Tropical Cyclones Tracks Forecast by Statistical Method

Jean Marc Rabeharisoa[1], Miloud Bessafi [2], Adolphe Ratiarison[1] and Olivier Rakotovazaha[1]

[1] Suspensions Rheology Laboratory, Department of Physics,

University of Antananarivo, Madagascar

[2] Laboratory for Energy, Electronics and Processes, La Réunion

Abstract

Tropical cyclone tracks are predicted by an analog ensemble forecast model. Forecast errors are measured by calculating the distance between the real position and forecast position, for a time lag of six hours. Principal Axis Tree (PAT) is inserted in the self-adapting model. Local Linear Wavelet Neural Network is used on three basins (Australian, South West of Indian Ocean and Atlantic basin). This neural networks family is trained by Particle Swarm Optimization algorithm. When compared with the arithmetic mean, Local Linear Wavelet Neural Network has advantage on forecast errors homogeneity. In general, Local Linear Wavelet Neural Network is equivalent to the arithmetic mean in forecast quality.

1. Introduction

Tropical cyclone tracks forecast methods divide generally in two big classes: statistical models and numerical models. Since these last thirty years, some considerable progress have been gotten with the meteorological models with the advent of the computers more and more effective in memory capacity and calculation speed, the integration of sophisticated preliminary treatment model to establish an initial state of the atmosphere (data assimilation). The majority of tropical cyclones tracks forecasts are type CLIPER (Climatology Persistence) introduced by Neumann [1]. Fraedrich K. and al [2] have used the self adapting model of Oliviers Sievers [3], in the australian basin by adapting metric weights. Compared to its reference model (CLIPER), they found the good performance in forecast error. Currently, the quality of cyclone tracks forecasts remains below expectations of the population, of the collectivities to warning systems taking into account the devastating impact (of human and material damages) and financial costs that lead these phenomena (populations displacement, infrastructure destruction, agriculture devastation,…).

The present study aims to improve the track forecast of cyclone. A new technique of analog selecting so-called Principal Axis Tree (PAT) is inserted in the self adapting model [3]. In the second time, Local Linear Wavelet Neural Network (LLWNN) whose training is done with the Particle Swarm Optimization (PSO) algorithm is used as calculation model.

The paper is organized as follows. The Principal Axis Tree is introduced in Section 2. The LLWNN is described in Section 3. The Particle Swarm Optimization algorithm is summarized in Section 4. The results on cyclone track forecast are shown in section 5. A short discussion is given in Section 6. Finally, concluding remarks are derived in the last section.

2. Principal Axis Tree (PAT)

Principal Axis Tree (PAT) permits to partition data set in an efficient manner in term of speed for the nearest neighbour determination. This nearest neighbour algorithm has been developed by James Mac Names [4]. This research algorithm is based on a very fast tree pruning, thanks to its power of elimination criteria. Principal component analysis is used to build an efficient search tree. The principal axis is defined as the principal component with the largest eigenvalue. The objective is to partition the dataset along the principal axis into n_c distinct regions such that each region contains roughly the same number of points. The process is repeated for each subset of point recursively until each subset contains fewer than n_c points.

The fast nearest neighbour algorithm consists of two parts:

- Principal Axis Tree Construction,
- Principal Axis Tree Search.

a. Principal Axis Tree construction

Schematically, it takes place in 4 stages:

1. Define nz as the number of points assigned to the node in progress. If $n_z < n_c$, the node is said terminal node and its treatment is finished, otherwise go to the second stage.

2. Construct the principal axis for the points in progress and calculate the projections of all these points on the principal axis,

3. Partition the set of the projected points in $\boldsymbol{n_c}$ distinct regions such that every region contains the same number of points (n_z / n_c or $n_z / n_c + 1$).

4. Assign to every created region a node label. There are $\boldsymbol{n_c}$ distinct nodes.

A tree, for which the set of the points is assigned to the root, is then obtained. These points are separated in different regions.

b. **Principal Axis Tree search**

Via the principal axis that has been saved to every stage, the region that contains a given point can be determined and therefore the associated child node is known. This determination is done by projecting the given point on the associated principal axis and doing dichotomic search among the limits of $\boldsymbol{n_c}$ regions. From there, the algorithm tempts to eliminate the sibling nodes via elimination criteria.

