

Application of Artificial Neural Networks Based Monte Carlo Simulation in the Expert System Design and Control of Crude Oil Distillation Column of a Nigerian Refinery

Lekan T. Popoola¹, Alfred A. Susu²

¹Department of Chemical and Petroleum Engineering, Afe Babalola University, Ado-Ekiti, Nigeria

²Department of Chemical Engineering, University of Lagos, Akoka, Nigeria

Email: popoolalekantaofeek@yahoo.com, alfredasusu22@hotmail.com

Received 18 February 2014; revised 15 March 2014; accepted 22 March 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This research work investigated comparative studies of expert system design and control of crude oil distillation column (CODC) using artificial neural networks based Monte Carlo (ANNBMC) simulation of random processes and artificial neural networks (ANN) model which were validated using experimental data obtained from functioning crude oil distillation column of Port-Harcourt Refinery, Nigeria by MATLAB computer program. Ninety percent (90%) of the experimental data sets were used for training while ten percent (10%) were used for testing the networks. The maximum relative errors between the experimental and calculated data obtained from the output variables of the neural network for CODC design were 1.98 error % and 0.57 error % when ANN only and ANNBMC were used respectively while their respective values for the maximum relative error were 0.346 error % and 0.124 error % when they were used for the controller prediction. Larger number of iteration steps of below 2500 and 5000 were required to achieve convergence of less than 10^{-7} for the training error using ANNBMC for both the design of the CODC and controller respectively while less than 400 and 700 iteration steps were needed to achieve convergence of 10^{-4} using ANN only. The linear regression analysis performed revealed the minimum and maximum prediction accuracies to be 80.65% and 98.79%; and 98.38% and 99.98% when ANN and ANNBMC were used for the CODC design respectively. Also, the minimum and maximum prediction accuracies were 92.83% and 99.34%; and 98.89% and 99.71% when ANN and ANNBMC were used for the CODC controller respectively as both methodologies have excellent predictions. Hence, artificial neural networks based Monte Carlo simulation is an effective and better tool for the design and control of crude oil distillation column.

How to cite this paper: Popoola, L.T. and Susu, A.A. (2014) Application of Artificial Neural Networks Based Monte Carlo Simulation in the Expert System Design and Control of Crude Oil Distillation Column of a Nigerian Refinery. *Advances in Chemical Engineering and Science*, 4, 266-283. <http://dx.doi.org/10.4236/aces.2014.42030>

Keywords

Neuron, Monte Carlo Simulation, Crude Oil Distillation Column, Artificial Neural Networks, Architecture, Refinery, Design, Control

1. Introduction

Neural networks were inspired by the power, flexibility and robustness of the biological brain. They are computational analogs of the basic biological components of a brain—neurons, synapses and dendrites. Artificial neural networks (ANN) consist of many simple computational elements (summing units—*neurons*—and weighted connections—*weights*) that work together in parallel and in series [1]. An ANN has the ability to learn relationships between given sets of input and output data by changing the weights. This process is called training the ANN [2]. The most well known training algorithm is the Back Propagation (BP) algorithm ([3] [4]). It minimizes the total sum of square error, which is the difference between the desired and actual output, using the gradient descent method. One of the most important properties of a trained ANN is its ability to generalize, which means that ANN can generate a satisfactory set of outputs from inputs that are not used during its training process [5]. The performance of the ANN model is a function of several design parameters such as the number of hidden layers, the number of hidden neurons in each hidden layer, the size of the training set and the training parameters. Theoretical work in ANN has shown that a single hidden layer is sufficient to approximate any complex nonlinear function under quite general conditions. While too many hidden neurons can hinder the ANN's ability to generalize data not seen during training by causing over-fitting, too few hidden neurons can cripple its ability to learn the mapping at hand [6].

An important stochastic and probabilistic tool that can be used in simulating the artificial neural networks model is the Monte Carlo simulation. Monte Carlo means using random numbers in scientific computing. More precisely, it means using random numbers as a tool to compute something that is not random. Monte Carlo analysis is a computer-based method of analysis developed in the 1940's that uses statistical sampling techniques in obtaining a probabilistic approximation to the solution of a mathematical equation or model [7]. Numerical simulations of stochastic processes have become an important task in many engineering fields. Monte Carlo approaches are particularly suitable tools for those simulation purposes. Their usefulness in diverse engineering applications and other fields has been well established over a period of decades [8].

Zhang *et al.* [9] studied the microscopic structures of the binary mixture of methanol-hexane under different conditions by the Monte Carlo (MC) method. Alexander *et al.* [10] presented a spatial model using Monte Carlo simulation for the mean and correlation of highly dispersed count down and applied it to individual-level counts of the nematode *Wuchereria bancrofti*, a parasite of humans which causes the disease lymphatic filariasis. Gilks *et al.* [11] rejuvenated particles based on Markov chain Monte Carlo. Khu *et al.* [12] investigated the reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modelling. Kuo *et al.* [13] used the data from Monte Carlo simulation to verify the proposed method integration of ART2 neural networks and genetic K-means algorithm for analyzing web browsing paths in electronic commerce. Yuxiang *et al.* [14] investigated the diffusion behavior of methanol in different critical media (*n*-pentane, *n*-hexane, *n*-heptane and acetone) using Monte Carlo (MC) method. Yeh *et al.* [15] proposed a methodology based on Monte Carlo simulation and ANN to estimate the reliability of a threshold voting system, which is a generalization of k-out-of-n systems. Sugiyama [16] reviewed three software packages for Monte Carlo simulation/risk analysis on a spreadsheet. A Monte Carlo particle model associated with neural networks for tracking problem had been examined [17]. A general regression neural network and Monte Carlo simulation model for survival and growth of Salmonella on raw chicken skin as a function of serotype, temperature and time for use in risk assessment had been developed [18]. Liu [19] investigated volumetric estimation of existing accumulated oil and gas in reservoir using Monte Carlo simulation. Safdari *et al.* [20] used artificial neural networks and Monte Carlo simulation in terms of uncertainty for the prediction of budget deficit in Iran.

Though previous research works had applied artificial neural networks in oil refineries ([21]-[25]), none had applied artificial neural networks based Monte Carlo simulation in the expert system design and control of crude oil distillation column of a refinery as far as literature review is concerned. This makes this research work to be

first of its kind to apply the method in oil refinery. An expert system is a computer system employing expert knowledge to attain high levels of performance in solving the problems within a specific domain area [26]. Today, expert systems have demonstrated their potential, gained credibility and are being widely used to solve a variety of problems in industry and government [27]. One of these complex problems for the control of which an expert system is amenable, is a crude oil distillation column [25]. The crude separation process involves many complex phenomena which have to be controlled in its best placement [23]. Controlling distillation column starts by identifying controlled, manipulated and load variables. Controlled variables are those variables that must be maintained at a precise value to satisfy column objectives [24]. The stages involved in transforming crude oil into finished products using crude oil distillation column in a petroleum refinery had been discussed ([19] [25] [28] [29]). **Figure 1** shows a typical configuration of a crude oil distillation system.

Where ADU and VDU are atmospheric and vacuum distillation columns; TPA, MPA and BPA are the top, middle and bottom pump-arounds of ADU; LGO and HGO are light gas oil and heavy gas oil for the atmospheric distillation column; VLGO and VHGO are light gas oil and heavy gas oil for the vacuum distillation column.

The objective of this research work is to develop MATLAB computer program for the algorithms of the artificial neural networks based Monte Carlo simulation and test it on existing running data of a crude oil distillation column in a Refinery. Also to compare the computed results with the real data of the running crude oil distillation column of the Refinery using only artificial neural networks (ANN) and artificial neural networks based Monte Carlo (ANNBMC) simulation separately. The results obtained when only artificial neural networks model was used for expert system design and control of the examined crude oil distillation column of the refinery had been presented elsewhere [25].

2. Scope

The network training is accomplished based on the standard method of error back-propagation. The search for the optimum adjustment of the weights and biases is realized with the aid of a gradient descends method operating with a generalized delta rule. Monte Carlo simulation is used as a stochastic and probabilistic tool in generating random numbers used in adjusting the weights in the neural network.

3. Modelling Equations

3.1. Network Configuration and Data Conditioning

The output signal y_k generated by neuron k of the ANN is given as [30]:

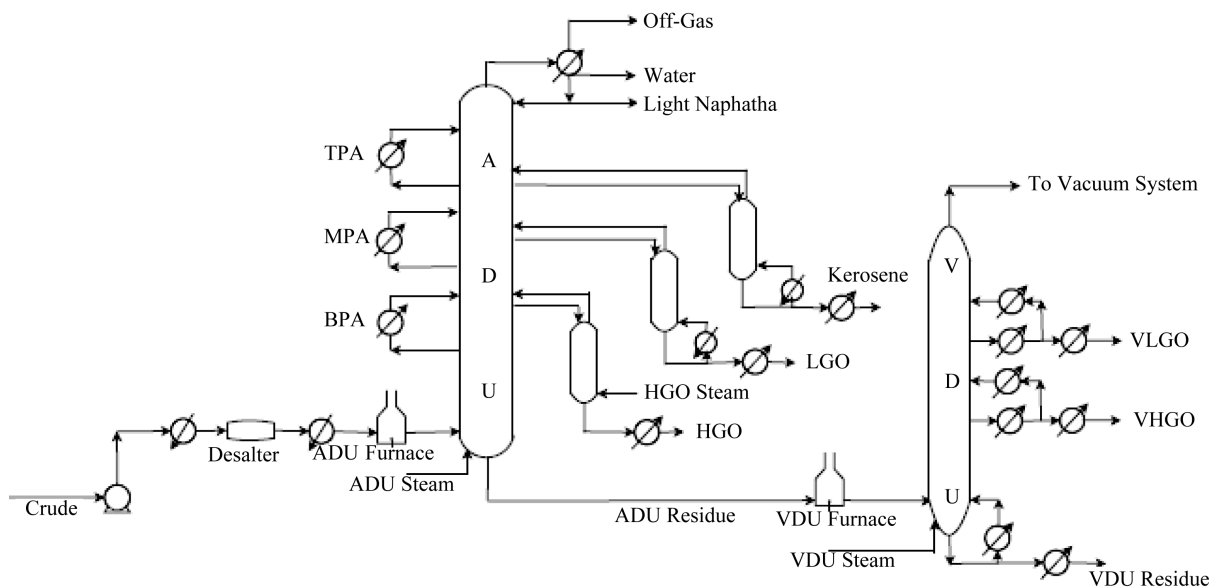


Figure 1. Schematic diagram of crude oil distillation systems [19].

$$y_k = \varphi \left(\sum_{j=1}^m w_{kj} \cdot x_j + b_k \right) \quad (1)$$

where y_k = Output signal; φ = Activation function; m = Total number of inputs to the neuron; j = Input; w_{kj} = Synaptic weight of input j for neuron k ; x_j = Input Signal; b_k = Bias value of neuron k .

