Artificial Neural Network Modelling, Simulation and Prediction of Gas Production

Lydia Tamuno-Dieyepriye Philip [1], Chinonso Martins Okafor [2], Tochukwu Divine Eluwa [3], Salisu Salihu Alhassan [4], Abdulwahab Giwa [5]

[1] *GANKCO Energy Services*
17 Market Lane, Rumuomasi, Port Harcourt, Rivers State, Nigeria

[2] *CypherCrescent Limited*
76A Fiddil Avenue, Trans Amadi Industrial Layout, Off Ordinance Road, Port Harcourt, Rivers State, Nigeria

[3] *Chevron Nigeria Limited, Warri, Nigeria*
Escravos, Ugborodo, Delta State, 331106, Nigeria

[4] *Abubakar Tafawa Balewa University*
Dass road, P. M. B. 0248, Bauchi, 740272, Nigeria

[5] *Federal University Dutsin-Ma*
P. M. B. 5001 Dutsin-ma, Katsina State, Nigeria

DOI: 10.22178/pos.121-23

LCC Subject Category: T1-995

Received 16.07.2025
Accepted 27.08.2025
Published online 31.08.2025

Corresponding Author:
Abdulwahab Giwa

Abstract. Recently, exploration, drilling and production in oil and gas sources have become challenging owing to the complexity of the system. Industry developers established these sources several years ago, and their production histories now differ. To that end, the production and administration of oil and gas resources have necessitated the application of an advanced method of data processing, referred to as an artificial neural network, to solve these challenges. An Artificial Neural Network (ANN), a type of Artificial Intelligence (AI), is a network of interconnected nodes inspired by the structure of neurons in the brain. One primary goal of neural networks is to solve the complex problems of the oil and gas industry that cannot be easily addressed using traditional modelling tools. This method typically helps decision-makers improve choices and reduce non-productive time and costs. In this research, we developed an artificial neural network (ANN) model of a gas production system in MATLAB to model, simulate, and predict gas production. We trained the network on field gas-production data, using temperature and pressure as input parameters, and applied various training algorithms. We varied the number of hidden-layer neurons and the delays in the model, which produced 35 distinct outputs. The predicted outputs demonstrated excellent performance, achieving a correlation value of 0.98 and a mean squared error of less than 2%. Furthermore, the statistical error metrics showed excellent agreement between the ANN predictions and field report data. Thus, the results indicated that the ANN model could be applied to predict gas production accurately from a flow station.

Keywords: Artificial neural network (ANN); MATLAB; hidden layers; training algorithm; mean square error (MSE).

INTRODUCTION

In today's highly competitive environment, oil and gas production and management necessitate the use of cutting-edge technology. These tools enable the reduction of the cost of hydrocarbon resource exploration, production, and management. Engineers continually strive to keep pace with the latest advancements in information technology. Employing computers in the workplace, incorporating sophisticated simulation models in decision-making processes, and digital control and monitoring of equipment that were considered cutting-edge only a few years ago are now standard operating procedures. The term "Advanced Technologies" has a fluid meaning [1], and one of its components is an artificial neural network.

Artificial neural networks and fuzzy set theory, with their applications in artificial intelligence, have taken on a new meaning in recent years, giving the phrase "Advanced Technologies" a new meaning. These tools provide engineers and scientists with the foundation for developing intelligent machines. These types of tools have significant potential in hydrocarbon exploration, production, and management. Although the expert system is only one member of the artificial intelligence family, it has been used as a synonym for artificial intelligence. In fact, many AI researchers believe that neural networks have accomplished considerably more in their short lifetimes than expert systems have in their entire lifetimes. The utilisation of neural networks, which are non-algorithmic, non-digital, intensely parallel, and distributive information processing systems, is increasing every day.

In recent years, several articles on the application of neural networks in the petroleum industry have appeared in the literature. This collection comprises two types of articles. Those who use neural networks to study formation lithology from well logs, and those who utilise neural networks to choose a reservoir model for use in traditional well test interpretation studies. The use of a fault-tolerant procedure to automate these processes, which are traditionally conducted by log analysts and reservoir engineers, could be beneficial.

Oil/gas exploration, drilling, production, and reservoir management are complex these days because most conventional oil and gas sources have already been discovered and have been producing for many years. As a result, petroleum engineers are utilising advanced tools, such as artificial neural networks (ANNs), to support decision-making and reduce non-productive time and costs.

