International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 7 Issue 4, April - 2023, Pages: 1-3

Wind Speed Forecast Tool for Wind Power Plants in Madagascar

ANDRIAMAHITASOA Bernard Andriamparany, RAKOTOARIMANANA Liva Graffin, MANDIMBY Junior Zoë Jean Tigana, RAFANJANIRINA Eulalie Odilette, RANDRIAMANANTANY Zely Arivelo, RAKOTOMALALA Minoson

Institute for Management of Energy (IME), University of Antananarivo, PB 566,
Antananarivo 101, Madagascar
Email: b.andriamahitasoa@gmail.com

Abstract: *Since wind energy is an exploitable source, its exploitation requires knowledge about reserve of load capacities. Upcoming load capacities can be available thanks to the data from measuring stations in the region of Antananarivo. This paper aims at presenting non-linear short-term forecasting techniques of wind speed. It proposes new design model based on Artificial Neural Networks (ANNs) techniques for wind speed forecast. For the wind speed forecast, Classical neural networks, Bayesian neural networks and Gaussian process models is used. Each model runs with two year-wind speed data training and one-year data for testing. The Gaussian process model has shown the best performance and it can predict 3-day-horizon wind speed during one year with less than 29% accuracy. This model can be a useful tool to develop wind power plants, particularly in Madagascar.*

Keywords— Bayesian neural networks, classical neural networks, Gaussian process, wind speed forecasting

1. INTRODUCTION

Wind energy is among the most promising widespread sources which can help face energy need, especially in the Island of Madagascar, mostly in the north-western region and south-east coasts. Hybrid source with existing interconnected networks could meet the energy needs of these regions since the use of this renewable energy has various advantages.

The exploitation of this energy source requires knowledge of its reserve capacity. In most cases, this is not available because there are no measuring stations. This study aims at presenting non-linear forecasting techniques of the wind speed in order to integrate it into the existing Interconnected Network (IR) on the island. Artificial neural networks (ANNs) are widely used tools in forecasting. It is an interesting alternative to traditional statistics for existing data processing. Two of these techniques consist of the use of Bayesian model Neural Networks (NNs) and Gaussian Process (GP) [1], [2].

This paper develops an adaptive short-term wind speed prediction scheme using NN and GP. These prediction methods allow to find wind speed evaluation at time ($t + \Delta t$). For the training, 1095-days wind speed data between 2003 and 2005 in Antananarivo is used. These collected data come from the Geophysical Institute and Observatory of Antananarivo (IOGA).

2. STUDY CONTEXT

Neural networks are the nonlinear models used for regression [3]. They are flexible models even when designing an NN for solving particular application, [4], [5]. NNs can approximate any continuous function to an adequate accuracy if the number of hidden neurons is sufficient [6].

In this study, to estimate the wind speed, the first method consists in the use of neurological approach which is based on classical technique and Bayesian inference and after, the Gaussian Process approach. In fact, MacKay developed the Bayesian method for NNs offering significant advantages over classical learning [7],[8]. However, it is important to make several approximations with Bayesian approach. By contrast, GPs are powerful methods for regression whereas most calculations are analytically feasible [9].

3. NEURAL NETWORK

3.1 Classical network

For Classical network, Fig. 1 shows NNs, the Multi-Layer Perceptron (MLP) structure. This MLP structure consists of an input layer, one or more hidden layers and an output layer. Input layer collects all vectors xi of the inputs of the model while the output layer reports that of y. In our case, y is the output that corresponds to the next day's wind speed forecast.

