

## Comparison Of Neural Network And Differential Evolution In Estimation Of Air Quality Using Mean Square Error

Ima O. Essiet

Department of Electrical Engineering, Bayero University, PMB 3011, Kano, Nigeria

---

**Abstract:** Softcomputing techniques are fast becoming reliable and efficient means of prediction and estimation. This has made their application more wide spread in recent years. With the growing need for intelligent devices and systems comes the need to explore these techniques even further. This paper applies neural networks and differential evolution (two of the most effective softcomputing algorithms) to the estimation of air quality and compares the accuracy of their results using the mean square error (MSE) method. Air pollution is an ever increasing menace in major cities around the world. Air contaminants such as those from motor vehicles and industrial wastes are the most common forms of pollutants. The health implications of inhaling contaminated air are evident in the growing number of cases of lung cancer and tuberculosis. Since these contaminants are invisible to the naked eye, it becomes necessary to implement an algorithm which can accurately identify them especially when their concentration becomes a threat to human health. The aim of this paper is to develop an effective algorithm to achieve this by comparing the efficacy of both neural networks and differential evolution in the determination of the concentration of air pollutants. The air component markers being analysed include oxides of carbon, nitrogen, sulphur and also ammonia. The study also intends to identify the most potent sources of air pollution by analysing air samples obtained at various locations within Kano city in Nigeria.

**Keywords:** feedforward neural network, air pollutants, differential evolution, MSE, metal oxide semiconductor

---

### I. Introduction

Rapid industrialization and technological innovation have led to an increase in air pollution all around the world. It has been observed that frequent exposure to air pollutants is likely to result in heart disease, respiratory illness and other health risks, which eventually leads to high mortality rates (Bassel, 1981, Odigure, 2008 & Ediagbonya, 2013). Major sources of air pollution in major cities include carbon monoxide (CO) fumes from vehicle exhausts, refinery and industrial emissions, waste incineration and occasional bush and forest burning. In Kano metropolis, Nigeria, Nabegu established that improper management of municipal solid waste is another contributing factor to environmental air pollution. It was established in the study that 80% of the population in Kano metropolis do not have access to waste collection services. Also, of the 20% that is collected, 92% of the population considered the service to be extremely poor (Nabegu, 2010). With rising population figures and these poor statistics, there is little wonder why air pollution is increasingly becoming a menace. In addition to comparing the data mining capabilities of neural networks and differential evolution, this work aims to implement softcomputing algorithms which can determine the concentration of the specified pollutants in realtime.

Differential evolution (DE) is a metaheuristic softcomputing technique which is particularly suited to multi-modal or multi-objective function problems (popularly termed non-deterministic polynomial-time or NP-hard problems). It was first proposed by Kenneth Price and Rainer Storn in 1995 when they used it to successfully solve a 33-dimensional Chebyshev polynomial (Price, Storn & Lampinen, 2006). Since then, it has found widespread application in various fields such as in economics, engineering and science to mention a few (Ardia, Boudt et al, 2011). DE is especially suitable for numerical optimization problems which have no particular defining derivative function. It is for this reason that the algorithm is termed multi-modal and derivative-free. The algorithm is said to have been successfully implemented when desired objectives have been maximised while simultaneously minimising constraints.

Artificial neural networks (ANNs) are an extremely powerful and reliable tool for implementing artificial intelligence. They have been used extensively for many years for data classification and pattern recognition. The key difference between ANNs and conventional computers with regard to data processing is that while computers require a specific set of instructions to successfully solve a problem, neural networks self-organise their internal components (neurons, axons and dendrites) to obtain the most suitable solution. This approach is based on the human brain's method of data processing which gives an appropriate response based on its ability to either excite or inhibit neural pathways based on a set of firing rules. ANNs have the ability to make sense of extremely complicated or imprecise data by detecting trends and patterns that are too complex to

be seen by humans and available computer techniques (Buhari, 2011). The first artificial neuron was applied by neurophysiologist Warren McCulloch and logician Walter Pitts in 1943.