The activation function (φ) is required to be nonlinear and monotonically increasing from zero to unity. The logistic sigmoid function is given as [31]:

$$\varphi(x) = (1 + \exp(-x))^{-1} \quad (2)$$

where $\varphi(x)$ = Sigmoidal function of input x .

The derivative of the logistic sigmoidal function which must be computed millions of times during the network training is [31]:

$$\varphi'(x) = \varphi(x) \cdot (1 - \varphi(x)) \quad (3)$$

where $\varphi'(x)$ = Derivative of the sigmoidal function of input x .

The signal flow through the network is summarized as [32]:

$$y = \varphi \left(\sum_{j^{(3)}=1}^{j^{(3)}} \left[w_{j^{(4)}j^{(3)}} \cdot \varphi \left(v_{j^{(3)}} \right) \right] + b_{j^{(4)}} \right) \quad (4)$$

$$v_{j^{(3)}} = \sum_{j^{(2)}=1}^{j^{(2)}} \left[w_{j^{(3)}j^{(2)}} \cdot \varphi \left(\sum_{j^{(1)}=1}^{j^{(1)}=m+1} \left[w_{j^{(2)}j^{(1)}} \cdot x_j \right] + b_{j^{(2)}} \right) \right] + b_{j^{(3)}} \quad (5)$$

where $J^{(L)}$ denotes the number of neurons $j^{(L)}$ in layer L , $w_{j^{(L)}j^{(L-1)}}$ represents the weight for the signal from neuron $j^{(L-1)}$ in layer $L-1$ to neuron $j^{(L)}$ in layer L , and $b_{j^{(L)}}$ is the bias of neuron $j^{(L)}$ in layer L .

The original raw data values z_j are transformed into the input signals x_j using the equation

$$x_j = (z_j - \bar{z}) \cdot \sigma_z^{-1} \quad (6)$$

where \bar{z} denotes the mean value of the raw data and σ_z^{-1} is the inverse of the standard deviation of the raw data.

The generated process values (p) are determined using the equation

$$p = \Phi^{SN^{-1}}(y) \cdot \sigma_z + \bar{z} \quad (7)$$

where $\Phi^{SN^{-1}}(y)$ is the inverse function of the standard normal distribution of the network output y , σ_z and \bar{z} are the standard deviation and mean values of the raw data z .

3.2. Artificial Neural Networks Back-Propagation Algorithm

The network training is accomplished based on the standard method of error back-propagation [33]. The free values of the neural network (weights and biases) are adjusted so that the network is capable of reproducing the training data with a sufficient precision. Thus, for each sequence of process values $z(t_{n-r}), \dots, z(t_{n-1})$ the network is intended to generate a prognosis P_n for the subsequent process value $z(t_n)$ with a minimum prediction error given as [2]:

$$e = z(t_n) - p_n \quad (8)$$

and error energy given as [34]:

$$E = \frac{1}{2} \cdot e^2 \quad (9)$$

Specifically,

$$\sum_{n=r+1}^N (z(t_n) - p_n)^2 \Rightarrow MIN \quad (10)$$

in which $N = \text{Length of the observed process record (training data)}$; $r = \text{Order of the tapped delay line memory}$.

The search for the optimum adjustment of the weights and biases is realized with the aid of a gradient descend method operating with a generalized delta rule. For each predicted process value the prediction error of the neural network is retraced through the complete network (back-propagation) to compute changes of the weights and biases. This is done iteratively, as described subsequently, until the prediction error approaches the global minimum.

One sequence of $r + 1$ successive process values $z(t_{n-r}), \dots, z(t_{n-1}), z(t_n)$ is randomly selected from the training data with the aid of a discrete uniform distribution over the $N - r$ possible choices. Then, the error signal $e(q)$ in the current iteration step q is determined with Equation (9) and is used to compute the local gradients $\partial E / \partial w_{kj}$ in the weight space, proportional to which the weights and biases are to be changed. Let $v_{j^{(L)}}(q)$ be the argument of the activation function $\varphi_{j^{(L)}}$ in neuron $j^{(L)}$ of layer L and $y_{j^{(L-1)}}(q)$ denotes the neuron output of neuron $j^{(L-1)}$ in the previous layer $L - 1$. The new weights for the next iteration step are stated as [30]:

$$w_{j^{(L)}j^{(L-1)}}(q+1) = w_{j^{(L)}j^{(L-1)}}(q) + \alpha \cdot \left[w_{j^{(L)}j^{(L-1)}}(q-1) \right] + \eta \cdot \delta_{j^{(L)}}(q) \cdot y_{j^{(L-1)}}(q) \quad (11)$$

where

$$\delta_{j^{(L)}}(q) = e(q) \cdot \varphi'_{j^{(L)}}\left(v_{j^{(L)}}(q)\right) \quad (12)$$

for the neuron $j^{(L)} = 1$ in the output layer, and

$$\delta_{j^{(L)}}(q) = \varphi'_{j^{(L)}}\left(v_{j^{(L)}}(q)\right) \cdot \sum_{j^{(L+1)}=1}^{j^{(L+1)}} \delta_{j^{(L+1)}}(q) \cdot w_{j^{(L+1)}j^{(L)}}(q) \quad (13)$$

for the neurons $j^{(L)}$ in layer L . Therefore, the biases $b_{j^{(L)}}$ are treated as weights $w_{j^{(L)}j^{(L-1)}}$ for constant input signals $y_{j^{(L-1)}} = +1$ from the previous layer $L - 1$. The parameters α and η are introduced to control the numerical behaviour of the iteration. Whereas the learning rate $\eta > 0$ determines the degree with which the actual error gradients effect the weight change, the momentum factor $\alpha \in [0, 1)$ acts as a delay parameter in the weight adjustment.

When the weight adjustment in iteration step q is completed, the next sequence of $r + 1$ successive process values $z(t_{n-r}), \dots, z(t_{n-1}), z(t_n)$ is randomly selected to proceed with the weight adjustment in iteration step $q + 1$. This procedure of iteratively adjusting the weights and biases is referred to as sequential training mode, which possesses the advantage of being stochastic in nature. This induces a good performance in the search for the global minimum of the objective function.

3.3. Monte Carlo Simulation

A set of weighted particles (samples), drawn from the posterior distribution of the model parameters, is used to map integrals to discrete sums. More precisely, the posterior can be approximated by the following empirical estimate [35]:

$$\hat{p}(\theta_{0:t} | y_{0:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{\theta_{0:t}^{(i)}}(d\theta_{0:t}) \quad (14)$$

where the random samples $\{\theta_{0:t}^{(i)}; i = 1, \dots, N\}$, are drawn from the posterior distribution and $\delta(d)$ denotes the Dirac delta function. Consequently, any expectations of the form [36]:

$$E(g_t(\theta_{0:t})) = \int g_t(\theta_{0:t}) p(\theta_{0:t} | y_{1:t}) d\theta_{0:t} \quad (15)$$

may be approximated by the following estimate [37]:

$$\overline{E(g_t(\theta_{0:t}))} = \frac{1}{N} \sum_{i=1}^N g_t(\theta_{0:t}^{(i)}) \quad (16)$$

where the particles $\theta_{0:t}^{(i)}$ are assumed to be independent and identically distributed for the approximation to hold.

According to the law of large numbers [34],

$$\overline{E(g_t(\theta_{0:t}))} \xrightarrow[N \rightarrow \infty]{a.s.} E(g_t(\theta_{0:t})) \quad (17)$$

where $\xrightarrow[N \rightarrow \infty]{a.s.}$ denotes almost sure convergence. Moreover, if the posterior variance of $g_t(\theta_{0:t})$ is bounded, $\text{var}_{p(\cdot|d_{1:t})}(g_t(\theta_{0:t})) < \infty$, then the following central limit theorem holds [38]:

$$\sqrt{N} \left(\overline{E(g_t(\theta_{0:t}))} - E(g_t(\theta_{0:t})) \right) \xrightarrow[N \rightarrow \infty]{} N \left(0, \text{var}_{p(\cdot|d_{1:t})}(g_t(\theta_{0:t})) \right) \quad (18)$$

where $\xrightarrow[N \rightarrow \infty]{} \rightarrow$ denotes the convergence in distribution.

3.4. Algorithm and Architecture for the Design and Control of Crude Oil Distillation Column Using Artificial Neural Networks Based Monte Carlo Simulation

The algorithm for design and control of crude oil distillation column using artificial neural networks based Monte Carlo Simulation stated in sub-sections 3.0.1, 3.0.2 and 3.0.3 above is shown in **Figure 2**.

The inputs to the column are crude oil and steam flow while the outputs are Naphthalene, Kerosene, Light Diesel Oil and Heavy Diesel Oil. Some quantity of Naphthalene, Kerosene and Light Diesel Oil (reflux flows) are returned into the column while stripping (distillate flows) is sent to the storage tank.

