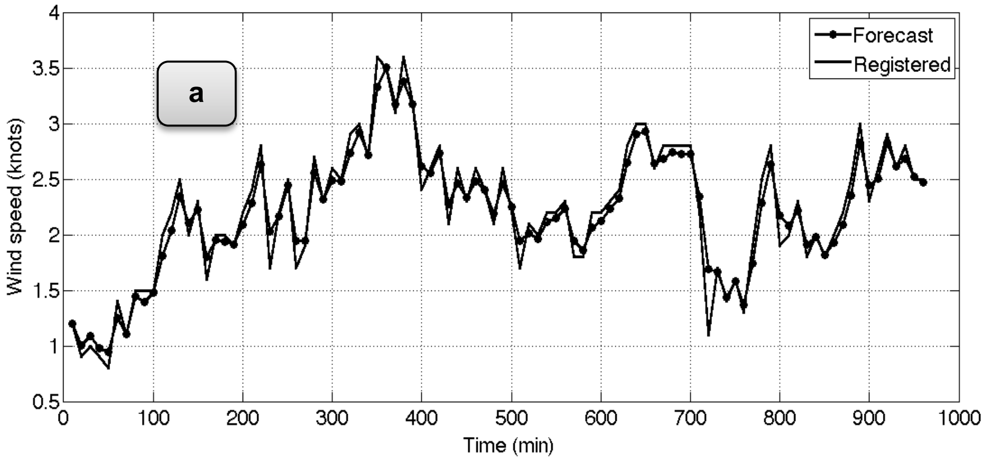
Figure 4 Forecast and measured daily-average wind speeds.

3.2 Ten minute data

Different data sets available from NIWA (2012) consists of wind speeds registered at six different stations in New Zealand over a period of one year (2010) at intervals of ten minutes. A total of 6940 instantaneous wind speed values were available from each station. While daily-averaged wind speeds (Figure 4) fluctuated around an almost constant value, the running average of the ten-minute measurements, shown in Figure 5, is affected by low frequency fluctuations. In order to take into account this additional complexity, the structure of the ANN was adjusted. In particular, the number of neurons in the second hidden layer was increased from 10 to 15, and the input vector size was increased from seven to eight data points. The same activation functions, training algorithm and the convergence criterion as selected in Section 3.1 were used.

Different input vectors were tested employing sets of data from all the six stations independently, and similar results were achieved. The results achieved for one of these are presented below as an example.

We are interested in predicting several time steps ahead. Therefore, the output vector was increased from one to five consecutive values (i.e. forecasting for up to the next 50 minutes). Figures 5(a) and 5(b) show the forecast values $s_{k+1} = f(s_{k-7}, \ldots, s_k)$ and $s_{k+2} = g(s_{k-7}, \ldots, s_k)$, corresponding to one and two steps ahead, and for 10 and 20 minutes ahead, respectively. For 10 minutes ahead, the ANN is able to correctly predict the sign of the wind velocity change (i.e. increasing or decreasing) in 98,5% of the cases, and the amplitude of the change is computed with an uncertainty $U = 5\%$. For more than 10 minutes ahead, the uncertainty of the ANN increases almost linearly. In particular, Table 1 shows R and U values for the 10, 20, 30, 40 and 50 minutes ahead forecasts.



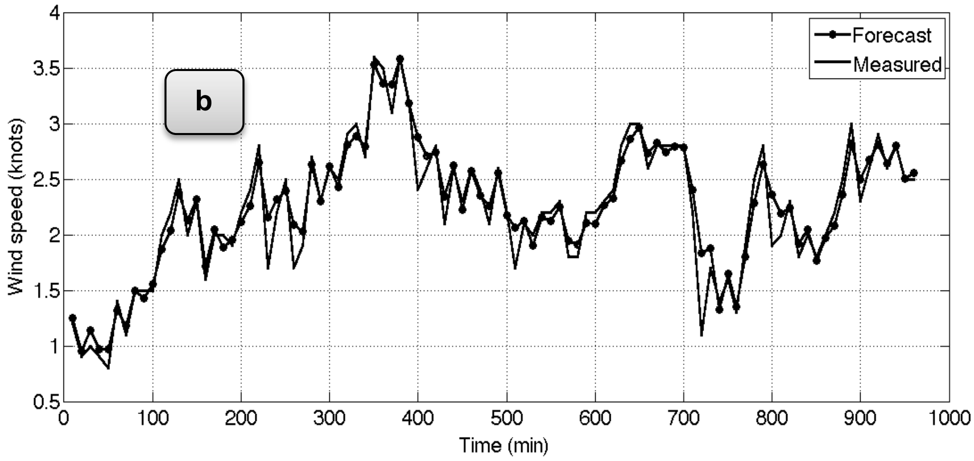


Figure 5: Measured and forecast wind speeds for one (a) and two (b) steps ahead.

