Modeling The Results Of A Perceptron And Neuro-Fuzzy Neural Network Simulation (ANFIS)

LALAOHARISOA Sahoby[1,1], VELO Jérôme[2], MANASINA Ruffin[3], RANDRIANANTENAINA Todihasina Roselin[4], RATIARISON Adolphe Andriamanga[5]

Atmosphere, Climate and Oceans Dynamics Laboratory[1,2,3,4,5]

University of Antananarivo-Madagascar[1,5], University of Toamasina-Madagascar[2], University of Mahajanga-Madagascar[3,4],

sahoby.lalaoharisoa@univ-antananarivo.mg[1], zjvelo@gmail.com[2], ruffin.manasina@gmail.com[3], todyroselin@yahoo.fr[4]

Adolphe.ratiarison@univ-antananarivo.mg[5]

Abstract – **The objective of this work is to model simulation data of a dust devils in Comsol using neuro-fuzzy methods (ANFIS: Adaptive Neuro Fuzzy Inference Systems) and perceptron neural networks. Since the number of simulations performed was insufficient, we used the Spline function to increase the amount of data. The results show that neuro-fuzzy is more effective than perceptron neural networks. The obtained models are excellent, with a Nash -Sutcliffe criterion value above 90%.**

Keywords – **Modeling, Simulation Data, Perceptron Neural Networks, Neuro-Fuzzy, Spline.**

I. INTRODUCTION

Dust devils are phenomena that are frequently observed in dry and sunny regions. We conducted a simulation using Comsol Multiphysics [1]. This simulation starts with air already in rotational motion. We used the LES-Smagorinsky method to solve the partial differential equations governing this phenomenon. Temperature and pressure data were obtained from the 5 simulations we conducted by varying the fluid's rotational speed. According to our simulations during our thesis [1], we observed that the maximum temperature and pressure vary depending on the rotational speed. As a result, to model this temperature, we utilized the perceptron neural network and neuro-fuzzy (ANFIS) method, based on pressure and rotational speed. The objective of this work is to compare the results obtained by two methods: the perceptron neural network and the neuro-fuzzy (ANFIS). Before applying these methods, the data are curve fitted to increase the amount of data used (data interpolation).

The remainder of the paper is organized as follows: section III presents an overview of multilayer Perceptron artificial neural networks with the results obtained. Section IV describes the implementation of neuro-fuzzy (ANFIS) as well as the presentation of the model. Prior to concluding, we will compare the two methods Ease of Use

II. INTERPOLATION OF DATA

During our thesis work [1], we simulated a rotating fluid in a parallelepiped. We multiplied this simulation five times, always recovering the maximum value of temperature and pressure for a given rotation speed. Then, these maximum temperature and pressure values from these simulations will be grouped in Table I below.

TABLE I: SIMULATION DATA

Angular velocity (rad/s)	Temperature (K)	Pressure (Pa)
π	301.057	84359.84
$\pi/2$	303.586	79358.25
$\pi/4$	306.379	73638.42
$\pi/6$	309.979	71824.48
$\pi/35$	312.45	68656.07

In order to reach the final objective, it is necessary to multiply the number of these points. This action leads us to interpolate these data. Interpolation is a technique to find a continuous function passing through a set of given data points. The interpolation method utilized in this study is the "interp1" function with the "spline" method. The "interp1" function is used to interpolate the data in Table I to one dimension. The "spline" is a particular method of interpolation which consists in fitting segments of polynomials of degree k to subsets of consecutive data points, where k is a positive integer.

This interpolation provides us with the approximate values of the simulation data presented in blue, and the simulated data in orange. (See figure 1).
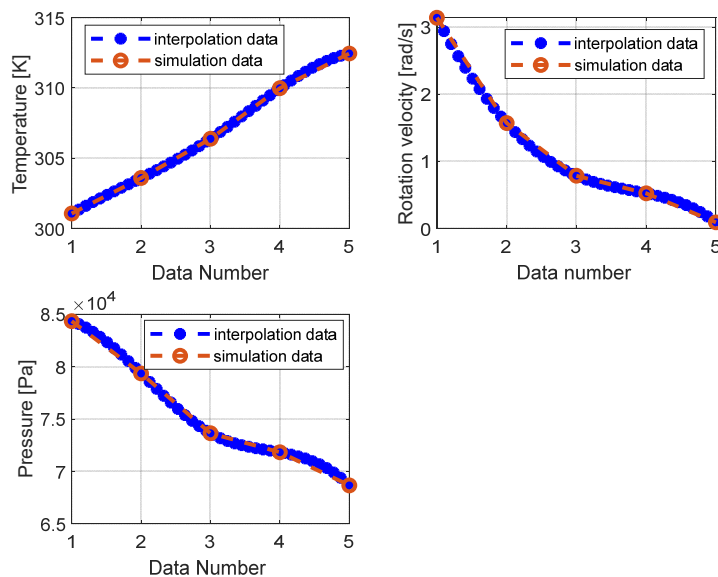

