UJEAS

UNIZIK JOURNAL OF ENGINEERING AND APPLIED SCIENCES

UNIZIK Journal of Engineering and Applied Sciences 2(1), June 2023, 143-158
Journal homepage: https://journals.unizik.edu.ng/index.php/ujeas
PRINT ISSN: 2992-4383 || ONLINE ISSN: 2992-4391

Modeling and classification of oil in a multilayer artificial neural network

Aina, E.A.[1]; Odo, K.O[2*] and Omosun, Y[3]
[1,2,3]Electrical and Electronic Engineering Department, Michael Okpara University of Agriculture, Umudike, Abia State.
[*]Corresponding Author's E-mail: kayceebby@yahoo.co.uk

Abstract
This research work presents modeling and classification of oil in a multilayered artificial neural network. Oil spill is one of the major sources of pollution to the sea which can be accidental or deliberate. In order to avoid this menace in our environment, early detection of oil spills and quick interventions are of paramount importance. In this research work, oil spills classification system based on laser fluorosensor spectra data was modeled and simulated. Artificial Neural Network (ANN) toolbox in Matlab/Simulink with MLP (multi-layer perceptron) based supervised architecture was used for the simulation. The network was trained to understand numerous spectra data of laser fluorosensor for different oil spill products (light oil, medium oil, and heavy oil) and other backgrounds (water, sand and stone). The trained network was tested using data set to the network. It was found that the ANN with MLP based supervised architecture performed well when the number of neurons in hidden layers is the same and an average of 100% classification result was achieved. It was also observed that the network behaved badly and could not generalize well when the number of neurons in the two hidden layers differs. The performance, accuracy and precision were very poor in all the cases where the two hidden layers have different number of neurons.

Keywords: Mean square error, Oil classification, Artificial neural network, multi-layer perceptron, Hidden layers

1. Introduction

The effect of the oil spill on the sea ecosystem has claimed the lives of so many aquatic animals. Whatever the source of oil spill is, oil spill pollution will continue to occur, therefore, in order to lessen its effect, the improvement of its detection and continuous monitoring are the most important issues to effectively plan countermeasure responses (Akkartal *et al.*, 2008). Classification of oil spills into various oil types is important because it helps the response team to know the type of instrument to be deployed during clean-up processes. Oil spills are often classify into light, medium and heavy oil because the instruments use to clean up all light oils are the same and the instruments used to clean up all medium crudes are the same and also the instruments use to clean up all heavy oils are the same. Since the major reason for oil spill classification is to know the type of instruments to be used during clean up, it is therefore reasonable to limit the classification of oil spills to light, medium and heavy oils, (Maya, *et al.*, 2014).

Oil spills occur after oil transportation, oil drilling and accidental collision or sinking of oil tankers, failures in pipelines and oil rigs, etc. Small spills are easier to handle effectively with existing technology. Size matters in oil spills and large spills are more important. On the other hand, recent studies have shown that major spill incidents have been fewer in number; the broad public exhibits a "memory" on the major spills, but generally remains unaware that minor spills happen daily, (Panagiota et al, 2021). This research work will help the oil spill monitoring team to respond fast to oil spills incidences in Nigeria, hence, preventing the spread of oil spills and the damages that it can cause to the environment. The research work will also help the oil producing states, areas and communities in Nigeria, to avoid the pollution caused by oil spills in their environments to a large extent.

In this research work, a classification system was therefore simulated by training an MLP neural network with laser fluorosensor spectra data. The effect of different number of hidden layers and different number of neurons in hidden layers was also checked on the performance and accuracy of the network. The primary goal in oil spill detection is to positively distinguish oil from the background, i.e. water, ice etc. Once oil is detected, it can be further classified into various oil types. Once classified, actions can be taken to respond to the oil spill and to model the oil spill drift and spreading. The Pearson Correlation Coefficient, Principal Component Analysis (PCA) and Artificial neural network have been widely used for oil spill detection and classification (Maya *et al.*, 2014).

Artificial neural network is a kind of simulation system that simulates the information processing of human brain and can be self-organized and self-adaptive. They are multi-layer network of neurons that are used to classify things and make predictions. The network comprises of input layer, hidden layer and output layer.
There are three types of an artificial neural network models which are: Perceptron network, Multi-layer Perceptron (MLP) network and Self-organizing maps (SOM) network. Perceptron network can classify linearly separable input vectors very well. The training technique used is called Perceptron learning rule. They are only suitable for problems in pattern classification. Perceptron has only two layers, namely one input and one output.

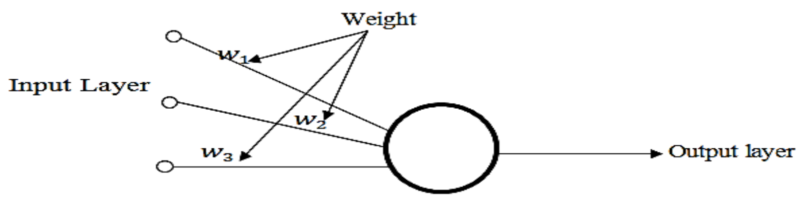

Figure 1: Perceptron model of an artificial neural network (Aharkava *et al*., 2010)

A multi-layer Perceptron has at least three layers; one input layer, one hidden layer and one output layer. MLP can have more than one hidden layer and many neurons in each of the hidden layer. The complexity of MLP depends on the number of hidden layers. MLP utilizes supervised learning technique called back propagation for training (Rosenblatt *et al.*, 2009). Its multiple layers and non-linear activation distinguish it from a linear Perceptron. It can classify any form of data and data that is not linearly separable (Cybenko, 2011).

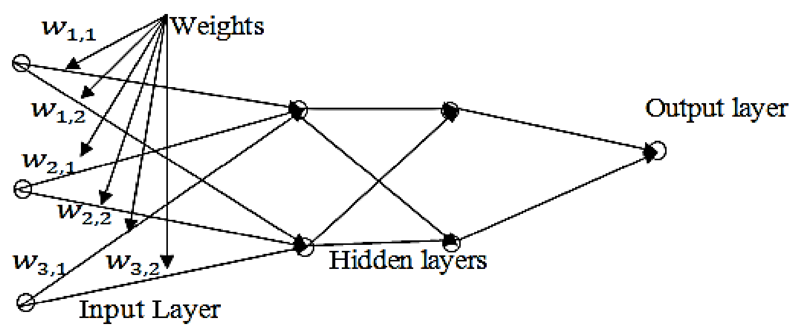

Figure 2: Multi-layer Perceptron model of an artificial neural network (Gil *et al.*, 2010)

In self-organizing map (SOM), the network is trained using unsupervised learning process. It differs from Perceptron and MLP as it apply competitive learning process as opposed to error-correction learning such as back propagation with gradient descent (Bishop, 1995). Also, SOM uses neighborhood function to preserve the topological properties of the input space.

