Wind Resource Assessment and Forecast Planning with Neural Networks

Nicolus K. Rotich[*], Jari Backman, Lassi Linnanen, Perfilieve Daniil

Laboratory of fluid dynamics
Lappeenranta University of Technology, Lappeenranta, Finland
e-mail: nicolus.rotich@lut.fi

ABSTRACT

In this paper we built three types of artificial neural networks, namely: Feed forward networks, Elman networks and Cascade forward networks, for forecasting wind speeds and directions. A similar network topology was used for all the forecast horizons, regardless of the model type. All the models were then trained with real data of collected wind speeds and directions over a period of two years in the municipal of Puumala, Finland. Up to 70th percentile of the data was used for training, validation and testing, while 71–85th percentile was presented to the trained models for validation. The model outputs were then compared to the last 15% of the original data, by measuring the statistical errors between them. The feed forward networks returned the lowest errors for wind speeds. Cascade forward networks gave the lowest errors for wind directions; Elman networks returned the lowest errors when used for short term forecasting.

KEYWORDS

Wind, Resource, Assessment, Forecasting, Artificial, Neural, Networks

INTRODUCTION

Several wind power prediction models have been developed in the recent past. However, different models are suitable for various types of situations, depending on the nature of the required forecast. Some models are better suited for long-term forecasting while others are better for short term forecasting. The suitability of a model can be assessed by the number of time steps into the future, a model can forecast while still retaining its robustness on the predicted outputs, without losing its generalization ability. Generalization is the ability to produce accurate results even for input data set that the model has not 'seen' i.e. not used in the training of the model [1]. In general, three approaches of wind forecasting methods have been well documented so far (2012); the numerical weather prediction models (NWP), physical systems approach, and the statistical approaches.

The numerical weather prediction (NWP) system simulates the atmosphere by numerically integrating the equations of motion starting from the current atmospheric states. This is done by mapping the real world on to a discrete 3-D computational grid that divides the globe into numerous polygonal patterns of certain dimensions e.g. 60 km^2 [2].

---

[*] Corresponding author

Physical systems, model the dynamics of the atmosphere by parameterization of the planetary boundary layer (PBL) concept, also known as the atmospheric boundary layer (ABL). ABL is the lowest part of the atmosphere that is in continuous contact with the surface of the earth. Here, the physical quantities e.g. velocity, temperature and moisture (of the wind/air) are turbulent and vertical mixing is stronger. The physical systems are further broken down into two, the numerical simulations and diagnostic models, which are both based on parameterization of the planetary boundary layer flow. Some of the numerical models that have been developed based on parameterization of the planetary boundary layer are; Fifth-generation Mesoscale Model (MM5), Weather Research and Forecasting (WRF) model and Regional Spectra Model (RSM), discussed by [3]. Examples of diagnostic models are the Prediktor and Previento, developed by Landberg at the National Laboratory in Risø, Denmark in 1993 [2], and University of Oldenburg, Germany [4].

Generally, statistical systems are implemented based on built and trained models using real data (specific to the location in which data is collected) over a number of discrete periodic cycles. The difference between the predicted output and the required output (error) is minimized by fine-tuning it to a level which can be used for nowcasting and/or forecasting. The statistical systems are divided into three, Wind Power Prediction Tool (WPPT), Fuzzy Logic (FL) and Artificial Neural Networks (ANN). WPPT is a statistical tool developed and operated by the Danish national laboratories for weather forecasting. The WPPT is based on an autoregressive eXogeneous (ARX) input type model, where wind speed and therefore power is described as a non-linear, non-stationary and time-varying stochastic process representing the dynamics of the atmosphere. The second statistical approach is that which treat future wind speeds as vague or indistinct and thus tries to solve by reasonable approximation with fuzzy logic concept. Such system has been developed and is currently operated for short term predictions by Ecole des Mines de Paris, France.

Artificial Neural Networks (ANN), also referred to as neurocomputing, is the third statistical approach which is one of the most recently developed methods for accurate forecasting. The objective of this study was to analyse the quality and quantity of the collected data, develop forecasting models using artificial neural networks which would enable general future planning, given previous data of wind speeds and directions. The models in the study could then be used to make important decisions pertaining sitting and developing wind power farms at the study location.

METHODOLOGY

The Artificial Neural Network Project Cycle

A successful artificial neural network project (ANN), like project cycles in other disciplines, constitute a number of phases, namely; problem definition and formulation, system design, realization, verification, implementation, and system maintenance phase. The last two phases (system implementation and maintenance) involves embedding the obtained networks in an appropriate working system e.g. hardware or a packaged program that can be installed to run in a computer. This paper is only confined to the first four steps of the project cycle. Figure 1 below shows various stages of an ANN project cycle and the study scope.
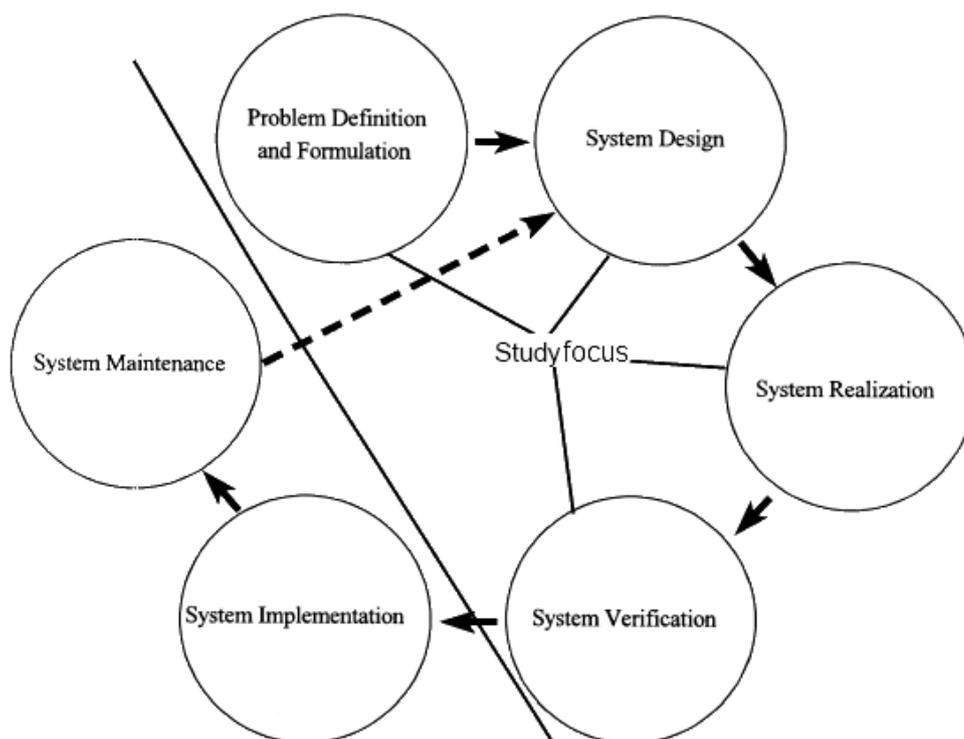
Figure 1. The project cycle of an ANN project, based on [5]