1. If the elimination criteria is satisfied, the sibling node is eliminated and the analysis goes back to the related node ;
2. If the criterion is not satisfied, the algorithm takes down in the nearest sibling node for an analysis and we take to the first stage of the first part.

Partial Distance Search is performed on the points of the remaining nodes, to have nearest neighbours of the given point. Let's recall that, by construction, a terminal node contains less than $\boldsymbol{n_c}$ points.

In summary, the process begins with the root, tempts to eliminate a section of the tree via the elimination criteria and takes down toward the terminal node that is correspondent to the query point.

The partial distance algorithm requires defining a distance formula.

3. Local linear wavelet neural network

The term of wavelet designates a function that oscillates for a time (if the variable is time) or an interval of finite length (if variable is type space). Beyond, the function decreases very quickly toward zero.

Wavelets are in the following form:

$$\Phi = \left\{ \Phi_j = \frac{1}{\sqrt{d_j}} \varphi\left(\frac{t \pm T_j}{d_j} \right), T_j, d_j \in \mathbb{R}^{nr} \ et \ j \in \mathbb{Z} \right\} \quad (1)$$

There are a family of functions generated from one single function $\varphi(t)$ by the operation of translation and dilatation. $\varphi(t)$ which is localized in both time space and frequency space, is called a mother wavelet. T_j and d_j are named translation and scale respectively.

In the standard form of wavelet neural network, the output of wavelet neural network is given by [5]:

$$y = \sum_{i=1}^{\ell} W_i \Phi_i(t) \quad (2)$$

$$y = \sum_{i=1}^{\ell} W_i \left| d_i \right|^{-\frac{1}{2}} \varphi\left(\frac{t - T_i}{d_i}\right) \quad (3)$$

Where,

Φ_i is the wavelet activation function of *ith* unit of the hidden layer,

w_i is the weight connecting the *ith* unit of the hidden layer to the output layer unit,

t represents the input to the wavelet neural network model,

ℓ is the number of hidden layer unit.

In nr dimensional input space, the multivariate wavelet basis function can be calculated by the tensor product of nr single wavelet basis functions as follows:

$$\varphi(t) = \prod_{i=1}^{nr} \varphi(t_i) \quad (4)$$

In order to take advantage of the local capacity of the wavelet basis functions while not having too many hidden units, Local Linear Wavelet Neural Network (LLWNN) is proposed [5,6] (Fig.1). LLWNN is an alternative type of wavelet neural network.

Its output in the output layer is given by:

$$y = \sum_{i=1}^{\ell} (w_{i,0} + w_{i,1} t_1 + \ldots + w_{i,nr} t_{nr}) \, \Phi_i(t) \tag{5}$$

$$y = \sum_{i=1}^{\ell} (w_{i,0} + w_{i,1} t_1 + \ldots + w_{i,nr} t_{nr}) \left| d_i \right|^{-\frac{1}{2}} \varphi\left(\frac{t - T_i}{d_i}\right) \tag{6}$$

The motivation for introducing the local linear models into a WNN are as follows:

1. Local linear models have been studied in some neuro-fuzzy systems and shown good performances [7].

2. Local linear models should provide a more parsimonious interpolation in high dimension spaces when modelling samples are sparse [8].