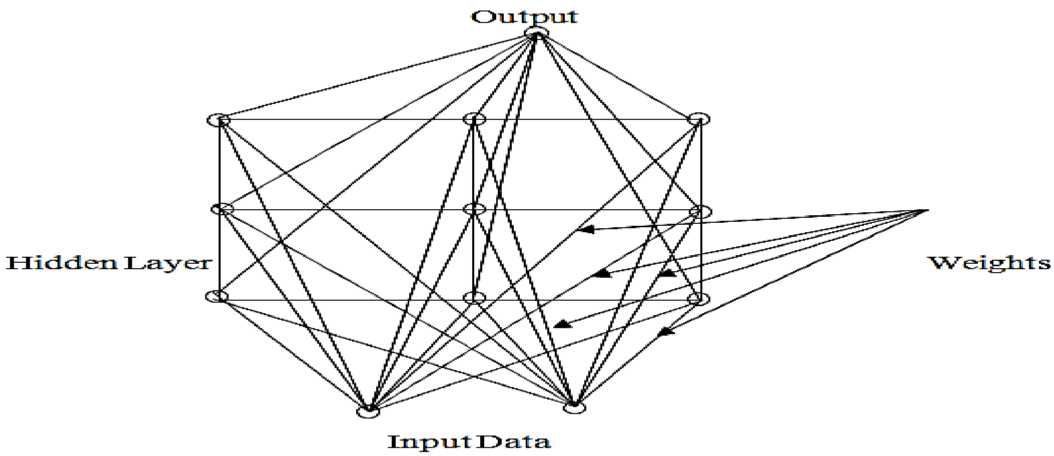

Figure 3: Self-organizing map (SOM) artificial neural network (Gil *et al.*, 2010)

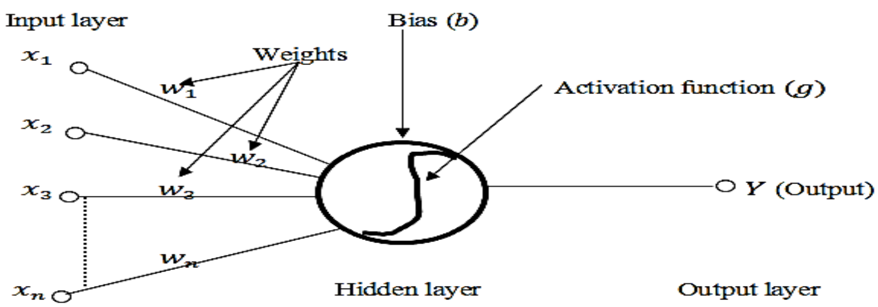

Figure 4: A typical artificial neural network model (Aharkava *et al.*, 2010)

The general algorithm for an artificial neural network as stated in (Aharkava *et al*., 2010) is given as
Y (Predicted Probability) = $g[(\sum_{i=1}^{n} x_i w_i) + b]$ (1)
Where Y is the predicted output signal, g is the activation function, w is the weight, x is the input signal and b is the bias.

Panagiota et al, (2021) reviewed Oil Spill Modeling. Several oil spill simulation models exist in the literature, which are used worldwide to simulate the evolution of an oil slick created from marine traffic, petroleum production, or other sources. These models may range from simple parametric calculations to advanced, new-generation, operational, three-dimensional numerical models, coupled to meteorological, hydrodynamic, and wave models, forecasting in high-resolution and with high precision the transport and fate of oil. Vahabi and Selviah (2019) researched on convolutional Neural Networks to Classify Oil, Water and Gas Wells Fluid Using Acoustic Signals. Their work investigated the classification algorithms that identify the fluid type in oil, water and gas pipes using acoustic signals. The datasets analyzed in their study were collected from real oil, water and gas well pipes under the sea where there is no controlled environment and data contains lots of noisy signals due to unpredicted events under the sea.

Nikolas (2018) reviewed different commercial softwares used for oil spill simulation. The paper explained the essence of oil spill simulation, to predict the horizontal movement of surface oil slick, the vertical distribution of oil particles, the concentration in the water column and the mass balance of those spills. The paper presented some Commercial softwares used to simulate oil spill, such as OILMAP, TRANSAS, OILFLOW2D, OSCAR, ANSYS. These softwares have the capacity to predict the horizontal movement of surface oil slick, the vertical distribution of oil particles, the concentration in the water column and the mass balance of spilled mentioned above. From all the literatures reviewed, it can be seen that none of the authors has used laser fluorosensor spectra data to train a multi-layer Perceptron (MLP) artificial neural network for the simulation of oil spills. Also, none of the authors checked the effect of different number of hidden layers and the number of neurons in hidden layers on the performance and accuracy of MLP network. Therefore, the purpose of this research work is to analyze and simulate a remote sensing system that will detect oil spills and classify oil spills into different products.

2.0 Materials and methods

The materials required in this research work are Laser fluorosensor spectra data of different oil spill products and other backgrounds collected from national oil spill detection and response agency in Nigeria (NOSDRA), Multi-Layer Perceptron (MLP), Artificial Neural Network, MATLAB/SIMULINK software and HP laptop.

2.1 Methods

The laser fluorosensor spectra data was tabulated and analyzed with Microsoft excel. The oil spills classification system was first modeled using a Multi-Layer Perceptron (MLP) model of an Artificial Neural Network technique. Then a MLP network training flow chart/oil spill classification scheme was developed. The oil spills classification system was simulated using an Artificial Neural Network toolbox in Matlab/Simulink with MLP based supervised architecture. A back propagation learning algorithm with an optimizer based on gradient descent method was used during the training of the network. The trained network was then used to distinguished oil spills from various backgrounds and also classified oil spills into different products. The performance, accuracy and precision of the trained network were evaluated by mean square error.

2.1.1 Multi-layer Perceptron (MLP) Model

The MLP model used in this research work has one input layer that receives external input, two hidden layers where transformation is taking place and one output layer which generates the classification results. Each neuron in the input and the hidden layers is connected to all neurons in the next layer by weighted connections (Gil *et al*., 2010). The neurons of the hidden layers compute weighted sums of their inputs and add a bias. The resulting sums are used to calculate the activity of the neurons by applying a sigmoid activation function (Gil *et al*., 2010).

Each neuron n_j is associated with a weight vector, $w_j \in R^n$.

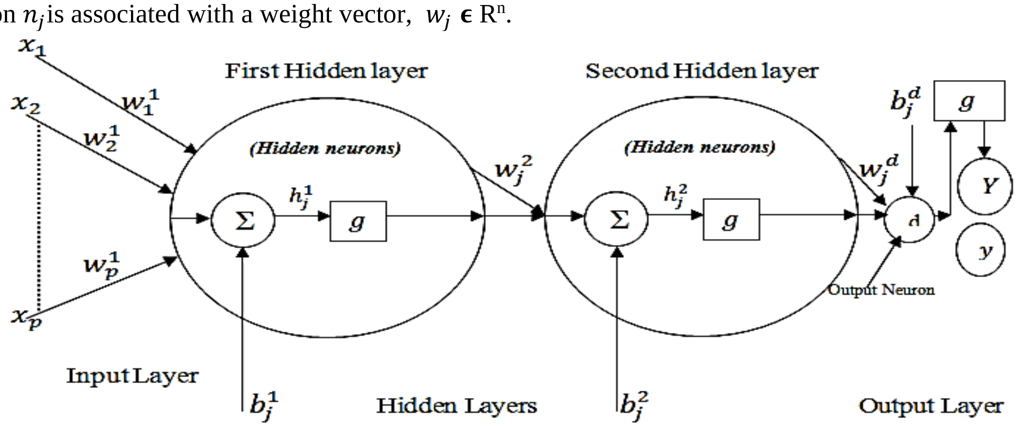

Figure 5: Multi-Layer Perceptron model with one input layer, two hidden layers and one output layer (Gil *et al*., 2010)

x_i= Input sample

w_i^1= Connection weight of a neuron in the first hidden layer

h_j^1 = The weighted sum plus the bias b_j^1 to each neuron in the first hidden layer

w_j^2= Connection weight of a neuron in the second hidden layer

h_j^1 = The weighted sum plus the bias b_j^2 to each neuron in the second hidden layer

w_j^d= Connection weight of neuron d in the output layer

b_j^d = The bias for the neuron d in the output layer.