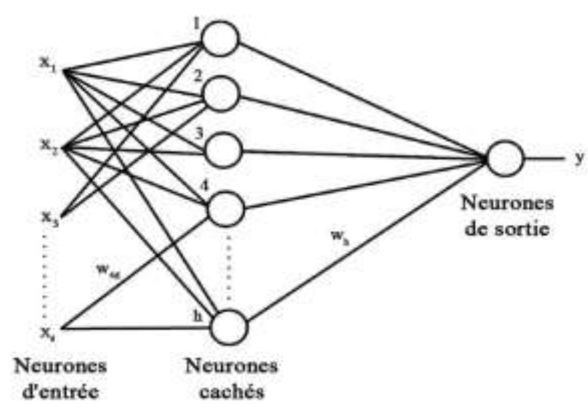

Fig. 1. *MLP with inputs xi, hidden neurons h and output y for the next day's wind speed forecast.*

Every neuron of a given layer, except the last layer emits a connection towards each neuron of the neighboring layer. Hidden layer neurons are characterized by activation function which is usually a hyperbolic tangent function:

$$f(x) = (e^x - e^{-x})/(e^x + e^{-x}) \quad (1)$$

Therefore, having x_i inputs and h hidden neurons, the linear output y is expressed by the equation:

$$y = f(v) = y(x_i; w) \quad (2)$$

Where V called "potential" which is related to the bias w_0 using the formula:

$$v = w_0 + \sum_{j=1}^{h} \sum_{i=1}^{d} w_{ji} x_i \quad (3)$$

These equations allow to estimate the parameters x_i of the NNs (whose weights are w_{ji}) during the learning phase. The generalization phase consists in evaluating the ability of the NNs to generalize which means giving correct predictions when confronted with new input data.

In the case of a complex model, the learning can lead to bad predictions and this is called "overfitting". However, the Bayesian approach offers advantages on this complexity control.

3.2 Bayesian approach

The Bayesian approach is used to determine the probability distribution function known as pdf. This pdf function represents the trust degrees taken through different values of weights. Neural parameters estimation of Bayesian inference consists in determining forward probability distribution, from approximate probability distribution and likelihood function by using Bayes rule according to equation such as, [8]:

$$P(a \setminus b) = P(b \setminus a)P(a)/P(b) \quad (4)$$

Thus, Bayesian methods report a complete distribution for the parameters of the NNs. Bayesian inference is based on these parameters:

- Data: Training (n samples) and testing (n^* samples)

$$D = \{x_i, t_i\}_{i-1}^{n} \text{ and } D^* = \{x_i{}^*, t_i{}^*\}_{i-1}^{n^*}\ t_i = y(x_i; w) + \varepsilon_i \quad (5)$$

- Model: $\varepsilon_i N(0, \sigma^2), \varepsilon_i$ follows the Gaussian distribution mean zero. Variance $\sigma^2 y$ is given by deterministic function vector x_i of inputs with additional Gaussian noise.

- Likelihood function: Like Gaussian noise, the likelihood function is also Gaussian,

$$p(t|x, w) = \left(\frac{1}{2\pi\sigma^2}\right) exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(t_i - y(x_i, w)\right)\right) \quad (6)$$

- Guessing function: *p(w)*

- Forward function: Bayes rule

$$p(w|x, t) = p(t|x, w)p(w)/p(t|x,) \quad (7)$$

- Forecasts: The forecasts x^* are all value averages with their weights according to their forward probability.

$$p(y^*|x^*, x, t) = \int p(y^*|w, x^*).p(w|x, t).dw \quad (8)$$

In case of NNs with large weight dimension, weight evaluation cannot be analytically performed. MacKay proposed approximation called obviousness framework to overcome this issue, [7].

