Using Reservoir Computing for Forecasting of Wind Power Generated by a Wind Farm

Bruna Cavalcanti Galle de Aguiar and Mêuser Jorge Silva Valença
Polytechnic School of Pernambuco
University of Pernambuco
Recife, Brazil
Email: {bcga, meuser}@ecomp.poli.br

***Abstract*—One of the main challenges today is the growing global energy demand. In order to meet this need, the most widely used energy sources are oil, natural gas and coal. The main problem with these sources is due to the fact that, besides being extremely polluting, they are non-renewable sources. Therefore, renewable sources are becoming essential for humanity. Among many of them, the wind is the most promising choice. Wind farms have their potential directly related to the wind power, which requires good estimates of this variable in order to build effective strategies and plans. However, this task presents great difficulties due to the complex characteristics of the wind, such as the high variability of its velocity and direction. This paper aims to use the technique of Reservoir Computing for the prediction of wind power generated by a wind farm and compare its performance with the one produced by the Multi-Layer Perceptron, another type of artificial neural network and the most widely used for this purpose. At the end, it will be possible to analyse the results and conclude which one is more appropriate for predicting wind power.**

Keywords*—*Reservoir computing; forecasting of wind power, artificial neural network, MLP.

I. INTRODUCTION

One of the major challenges today is the growing global energy demand. In order to meet this need, the most widely used energy sources are petroleum, natural gas and coal. The main problem with these sources is due to the fact that, besides being extremely polluting, they are non-renewable sources, i.e., will be exhausted from nature within a few years. According to the International Energy Agency (IEA), if we do not reduce the average consumption recorded in recent decades, the world reserves of oil and natural gas will be exhausted in 100 years and those of coal in 200 years [1]. Thus, the use of renewable energy sources has become essential. They will also help to combat environmental degradation.

Renewable energy, for the reasons cited above, is becoming increasingly important to humanity. The main advantage of renewable energy are: it is clean, safe, abundant and, therefore, does not impact the environment in a negative way. Among the various sources available in the world, the wind is the most promising choice. This is explained due to its constant availability anywhere and its production is now considered cost competitive [2].

Due to the randomness of wind generation, it is not possible to guarantee a fixed amount of energy to the electrical system. In addition to that, there is the increasingly high investment of several governments in this type of energy in order to meet the high power consumption in recent years. Thus, in order to help countries whose energy matrix now includes the wind as an alternative source, the forecast wind power has proven to be crucial for developing strategies and appropriate, efficient and inexpensive planning. This forecasting depends mainly on wind power. Various models are used to perform it, several of them including artificial intelligence. This work aims to use an architecture of Artificial Neural Network (ANN) called *Reservoir Computing* (RC) and analyse its performance.

Although there are already models using the most common types of ANNs, such as Multi-Layer Perceptron (MLP), the *Reservoir Computing* was chosen for having an architecture in which artificial neurons are interconnected and organized in a more similar way to the human brain (a metaphor that is the origin of ANNs, as the name implies)[3].

This characteristic of the RC architecture allows this technique to represent systems with dynamic behaviour, which are difficult to be represented in neural networks, such as MLP [4]. Therefore, learning the dynamic characteristics which represent the temporal series of the wind power becomes a more suitable task for the RC.

Due to this, it is expected that its performance on forecasting is better than that obtained by the other ANNs. Hence, the prediction would be more accurate and increase efficiency in the planning and use of wind energy, encouraging its use in many place and, at the same time, preserving the environment.

This paper is organized as follows. Section II introduces the Reservoir Computing technique, its structure and how it is created, used and trained. Section III presents the methodology used during this work, such as the database used and the ANNs configurations. Results can be found in Section IV. Conclusions and future works are given in Section V.

II. RESERVOIR COMPUTING

In addition to feed forward architectures, such as the MLP, widely used in time series forecasting, Recurrent Neural Networks (RNN) began to emerge. In this new type of network, there is the addition of recurrent connections to existing feed forward architectures. These connections transform the system into a complex dynamic system and one more suitable for solving temporal problems. In the case of this project, it becomes an attractive option because the problem to be solved, the forecast of wind power, is temporal in nature. Figure 1 shows the structure of a feed forward and recurrent network [5].

The RNNs are computational models capable of creating internal memory required to store the history of input patterns through their recurrent connections [5].