Σ = Transfer function

g = Activation function

Y = Predicted output

y = Actual output

The neuron outputs quantity is expressed by relation

$$Y_{pd} = g(m_{pd}) = \frac{1}{[1+ exp^{(-m_{pd})}]} \quad (2)$$

Weights from the second hidden layer is given by (Singh *et al*., 2008) as

$$w_j^d(t+1) = w_j^d(t) + \Delta w_j^d \quad (3)$$

MLP network is trained by following the processes;

a. Initialize the weights to small random values.
b. Choose an input vector and propagate it forward. This yields values for v_j, f_j and Y, the outputs from the first hidden layer and the second hidden layer and output layer respectively.
c. Compute mean square error
d. Update weights

Since the input layer, the hidden layers and the output layer are all vectors, then, matrices can be used to summarize all the equations. The artificial neural network designed in this research work has six input signals, two hidden layers and six outputs. Each of the hidden layers has seven neurons which are arranged vertically. The structure is represented in Figure 6

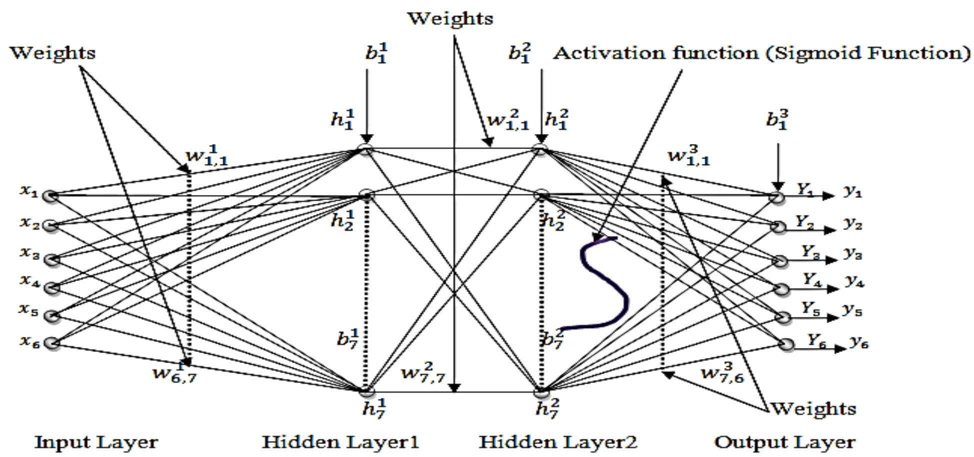

Figure 6: Artificial Neural Network Model (Multi-Layer Perceptron supervised based architecture); with six inputs, two hidden layers and six outputs.

Figure 6 represents the architecture of the artificial neural network used in this research work. It has six inputs (x_1, x_2... x_6), two hidden layers; the first hidden layer has (h_1^1, h_2^1…h_7^1) neurons and the second hidden layer has (h_1^2, h_2^2…h_7^2) neurons and the output layer has six outputs (y_1, y_2…y_6). In the first hidden layer are weights ($w_{1,1}^1$, $w_{1,2}^1$ $w_{1,3}^1$, ….$w_{1,7}^1$, $w_{2,1}^1$, $w_{2,2}^1$ $w_{2,3}^1$, ….$w_{2,7}^1$, $w_{3,1}^1$, $w_{3,2}^1$ $w_{3,3}^1$, ….$w_{3,7}^1$, $w_{4,1}^1$, $w_{4,2}^1$ $w_{4,3}^1$, ….$w_{4,7}^1$, $w_{5,1}^1$, $w_{5,2}^1$ $w_{5,3}^1$, ….$w_{5,7}^1$, $w_{6,1}^1$, $w_{6,2}^1$ $w_{6,3}^1$, ….$w_{6,7}^1$).

Similarly, in the second hidden layer are weights ($w_{1,1}^2$, $w_{1,2}^2$, $w_{1,3}^2$…..$w_{1,7}^2$, $w_{2,1}^2$, $w_{2,2}^2$, $w_{3,3}^2$…..$w_{2,7}^2$, $w_{3,1}^2$, $w_{3,2}^2$, $w_{3,3}^2$…..$w_{3,7}^2$, $w_{4,1}^2$, $w_{4,2}^2$, $w_{4,3}^2$…..$w_{4,7}^2$, $w_{5,1}^2$, $w_{5,2}^2$, $w_{5,3}^2$…..$w_{5,7}^2$, $w_{6,1}^2$, $w_{6,2}^2$, $w_{6,3}^2$…..$w_{6,7}^2$, $w_{7,1}^2$, $w_{7,2}^2$, $w_{7,3}^2$…..$w_{7,7}^2$). And the the output layer are weights ($w_{1,1}^3$, $w_{1,2}^3$, $w_{1,3}^3$…..$w_{1,6}^3$, $w_{2,1}^3$, $w_{2,2}^3$, $w_{3,3}^3$…..$w_{2,6}^3$, $w_{3,1}^3$, $w_{3,2}^3$, $w_{3,3}^3$…..$w_{3,6}^3$, $w_{4,1}^3$, $w_{4,2}^3$, $w_{4,3}^3$…..$w_{4,6}^3$, $w_{5,1}^3$, $w_{5,2}^3$, $w_{5,3}^3$…..$w_{5,6}^3$, $w_{6,1}^3$, $w_{6,2}^3$, $w_{6,3}^3$…..$w_{6,6}^3$, $w_{7,1}^3$, $w_{7,2}^3$, $w_{7,3}^3$…..$w_{7,6}^3$).

Also biases are applied to neurons in each hidden layer and the output layer. The biases in the first hidden layer are (b_1^1, b_2^1,…. b_7^1). Similarly the biases in the second hidden layer are (b_1^2, b_2^2,…. b_7^2). And the biases in the output layer are (b_1^3, b_2^3,…. b_6^3). The activation function used in this research work is a sigmoid function. The weighted sum of all the neurons in the first hidden layer are represented with the matrices in equation 4.