Table 1: R and U for one to five steps ahead.

	10 min	20 min	30 min	40min	50min
R	0.91	0.88	0.87	0.81	0.78
U	5%	7%	11%	15%	20%

3.3 Three-seconds data

Wind data from a station installed on the Newcastle University research vessel allowed a higher frequency analysis than discussed in Sections 3.1-2. The velocity components along the three spatial directions were measured every three seconds for up to a total of 1200 samples. We are interested only in wind variation with frequencies up to 0.01 Hz. In fact, in order to be able to react to a wind oscillation with a change of the vessel course, the wind oscillation should have a period of longer than at least one minute.

Because an ANN can learn from a smoother time series more easily than it can from a highly-oscillating time series (Haykin, 1999), it is convenient to effectively smooth the input data up to the frequencies of interest. Therefore, we used moving averages over 20 values, corresponding to an average over one minute and 0.017 Hz.

The same number of hidden layers and neurons, activation functions, and training algorithm as in Sections 3.2 were used. The signal, which is smoother than the previous ones, allowed for using input vectors of up to 60 values (20 measurements per velocity component) without increasing the error. This leads to more connections (i.e. synaptic weights) due to the larger sizes of the input and output vectors.

This ANN showed a better performance ($R = 0.96$) than did the models presented in Sections 3.1-2. This is due to the additional velocity components being used as input, which thus provide more coherent information to the ANN, and also to the smoother trend of the signal. However, the resulting larger number of synaptic weights led the training process to be interrupted by a validation stop. Therefore, the number of validation checks was increased from six to ten, allowing the training to stop because the minimum gradient was reached. This change resulted in enhancing the performance to yield $R = 0.98$ for one step ahead. The training was performed on the first 75% of the 1200 samples, resulting in 900 samples equivalent to 45 minutes. We investigated the possibility of starting with a shorter training and being able to increase it in real time. The first forecast was based on only five minutes of training. Then, after every 90 seconds, the training was updated with a training set extended with the last 30 measurements. This approach was tested until the whole 45 available minutes were used for the training and this showed an increasing performance during the operations. In particular, after 20 minutes of training, a correlation coefficient $R = 0.98$, with an uncertainty $U = 1.5\%$, was achieved. In this case where the output is a vector with the next three velocity components, the uncertainty was computed considering the three velocity components independently. This level of performance did not increase further extending the training up to 45 minutes.

This network was also used in an iterative way to produce forecasts for the following minutes, according to Eq. (4). Using the network iteratively to forecast a long sequence of future values, it was found that error propagation will lead to either a flat or a highly oscillating forecast. However, the longer the training, the slower the error propagates. Thus, after adequate training, it is possible to have a reliable forecast for several steps ahead.

Figure 6 shows the number of future values that are iteratively forecast with an error which is smaller than 20%. The graph shows a general growing trend: given 16.5 minutes of training, the ANN can make a forecast with an error smaller than 20% for the more than one minute ahead, and given 30 minutes of training or more, about 1.5 minutes ahead is forecast.

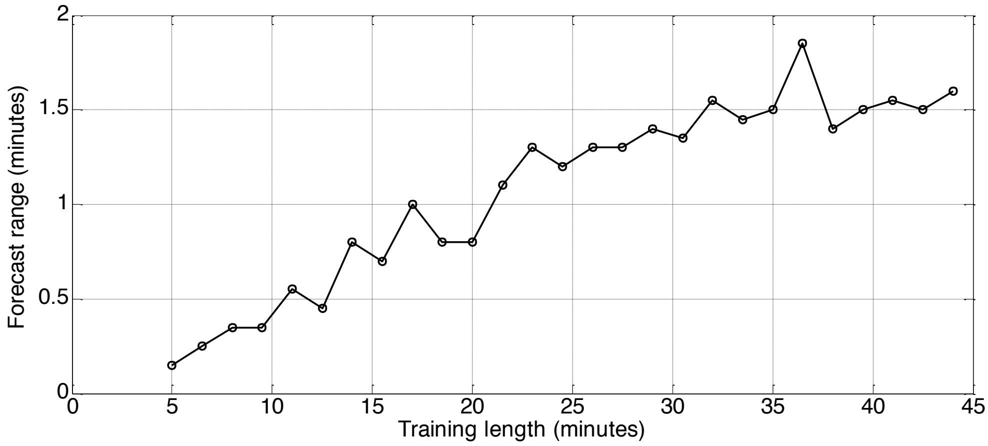


Figure 6. Forecast range with error lower than 20% versus training length.