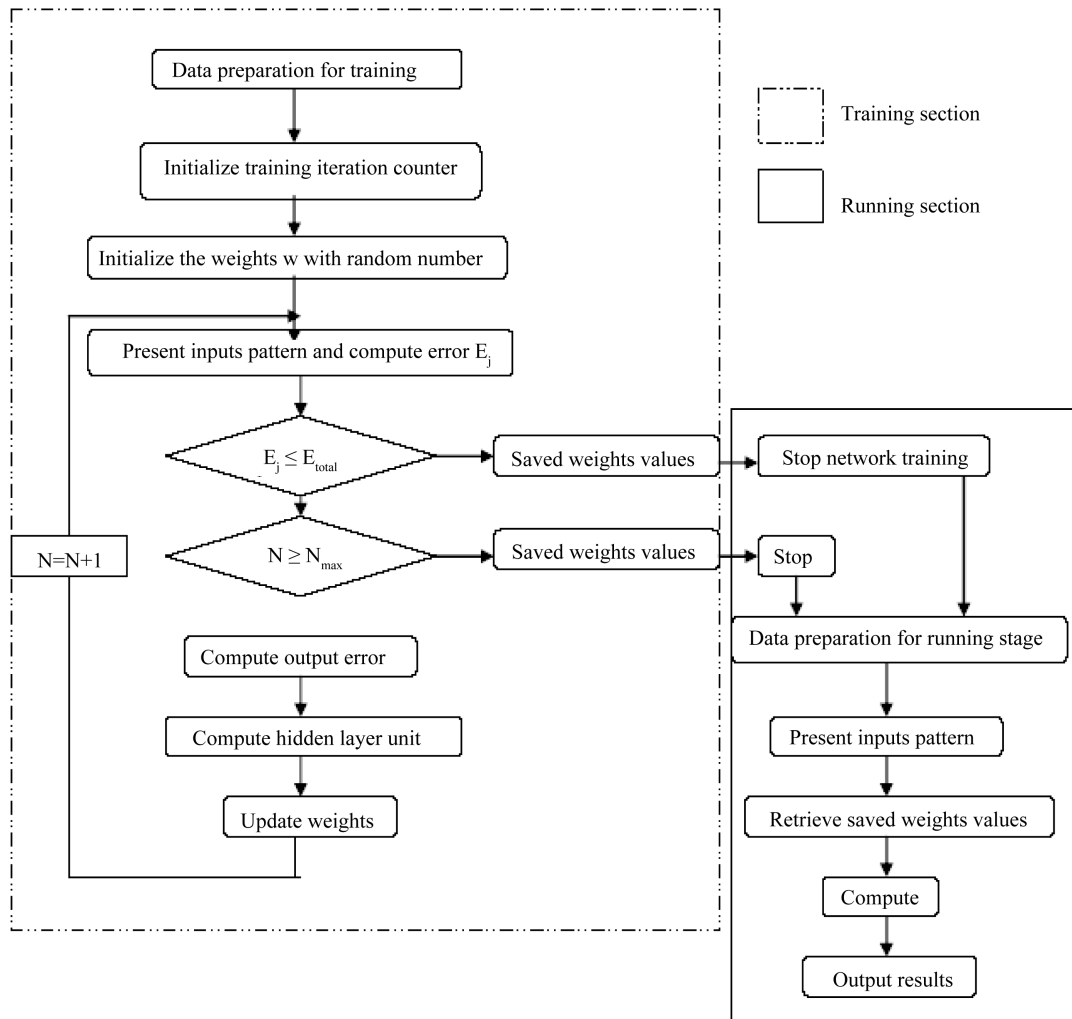


Figure 2. Algorithm for design and control of crude oil distillation column using artificial neural networks based Monte Carlo simulation [2].

The neural networks based Monte Carlo simulation architecture for the design of the crude oil distillation column (CODC) is fourteen inputs with one hidden layer (nine nodes) and seven outputs (14-1-7) making a total of 30 nodes distributed over the three layers. The inputs to the network for the design are feed temperature of crude oil, kerosene flow ratio, AGO flow ratio, diesel flow ratio, crude oil flow rate, API gravity of crude oil, sulphur content of crude oil, compositions of C₂, C₃, i-C₄, n-C₄, i-C₅, n-C₅ and Cyclo-pentane in crude oil represented respectively as I₁, I₂, I₃, I₄, I₅, I₆, I₇, I₈, I₉, I₁₀, I₁₁, I₁₂, I₁₃ and I₁₄. The outputs from the NN architecture are temperatures at which 100% (T₁₀₀) of Kerosene, 90% (T₉₀) of Diesel and 10% (T₁₀) of AGO are distilled; and naphtha, kerosene, diesel and AGO flow rates represented respectively as O₁, O₂, O₃, O₄, O₅, O₆ and O₇. For the neural networks based Monte Carlo controller, the inputs to the networks are feed flow rate, feed temperature, top temperature, bottom temperature, reflux temperatures 1, 2 and 3; bottom flow, distillate flow 1 (Naphthalene), distillate flow 2 (kerosene), distillate flow 3 (Light Diesel Oil), distillate flow 4 (Heavy Diesel Oil) and top pressure represented as i₁, i₂, i₃, i₄, i₅, i₆, i₇, i₈, i₉, i₁₀, i₁₁, i₁₂ and i₁₃ respectively. The outputs are stripping steam to main column, LDO stripper, HDO stripper, reflux flow 1 (Top Pump around), reflux flow 2 (Kerosene Pump around) and reflux flow 3 (Light Diesel Oil Pump around) represented as o₁, o₂, o₃, o₄, o₅ and o₆ respectively. The architecture for the neural networks based Monte Carlo controller becomes thirteen (13) inputs with one hidden layer (nine nodes) and six (6) outputs (13-1-6) with a total of 28 nodes distributed over the layers. **Figure 3** and **Figure 4** show the neural network architecture for the design and control of CODC respectively.

4. Results

In this research work, artificial neural networks (ANN) model and artificial neural networks based Monte Carlo

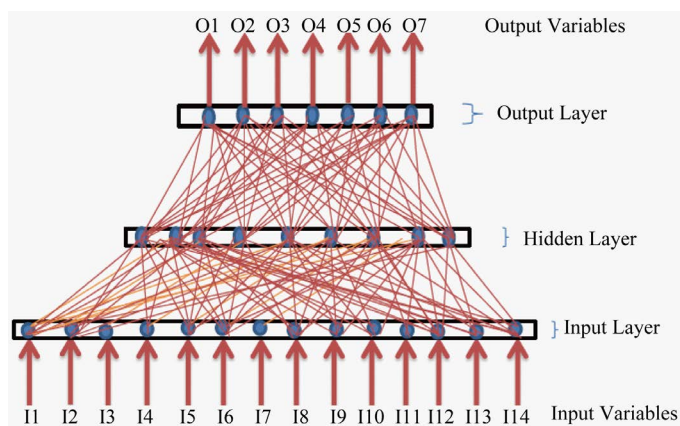


Figure 3. Architecture for the design of crude oil distillation column [25].

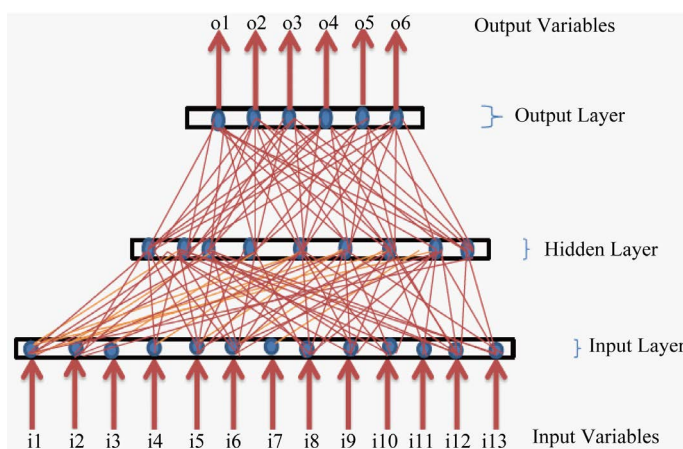


Figure 4. Architecture for crude oil distillation column controller [25].

(ANNBMC) simulation were developed separately for both the design and controller of the crude oil distillation column (CODC) to check for the accuracies and differences in their outputs from the network architectures used. They were both validated using experimental data obtained from functioning crude oil distillation column of Port-Harcourt Refinery, Nigeria. Out of the one-hundred and thirty (130) experimental data sets obtained, ninety percent (90%) were used for training the network while the remaining ten percent (10%) were used for testing the network to determine its prediction accuracy. MATLAB program was written for the neural networks model and artificial neural networks based Monte Carlo simulation. **Table 1(a)**, **Table 1(b)** and **Table 1(c)** show the test comparison results obtained between the values from the trained ANN and ANNBM networks for the design of CODC while **Table 2(a)** and **Table 2(b)** show the results obtained for the controller of the CODC compared with the experimental data of the CODC obtained from the Port-Harcourt Refinery, Nigeria. **Figure 5** and **Figure 6** represent the plots of the training error against the iteration number when ANN models and ANNBM

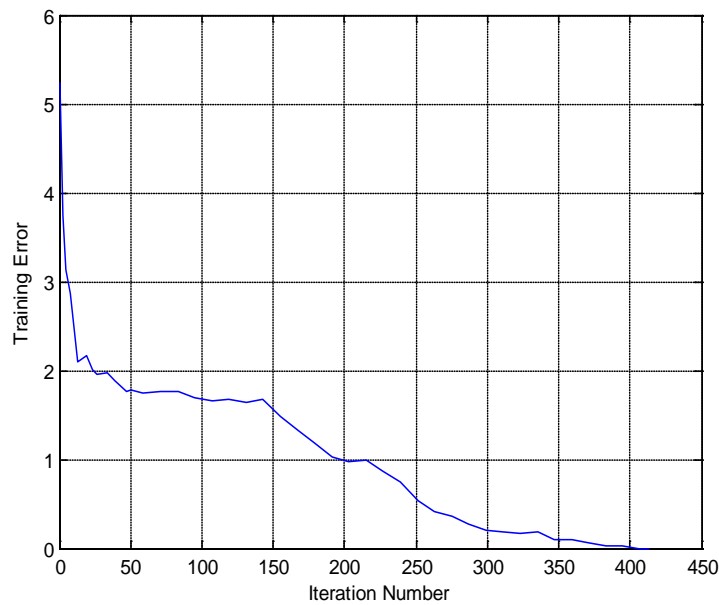


Figure 5. Training error vs iteration number for CODC design using ANN with 10 hidden neurons.