Problem definition and formulation

The overall view of this phase, together with the rationale has been partially covered in the first two chapters. The outstanding part is specific problem definition and formulation which entails explaining the kind of data available and what was required out of it. The problem involved two non-linear, non-stationery, univariate vectors of wind speeds and directions collected over a period of 2 years (from 1.11.2009 up until 30.10.2011). The data sampling intervals was 10 minutes. It was taken at a height of 60 m from the ground in the municipal region of Puumala. Puumala municipality is strategically situated along a 3,000 km shoreline at the southern Saviona region of Eastern Finland. Its location makes it prone to offshore winds that can be harnessed for wind power. The fundamental end results of the project were to construct the three common types of ANNs namely; feed forward, cascade feed forward and Jordan Elman neural networks for wind speeds and directions forecasting, and to test the networks by comparing and assessing their mean square error (MSE) and sum squared error (SSE) as the convergence criteria, during training and upon forecasting. Procedurally, the models were used in making a one step ahead hourly forecasts with 10 minute intervals, daily forecasts with hourly averages, weekly forecasts with half-daily averages and monthly forecasts with daily averages and the convergence criteria also measured for this forecasting step and the results presented and discussed.

System design

System design phase usually starts with data collection, and pre-processing, which can be done within or outside the computation environment. Selection of simulation parameters is the second process before model construction begins. The data used herein was provided by Lappeenranta University of Technology (LUT), and granted the author

with permission to use as part of this paper. System design therefore began from data pre-processing i.e. data averaging, subdivision of data into training, validation and testing sets, normalization (scaling) and backward/forward shifting in time into various lagged variables, in a process referred to as 'sliding window technique' used as inputs/outputs of the networks.

Data Pre-processing

Wind speed and direction vectors of length (104,043) were periodically averaged into the required time periods. To get hourly data, 6-ten minute measurements were averaged. Similarly to obtain daily means of wind speeds and directions, 24-hourly averages were taken. Averaging is followed by normalization of the vector. There are a number of ways to normalize data; here we used the reciprocal which scales the data to a range of 0 to 1, before subdividing into three parts; 70% for training, 15% for validation and 15% for system testing. Lagged variables (sliding windows) were then created conforming to the desired inputs and outputs; for hourly forecasts, six 10-minute interval outputs were required, for daily forecasts 24 outputs of hourly intervals, weekly interval required 7 outputs of daily averaged values, and monthly interval 30 outputs of daily averages.

Models construction

In general three classes of models were constructed; the feed forward neural networks (FFNN), Jordan Elman neural networks (JENN), and Cascaded feed forward neural networks (CFNN), (Figures 2-5). For each class of models above, lagged variables of wind speeds and directions were separately used as inputs to the networks. Four sub models were then constructed corresponding to the forecast horizons (hourly, daily, weekly and monthly), making a total of 24 models built.
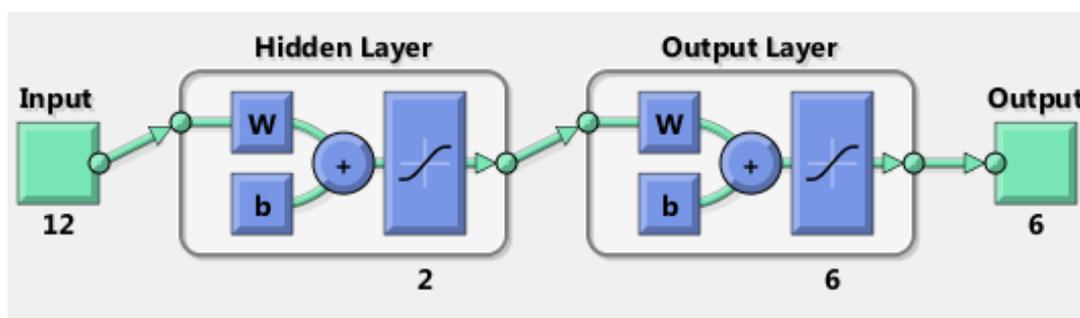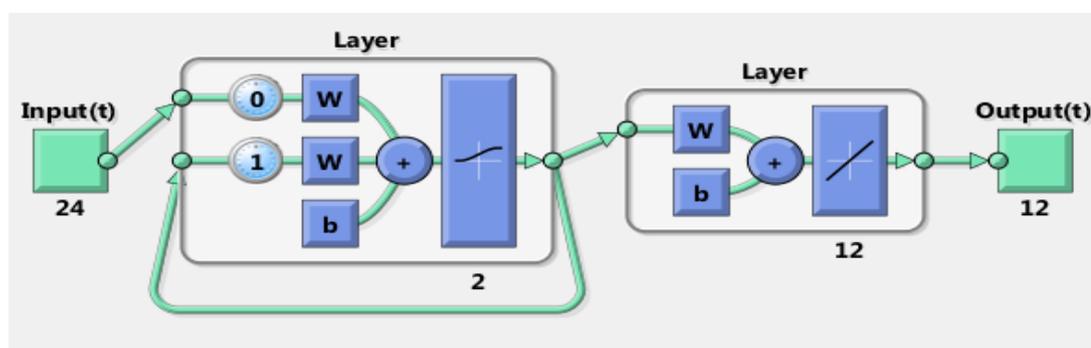

Figure 2. FFNN used for hourly forecasts

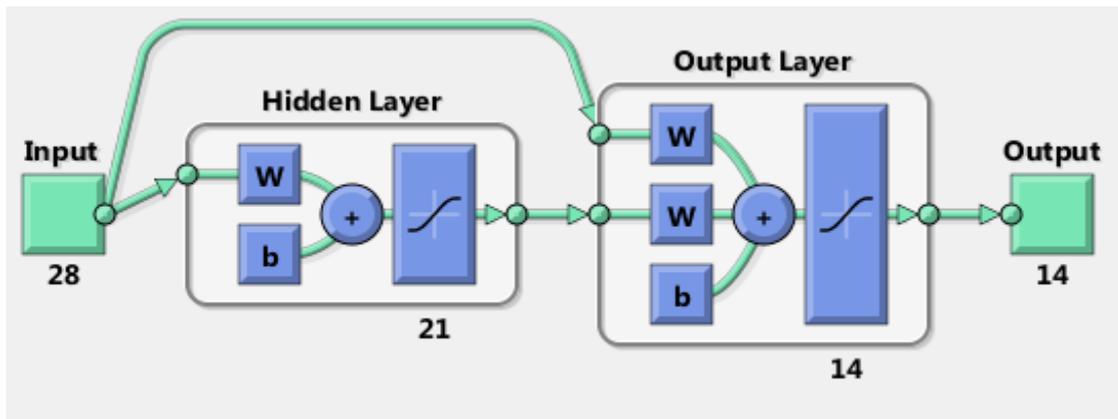Figure 3. JENN used for half-day forecasting

