



# **Neural Fuzzy Inference System-Based Weather Prediction Model (NFIS-WPM) Experiment**

## **Introduction**

In this experiment, the researcher is trying to implement a neural fuzzy inference system-based weather prediction model (NFIS-WPM) that is based on a study made by five Chinese PhD (Hu, Xue, Xiakua, Zhang, & Lu, 2014). It proposes a fuzzy weather prediction model based on neural networks. This means that the weather should have some discrete attributes that will be rated in percentage according to the prediction. In this report, the researcher is noting every process done toward that.

It is assumed that the reader has a copy of (Hu, Xue, Xiakua, Zhang, & Lu, 2014) and aware of its content. There will be many reference points to that study, but the whole experiment is based on that.

## **Base theory**

As (Hu, Xue, Xiakua, Zhang, & Lu, 2014) said in their introduction, "fuzzy logic and artificial neural networks are important in the intelligent control of complex system" (Neural Fuzzy Inference System-Based Weather Prediction, p. 2). They are the fuzzy logic and ANNs, used together to solve problems based on ordered classes to predict some fuzzy percentage, or to be more accurate, a decimal value between 0-1 that will reveal the binomial prediction.

As the base theory says: "Traditionally, ANNs use crisp weather variable as its inputs to predict crisp weather conditions, and in this paper, we use fuzzy weather variables as its inputs and predict the fuzzy weather condition instead. Our neural fuzzy inference system-based weather



prediction model (NFIS-WPM) could infer a new reasonable fuzzy variable output according to the fuzzy variable inputs by using the algorithm we proposed in this paper and then we use this approach in weather forecasting. In addition, NFIS-WPM is based on another technique that combines ANNs and fuzzy logic. This technique is different from the previous three techniques since it embeds neural network into fuzzy logic engine with fuzzy input and fuzzy output values. In other words, the first approach and the new approach operate in opposite ways. Besides, this new technique is specifically designed for fuzzy inference problems and the others have different goals." (Hu, Xue, Xiakua, Zhang, & Lu, 2014, p. 2).

## **Requirement**

This experiment is run using:

### **1- Python scripting language 3.6**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. (Gutttag, 2016). Python is widely used in data processing and AI application for its wide variance libraries.

### **2- TensorFlow Library**

Based on their official website: "TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains" (TensorFlow, n.d.)



### **3- Numpy Library**

NumPy is the fundamental package for scientific computing with Python. It contains a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, and useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. (NumPy, n.d.)

### **4- Keras Library**

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research. (Keras, n.d.).

### **5- Anaconda**

Anaconda Enterprise is an AI/ML enablement platform that empowers organizations to develop, govern, and automate AI/ML and data science from laptop through training to production. It lets organizations scale from individual data scientists to collaborative teams of thousands, and to go from a single server to thousands of nodes for model training and deployment (ANACONDA, n.d.).

### **6- Spyder IDE**

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used



as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software (Spyder, n.d.).

## Preprocessing

The processed data is dataset of makkah weather details done from 2008 till mid 2018.

The data has been converted into discrete values based on the study (Hu, Xue, Xiakua, Zhang, & Lu, 2014, pp. 10-11) as the following:

**Table 1. Fuzzy classification of atmospheric pressure.**

Atmospheric Pressure Rating	Moderate	Lower Slightly	Lower	Lowest
Atmospheric pressure value (hPa)	>940	[930, 940]	[920, 930)	<920

So, for pressure, a mean pressure of average sea pressure and average station pressure is calculated, the it is divided into its categories accordingly using MS Excel conditional formulas. It is recognizable that all pressure parameters in Makkah dataset are moderates except one outlier that is on 14/7/2011 were it was at its lowest. After ignoring the outlier, it appears that there is no considerable variance in pressure according to the dataset. For that, pressure is ignored in this experiment.

**Table 2. Fuzzy classification of dry and wet bulb temperature.**

Dry and Wet Bulb Temperature Rating	Lowest	Lower	Moderate	Higher	Highest
Dry and wet bulb temperature value (°C)	<-10	[-10, 5)	[5, 30)	[30, 45]	>45

For the temperature, our data set is considering Makkah. Accordingly, the records show that there are no lowest, lower, nor highest temps. So, we have distributed the temperature between moderate and higher according to table 2.



**Table 3. Fuzzy classification of relative humidity.**

Relative Humidity Rating	Dry	Less Dry	Less Humid	Humid
Relative humidity value (%)	[0, 30)	[30, 50)	[50, 70)	[70, 100]

Humidity mean is distributed according to table 3.

**Table 4. Fuzzy classification of wind speed.**

Wind Speed Rating	Calm	Light Air	Light Breeze	Gentle Breeze	Moderate Breeze
Wind speed value (MPH)	(0, 2)	[2, 4)	[4, 7)	[7, 11)	[11, 17)

Wind speed mean is distributed according to table 4.

**Table 5. Fuzzy classification of precipitation.**

Precipitation Rating	No Rain	Light Rain	Moderate Rain	Heavy Rain	Rainstorm
Precipitation value (mm)	0	(0, 10)	[10, 25)	[25, 50)	[50, +∞)

Precipitation mean is distributed according to table 5.

After distributing the parameters, we have created a Python function that will create a sequential model of 3 layers; input, hidden, and output layer. After many trials, we concluded that it best fits eight neurons for the input layer and hidden layer. The output layer will depend on the predictable result parameters.

Using that, It appears that the average best accuracy is obtained by fifty five training rounds.

## Results

- 1- The accuracy average obtained for temperature prediction is about 92.65%.
- 2- The accuracy average obtained for wind speed prediction is about 72.81%.



- 3- The accuracy average obtained for precipitation prediction is about 90.02%,
- 4- The accuracy average obtained for humidity prediction is about 80.09%.
- 5- The average accuracy for the algorithm is about 83.89.

## References

- ANACONDA. (n.d.). *What is Anaconda*. Retrieved 1 20, 2019, from ANACONDA: <https://www.anaconda.com/what-is-anaconda/>
- Guttag, J. (2016). *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press.
- Hu, J., Xue, S., Xiakua, Z., Zhang, S., & Lu, W. (2014). Neural Fuzzy Inference System-Based Weather Prediction. *Atmosphere*, 788-805.
- Keras. (n.d.). *Keras Home Page*. Retrieved 1 20, 2019, from Keras: <https://keras.io/>
- NumPy. (n.d.). *NumPy Home Page*. Retrieved 1 20, 2019, from NumPy: <http://www.numpy.org/>
- Spyder. (n.d.). *Spyder Home Page*. Retrieved 1 20, 2019, from Spyder: <https://www.spyder-ide.org/>
- TensorFlow. (n.d.). *TensorFlow Home Page*. Retrieved 1 20, 2019, from TensorFlow: <https://www.tensorflow.org/>



## Index

Neural Fuzzy Inference System-Based Weather Prediction Model (NFIS-WPM) Experiment .....	1
Introduction .....	1
Base theory .....	1
Requirement.....	2
1-Python scripting language 3.6.....	2
2-TensorFlow Library.....	2
3-Numpy Library.....	3
4-Keras Library.....	3
5-Anaconda .....	3
6-Spyder IDE .....	3
Preprocessing .....	4
Results.....	5
References .....	6
Index.....	7