In 2001, a new proposal for the design and training of RNNs was suggested independently by Wolfgang Maass [6] called Liquid State Machine (LSM) and Herbert Jaeger [7] called Echo State Networks (ESN). Verstraeten proposed the unification of these two approaches into a single term called Reservoir Computing. Since then, RC began to be adopted in the literature as a generic name for learning systems that consist of a recurrent network dynamics with simple computational nodes combined with a simple output function [8].

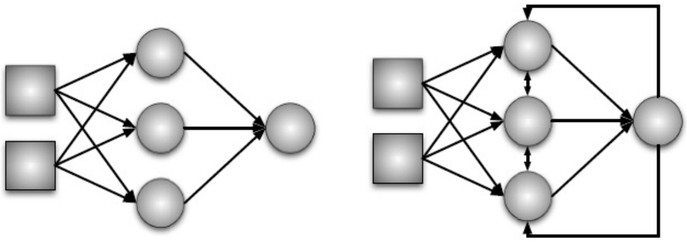

Fig. 1: Structure of a feed forward (left) and recurrent network (right)

[Fonte: [4]]

A reservoir computing system consists of two main parts: a reservoir and a linear output layer. The reservoir is a non-linear dynamical system with a recurrent topology composed of processing nodes. The connections between nodes are randomly generated and are globally rescaled in order to achieve a proper dynamic state. An important property of the RC is that the reservoir has fixed weights, that is, its training is not necessary. Only the output layer is trained, therefore, it has an output function. This function can be, for example, a classifier or linear regression algorithm [4].

The fact that only the output layer needs to be trained allows the use of the same reservoir for the solution of different tasks simultaneously by keeping the same inputs.

An interesting feature of RC is one based on ESNs, a property called echo. This property defines the effects of a previous state x (n) and an input value at a future state x (n +1) should gradually decrease with the passage of time k (i.e. k $\rightarrow$ infinity) and should not persists, or be amplified.

Due to recurrent connections, information on past entries is stored in the network. Because of this, the network contains a rich set of non-linear transformations and mixtures of input signals of past and present (called echoes) times.

A. Creating and Using Reservoirs

In the following text, it is assumed that the RC system consists of N reservoir nodes, M inputs and P outputs.

1) Creating the input and reservoir connections:

1) Construct an MxN input to reservoir weight matrix W_{in}. The weights are drawn from a random distribution or discrete set. If all input signals are fed to all reservoir nodes, then all elements of this matrix are non-zero. Otherwise, there will be null elements.
2) Construct an NxN reservoir interconnection weight matrix W_{res}. The values for the weights are again drawn from a distribution or a discrete set of values (e.g.,[-1, 1]).
3) Rescale the weight matrix globally such that the reservoir has a suitable dynamic excitability. The most common way to do this is to tune the spectral radius of W_{res}. The spectral radius of a matrix is its largest absolute eigenvalue. A value close to 1 is usually proposed as a good starting point for optimizations of ESNs.

2) Simulating the Reservoir and training and testing read-out:

1) Construct a dataset D and split it in 3 sets: training, cross validation and tests.
2) The network state at time k is denoted as x[k] and an input at the same time as u[k]. For every sample, we initialize x[0] = 0. Before starting the training, the first 100 cycles are called warm up. During the warm up, the neural network forgets its initial states and loses the influence of value zero. As soon as the warm up finishes, the training is started and the neural network is simulated recursively.
3) After every sample is simulated, state matrices of the training set are concatenated into a large state matrix A.
4) Compute the output weights. In this project, to perform this calculation, we used the Moore-Penrose [9] generalized matrix or pseudo-inverse due to the fact that the matrix A is not square. These calculations will be done automatically by a Java routine called JAMA.
5 After a sample entry is trained, cross-validation is performed in order to check if the training can now be finalized. The Mean Square Error (RMSE) is calculated and stored for each cross-validation done.
6) After training, the network is simulated with the test set and the Mean Absolute Percentage Error (MAPE) is calculated. These errors are stored for subsequent statistical tests. In this project, we used the Normalized Mean Absolute Error (NMAE).

Through the NMAEs calculated for network performance with the RC and MLP techniques, it will be possible to compare which architecture is the best choice for the prediction of wind power.