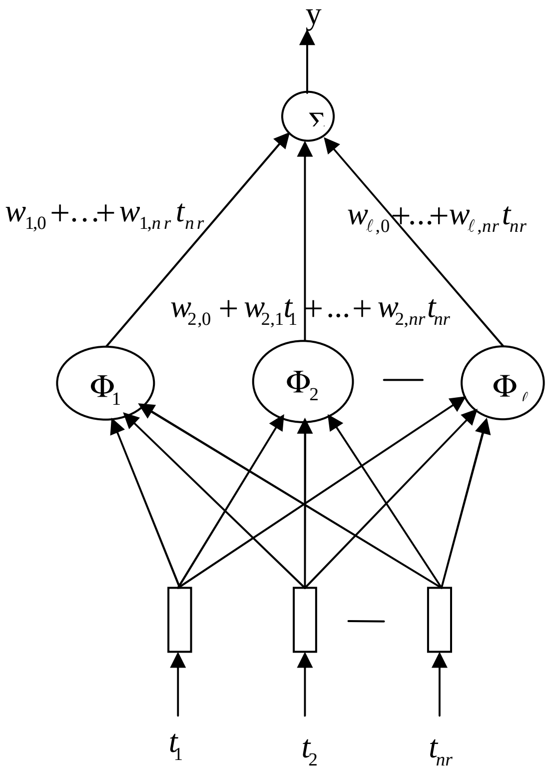

FIG.1: Local Linear Wavelet Neural Network.

4. Particle Swarm Optimization (PSO)

Particle Swarm Optimisation (PSO) has been developed by Kennedy and Eberhart [9,10]. PSO puts in plays particles groups in the form of vectors moving in research space.

Each particle is characterized by its position $X_i(\tau)$ and position changement vector $v_i(\tau)$ (named velocity) in a given instant τ.

PSO is based on two simple rules [11]:

1. Every individual remembers the best point (the closest to the objective) by which it passed during its evolution and tends to return to it.

2. Every individual is informed about the best point known within the population taken as a whole and tends to go there.

In particle position update, the direction of its movement, its velocity, its best position and the best position of its neighbours are taken in account.

At each time $\tau+1$, velocity and position for particle *i* are given by [12,13]:

$$v_i(\tau+1) = \omega * v_i(\omega) + c_1 * R_1 * (Po_i - X_i(\tau)) + c_2 * R_2 * (P_g - X_i(\tau)) \tag{7}$$

$$X_i(\tau+1) = X_i(\tau) + v_i(\tau+1) \tag{8}$$

Where,

R_1 et R_2 are uniformly distributed random number in [0..1],

c_1 and c_2 are positive constante, usualy

$$c_1 + c_2 \le 4 \tag{9}$$

ω : inertia weight,

Po_i is the best position of particle i

$X_i(\tau)$ is the current position of the particle i

P_g is global best position.

The algorithm runs as a convergence criterion was not met. This may be:

- A fixed number of iteration ;
- According to the fitness
- When the change in velocity is close to zero

5. Results : Tropicals cyclones tracks forecast:

Forecast error [2,3] is the following form :

$$E = 111\ \arccos[\sin(y_0) * \sin(y_f) + \cos(y_0) * \cos(y_f) * \cos(x_0 - x_f)] \qquad (10)$$

(x_0, y_0) is the observed position and

(x_f, y_f) is the forecast position.

Model skill, compared with a reference model, is:

$$s_k = (\langle E_{ref} \rangle - \langle E \rangle) / \langle E_{ref} \rangle \qquad (11)$$

$\langle E_{ref} \rangle$: Mean error obtained with the reference model.

$\langle E \rangle$ Mean error obtained with the model. Positive skill indicates that model have lower errors than reference and vice versa.

Variation coefficient of forecast errors [14] is

$$cv = \frac{\sigma}{\varpi} \qquad (12)$$

Where

σ is the standard deviation of error forecast

ϖ is the mean error

It indicates the homogeneity of forecast errors. With small value of cv, forecast errors ensemble is more homogeneous.

Analog selection PAT has been used. Local Linear Wavelet Neural Network has been compared to the arithmetic mean (Fig. 2, Fig. 3, Fig. 4¸ Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9¸ Fig. 10).

13 predictors have been used: current positions of analog cyclones (Longitudes and latitudes), zonal and meridional displacements for a time lag of 6 h up to 24 h and year day.

The number of predictor has been reduced with the principal component analysis (PCA), using matlab software, hence the network architecture {6-6-1} (6 inputs, 6 units of the hidden layer and one output).

PSO algorithm was employed to train the LLWNN model with this network architecture.

The objective function used is the root mean square error (RMSE),

$$RMSE=\sqrt{\frac{1}{Ne}\sum_{i=1}^{Ne}\left(y_1^i - y_2^i\right)^2} \qquad (13)$$

Where, y_1^i and y_2^i denote the target output and model output, respectively.

Ne is the number of the analog.