Figure 1: Data Interpolation

To model these results obtained, we must apply the method of artificial neural networks multilayer perceptron.

III. MULTILAYER PERCEPTRON ARTIFICIAL NEURAL NETWORK MODELLING

1-   Definition

According to Claude Touzet [4], artificial neural networks are highly connected networks of elementary processors operating in parallel. Each elementary processor calculates a unique output based on the information it receives. Any hierarchical structure of networks is obviously a network.

Neural network architectures can be divided into 4 main families [5]:

- Feed forwarded neural networks

- Recurrent neural networks (RNN)

- Resonance neural networks

- Self-organizing neural networks

In our case we used the feed forwarded neural network.

In the family of feed forward networks, we distinguish between single layer networks (simple perceptron) and multi-layer networks (multi-layer perceptron).

A multilayer perceptron (MLP) can be defined as the most widely used static neural network. The PMC is capable of approximating any continuous function provided that the number of neurons in the hidden layer is appropriately fixed [6].

2- Neural network optimization parameters

- *Neuronal architecture of the model*

For the temperature modeling we have partitioned the following neural network models: an input layer of 2 neurons (pressure and rotation speed), an output layer and three hidden layers of which the first one contains 10 neurons, the second one contains 21 neurons and the third one. The data is divided into two phases:

• learning phase, which represents 75% of the data;

• test phase, which represents 25% of the data.19 neurons (figure 2).

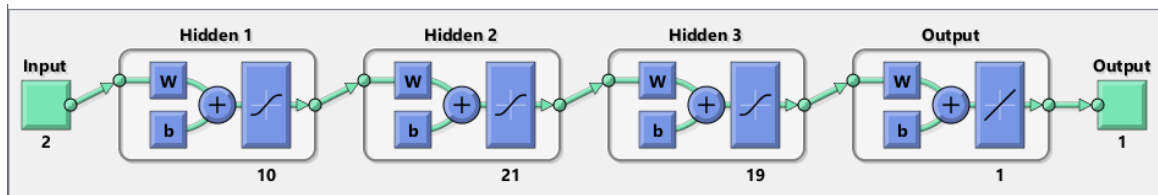The model's neural architecture is illustrated in Figure 2.

Figure 2: Neuronal architecture of the model

3- Model output

Figure 3 shows the curves of these selected models, the observation data and the approximate values obtained from the interpolation (colored in black). The curve colored in red shows the result of the simulation, the one colored in green shows the model learning and the one colored in blue shows the test.
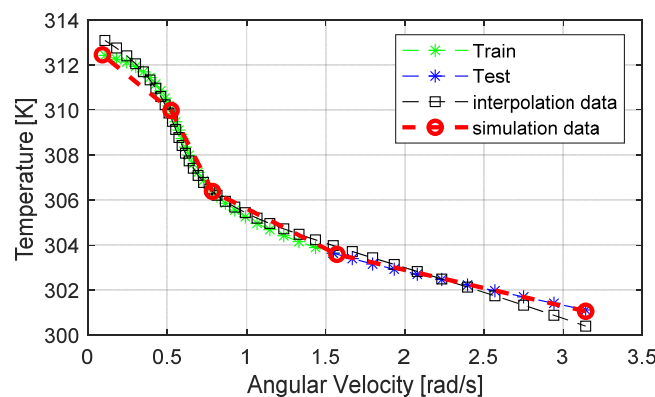