Figures 7-8 show the forecasts performed by the ANN for training periods of three and 33.5 minutes, respectively. In Figure 7, after three iterations, corresponding to a forecast for nine seconds ahead, the error was equal to 20% of the wind speed. This poor performance was significantly improved using a longer training periods in Figure 8, which shows that the ANN is able to forecast values for up to 1.5 minutes ahead.

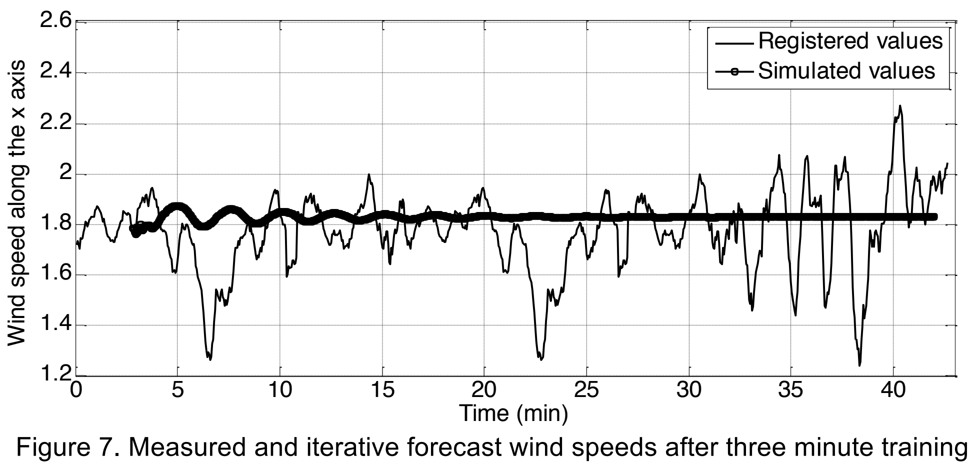


Figure 7. Measured and iterative forecast wind speeds after three minute training.

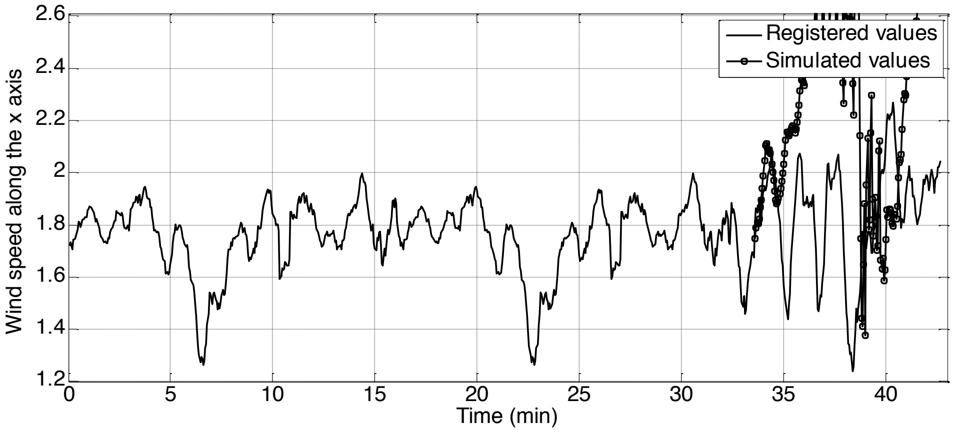


Figure 8. Measured and iterative forecast wind speeds after 33.5 minute training.

4. CONCLUSIONS

Reliable wind forecasts can enhance safety for navigation and performance during sailing yacht races. In this study we have presented ANN-based models for wind forecasts obtained at different time intervals, and that use the knowledge acquired from past measured wind speed values to forecast the near future wind speed. Simulations were carried out using daily averages, ten minute and three second measurements. For each of these sets of inputs, different feed-forward multi-layer networks were used. In fact, from daily averages down to three second measurements, the time histories showed smoother trends. This led to the use of a longer input vectors, in which input data stream length was optimised based on maximum learnability of the network. The use of longer input vectors requires more complex ANN structures, increasing the number of neurons in the hidden layers.