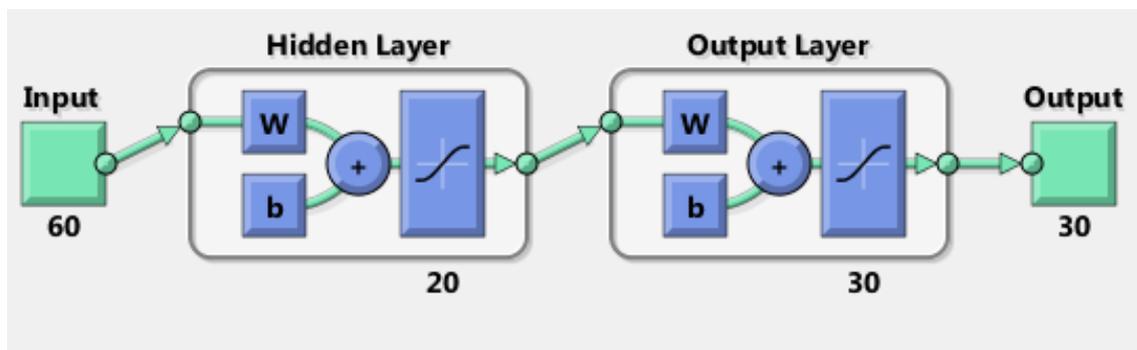Figure 4. Cascade feed forward neural network used for weekly forecasting

Figure 5. Cascade feed forward neural network used for monthly forecasting

To make them comparable, authenticable and more realistic, models of the same network topologies were constructed and used for the same forecast horizon, e.g. for hourly forecasting, a model with 12 inputs, 2 hidden neurons and 6 outputs, (denoted as 12:2:6), was used throughout for all model types (JENN, FFNN and CFNN). For daily forecasting: 24:2:12, weekly forecasting: 28:21:14 and monthly forecasts were performed with the largest model with a topology of 60:20:30.

System realization

The most interesting, challenging and critical phase of the study is to build the models. Tens of parameters are usually controlled during modelling with neural networks. However, not all of them have significant effects on the network's generalization ability. As a result, a number of modelling parameters are selected depending on the forecast horizon, degree of accuracy required, the speed at which the results are needed, among other factors. In most cases, applications used for modelling have inbuilt default settings e.g. MATLAB has readily available codes for quick modelling. In order to achieve a more meaningful model however, the modeller has to diligently select the parameters and optimize them according to some set rules and/or past experience. Noted parameters that influence network results are; the data size partitioning i.e. into training, validation and testing, type of data normalization used, input/output representation, network weight initialization, the learning rate, momentum coefficient, transfer function, convergence criteria, number of training cycles (epochs), hidden layer sizes, the training algorithm etc. For the current study, the following modelling parameters were considered

Input/output representation

In this study, the default normalization function was disabled to give room for custom defined normalization and denormalization; continuous, normalized variables between 0 and 1 were used as inputs and outputs representing wind speeds and directions, before denormalization to their original formats.

Transfer function (ζ)

The transfer functions used for this study were arrived at by trial and error methods starting from the presumption that data was scaled to the range of 0 to 1 and thus a sigmoidal transfer functions which possesses the distinctive properties of continuity and differentiability on the range ($-\infty$, $+\infty$) was necessary, an essential requirement of Back propagation learning [5]. A prior consideration was also given for the fact that a combination of hyperbolic transfer functions for both the hidden and the output layers yielded better recognition results [6].

Size of the hidden layer (H)

Nagendra and Khare in their study suggest that the rules failed to yield the 'optimal' size of hidden layer, inferring that the best way to obtaining the required hidden layer size is by iteratively adjusting the size while measuring the error during neural network testing [7]. In this study the neural networks should ideally be able to learn and 'understand' the fluid statics/dynamics of the atmosphere e.g. the effects of longitudinal and transverse wind velocity gradients, atmospheric temperature and pressure among other factors and assign appropriate weights to accurately forecast the future values. The final sizes of the hidden layer was arrived at by continuously iterating, while measuring the convergence criteria i.e. sum squared error (SSE) and mean squared error (MSE) during evaluation of the network. SSE and MSE were evaluated for one point per 'sliding window' and for 'one step ahead' forecasts and compared.

The training algorithm

Different training algorithms are good for different purposes, the predictive ability (which is the current subject), has been tested by Ghaffari and team, who concluded that the order of predictive ability of a network trained using above group of training algorithms is IBP, BBP followed by LM, QP and lastly GA. [8]. In this study, Bayesian regulation (BR) Back propagation algorithm was used for all the models. Lavenberg-Marquardt (LM) was also tried but it proved to take too long training time than expected. Both LM and BP training algorithms are implemented in MATLAB and can be invoked by a single command. Many training algorithms suffer from the problem of over fitting, a phenomenon in ANN, caused by overtraining, resulting in memorization of input/output, rather than basing them on the internal factors determined by the weights generated. This causes the network to respond poorly when presented with new data that was not used during training, thus losing the object orientedness, an important aspect of the network, also referred to as generalization. Bayesian regulation seems to train successfully has an inbuilt ability to get rid of this problem through automatic early stopping once the error starts to propagate. [9].

Network weight initialization

Several main techniques are currently used to get rid of premature saturation, a phenomenon that has been known to cause over fitting and affect network convergence [10, 11]. Nguyen and Widrow had suggested that initializing adaptive weights over a large number of training problems achieved major improvements in learning efficiency

[12]. Moallem and Ayoughi proposed three methods; increasing the number of hidden neurons, Weigend weight regularization and renewing saturated terms by adding anti-saturating terms [13]. Network weight initialization involves assigning predetermined optimum initial values for the weights to all existing connection links that help the network to converge faster. For the current study, Nguyen and Widrow weight initialization algorithm was used. In this algorithm, weight bias initialization values are picked between the intervals located randomly in the predetermined region i.e. -1 and 1. Nguyen and Widrow suggested that, if H is the number of units in the first layer, *Wbi* = 0.7*H*. *Wi* are chosen between -1 and 1 and the weights, *w* are assigned so that *w* = -*Wi/Wbi*, simply put as the uniform random values between -1 and 1 and is implemented in MATLAB [14] as a script file [15].

Learning rate (η)

A high learning rate is detrimental to the network as it poses a risk of overshooting while a slow learning rate takes too much time for the network to converge. The learning rate can be constant throughout, as was done in this study or can be made adaptive i.e. to vary with time, η(t). In the case of adaptive parameter, it can be made high in the beginning of the training or rather when the search is far away from the minimum; and smaller as the search reaches minimum. This parameter rate can be anything between 0 and 10. In all the networks created for this paper, the learning rates ranging from 0.01 to 3 gave satisfactory results.