III. METHODOLOGY

A. Database

The database used in the experiments was provided by the brazilian Operador Nacional de Sistena Elétrico (ONS) or National Operator of Electrical System. The ONS is the body responsible for the coordination and control of the operation of generation and transmission of electricity in the national interconnected system [10].

Data for average wind power are daily and the period in which they were measured and collected goes from December 1, 2011 until July 31, 2012. They were observed from 30 to 30 minutes and the available data are: average wind speed, direction and power. From all these attributes, we will only use the power of the wind. Through experiments, it was noted that only this variable achieves good results.

Each wind farm has an installed power capacity associated with it. This value indicates the maximum power that can be produced by the farm. The database used belongs to a wind farm with a capacity of 54.61 MW.

B. Pre-processing of data

The first step in the stage of pre-processing data is the normalization of values. This step aims to prevent high values from influencing too much the calculations of the ANN while low values go unnoticed. It is necessary to ensure that the variables at different intervals receive equal attention during the training. Moreover, the variables should have their values proportional to the boundaries of the activation function used in the output layer. If the activation function chosen is the logistic sigmoid, their values are limited between [0 and 1], then the data are usually normalized between [0.10 and 0.90] and [0.15 to 0.85] [11].

The normalization is calculated using the formula described in (1):

$$y = \frac{(b-a)(x_i - x_{min})}{(x_{max} - x_{min})} + a \tag{1}$$

where:

- y = normalized value;
- x_i = original value;
- x_{min} = minimum value of x;
- x_{max} = maximum value of x;
- a e b = limits chosen. In this work, a = 0.15 e b = 0.85.

C. Measure Network Performance

In case of wind power prediction, the usual error descriptors, such as MAPE and Mean Absolute Error (MAE), are given as a percentage of the installed capacity of a particular wind farm. In this work, it was defined that the network performance would be measured by the Normalized Mean Absolute Error (NMAE).

D. Predicting Wind Power with MLP

Although it is widely used in many researches, the MLP requires that several of its parameters are configurable and the choice of each directly influences the final outcome of the prediction.

Below are the main parameters of the MLP and Backpropagation algorithm:

- Number of neurons in the input layer;
- Number of neurons in the hidden layer (only one hidden layer);
- Number of neurons in the output layer;
- Activation function;
- Stopping criterion;
- Learning Rate;
- Momentum.

The number of entries varied in order to be possible to make an analysis of the impact of a larger amount of inputs, like 48 or a smaller number of inputs, such as 7 and if this alteration made any difference in the wind power prediction. As noted, it did not matter if the number of neurons in the input layer was 7 or 48. For the purpose of expediting the training, we chose the value of 7 inputs.

The output is always 48 wind power values, thus the predictions are one day ahead.

The algorithm used is the Backpropagation and the activation function chosen for the neurons is the logistic sigmoid. This function returns values in the interval [0, 1].

The stopping criterion used was cross-validation, with 50% of the set of values for training, 25 % for cross-validation and the remaining 25 % for testing.

Several tests were performed to define the learning rate, momentum and number of neurons in the hidden layer. The best results correspond to the values of 0.8 for the learning rate, 0.2 for the momentum and 6 neurons in the hidden layer.

The MLP used has been implemented in the JAVA programming language and in the Eclipse development environment [12].

E. Predicting Wind Power with RC

Like the MLP, the Reservoir Computing technique has several parameters that require configuration. Taking into account that it is a recent area of research, the choice of these settings can not be considered ideal and is often performed randomly. One way to do this is to evaluate each chosen value and determine if it was better or worse for the network performance. This process is repeated until a value is considered optimal, which does not necessarily means the best.

Below are the parameters whose settings were required to be defined during this project:

- Number of neurons in the input layer;
- Number of neurons in the output layer;
- Number of neurons in the reservoir;
- Activation function of the reservoir;
- Activation function of the output layer;
- Initialization of weights;
- Connection rate of the reservoir;
- Number of warm up cycles;
- Stopping criterion.

The number of inputs remains the same as used in the MLP, since it is necessary to keep this parameter with the same value of the prior neural network in order to perform statistical tests. The same applies to the number of outputs. The RC will have 48 neurons in the output layer to predict one day ahead.

The number of neurons in the reservoir is one of the parameters for which there is no fixed criterion that defines it. It was chosen randomly after checking the NMAE at the end of each training. It was observed that the ideal number of neurons in the reservoir was 20.