## II. Theoretical Background

The mean square error (MSE) method of data estimation is used to obtain the estimate of an unknown quantity  $U$  in terms of certain predetermined or observed data (Guo, Shamaï & Verdu, 2005 & Shannon, 1948). Assume  $A_1, A_2, \dots, A_n$  represent a sequence of random variables about which a set of observations are available, and  $U$  represents the unknown random variable. The task is to obtain the best possible estimate for  $U$  in terms of the known observations  $A_1, A_2, \dots, A_n$ . Assume,

$$\hat{U} = \Gamma(A_1, A_2, \dots, A_n) = \Gamma(\hat{A}) \tag{1}$$

Equation (1) represents an estimate for  $U$  in terms of the known observations.  $\Gamma(\cdot)$  can be either a linear or non-linear function of  $\hat{A}$ . The error of the above estimate is given by:

$$\varepsilon(\hat{A}) = U - \hat{U} = U - \Gamma(\hat{A}) \tag{2}$$

$|\varepsilon|^2$  is the square of the above error. Since  $\varepsilon$  is also a random variable,  $M\{|\varepsilon|^2\}$  is the mean square error (MSE). MSE minimization (MMSE) criterion for estimation involves minimizing the MSE by variation over all possible forms of  $\Gamma(\cdot)$ . With this criterion, the function estimator is chosen such that the MSE is at a minimum (Bhattad & Narayanan, 2004). Under the MMSE, the best estimator for the unknown  $U$  in terms of

$A_1, A_2, \dots, A_n$  is given by the conditional mean of  $U$  given  $\hat{A}$  such that:

$$\hat{U} = \Gamma(\hat{A}) = E\{U | \hat{A}\} \tag{3}$$

The MSE is given by:

$$\delta_\varepsilon^2 = E\{|\varepsilon|^2\} = E\left\{ \left| U - \hat{U} \right|^2 \right\} = E\left\{ \left| U - \Gamma(\hat{A}) \right|^2 \right\} \tag{4}$$

The minimum value of the MSE is given by:

$$\delta_{\min}^2 = E\left\{ \left| U - E(U | \hat{A}) \right|^2 \right\} \tag{5}$$

The measured air sample points form the observed data from which the MSE can then be estimated for both good and poor air quality given the desired values for both observation sets.

DE as an optimization algorithm is best suited to non-continuous, noisy and random-type problems (Zhang & Sanderson, 2009). Examples of such problems include parallel computing, multi-objective and constrained optimization, and filter design. DE as an optimizer performs the following in order to obtain the best possible solution to a problem:

- Initialization: This is the first step in the DE optimization process and includes *boundary specification* and *index generation*. Boundary specification entails setting upper and lower limits for parameters. This is done using upper and lower initialization vectors  $v_u$  and  $v_l$ . Index generation involves using vector parameters with values ranging from 0 – g. The d-th parameter of the m-th vector has an initial value:

$$x_{d,m,0} = rand_d(0,1) \cdot (v_{d,u} - v_{d,l}) + v_{d,l} \tag{6}$$

The random number generator  $rand_d(0,1)$  returns a uniformly distributed random number within the range  $[0,1)$ . The subscript 0 is added to indicate that the value being obtained is an *initial value*. DE treats all variables internally as floating-point values irrespective of their type.

- Mutation: This involves obtaining a trial vector population (child vector) from the existing or parent population. This vector is called the *mutant vector*,  $M$  and its purpose is to control the rate at which the vector population evolves. It is obtained according to the following relation:

$$M_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \tag{7}$$

Where F is the scaling factor (with an upper limit of 1.0), r0=base vector index, i=vector index, r<sub>1</sub>, r<sub>2</sub>=difference vectors

Equation (7) represents a form of mutation called differential mutation

- Crossover: This stage distinguishes DE from other evolutionary algorithms. While most EAs have the entire population moving together, DE allows only the best fitting candidates to move on to the next stage. This stage involves crossing each vector with a mutant vector  $N_{i,g}$  such that:

$$N_{i,g} = N_{j,i,g} = \begin{cases} M_{j,i,g} & \text{if } (rand_j(0,1) \leq C_r \text{ or } j = j_{rand}) \\ x_{j,i,g} & \text{otherwise} \end{cases} \tag{8}$$

Where C<sub>r</sub>=crossover probability (0,1)

If the random number is less than or equal to C<sub>r</sub>, the trial parameter is inherited from M<sub>i,g</sub>. Otherwise it is copied from x<sub>i,g</sub>.