Momentum coefficient (μ)

A high μ is likely to reduce the risk of getting trapped in the local minima. However, it runs the risk of overshooting just as a high learning rate does. This value, just like the learning rate, can be made adaptive, i.e. μ(t). It is set relatively high when the search is far away from the solution and lower as the search approaches the true minimum, depending on the error gradient [16]. For this project, the momentum coefficient between 0.0 and 1.0, as suggested by [17], produced satisfactory results.

Number of training cycles (Epochs)

An epoch is defined as a single presentation of each input/output data on the training set [16]. Epochs are set as one of the training parameters and are important in gauging the training time taken by a neural network to reach convergence and also to set the goal that determines the extent to which the network should be trained. For this study, the training epochs were set by trial and error, with a range of 100 to 1000. At most 1000 epochs for all the models built produced satisfactory results. The use of Bayesian regulation training algorithm also was used as a tool for setting the stopping time, making epoch setting just a supporting criterion.

System verification

This is the stage that this study is focused on, as it clearly distinguishes the variation in the original data to the predicted. It was made part of the modelling stage by supplying the model with the range of original data set, from the 70-85th percentile and comparing the model output to the last 85th to 100 percentile of the same data. The convergence criteria were then measured by determining the two statistical properties, i.e. the MSE and the SSE between the forecast and the target results, which were compared and reported for each model built. In the next section the quantitative results of the study were presented graphically and by tabulation.

Models performance measurement

In this study the mean square errors (MSE) and the sum square errors (SSE) were used to gauge the performance of the networks. The mean square error is the average of all the squares of individual errors between the model and the real measurements, and is given by:

$$MSE(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \qquad (1)$$

where N is the number of samples, x_i and y_i are measured and predicted values.

The sum square error (SSE) is the total summation of the individual squares of errors without averaging, and it gives an indication of the total magnitude of the error between the models and the measured results. SSE is given by:

$$SSE(x,y) = \sum_{i=1}^{N}(x_i - y_i)^2 \qquad (2)$$

In addition, MSE and SSE are useful in making comparisons between several models with same sets of data and same observations, N. In the event that more than one model is compared, one important indicator obtained is how better a model is, compared to the others. As seen, both MSE and SSE are dependent on the number of observations and so the quantities (orders) of errors are only significant, relative to those of other models and have units same as the square of the variable under question (m^2/s^2 for wind speed and sq. degree (o^2) for directions).

RESULTS

Models assessment for long-term forecasting

Tables below show the results of the models based on both MSE and SSE on training and upon simulation with totally new inputs, not used during training. Here we compare a column on the model outputs, to the corresponding column on the measured data. This is referred to as 1-point per sliding window. A 1- point per sliding window extends for the entire column length. Plotting a column on the target matrix versus a corresponding column on the model output matrix measures the generalization ability of the model with increasing forecast horizon on the long-term. The results for this exercise are shown in Tables 1-4. Tables 1 and 2 were used to assess long-term generalization ability for wind speed forecasting; similarly Tables 3 and 4 were used to test the generalization ability for wind directions on a long-term basis.

Table 1. The results of the models, assessing the generalization ability when used for long term forecasting of wind speeds (Hourly & Daily)

	HOURLY FORECASTS				DAILY FORECASTS			
MODEL	MSEt	SSEt	MSEv	SSEv	MSEt	SSEt	MSEv	SSEv
JENN	0.3801	27772	4.2E6	6.6E10	0.667	8106	8.689	22695
CFNN	0.3638	26582	1.7E7	2.7E11	0.667	8106	8.452	22076
FFNN	0.3807	27817	3767	5.9E7	0.688	8364	7.899	20632

Table 2. The results of the models, assessing the generalization ability when used for long term forecasting of wind speeds (Weekly & Monthly forecasts)

	WEEKLY FORECASTS				MONTHLY FORECASTS			
MODEL	MSEt	SSEt	MSEv	SSEv	MSEt	SSEt	MSEv	SSEv
JENN	2.104	2047.3	4.9982	1079.6	1.798	751.4	4.122	445.22
CFNN	2.56	2490.9	6.8416	1477.8	2.041	853.2	9.306	1005
FFNN	2.0931	2067	6.3329	1368	1.628	680.6	4.18	451.3

Table 3. The results of the models, assessing their generalization ability when used for long term forecasting of wind directions (Hourly & Daily forecasts)

	HOURLY FORECASTS				DAILY FORECASTS			
MODEL	MSEt	SSEt	MSEv	SSEv	MSEt	SSEt	MSEv	SSEv
JENN	6.4E3	4.7E8	8.8E6	1.3E11	4.2E3	5E7	1.3E4	3.4E7
CFNN	1.3E4	9.6E8	1.3E9	2.1E13	4.1E3	5E7	1.3E4	3.5E7
FFNN	1.6E4	1.1E9	1.7E4	2.7E8	3.3E3	4E7	1.3E4	3.5E7

Table 4. The results of the models, assessing their generalization ability when used for long term forecasting of wind directions (Weekly & Monthly forecasts)

	WEEKLY FORECASTS				MONTHLY FORECASTS			
MODEL	MSEt	SSEt	MSEv	SSEv	MSEt	SSEt	MSEv	SSEv
JENN	5.4E3	5.3E6	1.4E4	3.0E6	9.9E3	4.1E6	1.5E4	1.7E6
CFNN	6.6E3	6.4E6	1.3E4	2.8E6	6.2E3	2.6E6	1.2E4	1.2E6
FFNN	6.3E3	6.1E6	1.6E4	3.4E6	1.1E4	4.6E6	1.3E4	1.4E6

Models assessment for short-term forecasting

The short term usability of the models was assessed by measuring the relative error between the model output rows and the measured data, referred to as a sliding window. A sliding window is simply one set of inputs and outputs to a neural network model, e.g. for hourly forecasting with 10-minute interval data, a row of six model outputs are compared to the corresponding row in the real measured data matrix. Plotting and comparing the rows cutting across the model output matrix to those of the target matrix is what was referred to, as sample whole sliding window. This measures the generalization ability of the model on a short term basis, also commonly referred to as one-step-ahead forecasting. The results are presented in Tables 5 and 6. Table 5 assesses the generalization ability of the models when used for forecasting wind speeds; Table 6 presents same equivalent results for wind directions forecasting.