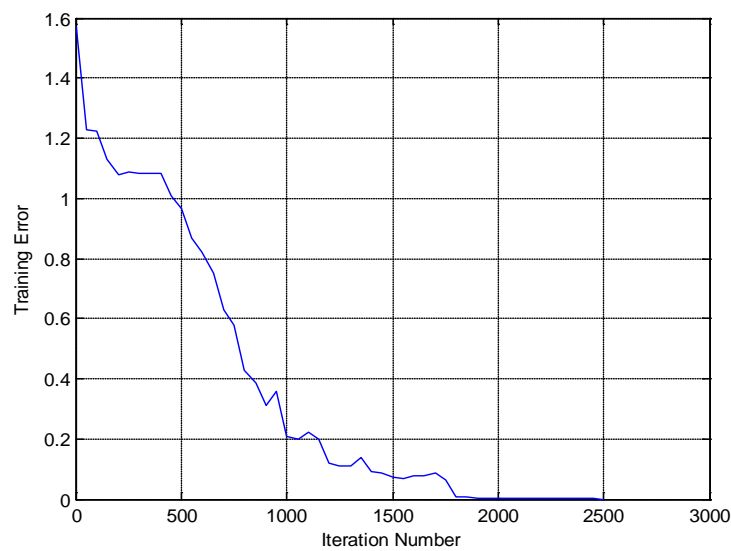


Figure 6. Training error vs iteration number for CODC design using ANNBM with 10 hidden neurons.

Table 1. (a) Test results obtained using ANN and ANNBMCM for crude oil distillation column design compared with experimental data from the refinery; (b) Test results obtained using ANN and ANNBMCM for crude oil distillation column design compared with experimental data from the refinery; (c) Test results obtained using ANN and ANNBMCM for crude oil distillation column design compared with experimental data from the refinery.

(a)

| Test No | T ₁₀ of AGO (°C) | | | | | T ₉₀ of Diesel (°C) | | | | | T ₁₀₀ of Kerosene (°C) | | | | |
|---------|-----------------------------|------------------|----------------------|-------------------|----------------------|--------------------------------|------------------|----------------------|-------------------|----------------------|-----------------------------------|------------------|----------------------|-------------------|----------------------|
| | Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) | Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) | Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) |
| 1 | 261 | 263 | 0.76 | 262.8 | 0.076 | 379 | 374 | 1.34 | 373.3 | 0.187 | 255 | 253 | 0.79 | 253.0 | 0 |
| 2 | 257 | 256 | 0.39 | 256.0 | 0 | 362 | 365 | 0.82 | 364.8 | 0.055 | 267 | 263 | 1.52 | 264.5 | 0.570 |
| 3 | 261 | 259 | 0.77 | 259.2 | 0.077 | 258 | 253 | 1.98 | 252.2 | 0.316 | 261 | 263 | 0.76 | 263.2 | 0.076 |
| 4 | 265 | 267 | 0.75 | 266.4 | 0.225 | 360 | 367 | 1.91 | 367.6 | 0.164 | 258 | 256 | 0.78 | 255.6 | 0.156 |
| 5 | 263 | 264 | 0.38 | 263.3 | 0.265 | 353 | 354 | 0.28 | 354.0 | 0 | 266 | 269 | 1.12 | 269.0 | 0 |
| 6 | 259 | 259 | 0.00 | 259.1 | 0.039 | 358 | 358 | 0.00 | 358.1 | 0.028 | 258 | 259 | 0.39 | 258.9 | 0.039 |
| 7 | 261 | 261 | 0.00 | 260.7 | 0.115 | 371 | 369 | 0.54 | 369.5 | 0.136 | 278 | 278 | 0.00 | 278.0 | 0 |
| 8 | 268 | 269 | 0.37 | 269.0 | 0 | 371 | 371 | 0.00 | 371.0 | 0 | 272 | 271 | 0.37 | 271.3 | 0.111 |
| 9 | 270 | 270 | 0.00 | 270.0 | 0 | 368 | 365 | 0.82 | 364.9 | 0.027 | 247 | 243 | 1.65 | 243.7 | 0.288 |
| 10 | 266 | 268 | 0.75 | 268.6 | 0.224 | 355 | 354 | 0.28 | 354.0 | 0 | 268 | 266 | 0.75 | 266.0 | 0 |
| 11 | 256 | 258 | 0.78 | 257.8 | 0.078 | 381 | 378 | 0.79 | 378.0 | 0 | 253 | 249 | 1.61 | 248.5 | 0.201 |
| 12 | 267 | 266 | 0.38 | 265.9 | 0.038 | 378 | 376 | 0.53 | 377.0 | 0.266 | 254 | 254 | 0.00 | 254.0 | 0 |
| 13 | 272 | 272 | 0.00 | 272.1 | 0.037 | 372 | 368 | 1.09 | 368.0 | 0 | 273 | 275 | 0.73 | 273.9 | 0.40 |

(b)

| Test No | Naphtha Flow Rate (m ³ /hr) | | | | | Kerosene Flow Rate (m ³ /hr) | | | | | Diesel Flow Rate (m ³ /hr) | | | | |
|---------|--|------------------|----------------------|-------------------|----------------------|---|------------------|----------------------|-------------------|----------------------|---------------------------------------|------------------|----------------------|-------------------|----------------------|
| | Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) | Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) | Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) |
| 1 | 205.2 | 205.6 | 0.20 | 205.5 | 0.049 | 142.8 | 143.9 | 0.76 | 143.8 | 0.070 | 42.7 | 42.9 | 0.47 | 42.9 | 0 |
| 2 | 210.0 | 209.8 | 0.10 | 209.5 | 0.143 | 139.3 | 138.9 | 0.29 | 138.9 | 0 | 42.0 | 41.8 | 0.48 | 41.9 | 0.239 |
| 3 | 201.9 | 201.3 | 0.30 | 201.7 | 0.199 | 142.9 | 142.3 | 0.42 | 142.1 | 0.141 | 45.5 | 45.6 | 0.22 | 45.5 | 0.219 |
| 4 | 203.1 | 203.5 | 0.20 | 203.0 | 0.246 | 138.4 | 139.8 | 1.00 | 139.9 | 0.072 | 42.6 | 42.3 | 0.71 | 42.2 | 0.236 |
| 5 | 207.1 | 206.8 | 0.15 | 207.0 | 0.097 | 143.8 | 142.4 | 0.98 | 142.1 | 0.211 | 44.3 | 44.3 | 0.00 | 44.3 | 0 |
| 6 | 203.5 | 203.8 | 0.15 | 203.7 | 0.049 | 141.1 | 141.9 | 0.56 | 142.0 | 0.071 | 41.9 | 41.8 | 0.24 | 41.8 | 0 |
| 7 | 201.5 | 201.3 | 0.10 | 201.3 | 0 | 142.8 | 142.9 | 0.07 | 143.0 | 0.070 | 43.1 | 42.8 | 0.70 | 42.9 | 0.234 |
| 8 | 204.3 | 204.6 | 0.15 | 204.6 | 0 | 138.7 | 138.7 | 0.00 | 139.1 | 0.288 | 41.6 | 41.9 | 0.72 | 41.8 | 0.239 |
| 9 | 202.0 | 201.7 | 0.15 | 201.7 | 0 | 142.9 | 143.3 | 0.28 | 143.1 | 0.140 | 43.8 | 43.9 | 0.23 | 43.9 | 0 |
| 10 | 203.8 | 203.8 | 0.00 | 203.9 | 0.049 | 143.9 | 145.6 | 1.17 | 145.8 | 0.137 | 42.2 | 42.8 | 1.40 | 42.7 | 0.234 |
| 11 | 204.8 | 203.6 | 0.59 | 203.6 | 0 | 145.1 | 144.9 | 0.14 | 144.2 | 0.483 | 44.6 | 44.6 | 0.00 | 44.4 | 0.448 |
| 12 | 202.9 | 202.9 | 0.00 | 202.8 | 0.049 | 144.1 | 144.2 | 0.07 | 144.2 | 0 | 42.5 | 42.3 | 0.47 | 42.3 | 0 |
| 13 | 205.7 | 204.3 | 0.69 | 203.9 | 0.196 | 143.7 | 142.5 | 0.84 | 142.4 | 0.070 | 43.5 | 43.8 | 0.69 | 43.8 | 0 |

(c)

| AGO Flow Rate (m ³ /hr) | | | | |
|------------------------------------|------------------|----------------------|-------------------|----------------------|
| Comp ^a | Exp ¹ | Err ^a (%) | Comp ^b | Err ^b (%) |
| 237.0 | 236.8 | 0.09 | 236.7 | 0.042 |
| 235.6 | 235.8 | 0.09 | 235.9 | 0.042 |
| 233.8 | 233.8 | 0.00 | 233.6 | 0.086 |
| 235.4 | 235.4 | 0.00 | 235.5 | 0.043 |
| 232.1 | 231.3 | 0.35 | 231.3 | 0 |
| 232.9 | 231.7 | 0.52 | 231.7 | 0 |
| 231.9 | 231.9 | 0.00 | 231.8 | 0.043 |
| 233.9 | 233.6 | 0.13 | 233.6 | 0 |
| 232.3 | 233.6 | 0.56 | 233.5 | 0.043 |
| 233.9 | 232.5 | 0.60 | 232.5 | 0 |
| 239.3 | 238.9 | 0.17 | 238.8 | 0.042 |
| 231.8 | 231.8 | 0.00 | 231.8 | 0 |
| 234.5 | 234.8 | 0.13 | 234.4 | 0.170 |

Hint: Comp^a = Computed Values for the CODC Design using Artificial Neural Networks (ANN); Comp^b = Computed Values for the CODC Design using Artificial Neural Networks Based Monte Carlo Simulation (ANNBMCM); Exp¹ = Experimental Values of the CODC Design; Err^a (%) = Error Percent between Exp¹ and Comp^a. Err^b (%) = Error Percent between Exp¹ and Comp^b.