- Selection: DE compares the trial vector to the target vector from which it inherits its parameters. If the trial vector has an objective function value equal to or less than that of the target vector, it replaces the target vector in the next generation. Otherwise, the target vector retains its position in the population for at least one more generation. The trial vector  $t_{i,g}$  is compared to the target vector  $x_{i,g}$  according to the relation:

$$x_{i,g+1} = \begin{cases} t_{i,g} & \text{if } f(t_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \tag{9}$$

Once a new population has been selected, the steps described above are repeated until an optimum is located or the specified maximum number of generations is reached.

The DE strategy employed in this work is the DE/rand/1/bin where *rand* denotes that a candidate for mutation is randomly chosen, *1* denotes that a single difference vector is used, and *bin* shows that crossover is according to independent binomial experiments.

Artificial neural networks (ANNs) function based on biological neurons. It is a known fact that humans have the ability to deduce or learn from past experiences. This ability is adopted by ANNs to draw conclusions based on patterns they recognise from data that is presented to them. They have been used extensively in time-series prediction in finance, medical diagnosis, and fraud detection (Papert & Minsky, 1969). It should be noted that neural networks are only as good as the training set used to generate them. As a result, the training stage of neural network development is vital. Training data also has to be updated regularly if the neural network results are to remain accurate (Alkon, 1989). Neural networks are suitable for prediction and estimation especially when input and output data patterns are well understood. Supervised and unsupervised learning are the two most common methods of training neural networks. The former involves adjusting network weights in order to minimize the error between desired and actual outputs. The latter involves training the neural network in such a way that similar input data would yield the same results. In other words, the neural network is allowed to organise itself since no dependent variable(s) is/are specified in the training data. Another effective technique of training neural networks is back-propagation (developed in the 1980s). It minimizes the error by taking the difference between actual and desired results and feeding this error back through the network while adjusting network weights (Lipmann, 1987). Figure 1 shows the structure of an artificial neuron.

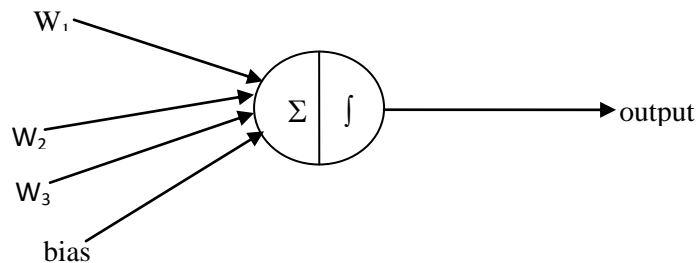


Figure 1. Artificial Neuron (Perceptron)

The neural network structure being used in this work is a feed forward neural network with learning by back-propagation. Feed forward neural networks allow data flow in only one direction (from input to output). A hidden layer makes the network more powerful by allowing it to recognize more patterns (Aleksander & Morton, 1990).

### III. Neural Network Simulation and DE Algorithm

The structure of the feedforward neural network to be used in the simulation of air samples consists of a single process layer to minimize overfitting. The transfer function used in the training process is the sigmoid or squashing function defined by the relation:

$$f(x) = \frac{1}{1 + e^{-\sum_i w_i x_i}} \tag{10}$$

Error back-propagation learning is enforced using the steep descent minimization method (Essiet, 2014). Weight and threshold coefficients between any two nodes *a* and *b* are adjusted according to the following expressions:

$$w_{ab}^{k+1} = w_{ab}^k - \eta \left( \frac{\partial E}{\partial w_{ab}} \right)^k \tag{11}$$

$$v_a^{k+1} = v_a^k - \eta \left( \frac{\partial E}{\partial v_a} \right)^k \tag{12}$$

Where  $\eta$  is the learning rate and  $E$  is the back-propagation error

Neural simulation is carried out using Neurosolutions version 5 training software. The structure of the neural network is shown in figure 2 below:

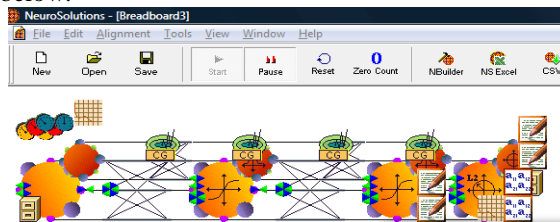


Figure 2. NeuroSolutions Breadboard Simulation of feedforward Neural Network

Air data samples were obtained using a Figaro TGS 2602 air quality monitor in three (3) separate locations around Kano metropolis namely Sabon Gari market, Kano City centre and Nassarawa GRA.