Table 5. The results of the models, assessing their generalization ability when used for short term forecasting of wind speeds

	HOURLY FORECASTS		DAILY FORECASTS		WEEKLY FORECASTS		MONTHLY FORECASTS	
MODEL	MSEv	SSEv	MSEv	SSEv	MSEv	SSEv	MSEv	SSEv
JENN	16.4211	98.266	0.3430	4.1156	2.8959	40.5432	3.1452	94.3548
CFNN	17.3025	103.8152	0.3492	4.1899	1.8416	25.7827	9.1991	275.9728
FFNN	16.4183	98.51	0.5430	6.5154	1.6656	23.3138	2.1257	63.7702

Table 6. The results of the models, assessing their generalization ability when used for short term forecasting of wind directions

MODEL	HOURLY FORECASTS MSEv	SSEv	DAILY FORECASTS MSEv	SSEv	WEEKLY FORECASTS MSEv	SSEv	MONTHLY FORECASTS MSEv	SSEv
JENN	9.6E3	5.8E4	4E3	4.7E4	2.8E4	4.0E5	1.5E4	4.4E5
CFNN	1.0E4	6.0E4	3.6E3	4.3E4	6.1E3	8.6E4	1.2E4	3.6E5
FFNN	3.0E4	1.8E5	3.7E3	4.4E4	2.5E4	3.4E5	1.8E4	5.3E5

Developing the criteria for choosing between different forecasting models

The core needs determines the criteria applied by the modeller in choosing between various types of models. A number of criteria used in this study to assist in making that choice are identified as the degree of accuracy needed, the forecast horizon for which the model is designed, and whether the model is usable for long-term or short-term forecasting. In this case, long-term forecasting can be hourly forecasts for a relatively long period of time e.g. several months ahead. With the kind of results presented in section 3.1 and 3.2 therefore, one can tell which model type has the lowest statistical error compared to other models, during training and upon verification i.e. with new inputs. It is also possible to tell which model is best suited for which forecast horizon, and which one is good enough for long/short term forecasting for both wind speeds and directions. Tables 7 and 8 summarize the obtained results, specifically answering the above important questions regarding the models.

Choice of wind speeds forecasting models based on generalization ability

Table 7. Making a choice between the models for use in forecasting wind speeds

MODEL	Generalization Error (MSE and SSE) Hourly L	S	Daily L	S	Weekly L	S	Monthly L	S	Score
JENN				✓	✓		✓		3
CFNN									0
FFNN	✓	✓	✓			✓		✓	5

Choice of wind directions forecasting models based on generalization ability

Table 8. Making a choice between the models for use in forecasting wind directions

MODEL	Generalization Error (MSE and SSE) Hourly L	S	Daily L	S	Weekly L	S	Monthly L	S	Score
JENN		✓	✓						2
CFNN				✓	✓	✓	✓	✓	5
FFNN	✓								1

Sample forecast results from selected networks

Sample plots for predicted and forecasted values versus measured data for two groups of networks models (FFNN and CFNN) are presented below for hourly, weekly and

monthly forecast horizons. Two important terminologies are emphasized *predicted* and *forecasted* results; the difference between predicted and forecasted variables should be noted. In statistical modelling, predicted variable usually refers to the output of data used for training, i.e. assessing how well the training data fits to the model output. Forecasting is the expected results into the future from a predictive model, for inputs that were not used during training of the model. Samples of predicted, measured, and forecasted results for hourly, weekly and monthly horizons are shown in Figures 6-11.

Hourly forecasting of wind speeds with FFNN
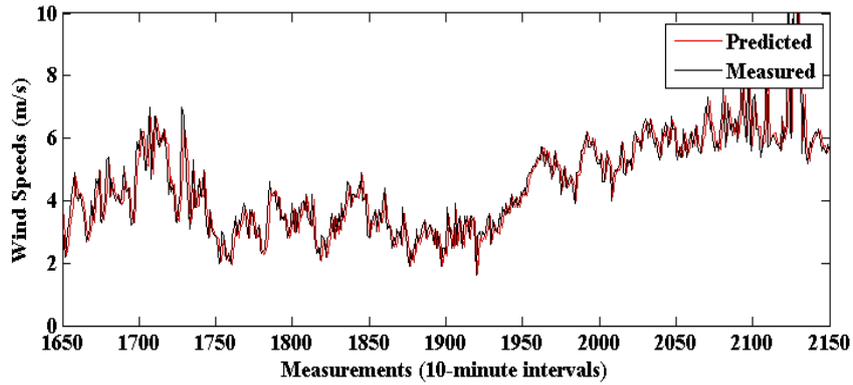
Figure 6. Comparing model outputs and the measured hourly wind speeds upon training
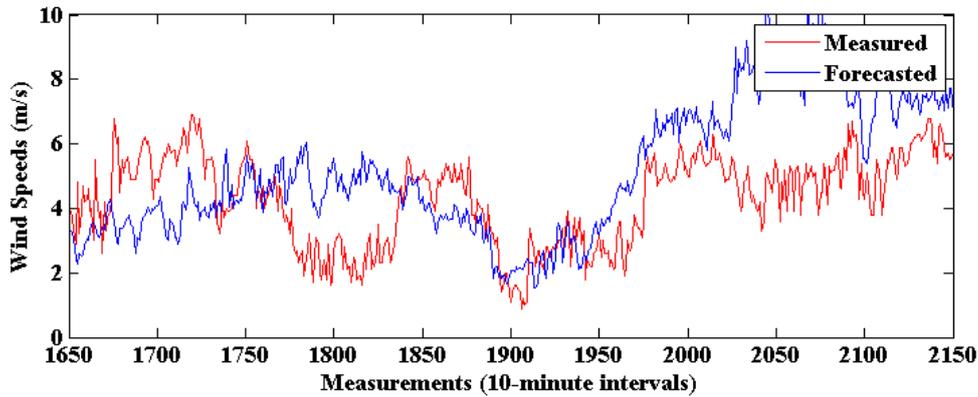

Figure 7. Comparing model outputs and the measured hourly wind speeds upon verification

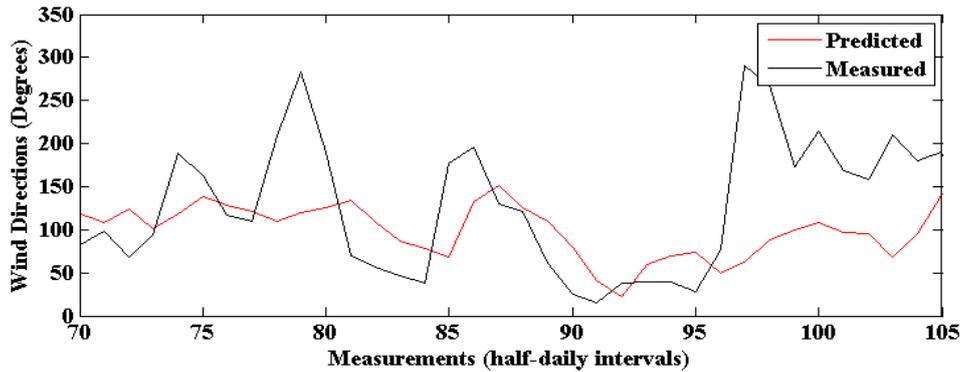### *Weekly forecasting of wind directions with CFNN*