$$\begin{bmatrix} h_1^1 \\ h_2^1 \\ h_3^1 \\ h_4^1 \\ h_5^1 \\ h_6^1 \\ h_7^1 \end{bmatrix} = \begin{bmatrix} w_{1,1}^1 & w_{2,1}^1 & w_{3,1}^1 & w_{4,1}^1 & w_{5,1}^1 & w_{6,1}^1 \\ w_{1,2}^1 & w_{2,2}^1 & w_{3,2}^1 & w_{4,2}^1 & w_{5,2}^1 & w_{6,2}^1 \\ w_{1,3}^1 & w_{2,3}^1 & w_{3,3}^1 & w_{4,3}^1 & w_{5,3}^1 & w_{6,3}^1 \\ w_{1,4}^1 & w_{2,4}^1 & w_{3,4}^1 & w_{4,4}^1 & w_{5,4}^1 & w_{6,4}^1 \\ w_{1,5}^1 & w_{2,5}^1 & w_{3,5}^1 & w_{4,5}^1 & w_{5,5}^1 & w_{6,5}^1 \\ w_{1,6}^1 & w_{2,6}^1 & w_{3,6}^1 & w_{4,6}^1 & w_{5,6}^1 & w_{6,6}^1 \\ w_{1,7}^1 & w_{2,7}^1 & w_{3,7}^1 & w_{4,7}^1 & w_{5,7}^1 & w_{6,7}^1 \end{bmatrix} \mathrm{X} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} b_1^1 \\ b_2^1 \\ b_3^1 \\ b_4^1 \\ b_5^1 \\ b_6^1 \\ b_7^1 \end{bmatrix} \quad (4)$$

The output vector v_j from the first hidden layer can also be determined as

$$v_j = g\begin{bmatrix} h_1^1 \\ h_2^1 \\ h_3^1 \\ h_4^1 \\ h_5^1 \\ h_6^1 \\ h_7^1 \end{bmatrix} = \frac{1}{[1+\exp\left(-h_{ij}^1(t)\right)]}\begin{bmatrix} h_1^1 \\ h_2^1 \\ h_3^1 \\ h_4^1 \\ h_5^1 \\ h_6^1 \\ h_7^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{[1+(e^{-h_1^1})]} \\ \frac{1}{[1+(e^{-h_2^1})]} \\ \frac{1}{[1+(e^{-h_3^1})]} \\ \frac{1}{[1+(e^{-h_4^1})]} \\ \frac{1}{[1+(e^{-h_5^1})]} \\ \frac{1}{[1+(e^{-h_6^1})]} \\ \frac{1}{[1+(e^{-h_7^1})]} \end{bmatrix} \quad (5)$$

Also the weighted sum of all the neurons in the second hidden layer are represented with matrices in equation 6

$$\begin{bmatrix} h_1^2 \\ h_2^2 \\ h_3^2 \\ h_4^2 \\ h_5^2 \\ h_6^2 \\ h_7^2 \end{bmatrix} = \begin{bmatrix} w_{1,1}^2 & w_{2,1}^2 & w_{3,1}^2 & w_{4,1}^2 & w_{5,1}^2 & w_{6,1}^2 & w_{7,1}^2 \\ w_{1,2}^2 & w_{2,2}^2 & w_{3,2}^2 & w_{4,2}^2 & w_{5,2}^2 & w_{6,2}^2 & w_{7,2}^2 \\ w_{1,3}^2 & w_{2,3}^2 & w_{3,3}^2 & w_{4,3}^2 & w_{5,3}^2 & w_{6,3}^2 & w_{7,3}^2 \\ w_{1,4}^2 & w_{2,4}^2 & w_{3,4}^2 & w_{4,4}^2 & w_{5,4}^2 & w_{6,4}^2 & w_{7,4}^2 \\ w_{1,5}^2 & w_{2,5}^2 & w_{3,5}^2 & w_{4,5}^2 & w_{5,5}^2 & w_{6,5}^2 & w_{7,5}^2 \\ w_{1,6}^2 & w_{2,6}^2 & w_{3,6}^2 & w_{4,6}^2 & w_{5,6}^2 & w_{6,6}^2 & w_{7,6}^2 \\ w_{1,7}^2 & w_{2,7}^2 & w_{3,7}^2 & w_{4,7}^2 & w_{5,7}^2 & w_{6,7}^2 & w_{7,7}^2 \end{bmatrix} \mathrm{X} \begin{bmatrix} \frac{1}{[1+(e^{-h_1^1})]} \\ \frac{1}{[1+(e^{-h_2^1})]} \\ \frac{1}{[1+(e^{-h_3^1})]} \\ \frac{1}{[1+(e^{-h_4^1})]} \\ \frac{1}{[1+(e^{-h_5^1})]} \\ \frac{1}{[1+(e^{-h_6^1})]} \\ \frac{1}{[1+(e^{-h_7^1})]} \end{bmatrix} + \begin{bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \\ b_4^2 \\ b_5^2 \\ b_6^2 \\ b_7^2 \end{bmatrix} \quad (6)$$

The output vector f_j from the second hidden layer can also be determined as

$$f_j = g\ (h_j^2) = g\begin{bmatrix} h_1^2 \\ h_2^2 \\ h_3^3 \\ h_4^4 \\ h_5^5 \\ h_6^6 \\ h_7^7 \end{bmatrix} = \frac{1}{[1+\exp\left(-h_{ij}^1(t)\right)]}\begin{bmatrix} h_1^2 \\ h_2^2 \\ h_3^3 \\ h_4^4 \\ h_5^5 \\ h_6^6 \\ h_7^7 \end{bmatrix} = \begin{bmatrix} \frac{1}{[1+(e^{-h_1^2})]} \\ \frac{1}{[1+(e^{-h_2^2})]} \\ \frac{1}{[1+(e^{-h_3^2})]} \\ \frac{1}{[1+(e^{-h_4^2})]} \\ \frac{1}{[1+(e^{-h_5^2})]} \\ \frac{1}{[1+(e^{-h_6^2})]} \\ \frac{1}{[1+(e^{-h_7^2})]} \end{bmatrix} \quad (7)$$

Finally, the weighted sum of all the neurons in the output layer are represented with matrices in equation 8

$$\begin{bmatrix} m_{d1} \\ m_{d2} \\ m_{d3} \\ m_{d4} \\ m_{d5} \\ m_{d6} \end{bmatrix} = \begin{bmatrix} w_{1,1}^3 & w_{2,1}^3 & w_{3,1}^3 & w_{4,1}^3 & w_{5,1}^3 & w_{6,1}^3 & w_{7,1}^3 \\ w_{1,2}^3 & w_{2,2}^3 & w_{3,2}^3 & w_{4,2}^3 & w_{5,2}^3 & w_{6,2}^3 & w_{7,2}^3 \\ w_{1,3}^3 & w_{2,3}^3 & w_{3,3}^3 & w_{4,3}^3 & w_{5,3}^3 & w_{6,3}^3 & w_{7,3}^3 \\ w_{1,4}^3 & w_{2,4}^3 & w_{3,4}^3 & w_{4,4}^3 & w_{5,4}^3 & w_{6,4}^3 & w_{7,4}^3 \\ w_{1,5}^3 & w_{2,5}^3 & w_{3,5}^3 & w_{4,5}^3 & w_{5,5}^3 & w_{6,5}^3 & w_{7,5}^3 \\ w_{1,6}^3 & w_{2,6}^3 & w_{3,6}^3 & w_{4,6}^3 & w_{5,6}^3 & w_{6,6}^3 & w_{7,6}^3 \end{bmatrix} \text{X} \begin{bmatrix} \frac{1}{[1+(e^{-h_1^2})]} \\ \frac{1}{[1+(e^{-h_2^2})]} \\ \frac{1}{[1+(e^{-h_3^2})]} \\ \frac{1}{[1+(e^{-h_4^2})]} \\ \frac{1}{[1+(e^{-h_5^2})]} \\ \frac{1}{[1+(e^{-h_6^2})]} \\ \frac{1}{[1+(e^{-h_7^2})]} \end{bmatrix} + \begin{bmatrix} b_1^3 \\ b_2^3 \\ b_3^3 \\ b_4^3 \\ b_5^3 \\ b_6^3 \end{bmatrix} \quad (8)$$