The goal of this study is to demonstrate that neural networks can address some of the most significant challenges faced in the petroleum industry. Engineers and researchers can utilise neural networks to address fundamental petroleum engineering challenges, as well as specific problems that traditional computing methods have been unable to solve. Specifically, this study aims to apply a neural network to predict gas production in a given gas flow station.

Theoretical Background

1) Neural Network Structure. An artificial neural network is an information-processing system that exhibits specific performance characteristics similar to those of biological neural networks [2]. Cells are the building blocks of all living things. Neurons are the fundamental components of the nervous system. A cell body, an axon, and dendrites make up a normal biological neuron, as shown in Figure 1.

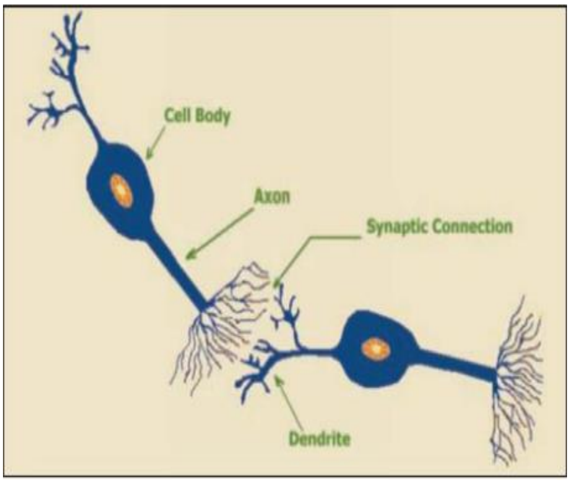

Figure 1 – Two bipolar neurons [2]

Dendrites allow information to enter the cell body. The cell body then sends an output through the axon to another receiving neuron, and the output from the first neuron becomes an input for the second one, and so on [2].

The human brain comprises between 10 and 500 billion neurons [3]. There are approximately 500 neural networks in each region that contain these neurons [4]. Approximately 100,000 neurons comprise each neural network, and these neurons are connected to thousands of additional neurons [2]. The complex behaviour of humans is attributed to this structure. Simple tasks like walking, collecting a cup of coffee, or moving one's hands require incredibly intricate calculations that the human brain can perform, but advanced computers cannot. At the same time, computer chip cycles run from nanoseconds to milliseconds, and human brain cycles span from 10 to 100 milliseconds. Despite this, the human brain is still capable of performing considerably more complex activities than computers, thanks to the advanced structures of the neurons.

Artificial neural networks (ANNs) are a simulation of the above-mentioned biological process. ANNs are created using mathematical models that make the following assumptions [2]:

1. Information is processed by elements known as neurons.

2. There are links between the neurons that allow information to pass through.

3. Each link in the connection has its own weight.

4. Once the neurons have received the inputs, they will use an action function to determine the outputs.

Figure 2 is a schematic of an artificial neuron; the outputs of other neurons are multiplied by the weights of the connection links and enter the neuron.

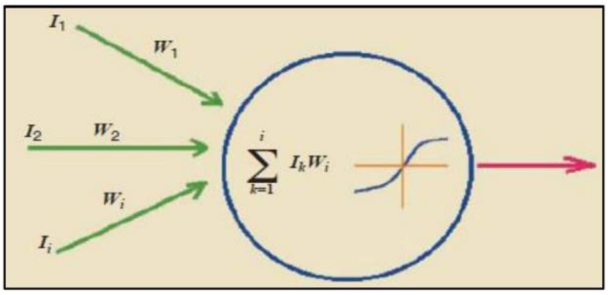

Figure 2 – Schematic of an artificial neuron [2]

The inputs are then combined, and the neuron's activation function is applied to generate an output. As a result, a neuron has multiple inputs but only one output. A neural network consists of an input layer, one or more hidden layers, and an output layer. The input and output layers, as the names suggest, are for inputs and outputs. The hidden layer is responsible for extracting features from the data [2]. ANNs can be simple three-layer networks, as illustrated in Figure 3, or more complex, as shown in Figure 4.

2) Mechanism of a Neural System. Training and test sets are the two categories into which a typical neural data processing algorithm separates the database. The training set builds the desired network for the model. Depending on the paradigm, this procedure involves the network adjusting the weights between its neurons or processing units using the desired result in the training set (supervised training).