Figure 3: Temperature model by perceptron neural network

4-   Model validation

We can conclude that the chosen model for the temperature is excellent, as the calculated correlation coefficient between the desired output and the calculated output is R=0.99992. The result is shown in Figure 4.
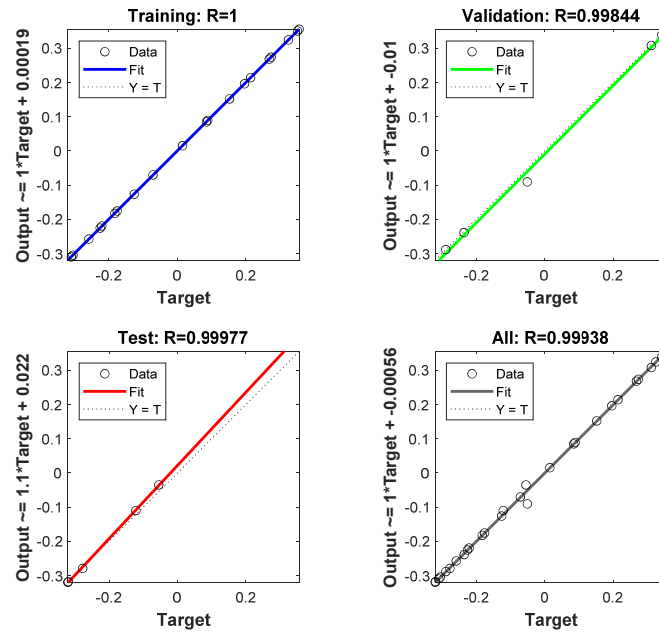

Figure 4: Validation of the neural network model

- *MSE value*

The graph in Figure 5 shows a decrease in MSE values as weights were improved during the network training that ended at 14 epochs based on the reduction in adaptive weight.
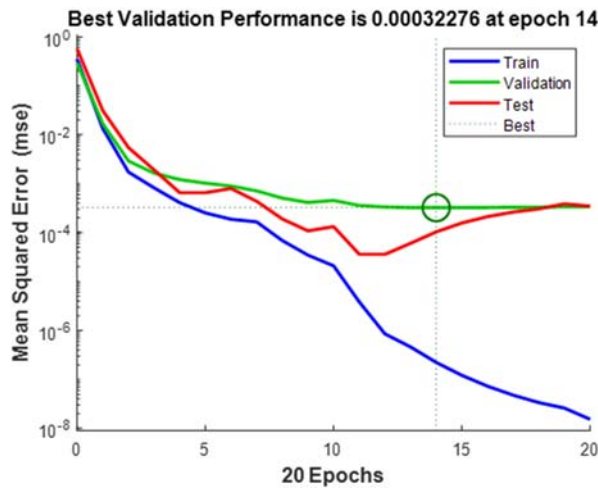

Figure 5: The performance of the neural network learning phase for the maximum temperature model.

After obtaining the model by the PMC method, we will model these same data by the ANFIS method.

IV. RESULTS USING THE ANFIS METHOD

1- ANFIS (Adaptive Network Based Fuzzy Inference System)

- Definition

Neuro-fuzzy systems are fuzzy systems trained by a learning algorithm inspired by the theory of neural networks.

The interest of building systems integrating neural networks and fuzzy inference systems (FIS) lies in their complementary characteristics [2].

ANFIS is a class of adaptive network proposed by Jang in 1993. It can be seen as a non-looped neural network for which each layer is a component of a neuro-fuzzy system and, as such, it is a universal "approximator" [2]

2- ANFIS model structure

ANFIS is a neuro-fuzzy adaptive inference system that consists of using a 5-layer MLP (Multilayer perceptron) neural network (Figure 6) for which each layer corresponds to the completion of a step of a Takagi Sugeno type fuzzy inference system [3].

For the ANFIS model, we also divided the data into 2 phases, 75% for the learning phase and 25% for the test.

Table II shows the different layers in Figure 6 with the number of neurons in each layer.