Figure 8. Comparing model outputs and the measured weekly wind directions upon training
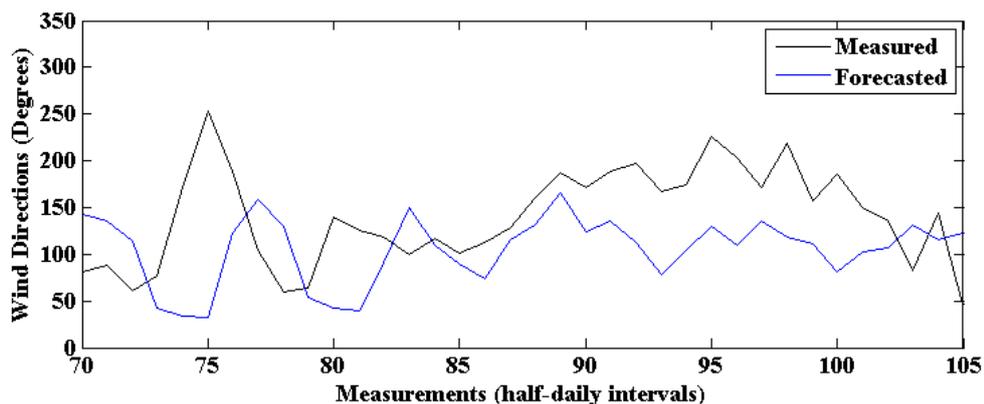
Figure 9. Comparing model outputs and the measured weekly wind directions upon verification

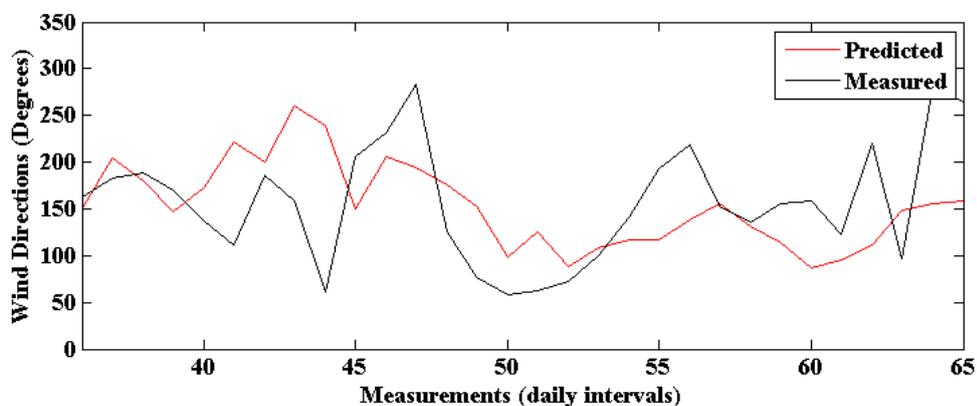## Monthly forecasting of wind directions with CFNN

Figure 10. Comparing model outputs and the measured monthly wind directions upon training
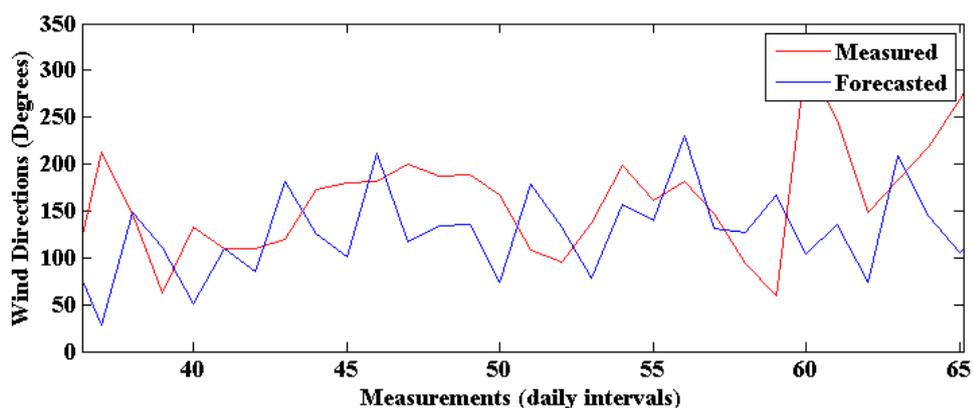

Figure 11. Comparing model outputs and the measured monthly wind directions upon forecasting

NB: The measured data on training is a part of the first 70% and on verification is part of the last 15% of the original data; therefore they are not the same set of data.

DISCUSSION

The results were obtained by taking 70% of the data, further divided into a second set of 70, 15 and 15%, and used for training, validation and testing. The last 30% of the original data was used for verification, i.e. by presenting the model with the second last 15% which was not used for training and assessing how the output from the models compares with the last 15% of the original data, as explained in section 2.5. The results from each of the models were organized and assessed in terms of the magnitude of the statistical error between the forecasted result and the real measured data. This was achieved by measuring the average of the squares of errors (MSE) and the total sum of the squared errors (SSE), for each model. The procedure was repeated for the two stages of data analysis, during training and upon verification, for *one-point per sliding window* and for a sample of *whole sliding window* and the overall error magnitude, as shown in Tables 1-6. The sliding window concept is explained; when data is converted into lagged variables, they form sliding windows of different sizes depending on the required inputs and outputs of the model.

To conduct 'mass' forecasting, the new inputs to the network must be in the form of the training inputs (same column size). In the same way, the outputs from the model have the same column size as the target matrix/vector. The success of the models was realized by measuring the relationship between the measured versus the model outputs (MSE & SSE).

In general, for each of the three types of models (Feed Forward, Jordan Elman and Cascade forward): 4 similar models (topologically) were built, corresponding to four forecast horizons: hourly, daily, weekly and monthly forecasting, of both wind speeds and directions. As an overall observation, the mean square error and the sum square errors, which were used as the convergence criteria were relatively lower during training, but shot up steadily upon simulation with new inputs.