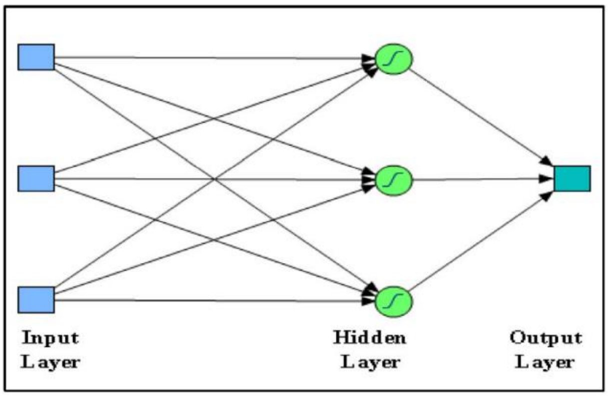

Figure 3 – Example of a simple network [5]

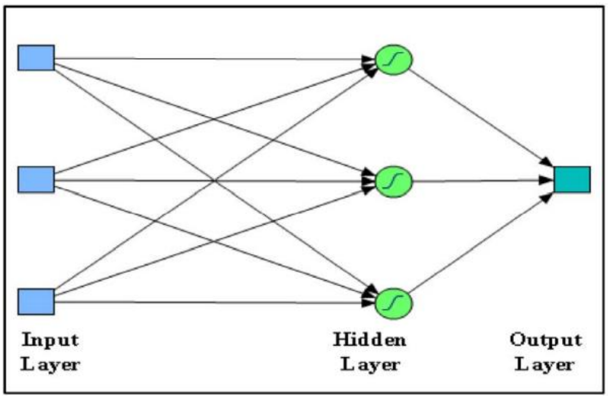

Figure 4 – Example of a complex network [5]

Applying the test set to the network for validation occurs after the network has "converged" on the data in the training set. It is crucial to remember that although the user has the appropriate test set output, the network has not seen it. Researchers perform this step to ensure the trained network remains resilient and intact, thereby allowing for a clearer understanding of the neural system's actual functionality.

A neuron is a nerve cell that has every nerve cell function. One of the most distinguishing characteristics of animals is their nervous systems, which include neurons. The body of the cell contains the nucleus. The nucleus is reached by at least one dendrite. In most situations, impulses are conducted toward the cell body by the branching, tapering processes of the nerve cell. Axons are nerve cell processes that carry impulses out from the cell body.

Nerve structures are made up of bundles of neurons known as nerve fibres. Nerves, in a simplified scenario, carry impulses from receptor organs (such as the eyes or ears) to effector organs (such as muscles or glands). A synapse is the point between two neurons in a neural pathway where the termination of one neuron's axon comes into proximity with the cell body or dendrites of another. At this point, a microscopic gap exists, and the relationship between the two neurons is one of contact only. The impulse that travels through the first neuron causes an impulse to travel through the second one. Signals enter the synapses. These are the parameters. They have been "weighted". That is to say, some signals are more powerful than others. Some signals stimulate (positive), while others inhibit (negative). The neuron adds together the effects of all weighted inputs. If the sum is equal to or greater than the neuron's threshold, the neuron fires (gives output). It is "all-or-nothing" in this case. A neuron either fires or does not fire.

Recent advancements in hardware have enabled the computer simulation of artificial neural networks. Although it may seem odd to simulate a parallel process on a sequential machine, numerous advantages have been identified. It has bought time for the fundamental goal of implementing neural networks in hardware, and it has shed light on flaws in previous models. Simulations have enabled us to understand better and improve the technology, as well as predict the performance of a specific neural network in a given application. Analogue neural network circuits have been built and tested in addition to simulations.

In neural computing, the artificial neuron is referred to as a Processing Element, or PE for short. Another name for this fundamental building piece is a node. The resemblance between these artificial neurons and their natural counterparts is minimal. They barely approximate biological neurons to a first-order degree. Neurons in the human brain carry out at least 150 distinct tasks, and Processing Elements models roughly three of those tasks. The PE is responsible for several essential tasks. The system evaluates every input signal to determine its strength. The next step is to calculate the sum of the input signals and compare it to a threshold. Lastly, it needs to determine the output. A PE should have many input signals, just as a neuron has many inputs (stimulation levels). They should all enter PE at the same time. In response, depending on some threshold level, a neuron either "fires" or "does not fire". The PE will be given a single output signal, similar to how a biological neuron works - many inputs, one output.