Table 2. (a) Test results obtained using ANN and ANNBMC for crude oil distillation column controller compared with experimental data from the refinery; (b) Test results obtained using ANN and ANNBMC for crude oil distillation column controller compared with experimental data from the refinery.

(a)

| Test No | Stripping Steam to main column (kg/hr) | | | | | LDO Stripper (kg/hr) | | | | | HDO Stripper (kg/hr) | | | | |
|---------|--|------------------|----------------------|-------------------|----------------------|----------------------|------------------|----------------------|-------------------|----------------------|----------------------|------------------|----------------------|-------------------|----------------------|
| | Comp ^c | Exp ² | Err ^c (%) | Comp ^b | Err ^d (%) | Comp ^c | Exp ² | Err ^c (%) | Comp ^b | Err ^d (%) | Comp ^c | Exp ² | Err ^c (%) | Comp ^b | Err ^d (%) |
| 1 | 6631 | 6632 | 0.015 | 6632 | 0 | 5603 | 5603 | 0 | 5602.5 | 0.009 | 796.1 | 795.8 | 0.038 | 795.2 | 0.075 |
| 2 | 6631 | 6632 | 0.015 | 6632 | 0 | 5603 | 5603 | 0 | 5603 | 0 | 795.9 | 795.7 | 0.025 | 795.2 | 0.063 |
| 3 | 6630 | 6630 | 0 | 6630 | 0 | 5603 | 5604 | 0.018 | 5603.5 | 0.009 | 796.7 | 796.2 | 0.063 | 795.9 | 0.038 |
| 4 | 6631 | 6631 | 0 | 6631 | 0 | 5605 | 5605 | 0 | 5605.2 | 0.004 | 797.5 | 797.7 | 0.025 | 797.7 | 0 |
| 5 | 6636 | 6635 | 0.015 | 6635 | 0 | 5613 | 5614 | 0.018 | 5613.9 | 0.002 | 800.9 | 800.6 | 0.038 | 801.1 | 0.063 |
| 6 | 6637 | 6637 | 0 | 6637 | 0 | 5613 | 5614 | 0.018 | 5614 | 0 | 800.8 | 800.7 | 0.013 | 800.9 | 0.025 |
| 7 | 6641 | 6640 | 0.015 | 6640 | 0 | 5614 | 5614 | 0 | 5614 | 0 | 802.8 | 802.4 | 0.050 | 802.4 | 0 |
| 8 | 6641 | 6640 | 0.015 | 6640 | 0 | 5616 | 5616 | 0 | 5616 | 0 | 804.2 | 804.4 | 0.025 | 803.9 | 0.062 |
| 9 | 6641 | 6640 | 0.015 | 6639 | 0.015 | 5616 | 5617 | 0.018 | 5617.6 | 0.011 | 804.1 | 804.3 | 0.025 | 804.3 | 0 |
| 10 | 6645 | 6644 | 0.015 | 6644 | 0 | 5622 | 5621 | 0.018 | 5622 | 0.018 | 804.9 | 804.7 | 0.025 | 804.8 | 0.012 |
| 11 | 6649 | 6648 | 0.015 | 6647 | 0.015 | 5622 | 5622 | 0 | 5622 | 0 | 806.1 | 806.8 | 0.087 | 807.1 | 0.037 |
| 12 | 6650 | 6649 | 0.015 | 6648 | 0.015 | 5622 | 5622 | 0 | 5621.7 | 0.005 | 809.6 | 809.7 | 0.012 | 809.4 | 0.037 |
| 13 | 6650 | 6649 | 0.015 | 6649 | 0 | 5623 | 5624 | 0.018 | 5623.8 | 0.004 | 809.0 | 809.8 | 0.099 | 808.8 | 0.124 |

(b)

| Test No | Reflux Flow 1 (m ³ /hr) | | | | | Reflux Flow 2 (m ³ /hr) | | | | | Reflux Flow 3 (m ³ /hr) | | | | |
|---------|------------------------------------|------------------|----------------------|-------------------|----------------------|------------------------------------|------------------|----------------------|-------------------|----------------------|------------------------------------|------------------|----------------------|-------------------|----------------------|
| | Comp ^c | Exp ² | Err ^c (%) | Comp ^b | Err ^d (%) | Comp ^c | Exp ² | Err ^c (%) | Comp ^b | Err ^d (%) | Comp ^c | Exp ² | Err ^c (%) | Comp ^b | Err ^d (%) |
| 1 | 374.6 | 374.2 | 0.107 | 374.1 | 0.027 | 801.5 | 801.8 | 0.037 | 801.7 | 0.013 | 374.1 | 374.2 | 0.027 | 374.2 | 0 |
| 2 | 374.1 | 374.1 | 0 | 374.1 | 0 | 801.5 | 801.8 | 0.037 | 801.8 | 0 | 372.3 | 372.8 | 0.134 | 372.9 | 0.026 |
| 3 | 374.7 | 374.3 | 0.107 | 374.3 | 0 | 801.3 | 801.8 | 0.063 | 801.6 | 0.025 | 370.4 | 370.4 | 0 | 370.5 | 0.0270 |
| 4 | 375.1 | 375.6 | 0.133 | 375.5 | 0.027 | 802.1 | 802.3 | 0.025 | 802.5 | 0.024 | 370.1 | 370.4 | 0.081 | 370.4 | 0 |
| 5 | 376.2 | 376.4 | 0.053 | 376.3 | 0.027 | 802.7 | 802.6 | 0.013 | 802.5 | 0.013 | 370.6 | 370.4 | 0.054 | 370.3 | 0.027 |
| 6 | 376.4 | 376.2 | 0.053 | 376.2 | 0 | 803.9 | 803.7 | 0.024 | 803.4 | 0.037 | 370.4 | 370.1 | 0.081 | 369.9 | 0.054 |
| 7 | 376.9 | 376.6 | 0.080 | 376.4 | 0.053 | 803.7 | 803.7 | 0 | 803.7 | 0 | 374.0 | 373.6 | 0.107 | 373.9 | 0.080 |
| 8 | 377.9 | 377.9 | 0 | 377.9 | 0 | 803.2 | 803.7 | 0.062 | 803.9 | 0.025 | 373.5 | 373.8 | 0.080 | 373.8 | 0 |
| 9 | 376.8 | 376.8 | 0 | 376.5 | 0.079 | 805.3 | 805.1 | 0.025 | 805.0 | 0.012 | 376.9 | 376.8 | 0.027 | 376.4 | 0.106 |
| 10 | 376.6 | 376.7 | 0.027 | 376.7 | 0 | 805.4 | 805.5 | 0.012 | 805.5 | 0 | 376.6 | 376.8 | 0.053 | 376.9 | 0.027 |
| 11 | 376.9 | 376.5 | 0.106 | 376.5 | 0 | 805.7 | 805.5 | 0.025 | 805.5 | 0 | 376.7 | 376.9 | 0.053 | 376.8 | 0.027 |
| 12 | 376.6 | 376.3 | 0.079 | 376.2 | 0.026 | 807.6 | 807.0 | 0.074 | 807.2 | 0.025 | 376.7 | 376.9 | 0.053 | 376.9 | 0 |
| 13 | 376.4 | 376.8 | 0.106 | 376.9 | 0.027 | 807.6 | 806.9 | 0.087 | 807.2 | 0.038 | 376.9 | 375.6 | 0.346 | 375.2 | 0.107 |

Hint: Comp^c = Computed Values for the CODC Controller using Artificial Neural Networks (ANN); Comp^d = Computed Values for the CODC Controller using Artificial Neural Networks Based Monte Carlo Simulation (ANNBMC); Exp² = Experimental Values of the CODC Controller; Err^c (%) = Error Percent between Exp and Comp^c. Err^d (%) = Error Percent between Exp and Comp^d.

were used for the design of the CODC with neural network architectures of 10 hidden neurons respectively. **Figure 7** and **Figure 8** depict the plots of the training error against the iteration number when ANN models and ANNBMC were used for the CODC controller prediction with neural network architectures of 10 hidden neurons respectively. **Figures 9-15** represent the comparative test results for the prediction of T₁₀ of AGO, T₉₀ of Diesel, T₁₀₀ of Kerosene, Naphtha, Kerosene, Diesel and AGO flow rates respectively as part of CODC design using ANN and ANNBMC. **Figures 16-21** show the comparative test results for the prediction of stripping steam to main column, LDO stripper, HDO stripper, reflux flows 1, 2 and 3 respectively as part of the CODC

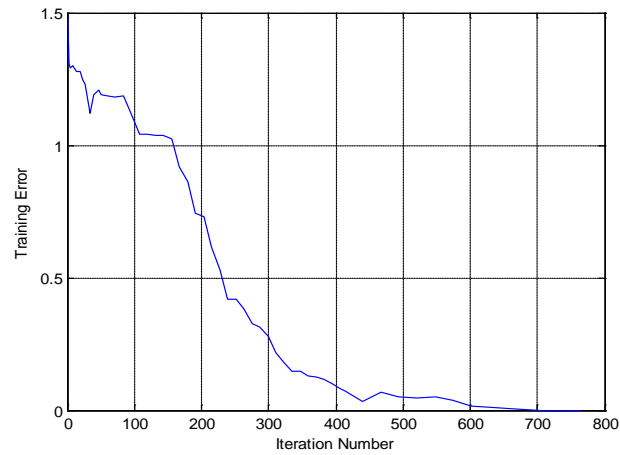


Figure 7. Training error vs iteration number for CODC controller using ANN with 10 hidden neurons.