As mentioned in Section II, the weights of the input layer to the reservoir and the weights of the reservoir are randomly generated from a random distribution.

The reservoir states are initialized to zero (0). Because of this, as also mentioned in Section II, it was decided to add to the network a phase called warm up. During the warm up, it is not necessary to find the weights of the output layer, or to calculate an output value. This warm up phase is done just to update the states of the reservoir and remove the dependence on the initial state. The number of cycles chosen for warm-up was 10.

The connection rate of the reservoir neurons was 20%. That is, only 20 % of the connections have weight values different from zero associated to them.

The stopping criterion used was also cross-validation, with 50% of the set of values for training, 25 % for cross-validation and the remaining 25 % for testing.

The activation function chosen in the reservoir was the logistic sigmoid. In the output layer, the selected function was linear one.

During this work, we implemented a Neural Network with the technique of RC in the programming language Java and the Eclipse development environment. Figure 2 is a synthesized form of how this ANN works.

F. Statistical Tests

After 30 trainings with each type of neural network [13], statistical tests were performed to assess which technique has the best performance in the prediction of wind power or if their results can be considered statistically equivalent.

Among various tests in the literature, there are the t-student

```
Set the number of neurons in the input layer ;
Set the number of neurons in the reservoir layer ;
Set the number of neurons in the output layer ;
Randomly generate the weights of Win matrix between
  -1 e 1;
Randomly generate the weights of Wres matrix between
  -1 e 1;
Normalize the weights of Wres matrix so that the
  spectral radius of the matrix is smaller than or equal to
  1;
while until the end of the number of warm up cycles do
  | updates the states of the neurons of the RC;
end
while until the stopping criterion is reached do
  | for each value of the input set do
  |   | updates the states of the neurons of the RC;
  | end
  | Calculates the Moore-Penrose matrix inverse to find
  | the weights connecting the output layer to the RC;
  | for each value of the cross-validation set do
  |   | updates the states of the neurons of the RC;
  | end
  | Calculates the output values of the RC;
  | Calculates the RMS;
  | Checks if the stopping criterion has been reached;
end
for each value in the set of tests do
  | updates the states of the neurons of the RC;
end
Calculates the output values of the RC;
Calculates the NMAE// in case of wind power
  prediction;
```

Fig. 2: Pseudo-code of RC

test and the Wilcoxon Rank-Sum test. To apply t-student test, it is necessary to begin using the Shapiro-Wilk test in order to check whether or not the data are normally distributed. If they are considered normal samples, then we performed an F test to verify if the variance can be considered originated from the same population. By passing these tests, we applied t-student test for unpaired data. If samples are not considered normal, nor pass the F test, then we apply the Wilcoxon Rank-Sum test.

IV. RESULTS

The training of the MLP and the RC was conducted with three settings, as mentioned in the previous section, until the best result was achieved. The following three settings were tested: 7, 24 and 48 entries. As it shown in table I, the NMAE did not vary significantly if the number of inputs were altered.

TABLE I: Executions for determining the ideal number of inputs

Number of input neurons	RC	MLP
7	18.04%	23.54%
24	18.47%	24.03%
48	18.23%	24.07%

30 simulations were performed for each topology and the average NMAEs can be found in Table II.

TABLE II: Average Normalized Mean Absolute Errors Values

Topology	Average NMAE
RC	18.02 %
MLP	24.47%

In addition to calculating the average normalized mean absolute error, a graphic was constructed to better show the behaviour of the expected values when compared to the ones obtained from the training of a RC neural network.

Analysing Figure 3, it is possible to observe that, even if the average normalized mean absolute error is low and considered a better result when compared to other techniques, the curve of the calculated values do not approach the one with the real values. When it is desired to predict the maximum power (peaks) or the minimum (the valleys), these values are not well forecast and it can jeopardize the planning and decision making of a wind farm.