Furthermore, just as real neurons are influenced by factors other than inputs, some networks provide a mechanism for such influences. Engineers sometimes refer to this additional input as a bias term or a forcing term, and they refer to it as a forgetting term when the system needs to unlearn previously learned information. The system assigns each input a relative weight, which determines its impact; this is analogous to the varying synaptic strengths of biological neurons. In the way they combine to produce an impulse, some inputs are more important than others. Weights are network adaptive coefficients that determine the intensity of the input signal. They could be thought of as a measure of the connection's strength. The network changes the initial weight of a PE in response to various inputs and according to its modification rules.

Mathematically, the inputs and their weights could be considered as vectors, such as (I1, I2,…In) and (W1, W2, ...Wn). The dot product, also known as the inner product, of the two vectors yields the total input signal. The outcome is a scalar rather than a vector. The inner product of two vectors can be used to calculate their geometric similarity. The inner product is maximum when the vectors point in the same direction; it is minimum when the vectors point in opposite directions (180 degrees). Signals can be positive (excitatory) or negative (inhibitory).

A positive input encourages the PE to fire, whereas a negative input inhibits the PE from firing. If some local memory is connected to the PE, the results of previous computations can be saved and the weights used as the process continues. The ability to change the weights enables the PE to modify or learn its behaviour in response to its inputs. Assuming that a network classifies a production well as "an injection well", connection weights that respond correctly to a production well are strengthened in subsequent iterations; those that react incorrectly to others, such as an injection well, are weakened until they fall below the threshold level. It takes more than simply changing the weights for production well recognition; the weights must be adjusted so that all objects are correctly identified and classified. Backpropagation is used when weight adjustments are made in preceding layers of feedforward networks by "backing up" from the outputs; this is an essential concept because backpropagation algorithms are used in a large percentage of all networks today.

Assuming that this processing element is combined with other PEs to form a layer of these nodes. Inputs can be connected to multiple nodes with varying weights, resulting in a series of outputs, one for each node. The connections, which roughly correspond to axons and synapses in a biological system, provide a signal transmission pathway between the nodes. Several layers can be linked together. The input layer is the layer that receives the inputs. It typically serves no purpose other than to buffer the input signal. The output layer generates the network outputs. Any other layer is referred to as a hidden layer because it is internal to the network and has no direct contact with the outside world. Researchers sometimes compare the hidden layer to a "black box" within the network system. However, they can still investigate it even though it is not immediately visible. There could be one or several hidden layers. The weights associated with each interconnect are multiplied by the connections. They represent analogue values. It is worth noting that there are far more connections than nodes. If every output from one layer is passed along to every node in the next one, the network is said to be fully connected. This description of neural system components is based primarily on Nelson and Illingworth's book [1].

3) Application of Artificial Neural Network in Petroleum Engineering. Some fundamental petroleum engineering problems, as well as specific ones that conventional computing has been unable to solve, can be addressed by neural networks. When engineering data for design and interpretation is insufficient, petroleum engineers may benefit from neural networks; this is particularly common in basins and fields that have been producing for a long time. Due to the high cost of coring, well testing, and other related expenses, there may be a lack of adequate engineering data. Neural networks have shown great promise in generating accurate analysis and results from large amounts of historical data that would otherwise appear to be useless or irrelevant in the analysis. Neural networks are a viable alternative in addressing many problems in petroleum engineering [6].

Literature Review

Some researchers have worked on the subject matter. For instance, authors [7] developed a neural network model to forecast U.S. natural gas supply up to the Year 2020. The developed neural network model served as both a short-term and long-term predictive tool for natural gas supply. The researchers found that the model could be used to investigate the quantitative impact of various physical and economic factors on future gas production. Authors [8] developed a neural network to predict the WAT for various hydrocarbon mixtures. The model was trained using different combinations of thermodynamic properties of twelve different hydrocarbon fluids and subsequently validated against experimental data. Results show that the ANN approach was able to predict the WAT more accurately than the existing models. The researchers found that the average absolute deviations (AAD%) for the ANN were lower than those of the existing models. They also discovered that using a combination of molecular weight, density, and activation energy as input parameters produced the most accurate predictions.