Therefore the predicted output is given by

$$\text{Y} = g \begin{bmatrix} m_{d1} \\ m_{d2} \\ m_{d3} \\ m_{d4} \\ m_{d4} \\ m_{d5} \end{bmatrix} = \begin{bmatrix} \frac{1}{[1+(e^{m_{d1}})]} \\ \frac{1}{[1+(e^{m_{d2}})]} \\ \frac{1}{[1+(e^{m_{d3}})]} \\ \frac{1}{[1+(e^{m_{d4}})]} \\ \frac{1}{[1+(e^{m_{d5}})]} \\ \frac{1}{[1+(e^{m_{d6}})]} \end{bmatrix} \quad (9)$$

Then,

$Y_1 = \frac{1}{[1+(e^{m_{d1}})]}$

$Y_2 = \frac{1}{[1+(e^{m_{d2}})]}$

$Y_3 = \frac{1}{[1+(e^{m_{d3}})]}$

$Y_4 = \frac{1}{[1+(e^{m_{d4}})]}$

$Y_5 = \frac{1}{[1+(e^{m_{d5}})]}$

$Y_6 = \frac{1}{[1+(e^{m_{d6}})]}$

Then the Mean square error (MSE) becomes

$$E = \frac{1}{2}\sum_{i=1}^{p}(y-Y)^2 = \frac{1}{2}\sum_{i=1}^{6}\left(\begin{bmatrix} y1 \\ y2 \\ y3 \\ y4 \\ y5 \\ y6 \end{bmatrix} - \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{bmatrix}\right)^2 \quad (10)$$

3.0 Results and Discussions

The dynamic behaviour of MLP model of an artificial neural network was simulated using an Artificial Neural Network toolbox in Matlab/Simulink with MLP based supervised architecture. The simulated MLP network architecture has six input samples in the input layer, two hidden layers and six outputs in the output layer. A Matlab program was written to train the network and to determine the mean square error of the training set, validation set and the test set and hence determine the output (predictions). The predicted value is gotten by the simulation of equation 2 and equation 9. The MSE value is gotten from the simulation of equation 10.

Table 1: The target substances and their corresponding target values (actual values)

Target (Substances)	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
	0	0	0	1	1	1
Target value	0	1	1	0	0	1
	1	0	1	0	1	0

The target value in the Table 1 corresponds to each of the training spectra of the target substances. The values are in form of readable binary files representing the actual output (expected output) of the network. During the training of the network, the total data set was divided into training set, validation set and test set. Seventy percent (70%) of the data was used as a training set, twenty percent (20%) was used as a validation set and ten percent (10%) was used as a test set. The learning rate was set to 0.05 and the epoch (number of iterations) was set to 1000. A back propagation learning algorithm with an optimizer based on gradient descent method was used during the training of the network. The parameter used in training the network is summarized in the Table 2

Table 2: The network training parameters

PARAMETERS	VALUE
Learning rate	0.05
Epochs	1000
Number of hidden layers	2, 3, and 4
Number of neurons in the hidden layers	(7,7), (7, 7, 7), (7, 7, 7,7), (7,8), (9,7), (8,10) and (10,10)
Training data set	70% of the total data set
Validation data set	20% of the total data set
Test data set	10% of the total data set
Learning Algorithm/Optimizer	Back propagation/Gradient descent

The results obtained when the network was trained with two hidden layers and each hidden layer having seven neurons is shown in Table 3.

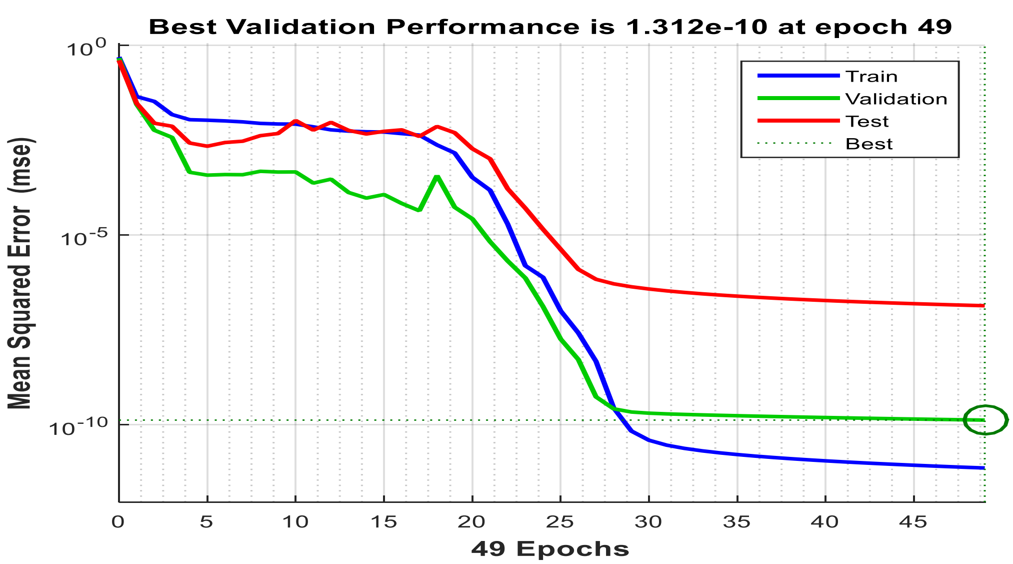

Figure 8: Graph of mean square error against 49 epochs

Table 3: Two hidden layers with seven neurons

No of Hidden layers/neurons	[7 7]					
Substance	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
Actual Value	0	0	0	1	1	1
	0	1	1	0	0	1
	1	0	1	0	1	0
Predicted Value	3.59E-08	-1.56E-07	-1.14E-07	1	1	1
	1.62E-07	1	1	3.32E-07	4.94E-07	1
	0.99999	2.49E-07	0.999999	-1.00E-07	0.999999	1.50E-07
MSETrain	0.000003					
MSEVal	0.00001					
MSETest	0.0004					
Training time	21sec					
% Classification achieved	100%					

Figure 8 shows the performance curve obtained when the network was trained with two hidden layers and each hidden layer having seven neurons. It was observed that the best validation performance was obtained when the epochs was forty nine (49) and the mean square error gotten from the simulation of equation 10 was $1.312 * 10^{-10}$. Since the means square error is so small and is even tending to zero, it indicates good performance. The results obtained when the network was trained with three hidden layers and each hidden layer having seven neurons is shown in Table 4.