The used mother wavelet is as follows:

$$\varphi(t)=\pm t \; e^{-\frac{1}{2}t^2} \qquad (14)$$

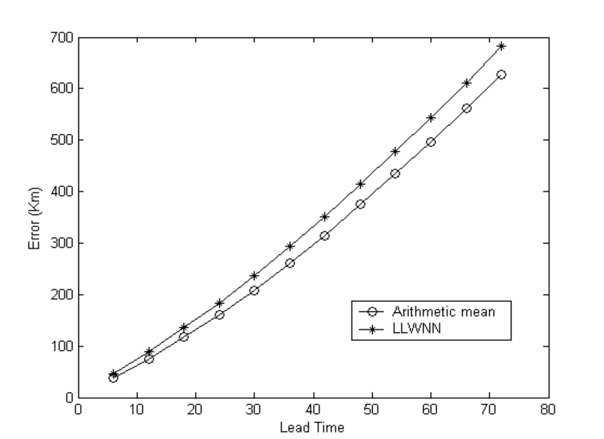

FIG. 2: Mean forecast errors changing with the lead time (h) in **australian** basin for 480 cyclones. The used selection analog is PAT.

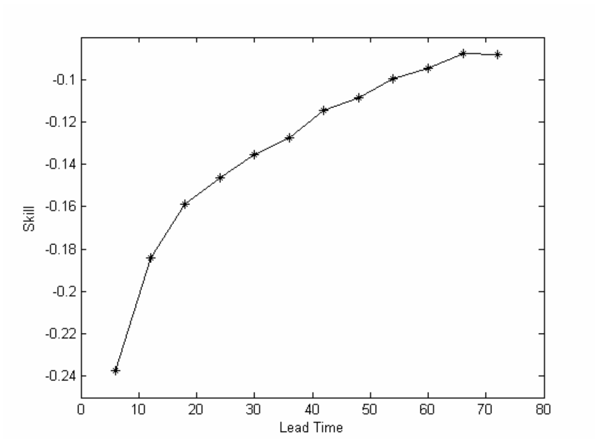

FIG. 3: Local Linear Wavelet Neural Network skill with reference to Arithmetic mean zero skill or reference model, in the australian basin, for 480 cyclones.

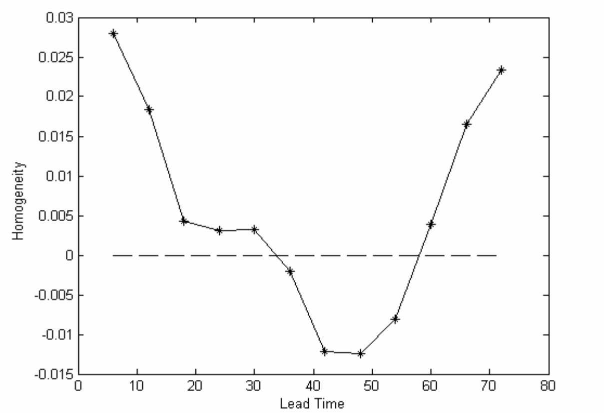

FIG. 4: Local Linear Wavelet Neural Network Forecast error homogeneity with reference to Arithmetic mean zero homogeneity or reference model, in the australian basin.

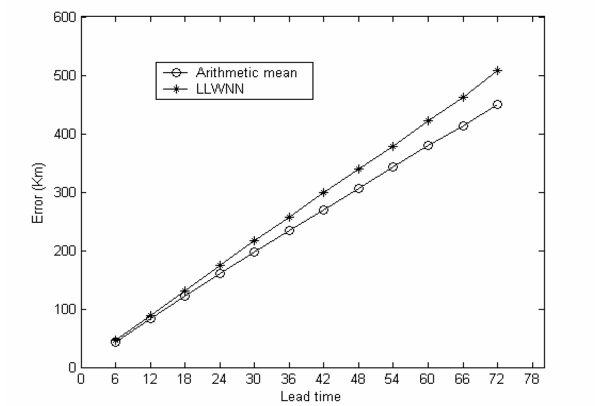

FIG. 5: Mean forecast errors changing with the lead time (h) in **the South West of Indian Ocean** basin for 494 cyclones. The used selection analog is PAT.