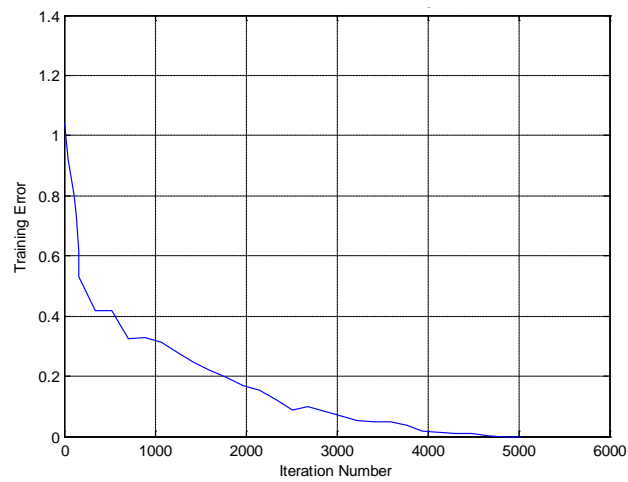


Figure 8. Training error vs iteration number for CODC controller using ANNBMC with 10 hidden neurons.

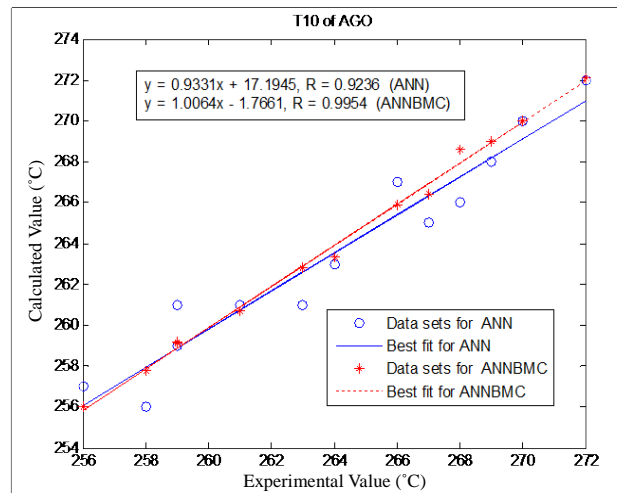


Figure 9. Comparative test results for the prediction of T_{10} of AGO using ANN and ANNBMC.

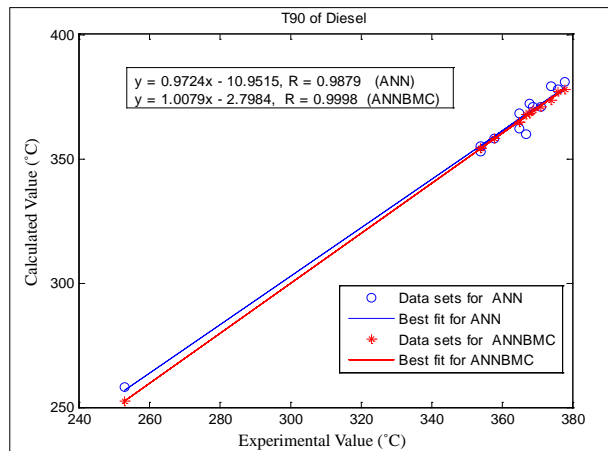


Figure 10. Comparative test results for the prediction of T₉₀ of AGO using ANN and ANNBMCM.

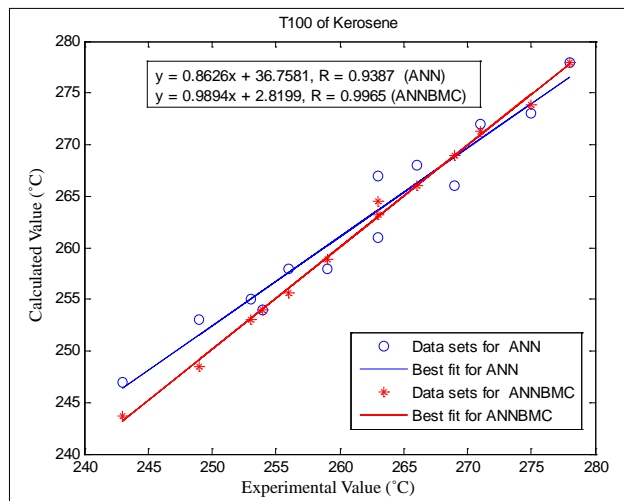


Figure 11. Comparative test results for the prediction of T₁₀₀ of AGO using ANN and ANNBMCM.

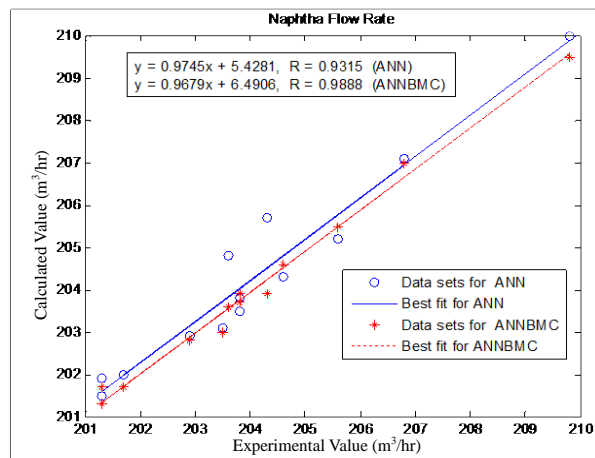


Figure 12. Comparative test results for the prediction of naphtha flow rate using ANN and ANNBMCM.

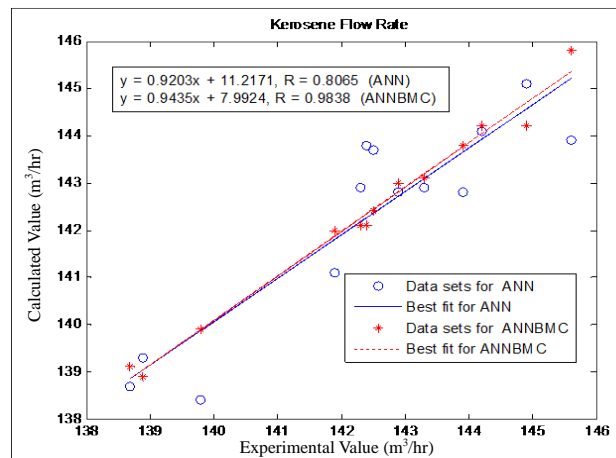


Figure 13. Comparative test results for the prediction of kerosene flow rate using ANN and ANNBMC.

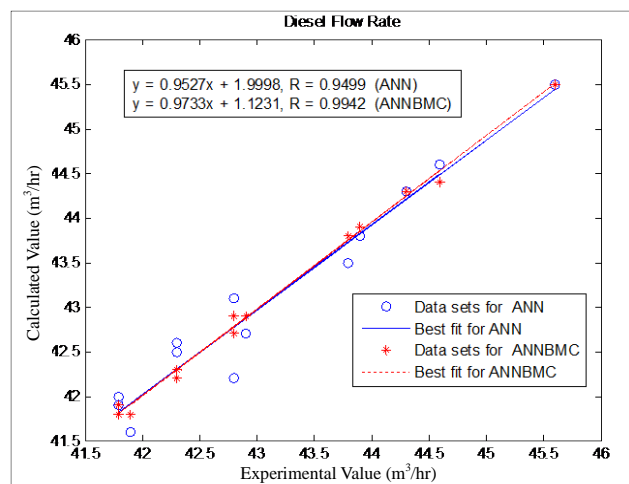


Figure 14. Comparative test results for the prediction of diesel flow rate using ANN and ANNBMC.

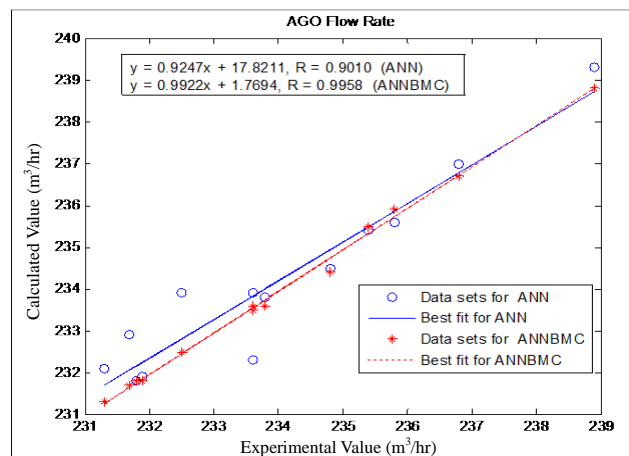


Figure 15. Comparative test results for the prediction of naphtha flow rate using ANN and ANNBMC.

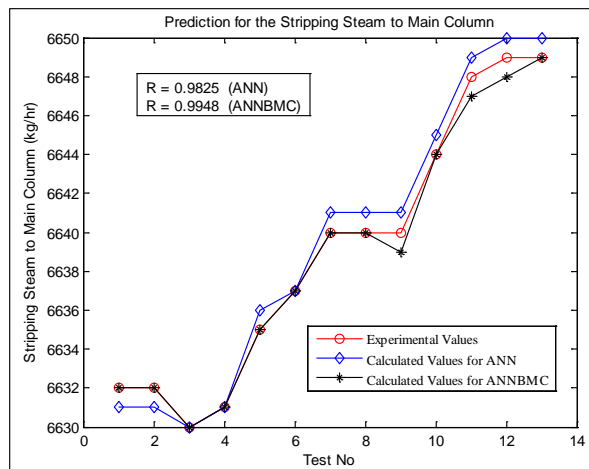


Figure 16. Comparative test results for the prediction of stripping steam to main column using ANN and ANNBMC.