Given daily wind speed averages, the ANN was able to predict with high accuracy (96% of the times) the trend for the next day, while however the absolute wind speed was poorly predicted. Decreasing the sampling periods to ten minutes and three seconds, the ANN was able to predict, with high accuracy, up to 20 minutes ahead (two steps ahead: $R = 0.88, U = 7\%$) and 1.5 minute ahead (30 steps ahead: $R = 0.8, U = 15\%$), respectively.

Also, it was investigated the possibility to reduce the initial ANN training to only five minutes and then to use a real time training, where the training is updated every 1.5 minutes with the most recent measurements. The performance of the ANN was found to increase until when the training was longer than 22.5 minutes and then remain almost constant, able to predict up to 1.5 minute ahead.

REFERENCES

Churchland P., Sejnowski T. (1992), “The Computational Brain”, MIT Press, Cambridge.

Foley A. M., Leahy P. G., Marvuglia A., McKeogh E. J. (2012), “Current Methods and Advances in Forecasting of Wind Power Generation”, Renewable Energy, vol. 37, issue 1, pp.1-8.

Haykin S. (1999), “Neural Networks, A Comprehensive Foundation”, Second Edition, Prentice Hall PTR Upper Saddle River, NJ, USA.

Hagan M. T., Menhaj M. B. (1994), “Training Feedforward Networks with the Marquard Algorithm”, IEEE Transactions on Neural Networks, vol. 5, issue 6, pp. 989-993.

Huang Z., Chalabi Z. S. (1995), “Use of Time-Series Analysis to Model and Forecast Wind Speed”, Journal of Wind Engineering and Industrial Aerodynamics, vol.56, issues 2-3, pp.311-322.

Kavasseri R. G., Seetharaman K., (2009), “Day-ahead wind speed forecasting using f-ARIMA models”, Renewable Energy, vol. 34, issue 5, pp. 1388-1393.

Kaastra I., Boyd M. (1996), “Designing a Neural Network for Forecasting Financial and Economic Time Series”, Neurocomputing, vol. 10, issue 3, pp. 215-236.

Khathib T., Al-Sadi S. (2011) “Modeling of Wind Speed for Palestine Using Artificial Neural Networks”, Journal of Applied Sciences, vol. 11, issue 14, pp. 2634-2639.

Kohonen T. (1990), “The Self-organizing Map”, Proceedings of the IEEE, vol. 78, issue 9, pp. 1464-1480.

Lawrence S., Giles C.L., Tsoi A.C., Back A.D. (1997), “Face Recognition: A Convolutional Neural-Network Approach”, IEEE Transactions on Neural Networks, vol. 8, issue 1, pp. 98-113.

Louka P., Galanis G., Siebert N., Karinotakis G., Katsafados P., Pytharoulis I., Kallos G. (2008), “Improvements in Wind Speed Forecasts for Wind Power Prediction Purposes using Kalman Filtering”, Journal of Wind Engineering and Industrial Aerodynamics, vol. 96, issue 12, pp. 2348-2362.

Masson V. (2000), “A Physically-Based Scheme for the Urban Energy Budget in Atmospheric Models”, Boundary-Layer Meteorology, vol. 94, issue 3, pp. 357-397.

Mohandes M. A., Rehman S., Halawani T. O. (1998), “A Neural Networks Approach for Wind Speed Prediction”, Renewable Energy, vol. 13, issue 3, pp. 345-354.

More A., Deo M. C. (2003) “Forecasting Wind with Neural Networks”, Marine Structures, vol. 16, issue 1, pp.36-49.

Morgan, Nelson, Bourlard, Herve A. (1995), “Neural Networks for Statistical Recognition of Continuous Speech”, Proceedings of the IEEE, vol. 83, issue 5, pp. 742-770.

NIWA 2012, National Climate Database, National Institute of Water and Atmospheric Research,available from http://cliflo.niwa.co.nz/ (visited December 2011)

Poggio T, Girosi F. (1990), “Networks for Approximation and Learning”, Proceedings of the IEEE, vol. 78, issue 9, pp. 1481-1497.

Zhang X., Zhan J., Li Y., Zhang R. (2011) “Short-term Forecasting of Wind Speed Based on Recursive Least Squares”, 2nd Annual Conference on Electrical and Control Engineering, ICECE 2011 – Proceedings, pp. 367-370.