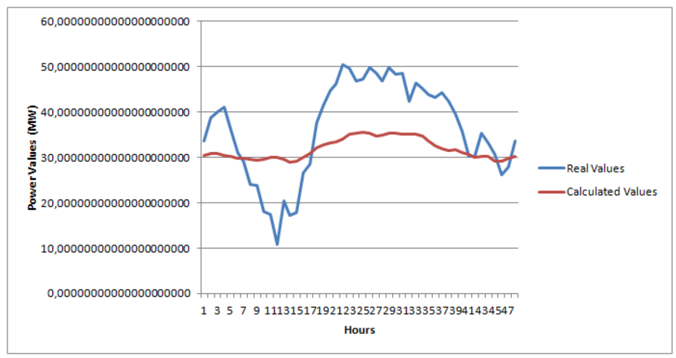

Fig. 3: Graphic comparing the curves of the calculated values with the real ones

Finally, for each set of 30 simulations, we performed statistical tests and the process began with Shapiro-Wilk test. The R software was used [14] and it defines the value for the significance level as 0.05. It was observed that the sample belonging to MLP network does not come from a normal population, i.e., that is not normally distributed, since the p-value calculated during the test was 0.005 for the MLP, thus less than the level of significance.

From this result, it became unnecessary to perform the Student t test, whose pre-condition for its use is that the samples were normal. For this reason, the next step was the application of Wilcoxon Rank-Sum test.

This test is non-parametric, that is, it is assumed that the data distribution is not regular. The result of this test showed that the p-value has a value much smaller than the significance level, hence, the null hypothesis is refuted, which states that the performance of neural networks with the RC and MLP architectures are considered statistically equal and concludes that the use of the proposed technique RC has a better performance.

V. CONCLUSION AND FUTURE WORK

This work aimed to predict the power generated by a wind farm for 1 day ahead using the technique of Reservoir Computing and compare it with the results provided by other existing prediction models. Since the MLP neural network is the most widely used in this type of application, it was the model chosen and its results were compared with the results of the RC technique.

In order to achieve this objective, a neural network with the RC topology was implemented and a database provided by the brazilian Operador Nacional de Sistema Elétrico (ONS) or National Operator of Electrical System was used. Several simulations were performed for both topologies and the results were compared.

Through statistical tests, it was proven that the performance with the RC is better than the one obtained by the MLP. This result opens a field of research for the study of dynamic neural networks, such as the RC for time-series forecasting.

As future work, further studies of the parameters of the RC will be conducted in order to find the best way to define them. Good and more accurate values can positively impact the RC's performance.

In addition to that, it is necessary to work with databases from across the country and the world. With different databases, it is possible to observe if the RC technique can still be considered the best solution.

Finally, since the graphic showed discrepancy between the calculated values and the real ones, a correction technique will be employed to ensure a more realistic representation of the real power curve.

REFERENCES

[1] W. de Cerqueira e Francisco. (2008) Energy sources [retrieved: August, 2012] [Online]. Available: http://www.mundoeducacao.com.br/geografia/fontes-energia.htm

[2] R. Albadó, *Wind Energy*, 1st ed. ArtLiber, 2002.

[3] B. Schrauwen, D. Verstraeten, and J. V. Campenhout, "An overview of reservoir computing: theory, applications and implementations," *Proceedings of the 15th European Symposium on Artificial Neural Networks*, 2007, pp. 471–482.

[4] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network trainning," *Computer Science Review*, vol. 3, 2009, pp. 127–149.

[5] A. F. Araújo, "A method for design and training of reservoir computing applied to time-series forecasting," Ph.D. dissertation, Federal University of Pernambuco, 2011.

[6] W. Mass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, 2002, pp. 2531–2560.

[7] H. Jaeger, "The ''echo state'' approach to analysing and training recurrent neural networks," 2001.

[8] D. Verstraeten, "Reservoir computing: computation with dynamical systems," Ph.D. dissertation, Ghent University, 2009.

[9] J. A. Fill and D. E. Fishkind, "The moore-penrose generalized inverse for sums of matrice," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, 1999, pp. 629–635.

[10] ONS. (2012) Operador nacional do sistema elétrico (national operator of electrical system) [retrieved: October, 2012] [Online]. Available: http://www.ons.org.br/home/

[11] M. J. S. Valença, *Fundamentals of Neural Networks*, T. Pereira, Ed. Livro Rápido, 2011.

[12] S. C. O. M. dos Santos, "A hybrid system based on neural network and ant colony," Master's thesis, Universidade de Pernambuco, 2010.

[13] N. Juristo and A. M. Moreno, *Basics of Software Engineering Experimentation*. Kluwer Academic Publisher, 2001.

[14] W. N. Venables, D. M. Smith, and the R Core Team, *An Introduction to R*, 2008.