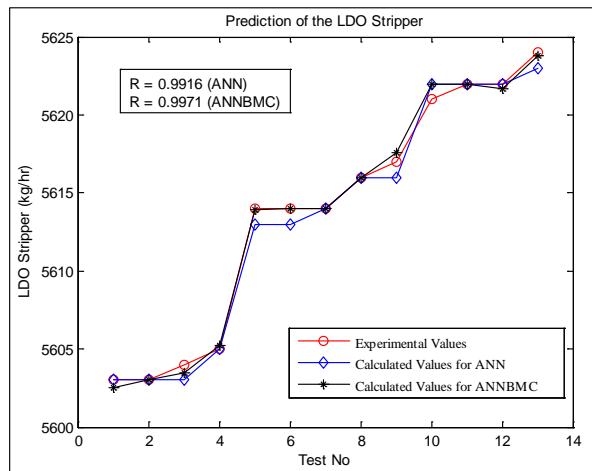


Figure 17. Comparative test results for the prediction of LDO stripper using ANN and ANNBMC.

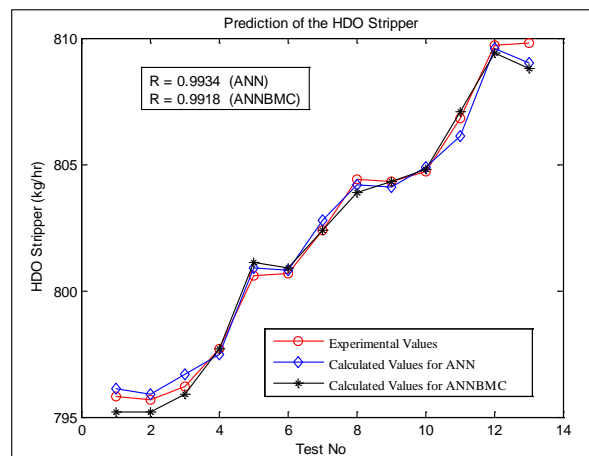


Figure 18. Comparative test results for the prediction of HDO stripper using ANN and ANNBMC.

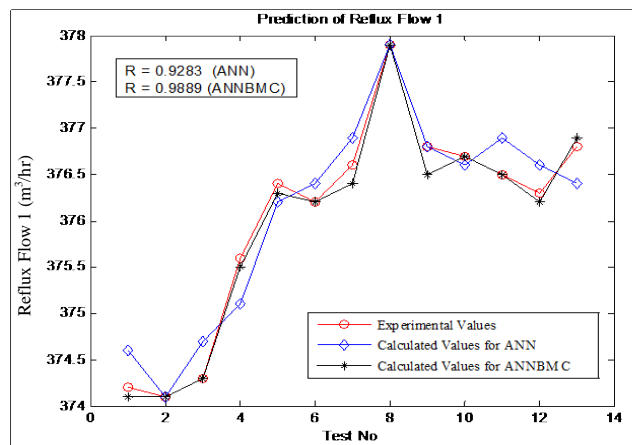


Figure 19. Comparative test results for the prediction of reflux flow 1 using ANN and ANNBMC.

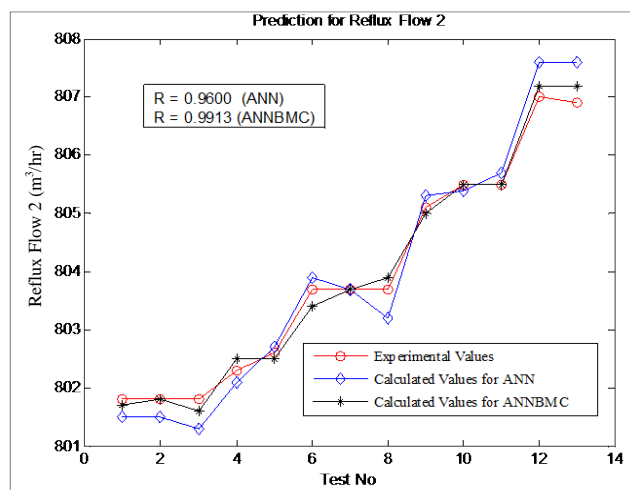


Figure 20. Comparative test results for the prediction of reflux flow 2 using ANN and ANNBMC.

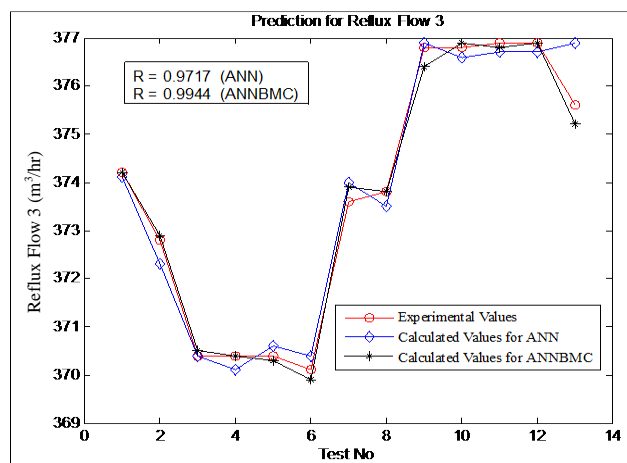


Figure 21. Comparative test results for the prediction of reflux flow 3 using ANN and ANNBMC.

controller prediction using ANN and ANNBMC.

5. Discussion of Results

The maximum training error obtained when artificial neural networks (ANN) model with architecture of 10 hidden neurons was used for the crude oil distillation column (CODC) design was below 6 while convergence was achieved below 400 iterations as shown in **Figure 5**. For the CODC design using ANNBMC, the training network converged to less than 10^{-6} below 2500 iterations with training error of less than 1.6 as shown in **Figure 6**. A maximum training error of less than 1.5 was achieved with convergence of below 700 iterations for the CODC controller using ANN only while the maximum training error was less than 1.2 and converged to a value of less than 10^{-7} after about 5000 iterations for the CODC controller using ANNBMC as depicted in **Figure 7** and **Figure 8** respectively. The prediction error decreases with increasing iteration number during the network training in all the plots of training errors against iteration numbers. Although larger number of iteration steps was required to achieve convergence of less than 10^{-7} when ANNBMC was used for both the design of the CODC and controller than when ANN only was used, lesser values of training errors were exhibited by the network architecture. The maximum relative errors between the experimental data and the calculated data obtained from the output variables of the neural network for CODC design were 1.98 error % and 0.57 error % when ANN only and ANNBMC were used respectively. For the CODC controller, the maximum relative errors between the experimental data and the calculated data obtained from the output variables of the network architecture were 0.346 error % and 0.124 error % when ANN only and ANNBMC were used respectively. This is an indication that minimum errors were achieved when ANNBMC was used for both the CODC design and controller prediction which corresponds to its training network architecture having lesser training error values with higher number of iteration steps in both task. Thus, using ANNBMC shows that a better training of the network with higher number of iteration steps reduces the relative error between the experimental data and calculated data.

The linear regression analysis performed between the experimental data obtained from the refinery and the calculated data obtained from the neural network architecture for the comparative test results for the prediction of T_{10} of AGO, T_{90} of Diesel, T_{100} of Kerosene, Naphtha, Kerosene, Diesel and AGO flow rates using ANN and ANNBMC are shown in **Figures 9-15** respectively. The correlation coefficients obtained for T_{100} of Kerosene, T_{90} of Diesel, T_{10} of AGO, naphtha, kerosene, diesel and AGO flow rates were 0.9387, 0.9879, 0.9236, 0.9315, 0.8065, 0.9499 and 0.9010 respectively when ANN only was used for the CODC design. The correlation coefficients obtained for T_{100} of Kerosene, T_{90} of Diesel, T_{10} of AGO, naphtha, kerosene, diesel and AGO flow rates were 0.9965, 0.9998, 0.9954, 0.9888, 0.9838, 0.9942 and 0.9958 respectively when ANNBMC was used for the CODC design. This analysis performed shows that the prediction accuracies for the CODC design were better when ANNBMC was used. The minimum and maximum prediction accuracies were 80.65% and 98.79%; and 98.38% and 99.98% when ANN and ANNBMC were used for the CODC design respectively. This is an indication that the network architecture for the ANNBMC was rigorously trained for better prediction accuracies and thus can be used to predict design variables (output variables) of the crude oil distillation column than using only ANN.

The linear regression analysis performed between the experimental data obtained from the refinery and the calculated data obtained from the neural network architecture for the comparative test results for the prediction of stripping steam to main column, LDO stripper, HDO stripper, reflux flows 1, 2 and 3 using ANN and ANNBMC are shown in **Figures 16-21** respectively. The regression coefficients executed between the experimental and calculated data were 0.9825, 0.9916, 0.9934, 0.9283, 0.9600 and 0.9717 for the stripping steam to main column, LDO stripper, HDO stripper, reflux flow 1 (Top Pump around), reflux flow 2 (Kerosene Pump around) and reflux flow 3 (Light Diesel Oil Pump around) respectively when ANN only was used for the CODC controller design. When ANNBMC was used for the CODC controller design, the regression coefficients were 0.9948, 0.9971, 0.9918, 0.9889, 0.9913 and 0.9944 for the stripping steam to main column, LDO stripper, HDO stripper, reflux flow 1 (Top Pump around), reflux flow 2 (Kerosene Pump around) and reflux flow 3 (Light Diesel Oil Pump around) respectively. The minimum and maximum prediction accuracies were 92.83% and 99.34%; and 98.89% and 99.71% when ANN and ANNBMC were used for the CODC controller respectively as both have excellent predictions. However, using ANNBMC for the CODC controller still predicts excellently well than ANN. These results also reflected in the minimal difference between the training error (<1.6 and <1.2)

of each of the network architecture for both the ANN and ANNBMC respectively for the CODC controller as shown in **Figure 7** and **Figure 8**. Excellent predictions for both methodologies resulted from their maintenance at particular values for various inputs of the network architecture. The little deviations between their output variables and that of the PID controller from the refinery resulted from their excessive usage by the PID controller to meet the product specifications (T_{100} of Kerosene, T_{90} of Diesel and T_{10} of AGO, naphtha, kerosene, diesel and AGO flow rates). Thus, the artificial neural networks based Monte Carlo simulation controller is effective for the predictions of the output variables and maximally relating the non-linear behaviour existing among various variables of the process.