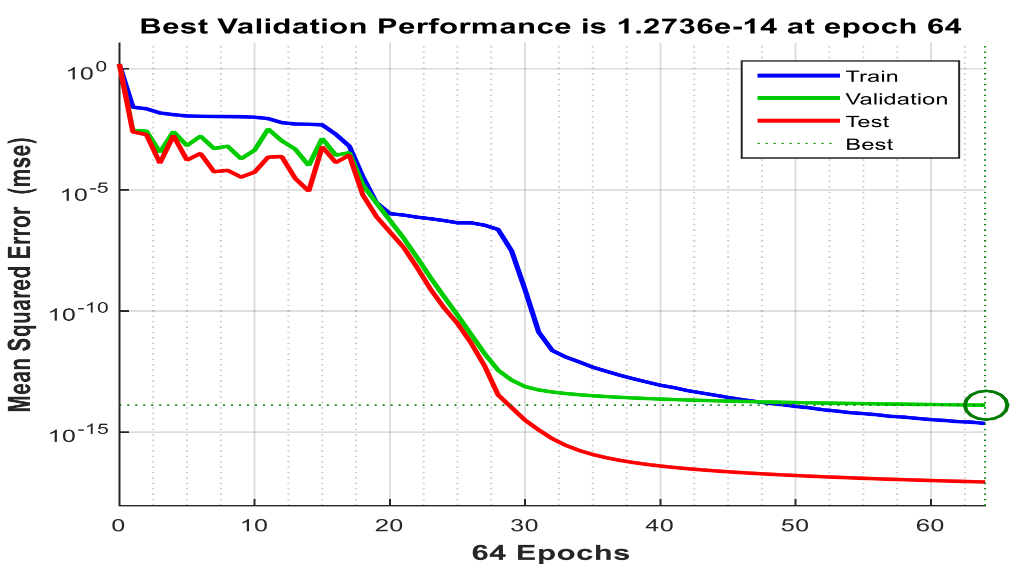

Figure 9: Graph of mean square error against 64 epochs

Table 4: Three layers with seven neurons

No. of Hidden Layers/Neurons	[7 7 7]					
Substance	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
	0	0	0	1	1	1
Actual value	0	1	1	0	0	1
	1	0	1	0	1	0
	5.96E-08	-7.41E-08	8.72E-08	1	1	1
Predicted value	1.24E-07	1	1	1.08E-07	3.15E-07	1
	1	9.65E-08	1	-1.71E-08	1	1E-09
MSETrain	0.00000006					
MSEVal	0.0000001					
MSETest	0.000000003					
Training time	32sec					
% Classification achieved	100%					

Figure 9 shows the performance curve obtained when the network was trained with three hidden layers and each hidden layer having seven neurons. It is seen that the best validation performance was obtained when the epochs was sixty four (64) and the means square error gotten from the simulation of equation 10 was $1.2736 * 10^{-14}$. The means square error here is also tending to zero and it also indicates good performance. The results obtained when the network was trained with four hidden layers and each hidden layer having seven neurons is shown in Table 5.

Table 5: Four hidden layers with seven neurons

No. of Hidden Layers/Neurons	[7 7 7 7]					
Substance	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
	0	0	0	1	1	1
Actual value	0	1	1	0	0	1
	1	0	1	0	1	0
	3.23E-08	7.86E-08	5.98E-08	1	1	1
Predicted value	8.51E-08	1	1	-7.35E-08	-9.05E-09	1
	1	7.15E-08	1	-1.79E-08	1	4.05E-08
MSETrain	0.000007					
MSEVal	0.00001					
MSETest	0.00005					
Training time	58sec					
% Classification achieved	100%					

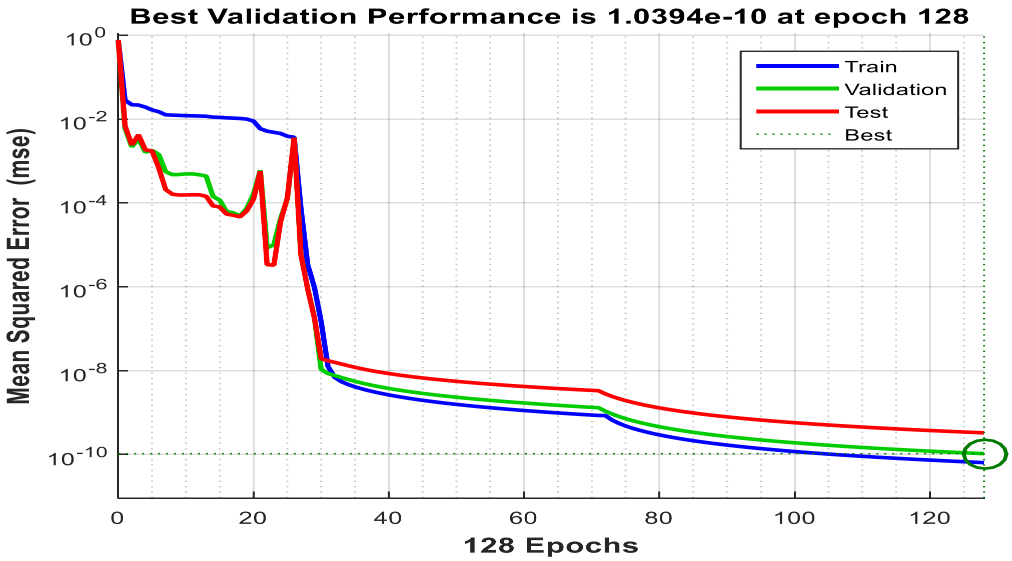

Figure 10: Graph of mean square error against 128 epochs

Figure 10 shows the performance curve obtained when the network was trained with four hidden layers and each hidden layer having seven neurons. It was observed that the best validation performance was obtained when the epochs were one hundred and twenty eight (128) and the means square error gotten from the simulation of equation 10 was$1.0394 * 10^{-10}$. The means square error is also small like the one obtained when the network was trained with three hidden layers and it also indicates good performance. The results obtained when the network was trained with two hidden layers and one of the hidden layers having seven neurons and the other eight neurons is shown in Table 6.

Table 6: Two hidden layers with seven and eight neurons

No. of Hidden Layers/Neurons	[7 8]					
Substance	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
	0	0	0	1	1	1
Actual value	0	1	1	0	0	1
	1	0	1	0	1	0
	0.6098	0.5878	0.0288	0.1035	0.4393	0.6515
Predicted value	0.5150	0.5148	0.1871	0.4042	0.1326	0.5968
	0.2744	0.2743	0.1049	0.2079	0.3571	0.3268
MSETrain	0.1025					
MSEVal	0.0200					
MSETest	0.0206					
Training time	18sec					
% Classification achieved	42%					