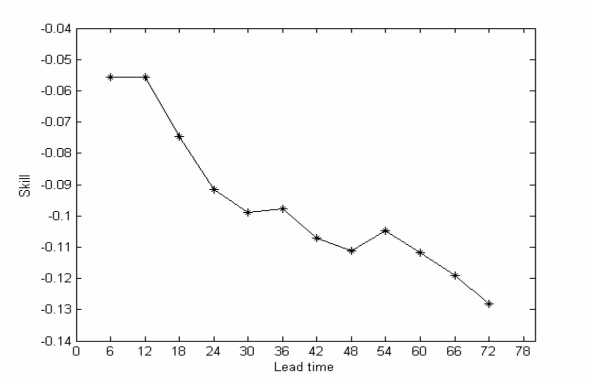

FIG. 6: Local Linear Wavelet Neural Network skill with reference to the Arithmetic mean zero skill or reference model, in the **South West of Indian Ocean** basin, for 494 cyclones.

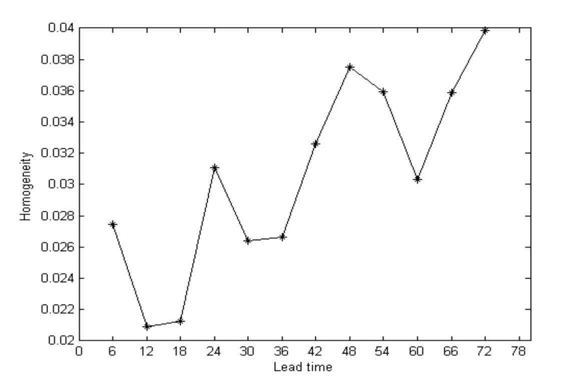

FIG. 7: Local Linear Wavelet Neural Network Forecast error homogeneity with reference to the Arithmetic mean zero error homogeneity or reference model, in the **South West of Indian Ocean** basin.

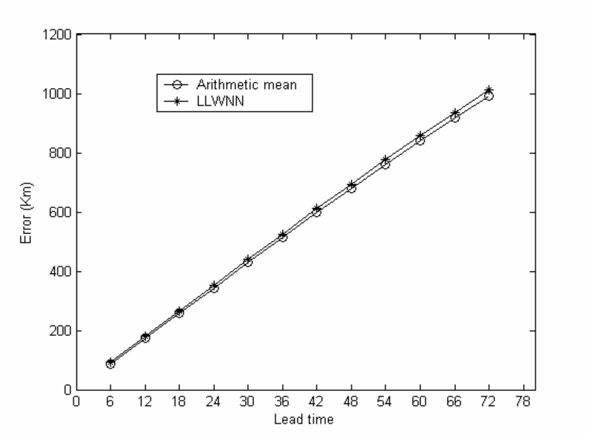

FIG. 8: Mean forecast errors changing with the lead time (h) in **atlantic** basin for 921 cyclones. The used selection analog is PAT.

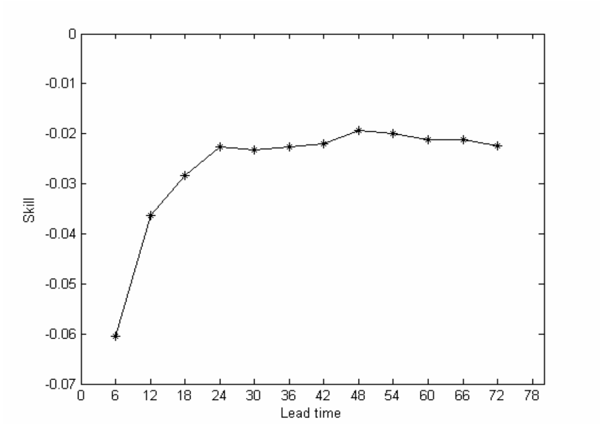

FIG. 9: Local Linear Wavelet Neural Network skill with reference to Arithmetic mean zero skill or reference model, in the Atlantic basin, for 921 cyclones.