3.3 Gaussian process

Regarding the NNs complexity and their implementation, Gaussian processes (GPs) are relatively recent developments for nonlinear models [9]. MacKay proposes the GPs fit to regression and approximation which can be done analytically. This is known as the Gaussian probability distribution generalization [7], [9]. A Gaussian distribution is defined by mean μ and covariance matrix $\sum$ such as:

$$\sum efg = (f_1, f_2, \dots, f_n) \sim N(\mu, \Sigma) \quad (9)$$

$$k(x, x) e\ g\ f(x) \sim GP\big(m(x), k(x, x)\big) \quad (10)$$

$$cov\left(f(x_p), f(x_q)\right) = cov(x_p, x_q) = exp\left(-\frac{1}{2}\left|x_p - x_q\right|^2\right) \quad (11)$$

GPs calculation formulation implies:

- Data: Training and testing

$$D = \{x_i, t\}_{i=1}^{n} \text{ and } D^* = \{x^*, t^*\} \quad (12)$$

- Forecasts: Gaussian distribution forecast for the test.

$$y^*|x^*, x, t$$

$$N\begin{pmatrix} \bar{y}^* = k(x^*, x)[k(x, x) + \sigma^2 I]^{-1} t, \\ cov(y^*) = k(x^*, x^*) - k(x^*, x)[k(x, x) + \sigma^2 I]^{-1} k(x, x^*) \end{pmatrix} \quad (13)$$

- Covariance matrix: with n training samples and n* of test. $k(x, x^*)$ represents the covariance matrix $n\ x\ n^*$

4. RESULTS

To assess the performance of the model, the code is run to evaluate output MAE and RMSE errors. 730 days wind speed data used to train the models and the remaining 365 days data for testing. Table 1 displays errors and Central Processor Unit (CPU) time comparisons from the model's training and test sets. Thus, first row of Table 1 shows that fitting is better both on learning and on testing for classical model made by 10 hidden neurons with 1-day horizon. For the Bayesian approach and GP, similar structure and horizon are implemented.

Table 1*:* Model's error and CPU time comparisons for training and test sets

Model	RMSE Training	MAE Training	RMSE Test	MAE Test	CPU Time (s)
Classic NN	0.2364	0.1881	0.2678	0.2188	1.77
Bayes NN	0.1870	0.1542	0.2119	0.1722	7.16
GP	0.1874	0.1544	0.2107	0.1711	111.63

According to Table 1, Bayesian approach clearly improves Classic NN's performance. However, this approach requires more CPU performance and more iterations to find optimal complexity. Above all models, GP model has the most improved test performance but it considerably requires CPU performance. Fig. 2 shows GP model test set predictions.

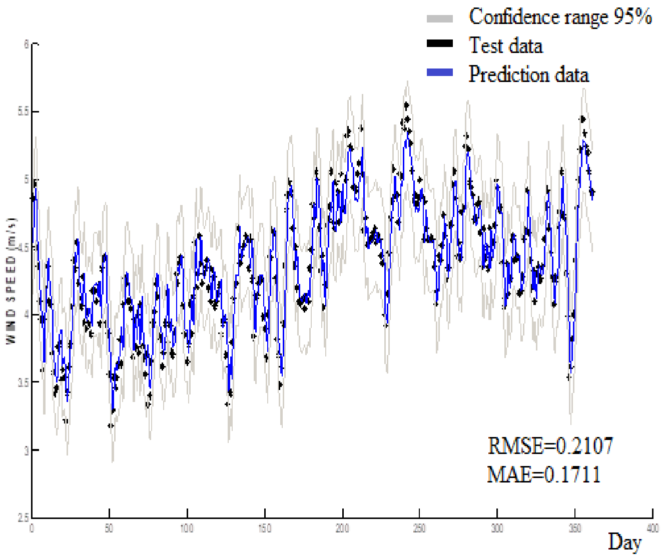

Fig. 2. *GP model wind speed forecast with 1-day horizon*

Since the GP model has shown the best model compared to the two others, further study successively with 3-days and 6-days' horizons is carried out to evaluate the forecast performance (Fig. 3). The GP model predicted wind speed for every 3-days' horizon and 29% errors occurred whereas the 6-days' horizon shows an error of 35%. The 3-days' horizon has the best and significant performance.