The TGS 2602 has high sensitivity to air contaminants (such as cigarette smoke) as well as hydrogen sulphide from waste materials. The sensor measurements are taken according to atmospheric temperature (°C)/% of relative humidity(RH). Measurements were generally taken at 30°C/60%RH to ensure uniformity of results.

The flowchart summarising the DE algorithm used is shown in figure 3:

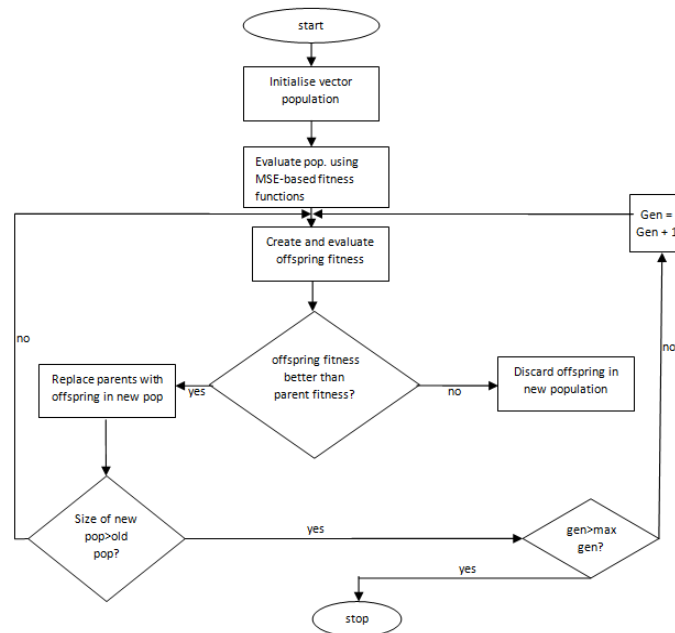


Figure 3. DE Algorithm for Air Quality Data Analysis

**IV. Discussion of Results**

The neural network and DE algorithm were both used to analyse air sample points obtained using the TGS 2602. The samples obtained at Sabon Gari market and Kano City centre were analysed as having poor quality while those obtained at Nassarawa GRA were good quality samples. 150 sample points were obtained for the poor quality class while 120 sample points were obtained for the good quality class. Data points were obtained in terms of the sensor resistance ratio at 30°C/60%RH. The sensor resistance is obtained according to the following relation:

$$R_{sensor} = \frac{V_{supply} - V_{output}}{V_{output}} \times R_{load} \tag{13}$$

Sensor resistance ratio values of between 0.80 and 0.95 were generally observed for good air quality while 1.20 - 1.65 was recorded for poor air quality. The confusion matrix results for neural network and DE respectively are as below:

**Table 1.** Confusion matrix showing accuracy of data mining results using neural network

	Good air quality (predicted)	Poor air quality (predicted)	Accuracy %
Good air quality (actual)	109	11	90.8%
Poor air quality (actual)	8	112	93.3%
Overall Accuracy			92.05%

**Table 2.** Confusion matrix showing accuracy of data mining results using differential evolution

	Good air quality (predicted)	Poor air quality (predicted)	Accuracy %
Good air quality (actual)	111	9	92.5%
Poor air quality (actual)	6	114	95.0%
Overall Accuracy			93.75%

The results obtained above show that DE was found to be slightly more accurate at predicting correct results for both sets of data compared to the neural network. Tabulated results comparing both algorithms using the proposed mean square error (MSE) method (cross-section of 30 trials) are shown below:

**Table 3.** Comparison of DE and Neural Network Using MSE (results obtained using MATLAB)

Session	Neural Network		Differential Evolution	
	Convergence rate (epochs)	MSE	Convergence Rate (epochs)	MSE
1	220	1.778E-15	117	1.888E-24
5	345	1.896E-09	168	1.954E-21
10	163	3.616E-17	202	1.432E-19
13	456	2.688E-15	196	2.623E-22
16	203	1.962E-12	216	2.851E-26
19	144	2.122E-20	200	1.081E-25
21	298	1.863E-23	114	2.869E-15
24	317	1.118E-21	109	2.452E-12
27	206	1.264E-21	120	1.814E-23
30	172	1.786E-22	153	1.659E-22