6. Conclusions and Recommendation

The expert system design and control of crude oil distillation column using artificial neural networks (ANN) model and artificial neural networks based Monte Carlo (ANNBMC) simulation of random processes had been done. MATLAB computer program had been written to simulate the artificial neural networks back-propagation algorithm of both methodologies for both the design and control of crude oil distillation column using experimental data of Port-Harcourt refinery, Nigeria. The design of the crude oil distillation column and its controller gave effective accuracies with minima error percent for their various output variables when ANNBMC was used than when ANN only was used. The ANNBMC controller is effective for the predictions of the output variables and maximally relating the non-linear behaviour existing among various variables of the process than ANN controller. Hence, artificial neural networks based Monte Carlo simulation is an effective tool for the design and control of crude oil distillation column.

It is highly recommended that computational complexity of both methodologies should be reduced as the training requires using many data for its accurate prediction. Also, there is need for the improvement in the network architecture used for the ANNBMC in order to give lesser training error with larger reduction in the iteration number as the training process requires a lot of time before convergence can be achieved.

References

- [1] Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, **2**, 359-366. [http://dx.doi.org/10.1016/0893-6080\(89\)90020-8](http://dx.doi.org/10.1016/0893-6080(89)90020-8)
- [2] Popoola, L.T., Babagana, G. and Susu, A.A. (2013) A Review of an Expert System Design for Crude Oil Distillation Column Using the Neural Networks Model and Process Optimization and Control Using Genetic Algorithm Framework. *Advances in Chemical Engineering and Science*, **3**, 164-170. <http://dx.doi.org/10.4236/aces.2013.32020>
- [3] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning Internal Representation by Error Propagation. In: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press, Cambridge, 1, 318-361.
- [4] Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*. Macmillan Publishing Company, New York.
- [5] Hansen, L.K. and Solomon, P. (1990) Neural Network Ensembles. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **12**, 903-1002. <http://dx.doi.org/10.1109/34.58871>
- [6] Popoola, L.T., Babagana, G. and Susu, A.A. (2013) Thrombo-Embolic Stroke Prediction and Diagnosis Using Artificial Neural Network and Genetic Algorithm. *IJRRAS*, **14**, 655-661.
- [7] Firestone, M., Fenner-Crips, P., Barry, T. and Bennett, D. (1997) *Guiding Principles for Monte Carlo Analysis*. Risk Assessment Forum, US Environmental Protection Agency, Washington DC.
- [8] Schuëller, G.I. and Spanos, P.D. (2001) Monte Carlo Simulation. *Proceedings of the International Conference on Monte Carlo Simulation*, Lisse, 17-19 July 2001, 24-36.
- [9] Zhang, X.G., Guo, X.Y., Zhong, B. and Peng, S.Y. (1998) Monte Carlo Simulation in Supercritical Methanol-Hexane System. *Chinese Journal of Chemical Engineering*, **14**, 413-418.
- [10] Alexander, N., Moyeed, R. and Stander, J. (2000) Spatial Modelling of Individual-Level Parasite Counts using the Negative Binomial Distribution. *Biostatistics*, **1**, 453-463.
- [11] Gilks, W. and Berzuini, C. (2001) Following a Moving Target: Monte Carlo Inference for Dynamic Bayesian Models. *Journal of the Royal Statistical Society: Series B*, **1**, 127-146.
- [12] Khu, S.T. and Werner, M.G.F. (2003) Reduction of Monte-Carlo Simulation Runs for Uncertainty Estimation in Hydrological Modelling. *Hydrology and Earth System Sciences*, **7**, 680-692. <http://dx.doi.org/10.5194/hess-7-680-2003>
- [13] Kuo, R.J., Liao, J.L. and Tu, C. (2005) Integration of ART2 Neural Network and Genetic K-means Algorithm for Analyzing Web Browsing Paths in Electronic Commerce. *Decision Support Systems*, **40**, 355-374.

<http://dx.doi.org/10.1016/j.dss.2004.04.010>

- [14] Jia, Y.X. and Guo, X.Y. (2006) Monte Carlo Simulation of Methanol Diffusion in Critical Media. *Chinese Journal of Chemical Engineering*, **14**, 413-418. [http://dx.doi.org/10.1016/S1004-9541\(06\)60093-1](http://dx.doi.org/10.1016/S1004-9541(06)60093-1)
- [15] Yeh, W.C., Yu, C.Y. and Lin, C.H. (2007) Evaluate Voting System Reliability using the Monte Carlo Simulation and Artificial Neural Network. *2nd International Conference on Wireless Broadband and Ultra Wideband Communications*, Sydney, 27-30 August 2007, 45-49. <http://dx.doi.org/10.1109/AUSWIRELESS.2007.32>
- [16] Sugiyama, S. (2008) Monte Carlo Simulation/Risk Analysis on a Spreadsheet: Review of Three Software Packages. *Foresight: The International Journal of Applied Forecasting*, **2008**, 36-41.
- [17] Pang, Z., Liu, D., Jin, N. and Wang, Z. (2008) A Monte Carlo Particle Model Associated with Neural Networks for Tracking Problem. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **55**, 3421-3429.
- [18] Oscar, T.P. (2009) General Regression Neural Network and Monte Carlo Simulation Model for Survival and Growth of Salmonella on Raw Chicken Skin as a Function of Serotype, Temperature and Time for Use in Risk Assessment. *Journal of Food Protection*, **72**, 2078-2087.
- [19] Liu, J. (2012) Predicting the Products of Crude Distillation Columns. Ph.D. Thesis, School of Chemical Engineering and Analytical Science, University of Manchester, Manchester.
- [20] Safdari, M. and Shamsoddini, M. (2012) Using Artificial Neural Networks and Monte Carlo Simulation in Terms of Uncertainty for Prediction of Budget Deficit in Iran. *Interdisciplinary Journal of Contemporary Research in Business*, **4**, 132-139.
- [21] Zilouchian, A. and Bawazir, K.H. (1999) Application of Neural Networks in Oil Refineries. *Proceedings of IEEE International Conference on Neural Networks*, New Orleans, 11-13 May 1999, 126-135.
- [22] Liau, L.C.K., Yangb, T.C.K. and Tsaib, M.T. (2004) Expert System of a Crude Oil Distillation Unit for Process Optimization Using Neural Networks. *Expert Systems with Applications*, **26**, 247-255.
- [23] Motlaghi, S., Jalali, F. and Ahmadabadi, M.N. (2008) An Expert System Design for a Crude Oil Distillation Column with the Neural Networks Model and the Process Optimization using Genetic Algorithm Framework. *Expert Systems with Applications*, **35**, 1540-1545. <http://dx.doi.org/10.1016/j.eswa.2007.08.105>
- [24] Tonnang, Z.E.H. (2010) Distillation Column Control Using Artificial Neural Networks. M. Sc Thesis, Microprocessors and Control Engineering, Department of Electrical and Electronics Engineering, Faculty of Technology, University of Ibadan, Ibadan.
- [25] Popoola, L.T., Babagana, G. and Susu, A.A. (2013) Expert System Design and Control of Crude Oil Distillation Column of a Nigerian Refinery using Artificial Neural Network Model. *International Journal of Research and Reviews in Applied Sciences*, **15**, 337-346.
- [26] McCarthy, J. (1984) Some Expert System Need Common Sense. Stanford University, Stanford.
- [27] Rich, E. (1994) Artificial Intelligence. McGraw Hill, New York.
- [28] Miller, W. and Osborne, H.G. (1938) History and Development of Some Important Phases of Petroleum Refining in the United States. The Science of Petroleum, Oxford University Press, London, 1465-1477.
- [29] Liebmann, K., Dhole, V.R. and Jobson, M. (1998) Integrated Design of a Conventional Crude Oil Distillation Tower Using Pinch Analysis. *Chemical Engineering Research and Design*, **76**, 335-347. <http://dx.doi.org/10.1205/026387698524767>
- [30] Beer, M. and Spanos, P.D. (2005) Neural Network Based Monte Carlo Simulation of Random Processes. *Proceedings of the 9th International Conference on Structural Safety and Reliability (ICOSSAR 2005)*, Millpress, Rotterdam, 9-16.
- [31] Lee, S.C. (2003) Prediction of Concrete Strength Using Artificial Neural Networks. *Engineering Structures*, **25**, 849-857. [http://dx.doi.org/10.1016/S0141-0296\(03\)00004-X](http://dx.doi.org/10.1016/S0141-0296(03)00004-X)
- [32] Bishop, C.M. (1995) Neural Networks for Pattern Recognition. Clarendon Press, Oxford.
- [33] Haykin, S. (1999) Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River.
- [34] de Freitas, J.F.G. (2001) Bayesian Methods for Neural Networks. Ph.D. Thesis, University of Cambridge, Cambridge.
- [35] Neal, R.M. (1995) Bayesian Learning for Neural Networks. Ph.D. Thesis, Graduate Department of Computer Science, University of Toronto, Toronto.
- [36] Fearnhead, P. (1998) Sequential Monte Carlo Methods in Filter Theory. Ph.D. Thesis, Department of Statistics, Oxford University, Oxford.
- [37] Doucet, A. (1997) Monte Carlo Methods for Bayesian Estimation of Hidden Markov Models. Application to Radiation Signals. Ph.D. Thesis, University Paris-Sud, Orsay.
- [38] Djuric, P.M. (1999) Monitoring and Selection of Dynamic Models by Monte Carlo Sampling. *IEEE Higher Order Statistics Workshop*, Ceasarea, 14-16 June 1999, 191-194.