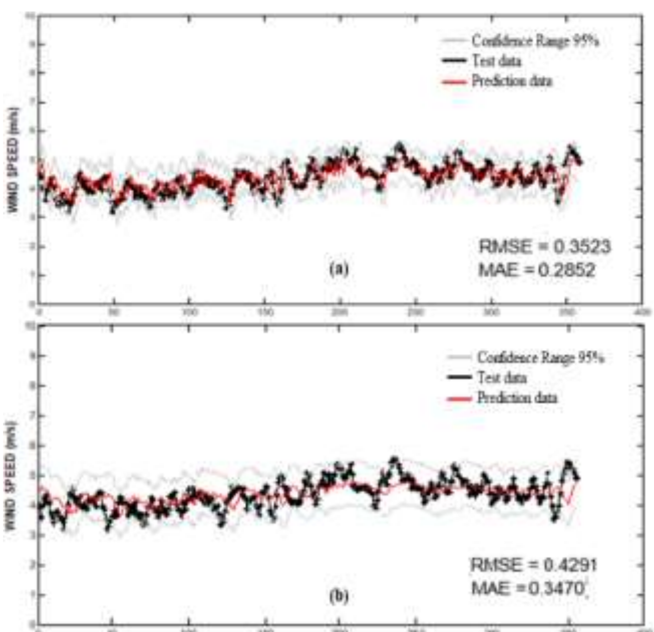

Fig. 3. *GP model for wind speed forecasts: (a) 3-days horizon, (b) 6-days horizon*

5. CONCLUSION

In conclusion, study presents three models for wind speed forecasting and compare their performance. Models use daily wind speed sample data from Antananarivo, Madagascar. Results showed that unlike classical NN techniques, Bayesian method can deal quite effectively with complex model to avoid overfitting training. Another drawback of the Bayesian NN is the chosen approximations number to quantify the weight parameters level integrals. By contrast, designed GP model is more robust. Indeed, it can predict wind speed forecast in Antananarivo about 365 days with 29% and 35% accuracies for every 3-days and 6-days horizons respectively. Future research will focus on improving GP model prediction performance and implement its test result in any hybrid energy sources software.

6. REFERENCES

[1] Blonbou, R., Monjoly, S., and Calif, R. (2010). Advanced tools for wind power integration into electrical networks. From Proceeding (702) Modelling, Identification, and Control – 2010. doi: 10.2316/p.2010.702-095

[2] Lauret, P., David, M., Calogine, D. (2011). Nonlinear Models for Short-time Load Forecasting, International Conference on Advances in Energy Engineering (ICAEE). doi:10.1016/j.egypro.2011.12.1109

[3] Bishop, C.M. (1995). Neural networks for pattern recognition. Oxford University Press, USA; 1st editio n, ISBN-10 : .9780198538646

[4] Hippert, H.S., Pedreira, C.E. and Souza, R.C. (2001) Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE Transactions on Power Systems, 16, 44-55. https://doi.org/10.1109/59.910780

[5] Zhang, G., Patuwo, B.E. and Hu, M.Y. (1998). Forecasting with Artificial Neural Networks: The State of the Art. International Journal of Forecasting, 14, 35-62. http://dx.doi.org/10.1016/S0169-2070 (97)00044-7.

[6] Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feed forward Networks Are Universal Approximators. Neural Networks, 2, 359-366. http://dx.doi.org/10.1016/0893-6080 (89)90020-8.

[7] MacKay, D. J. C. (1992). A practical Bayesian framework for back propagation networks. Neural Computation, 4, 448–472. doi: 10.1162/ neco.1992.4.3.448.

[8] Lauret P et al. (2008). Bayesian neural network approach to short time load forecasting. International Journal of Energy Conversion and Management 49:1156-1166. doi:10.1016/j.enconman.2007.09.009

[9] Rasmussen C.E., Williams C.K.I. (2006) Gaussian Processes for Machine Learning. The MIT Press. ISBN 026218253X. Massachusetts Institute of Technology.