From table 3, it can be seen that the convergence rate for DE is comparatively lower than that of the neural network. DE converged in as low as 109 epochs compared to the neural network which had the lowest convergence rate of 144 epochs (session 19). The MSE results also show that DE converged with less error between actual and processed output data values. MSE recorded as low as 1.081E-25 during session 19 while the lowest for neural network was 1.863E-23 during session 21. These results demonstrate that DE has proved to be faster and more accurate in this work in the determination of air quality compared to the feedforward neural network. Both algorithms developed in this work could be embedded in appropriate hardware for the realtime estimation of air quality.

## V. Conclusion

The aim of this work, which is to compare the performance of DE and neural network in the estimation of air quality using MSE, has been achieved. The results obtained clearly show that in this work, DE outperformed the neural network in terms of mining speed and accuracy of results obtained. Both algorithms have proved to be both easy to use and accurate in terms of data mining and are therefore highly recommended for applications involving noisy and random-type data points. They can therefore be used to implement realtime air quality estimation systems which can be installed at strategic places to provide data for use by environmental monitoring agencies. This would help to combat the risks associated with air pollution especially in major cities around the world.

## References

- [1]. Aleksander, I. & Morton, H. (1990). *An Introduction to Neural Computing*.
- [2]. Alkon, D. (1989). *Memory Storage and Neural Networks*. Scientific American.
- [3]. Ardia, D., Boudt, K. et al. (2011). Differential Evolution with DEoptim: An Application to Non-Convex Portfolio Optimization. *R Journal* , 27-34.
- [4]. Ayodele, J. A. (2007). Carbon Monoxide as Indoor Pollutant in Kano Metropolis. *Journal of Applied Science and Environmental Management* , 27-30.
- [5]. Bassel, F. (1981). *Air Pollution Control*. New York: John Wiley.
- [6]. Bhattad, K. & Narayanan. (2004). An MSE Based Transformation Chart to Analyze Iterative Decoding Schemes. *42nd Allerton Conference on Communication, Control and Computing*. Monticello, Illinois.
- [7]. Buhari, M. (2011). Neural Networks. In A. Dan-Isa, *Soft Computing and Metaheuristics* (pp. 10-33). Zaria: ABU Press.
- [8]. Dekoing, H. S. (1985). *Biomass Fuel Combustion and Health*. World Health Organization.
- [9]. Ediagbonya, T. &. (2013). Air Pollution and Respiratory Morbidity in an Urban Area of Nigeria . *Greener Journal of Environmental Management and Public Safety* , 10-15.
- [10]. Essiet, I. (2014). Neural Network for the Estimation of Ammonia Concentration in Breath of Kidney Dialysis Patients. *IOSR Journal of Computer Engineering* , 61-65.
- [11]. Guo, D., Shamaï, S. & Verdu, S. (2005). Mutual Information and Minimum Mean Square Error in Gaussian Channels. *IEEE Transactions on Information Theory* , 1261-1282.
- [12]. Holland, J. (1975). *Adaptation in Natural Artificial Systems*. Michigan: University of Michigan Press.
- [13]. Lipmann, R. (April 1987). An Introduction to Computing with Neural Networks. *IEEE ASSP Magazine* .
- [14]. Nabegu, A. (2010). An Analysis of Municipal Solid Waste in Kano Metropolis, Nigeria. *Journal of Human Ecology* , 111-119.
- [15]. Odigure, J. (2008). *Safety, Loss and Pollution Prevention in Chemical Process Industries*. Minna: Jodigs & Associate.
- [16]. Papert & Minsky . (1969). *Perceptrons: An Introduction to Computational Geometry*. Massachusetts: MIT Press.
- [17]. Price, K.V., Storn, R.M. & Lampinen, J.A. (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag.
- [18]. Shannon, C. (1948). A Mathematical Theory of Communication. *Bell System Technologies Journal* , 379-423.
- [19]. Zhang, J. & Sanderson, A.C. (2009). JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation* , 945-958.