To obtain good results with neural networks, data quantity is as important as data quality. A large amount of data is needed for training of the models. For this study two years data, seemed to limit the possibility of the models to adapt well and to develop accurate rules for generalization. It is possible that the relatively low quality results from both wind speed and direction models were as a result of the limited data quantity.
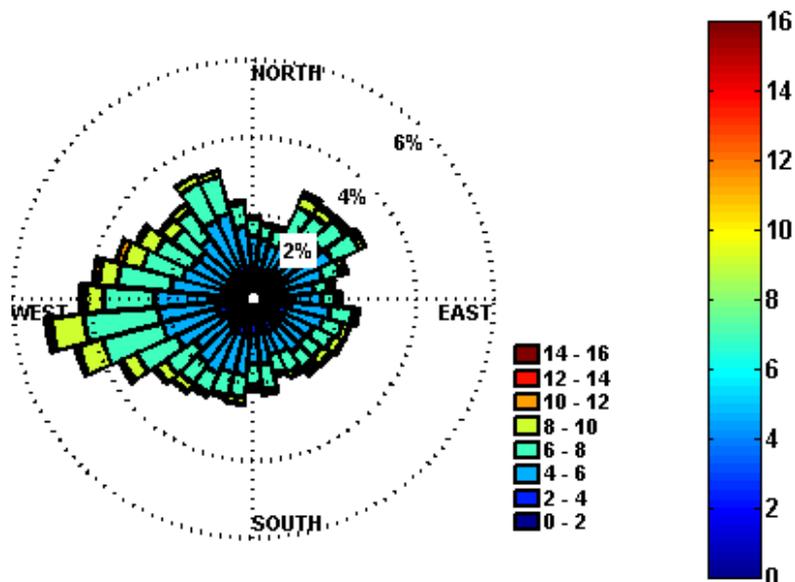

Figure 12. Plotted wind rose showing the prevalent wind speeds and directions in Puumala, Finland

The data quality and quantity used in the study were represented on a wind rose. Wind rose is a graphical representation of wind speeds and directions distribution for a particular location. Colour maps are usually used together with wind roses to give a quantitative feeling of the overall data distribution. Cool colours represent low values of the variables while warm colours represent medium values; hot colours shows peaks or highest values. Figure 12 is a wind rose representation for wind speeds and directions in Puulama, Finland. Figure 13 is a histogram showing wind speeds statistical distribution.
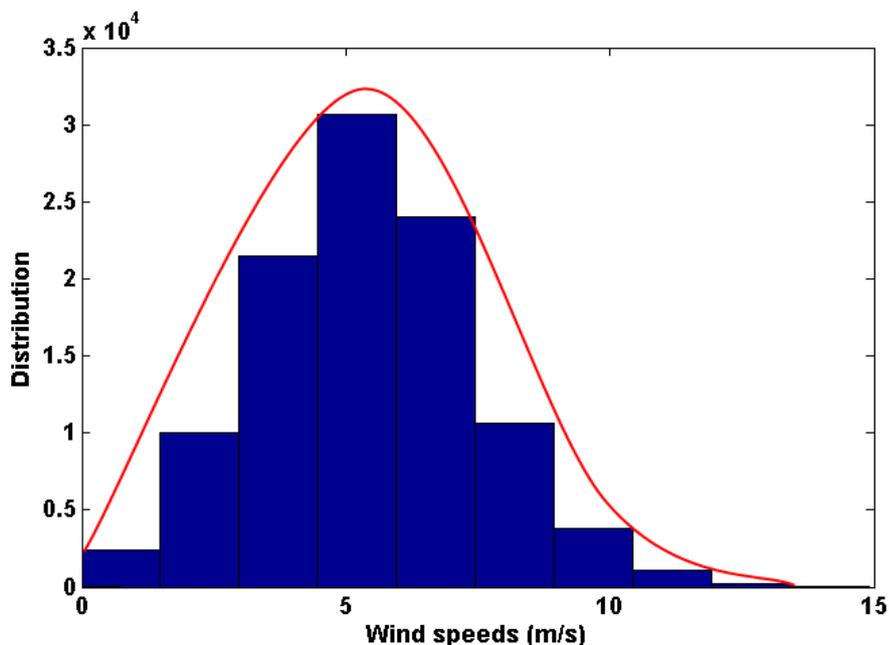

Figure 13. A histogram of wind speeds distribution

CONCLUSIONS

Quantitatively, based on the models' generalization ability, considering long-term and short-term forecasting, and by using both mean square error and sum squared error as the convergence criteria, the feed forward neural networks (FFNN) emerged as preferable type of models that may be used both for short-term and long-term wind speed forecasting, amongst other models tested. FFNN returned the lowest generalization error for 5 out of the 8 models built for wind speeds forecasting. On the other hand, cascade forward neural networks (CFNN) proved to be a better choice among the rest when applied for wind direction forecasting. CFNN returned the lowest generalization error in 5 out of the 8 models built for wind directions forecasting.

Qualitatively, hourly forecasting of wind speeds with FFNNs consistently returned the lowest generalization error both in the short term and in the long run. This adds up to the conclusions made by various researchers in the past. However, for wind directions CFNNs, which has less often been used compared to FFNN, returned the lowest generalization error when used both for weekly and monthly forecasting of wind directions. On a per-forecast-horizon basis, FFNNs returned the lowest generalization errors for hourly, weekly and monthly forecasts; while JENNs returned the lowest errors when used for forecasting of daily wind speeds. CFFNs gave the lowest errors when used for forecasting daily, weekly and monthly wind directions; while JENNs proved to be the best when used for hourly forecasting of wind directions. In addition, a combination of hyperbolic tangent transfer functions for both hidden and output layer returned better results for most of the models that were used for forecasting in this study.

Even though normalization would have reduced the range of the two sets of data; there is still a larger range between direction measurements, compared to those of speeds, even after normalization. It can be seen therefore, it is more difficult for the neural networks to train the sets of data with a large range in between, compared to training one with relatively small range. As a result, none of the models built can vividly be said to possess the ability to forecast wind directions, and thus opening up an opportunity for further research in this context. Nevertheless, FFNNs returned the lowest generalization errors for hourly, weekly and monthly forecasts; while JENNs returned the lowest errors when used for forecasting of daily wind speeds. CFFNs gave the lowest errors when used for forecasting daily, weekly and monthly wind directions; while JENNs proved to be the best when used for hourly forecasting of wind directions (Tables 5 and 6). In addition, a combination of hyperbolic tangent transfer functions for both hidden and output layer returned better results for most of the models that were used for forecasting in this study. All data were normalized to a range between 0 and 1; a logistic transfer function would have been expected to have a better performance on the data. On the contrary however, from the tests, a combination of hyperbolic tangent transfer functions for both hidden and output layer returned a relatively low error for most of the models.

However neural networks may be used to forecast natural phenomena e.g. wind speeds and directions, their 'intelligence' is limited to a relatively progressive change in the unique factors/rules developed and used by the networks during training. For instance, the training data of wind speeds and directions collected over a period of say 5 years can only be used for forecasting as long as the human, physical and environmental factors e.g. surrounding forests, buildings, terrain, etc., remain as is, or with minimal and gradual changes. This limits the use of implemented neural networks, as it would require re-training and review of relevant codes. This not only affects the neural networks used in ecological modelling but also many other research fields as well, and thus further research is called for in this area of study [18].