TABLE II: THE DIFFERENT LAYERS OF ANFIS

the different layers	Layers types	The number of neurons in a layer
layer 0 (input)	input (pressure et Angular velocity)	2
layer 1(inputmf)	The values	$9 \times 2 = 18$
layer 2(rule)	The rules	9
layer 3(outputmf)	The normalization	9
layer 4	Linearization of function	1
layer 5	Sum	1

The proposed architecture of the ANFIS model consists of 2 input parameters and an output value, as shown in Figure 6 below.
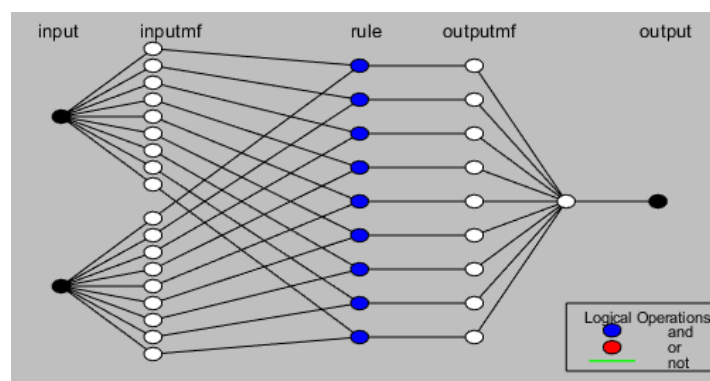

Figure 6: ANFIS 2-input network structure

3- Model validation

The model used shows that the similarity rate of the desired output and the calculated one is 99.936% for the training and

99.251% for the test (Figure 7). This allows us to conclude that the obtained model is good or even excellent.
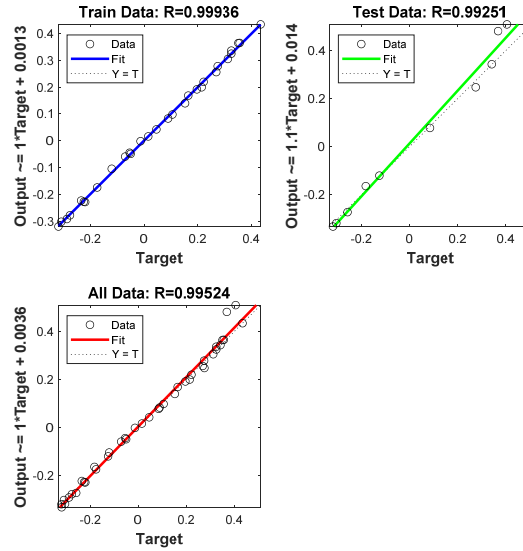

Figure 7: Validation of the ANFIS model

- Model mapping

The set of fuzzy inference rules that apply to the model structure is given in Figure 8. It contains Sugeno fuzzy rules.

The two rules of a first-order Sugeno fuzzy inference system can be expressed as shown in the following equations:

$$\text{IF x is } A_1 \text{ AND y is } B_1, \text{ THEN } f_1 = p_1 x + q_1 y + r_1$$

$$\text{IF x is } A_2 \text{ AND y is } B_2, \text{ THEN } f_2 = p_2 x + q_2 y + r_2$$

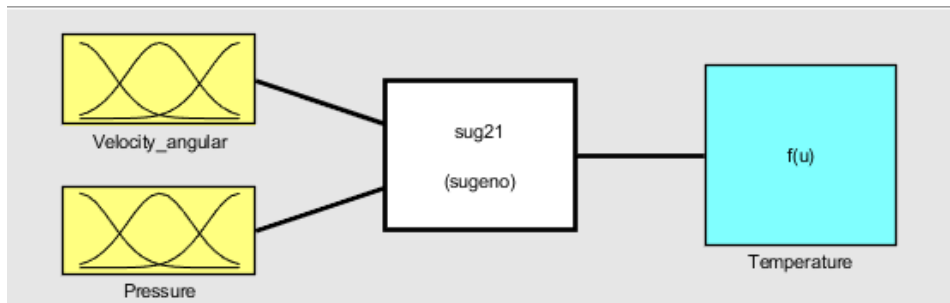In this work, we propose ANFIS with the learning scheme based on subtractive clustering (SC).