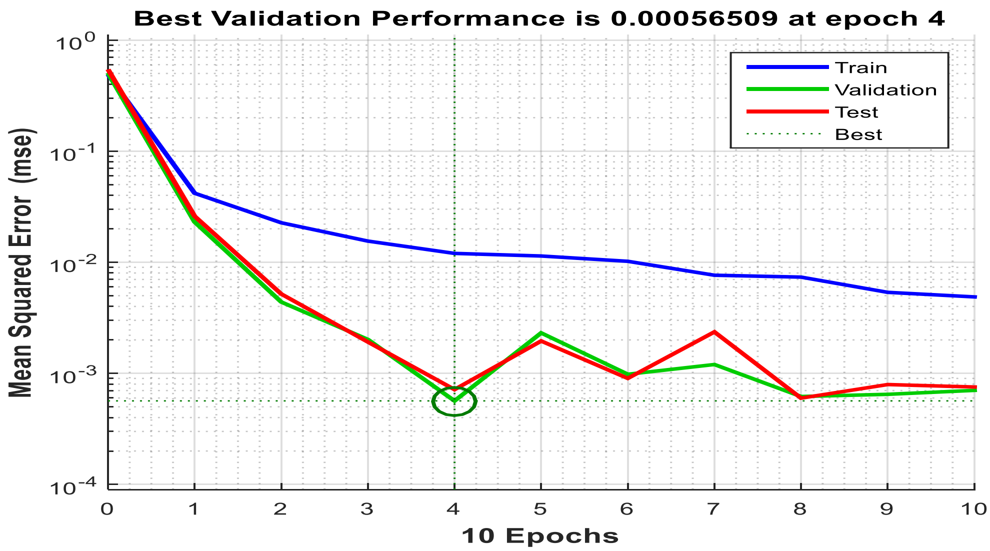

Figure 11: Graph of mean square error against 10 epochs

Figure 11 shows the performance curve obtained when the network was trained with two hidden layers and one of the hidden layers having seven neurons and the other eight neurons. It is seen that the best validation performance was obtained when the epochs were ten (10) and the mean square error gotten from the simulation of equation 10 was 0.00056509. The mean square error here is far larger than the ones obtained in Figures 8, 9 and 10 respectively and this indicates a very poor performance. The results obtained when the network was trained with two hidden layers and one of the hidden layers having seven neurons and the other nine neurons is shown in Table 7.

Table 7: Two hidden layers with seven and nine neurons

No. of Hidden Layers/Neurons	[9 7]					
Substance	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
	0	0	0	1	1	1
Actual value	0	1	1	0	0	1
	1	0	1	0	1	0
	0.3194	-0.1171	0.2006	-0.1106	-0.1256	0.6467
Predicted value	0.2320	-0.0557	0.2273	-0.1432	0.0832	0.7371
	0.4369	-0.01816	0.1686	-0.1575	0.1207	0.0454
MSETrain	0.1995					
MSEVal	0.1441					
MSETest	0.1511					
Training time	52sec					
% Classification achieved	7.2%					

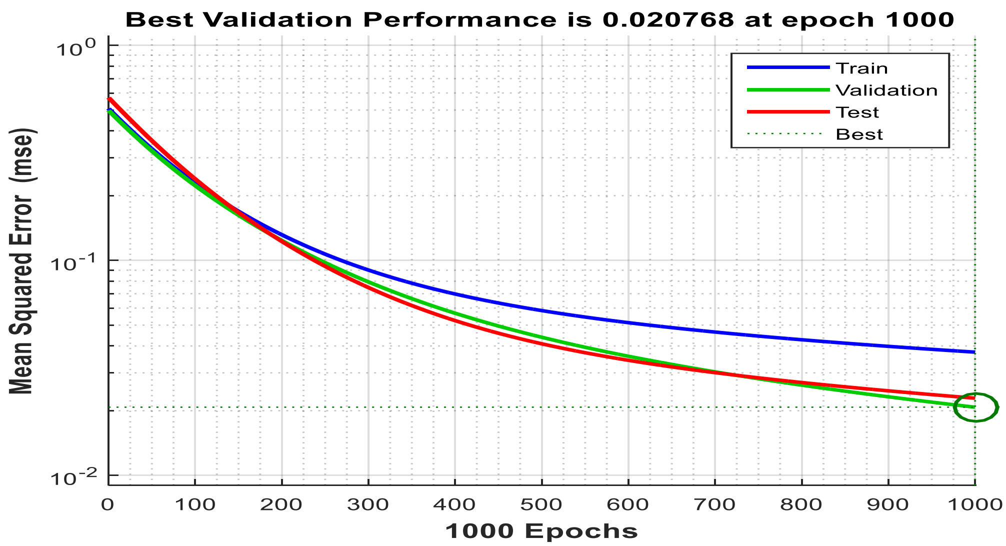

Figure 12: Graph of mean square error against 1000 epochs

Figure 12 shows the performance curve obtained when the network was trained with two hidden layers and one of the hidden layers having seven neurons and the other nine neurons. It was observed that the best validation performance was obtained when the epochs was one thousand (1000) and the mean square error gotten from the simulation of equation 10 was 0.020768. The mean square error in this case is really very high. The epochs (number of iterations) were exhausted and yet the validation curve has not flattened up (converged), this indicates a very poor performance. The results obtained when the network was trained with two hidden layers and one of the hidden layers having eight neurons and the other ten neurons is shown in Table 8.

Table 8: Two layers with eight and ten neurons

No. of Hidden Layers/Neurons	[8 10]					
Substance	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
	0	0	0	1	1	1
Actual value	0	1	1	0	0	1
	1	0	1	0	1	0
	-0.0612	0.2069	0.1151	-0.2417	0.1559	-0.0578
Predicted value	-0.0522	0.1050	0.2166	-0.2348	0.1484	-0.0643
	-0.2711	0.2003	0.1835	-0.2194	0.1326	-0.0731
MSETrain	0.1663					
MSEVal	0.2114					
MSETest	0.2817					
Training time	122sec					
% Classification achieved	5.7%					

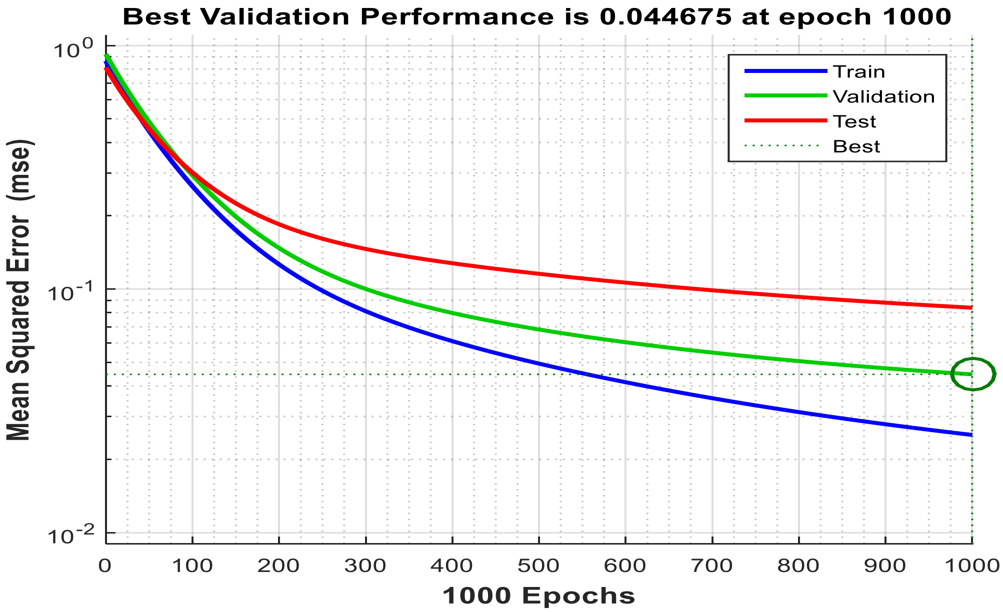

Figure 13: Graph of mean square error against 1000 epochs

Figure 13 shows the performance curve obtained when the network was trained with two hidden layers and one of the hidden layers having eight neurons and the other ten neurons. It is seen that the best validation performance was obtained when the epochs was one thousand (1000) and the mean square error gotten from the simulation of equation 10 was 0.044675. This is similar to what was obtained in Figure 12. The mean square error is also very high. The epochs (number of iterations) was exhausted and yet the validation curve has not flattened up (converged), this indicates a very poor performance. The results obtained when the network was trained with two hidden layers and each having ten neurons is shown in Table 9.