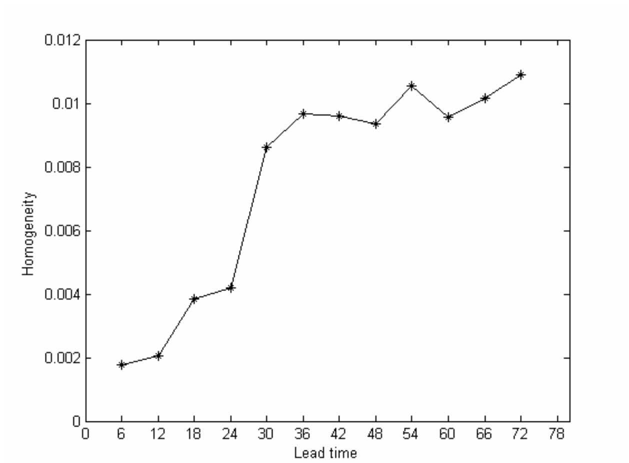

FIG. 10: LLWNN Forecast error homogeneity with reference to the Arithmetic mean zero forecast error homogeneity or reference model, in **atlantic** basin.

6. Discussion

On a test set for the analog selection PAT, compared to the arithmetic mean, Local Linear Wavelet neural Network gives average large forecast errors. But the advantage of using the Local Linear Wavelet Neural Network is the stability of forecast errors forecasts. Forecast errors obtained with Local Linear Wavelet Neural Network are more homogeneous compared to those of the arithmetic mean. This forecast stability of the Local Linear Wavelet Neural Network is due to the flexibility and the property of good approximation of neural network [15,16]

7. Conclusion

Local Linear Wavelet Neural Network allows having stable forecast. Forecast qualities with the mean arithmetic and with the Local Linear Wavelet Neural Network are equivalent. The use of PAT and LLWNN, for tropical cyclone tracks forecast is not still satisfactory.

References

[1] Neumann, *Bull. Am. Met. Soc*. **62**,1473 (1981).

[2] K. Fraedrich, C.C. Raible, F. Sielmann, *Bull. Am. Met. Soc.* **18**, 1 (2003).

[3] O. Sievers, K. Fraedrich, C.C. Raible, *Wheather and Forecasting* **15**, 623 (2000).

[4] J. Mc. Names, *IEEE Trans. on pattern analysis and machine intelligence,* **23** (9), 964 (2001).

[5] Y. Chen, J. Dong, B. Yang and Y. Zhang, *Proc. of the 5th World Congress on Intelligent Control and Automation*, (Hangzhou, P.R. China, 2004), pp. 1954-1957.

[6] Y. Chen, B. Yang, J. Dong. *Neurocomputing* **69**, 449 (2006).

[7] O. Nelles, R. Isermann. *Control Eng. Practice* **6**, 259 (1998)

[8] M. Boznar. *In intelligent information systems. IIS'97. Proc*., (Grand Bahama, Island Bahamas, 1997), pp.345-349.

[9] J. Kennedy and R.C. Eberhart, *Proc. 1995 IEEE Int. Conf. on Neural Networks, Perth, Australia* (IEEE Service Center, Piscataway, NJ IV, 1995), pp.1942–1948.

[10] R.C. Eberhart and J. Kennedy, *Proc. 6th Int. Symposium on Micro Machine and Human Science, Nagoya, Japan*, (IEEE Service Center, Piscataway, NJ, 1995), pp. 39-43.

[11] J. P. Rennard, *www.rennard.org/alife/french/ivafs.pdf*

[12] Y.H. Shi, R.C. Eberhart, *Proc. IEEE Int. Conf. on Evolutionary Computation, Anchorage, Alaska*, (IEEE Press, Piscataway, NJ, 1998), pp. 69-73.

[13] Y.H. Shi, R.C. Eberhart, *Proc. EP98, Evolutionary Programming VII, San Diego, USA*, (Springer-Verlag, New York, 1998), pp.591-600.

[14] E. Amiot, *Introduction aux probabilités et à la statistique*. Edited by Gaëtan morin éditeur ltée, (Boucherville, Canada, J4B 6G4 ,1990).

[15] T. Poggio and F. Girosi, *Proc. IEEE.* **78(9),** 1481 (1990).

[16] T. Poggio and F. Girosi, *Science.* **247,** 978 (1990).