Figure 8: Model mapping

4- Model output

Figure 9 shows us the result of the modeling of the temperature as a function of the rotation speed. In red, we have the simulation data, in green the modeling of the training data, in black the interpolated data, in blue the modeling of the tests. Visually, the curves are quasi-conformed.
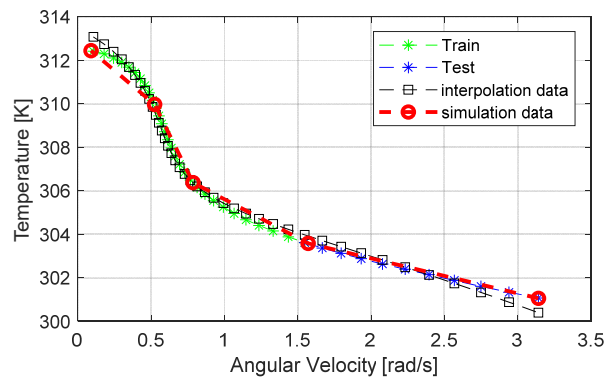
Figure 9: Temperature model by perceptron neural network

V. COMPARISON

1- NASH-Sutcliffe criterion

We used the NASH criterion to assess the performance of the model whose value range between -∞ and 1. We can calculate it using the following formula:

$$\text{NASH} = 1 - \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}} \tag{1}$$

y_i : Observed value

$\hat{y}_i$: Output data by the model

$\overline{y}$: The average of the observed data.

Kachroo (1986) gave the following scale (Table III) as to the values taken by the Nash criterion:

TABLE III: CLASSIFICATION OF MODEL QUALITY, THE NASH CRITERION.

Nash value (%)	Quality
Nash < 60	Mauvais
60 ≤ Nash < 80	Bon
80 ≤ Nash < 90	Très bon
Nash ≥ 90	Excellent

The Nash criterion values obtained by the neural network and ANFIS are shown in Table IV. They are both above 90%, so we have an excellent model.

TABLE IV: NASH CRITERION FOR EACH MODEL

	RNA	ANFIS
Nash	99,30%	99,35%

2- RMSE (Root Mean Squared Error)

In order to validate the obtained models, we calculated the root mean square error (RMSE) using a commonly used formula that is considered an excellent error measure for numerical predictions.

Table V represents the RMSE (root mean square error) value for each model. The calculated RMSE between the approximate value and the one obtained by both methods (ANFIS and RNA) is close to 0, which means that we have a good model. The formula for the RMSE is as follows:

$$RMSE = \sqrt{\frac{(y_i - \hat{y})^2}{N}} \qquad (2)$$

N: Data number

TABLE V: RMSE FOR EACH MODEL

	RNA		ANFIS	
	Training	Test	Training	Test
RMSE	0.2787	0.3497	0.2739	0.3280

Based on these results, we can conclude that the model obtained through the fuzzy neuro method is more reliable with a lower squared error than that obtained by RNA.

VI. CONCLUSION

This work led us to model the fluid temperature of a dust vortex using two methods: the perceptron neural network and the neuro-fuzzy (ANFIS). The input data for both methods are the pressure and the rotational speed of the fluid.

The calculated squared error values are small. The Nash criteria of each model are higher than 90%. This characteristic allows us to conclude that the models obtained by these two methods are excellent. Moreover, by comparing the two obtained results, we noticed that the modeling by neuro-fuzzy is more reliable.

REFERENCES

[1] S. LALAOHARISOA, Simulation numérique d'un tourbillon de poussiére par la méthode de LES Smagorinsky, Antananarivo, Madagascar: Université d'Antananarivo, 2021.

[2] R. ADELINE, R. GOURIVEAU et N. ZERHO, «Pronostic de défaillances : maîtrise de l'erreur de prédiction,» *7ème Conférence Internationale de MObilisation et SIMulation, MOSIM'08., Paris : France (2008),* pp. hal-00270725, version 1, 2008.

[3] A. E. DJOKHRAB , Planification et Optimisation de Trajectoire d'un Robot Manipulateur à 6 D. D. L. par des Techniques Neuro-Floues, UNIVERSITE MOHAMED KHIDER DE BISKRA : Mémoire , Faculté Des Sciences et De La Technologie , 2015.

[4] C. TOUZET, Des réseaux de neurones artificiels à la robotique coopérative, Université Aix Marseille: mémoire pour l'obtention d'HDR , 1998.

[5] Juri'Predis, «Juri'Predis;,» 24 mai 2022. [En ligne]. Available: https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-2-les-reseaux-de-neurones-artificiels.

[6] R. Rakotomalala, Deep learning : Perceptrons simples et multicouches, Lyon: Université Lumière Lyon 2, 2003.