Table 9: Two hidden layers with ten neurons

No. of Hidden Layers/Neurons	[10 10]					
	Light Oil	Medium Crude	Water	Heavy Oil	Sand	Stone
Substance						
	0	0	0	1	1	1
Actual value	0	1	1	0	0	1
	1	0	1	0	1	0
	9.7E-05	0.00001	0.00002	0.99994	1.00000	0.99997
Predicted value	1.5E-05	1.0000	1.00027	-0.00014	-4.8E-05	0.99997
	1.0000	1.9E-05	1.00009	-4.7E-06	1.00007	1.1E-05
MSETrain	0.00000003					
MSEVal	0.0000005					
MSETest	0.0000009					
Training time	42sec					
% Classification achieved	100%					

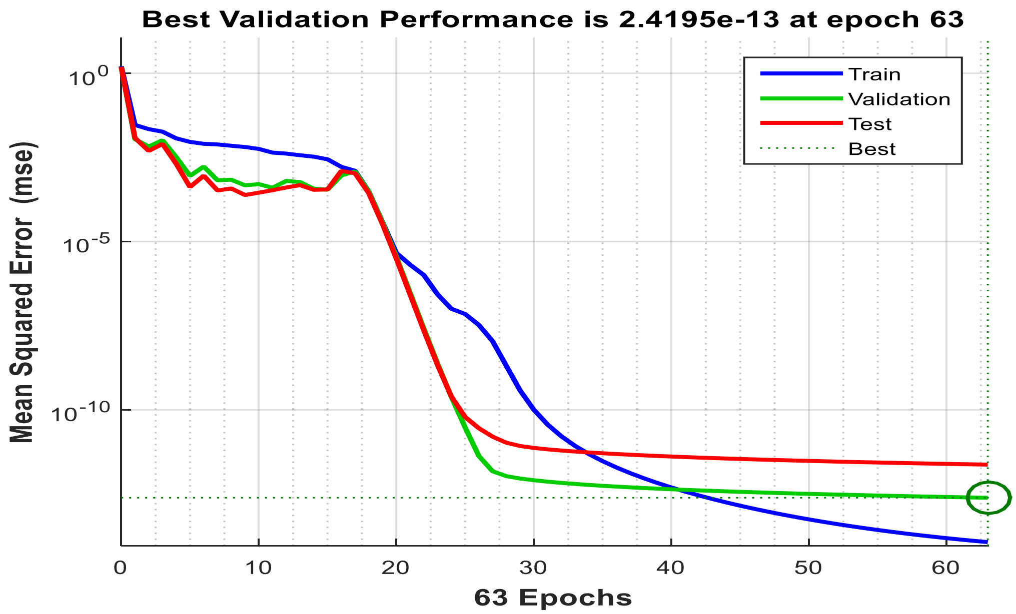

Figure 14: Graph of mean square error against 63 epochs

Figure 14 shows the performance curve obtained when the network was trained with two hidden layers and each having ten neurons. It was observed that the best validation performance was obtained when the epochs were sixty three (63) and the mean square error gotten from the simulation of equation 10 was $2.4195 * 10^{-13}$. The mean square error is again very small and the network converged at sixty three (63) epochs which again indicates a very good performance.

It was generally observed that as the number of the hidden layers increases with the number of neurons in each hidden layer remain the same (seven neurons), the time taken to successfully train the network increases but almost 100% classification was achieved in all cases. It was also observed that the network behaved badly and could not generalize well when the number of neurons in the two hidden layers differs. The performance, accuracy and precision were very poor in all the cases where the two hidden layers have different number of neurons.

4.0. Conclusion

The modelling and simulation of artificial neural network with multilayer layer perceptron based supervised architecture was carried out with spectra data under different number of hidden layers and different number of neurons in hidden layers and the results were analysed using Matlab. The result of the research work showed that the number of hidden layers in addition to the number of neurons in the hidden layers played an important role on the successful training of the network which results in the overall performance and accuracy of the network. The MLP performed well when the number of neurons in hidden layers is the same and an average of 100% classification result was achieved. An average of 18.3% classification result was achieved when the hidden layers have different number of neurons. It is therefore important to use the same number of neurons in hidden layers during the training of MLP network to avoid huge classification errors.

5.0 Recommendation

The laser fluorosensor spectra data used in this research work is limited to six substances which are light oil, medium crude, heavy oil, water, sand and stones. The modelling and simulation was carried out using artificial neural network (ANN). It is therefore recommended that other softwares should be used in modelling and classification of oil in order to compare the results obtained with that of ANN.

Acknowledgements

The support of National Oil Spill Detection and Response Agency in Nigeria (NOSDRA) is gratefully acknowledged for providing Laser fluorosensor spectra data of different oil spill products used for this research work.

References

Aharkava, L. 2010. Artificial Neural Networks and Self-Organization for Knowledge Extraction. *Master Thesis*, Charles University, Prague, 108.

Akkartal, A. and Sunar, F. 2008. The Usage of Radar Images in Oil Spill Detection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37: 271-276.

Brown, C., Fingas M., J. Monchalin, C. Neron and C. Padioleau, 2012. Airborne measurement of oil slick thickness. *Proceedings of the 29th Arctic and Marine Oil spill Program*.

Cybenko G. 2011. Approximation by superposition of sigmoid function; mathematics of control signals and system, 2(4), 303-314.

Bishop C. M. 1995. Neural networks for pattern recognition: *Oxford University Press*, New York, USA, 482.

Gil, D. and Johnson, M. 2010. Supervised SOM Based Architecture versus Multilayer Perceptron and RBF Networks. *Linkoping Electronic Conference Proceedings*, 48(5): 15-24.

Maya N., Jason L. and Yang G. 2014. Advances in remote sensing for oil disaster management. *State of the Art Sensors Technology for Oil Spill Surveillance:* Department of Geomatics Engineering, University of Calgary T2N 1N4, Alberta, Canada, Western Washington University, Disaster Reduction and Emergency Planning, Huxley College of the Environment, 516 High Street, Bellingham, Washington 98225-9085, USA, 8(1):236-255

Nickolas K. 2018. Review of oil spill simulation. 13th International conference on deregulated electricity market issues in South Eastern Europe. www.researchgate.net/publication/327974748

Panagiota K., Katerina S., George Z., Georgios G., and Georgios S. 2021. Oil Spill Modeling: A Critical Review on Current Trends, Perspectives, and Challenges. Journal of marine Science and Engineering. https://doi.org/10.3390/jmse9020181

Rosenblatt F. 2009. Principles of neurodynamics: Perceptrons and the theory of brain mechanism. *The American Journal of Psychology*, Washington D.C., 76(4):705-707

Singh, R. K. and Prajneshu. 2008. Artificial Neural Network Methodology for Forecasting Maize Crop Yield. *Agricultural Economics Research Review*, 21: 5-10.

Vahabi, N and Selviah, D.R. 2019. Convolutional Neural Networks to Classify Oil, Water and Gas Wells Fluid Using Acoustic Signals. Institute of Electrical/Electronics Engineers (IEEE).