With respect to wind energy planning specifically for the region under study, wind speed forecasting models seemed to produce relatively good results but only for shorter horizons (~ 6 hours) compared to those of wind directions; wind directions seemed accurate for a longer future period (~ 24 hours). In general the wind directions were skewed towards the western side, with a range between 235 and 300 ° measured from due north, while wind speeds were normally (Gaussian) distributed between (0 to 16 m/s), with 6-12 m/s as the persistent speeds for well over half of the test period (Figures 13 and 14). According to Aapo Koivuniemi an expert at TuuliSaimaa Oy, a Finnish company specializing in wind power production, produced electricity is naturally site and turbine specific. With the Finnish feed in tariff and typical modern approximately 110 m diameter rotor with 3 MW nominal power turbines, the very easiest sites can be profitable with about 6 m/s mean speed at 100 m height. Normal inland site might need a minimum of about 6.5-7 m/s to be an attractive investment opportunity. As for offshore, it makes a whole difference, because turbine foundations can become much more expensive (up to 2-3 times of the turbine price), and thus even 9 m/s speeds may not be enough to break even [19]. As Puumala lies along the shoreline, it can easily be concluded therefore, that the location was strategic and wind speeds were consistent, sufficient and reliable for considerable wind power generation.

NOMENCLATURE

H	Number of hidden layers in a neural network
N	The number of samples of data in error measurements
R	Coefficient of Correlation, *dimensionless fraction*

Greek letters

μ	Neural network momentum coefficient, *dimensionless constant*
η	Neural network learning rate, *dimensionless constant*
∞	Infinity
ζ	Neural network transfer function

Abbrevations

ABL	Atmospheric Boundary Layer
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARX	Autoregressive eXogeneous
BBP	Batch Back Propagation
BR	Bayesian regulation/regularization
CFFN	Chosen model
CFNN	Cascade Forward Neural Networks
FFNN	Feed Forward Neural Network
JENN	Jordan Elman Neural Network
L	Long term
LUT	Lappeenranta University of Technology
MATLAB	Matrix Laboratory
MLFFNN	Multi-Linear Feed Forward Neural Network
MM5	Fifth-generation Meso-scale Models
MSE	Mean Square Error
MSEt	Mean Square Error upon training
MSEv	Mean Square Error upon verification
S	Short term
SSEt	Sum Square Error upon training
SSEv	Sum Square Error upon verification

REFERENCES

1. Kavzoglu, T., Determining Optimum Structure for Artificial Neural Networks, *25th Annual Technical Conference and Exhibition of the Remote Sensing Society (RSS)*, September 8 - 10, pp. 675-682, Cardiff, UK, 1999.
2. Lange. M., *Analysis* of the uncertainty of wind power predictions, *Doctoral dissertation*, University of Oldenburg, Dept. Math & Natural Sciences, Germany, 2003.
3. Kwun, J.H., Kim, Y-K., Seo, J-W. & Jeong, J. H., Sensitivity experiments for winds prediction with planetary boundary layer parameterization, *Scientific and Technical Symposium on Storm Surges* (JCOMM2007SSS), Oct. 2 - 6, Seoul, Korea, 2007.
4. Focken, U., Lange, M., Waldl, H. P., Previento - A Wind Power Prediction System with an Innovative Up-scaling Algorithm, *Proceedings of EWEC*, pp. 826-829, June 2 - 6, Copenhagen, Denmark, 2001.
5. Basheer, I.A. and Hajmeer, M., Artificial neural networks: fundamentals, computing, design and application, *Journal of Microbiological Methods*, Vol. 43, pp. 3-31, 2000., http://dx.doi.org/10.1016/S0167-7012(00)00201-3
6. Karlik, B. and Olgac, A., Performance analysis of various activation functions in generalized MLP architectures of neural networks, *International Journal of Artificial Intelligence and Expert Systems,* Vol. 1 (4), pp. 111-122, 2010.

7. Nagendra, S. M and Khare, M., Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions, *Ecological Modelling*, Vol. 190, pp. 99-115, 2006., http://dx.doi.org/10.1016/j.ecolmodel.2005.01.062

8. Ghaffari, A., Abdollahi, H., Khoshayand, M. R., Bozchalooi, I. S., Performance comparison of neural network training algorithms in modelling of bimodal drug delivery, *International Journal of Pharmaceutics*, Vol. 327, pp. 126-138, 2006., http://dx.doi.org/10.1016/j.ijpharm.2006.07.056

9. Lisbôa, A., Pimentel, W., Martignoni W., Modelling fluid catalytic cracking with multivariate Statistics and artificial neural networks, *4ᵗʰ Mercosur Congress on Process Systems Engineering* (ENPROMER2005), Aug. 14-18, Rio de Janeiro, Brazil, 2005.

10. Lee, H.-M., Huang, T.-C., Chen, C.-M., Learning efficiency improvement of back propagation algorithm by error saturation prevention method, *International Joint Conference on Neural Networks*, 41 (3), pp. 1737-1742, 1999.

11. Lee, Y., Oh, S.H., Kim, M., An analysis of premature saturation in Back propagation learning, *Neural networks*, Vol. 6, pp. 719-728, 1993., http://dx.doi.org/10.1016/S0893-6080(05)80116-9

12. Nguyen, D. and Widrow, B., Improving the Learning Speed of 2-layer Neural Networks by Choosing Initial Values of the Adaptive Weights, *Proceedings of International Joint Conference on Neural Networks* (IJCNN), July 14 – 17, Vol. 3, pp. 21-26, San Diego, California, 1990.

13. Moallem, P. and Ayoughi, S. A., A complementary method for preventing hidden neurons' saturation in feed forward neural networks training, *Iranian Journal of Electrical and Computer Engineering*, Vol. 9 (2), pp. 127-133, 2010.

14. MathWorks, version 7.14.0.739 (R2012a), The MathWorks Inc., Natick, Massachusetts, USA, 2012.

15. Pavelka, A., and Procházka, A., Algorithms for Initialization of Neural Network Weights, *Sborník príspevku 12 rocníku konference MATLAB*, Vol. 2, pp. 453-459, 2004.

16. Fahlman, S. E., *An empirical study of learning speed in back-propagation networks*, Technical report, CMU-CS-88- 162. Pittsburgh: Carnegie Mellon University, USA, 1988.

17. Hassoun, M. H., *Fundamentals of Artificial Neural Networks*, Cambridge: The MIT press, USA, 1995.

18. Rotich, N., *Forecasting of wind speeds and directions with artificial neural networks*, Master's thesis, Lappeenranta University, Finland: Department of Energy Technology, 2012.

19. Koivuniemi, A., Aapo.koivuniemi(at)tuulisaimaa.fi, Re: Following up our previous conversation on wind energy. [Email] Message to Rotich, N. (nicolus.rotich(at)lut.fi), Sent Tuesday, April 16, 2013 3:54 PM, 2013.