Furthermore, the authors [9] assessed the application of proxy models generated through Artificial Neural Networks (ANNs) as a substitute for the flow simulator in the history matching process, demonstrating that the ANN can efficiently capture the nonlinearities of the problems. A synthetic reservoir with real characteristics was used to test the methodology. The researchers found that applying the ANN as a proxy model was promising and that they could achieve a good match with fewer simulations. Authors [10] used an Artificial Neural Network (ANN) for flow pattern identification, but with a pre-processing stage using natural logarithmic normalisation. This pre-processing stage helps to normalise the extensive data range and to reduce overlapping between flow patterns. The researchers extended the model's validity by using dimensionless inputs, enabling its application to horizontal pipes of various diameters, liquid densities, and viscosities. The concept was validated by building and testing the model using both experimental data and well-known multiphase flow models. An ANN model was built using three dimensionless parameters: the liquid Reynolds number, the gas Reynolds number, and the pressure drop multiplier. The developed model achieved an accuracy of more than 97% in classifying flow patterns across a wide range of flow conditions. Authors [11] developed a predictive data-driven model to understand well performance and forecast gas production using DTS data and daily flowing time as dynamic inputs.

The researchers took DTS measurements along the lateral side of well MIP-3H each day and upscaled them to the stage scale using an averaging method. The researchers trained a multilayer perceptron neural network (MLPNN) using stage-based daily DTS data and daily flow time to predict the next day's gas production. A sensitivity analysis was carried out by removing each stage DTS attribute from the input dataset to identify the most influential stages in predicting gas production. The sensitivity analysis (SA) reveals that several stages may carry higher weights in predicting gas production, while others have a lesser impact on prediction accuracy. Authors [12] carried out a study that focused on predicting oil production rates using the Levenberg-Marquardt backpropagation algorithm to train the Back Propagation Artificial Neural Network (BPANN) and Decline Curve Analytical Methods (DCAMs). The study considered 1600 data sets, with 70% for training and the remaining 30% for testing. The input parameters used were the gas production rate, tubing head pressure, and flowing bottom-hole pressure, with the crude oil production rate serving as the output. The developed BPANN model can predict oil production rates as a function of gas rates, production times, flowing bottom-hole pressures, and tubing head pressures.

Based on the literature review conducted, it was found that few studies are available on predicting gas production using artificial neural networks.

METHOD

The development of the neural network was carried out with the aid of MATLAB Neural Network Toolbox using the procedures outlined thus.

Selecting the Dynamic Time Series Methods. Prediction is a type of dynamic filtering that utilises past values of one or more time series to forecast future values. Dynamic neural networks, which include tapped delay lines, are used for non-linear filtering and prediction. This tool allows the solution of three types of non-linear time series problems, as shown in Figure 5.

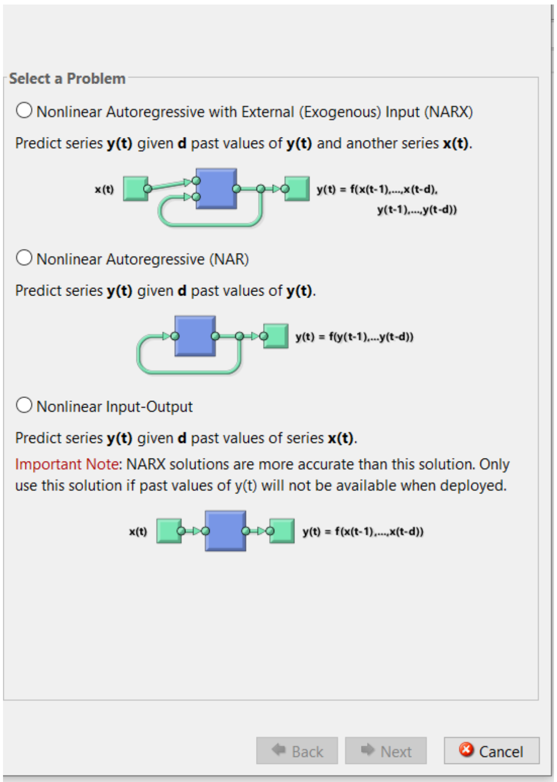

Figure 5 – Selection of a non-linear time series problem

The three types of non-linear time series models were employed in this study to predict gas production.

Selection of Data. At a particular time, after selecting the type of non-linear time series model, the researchers chose the data set for the neural network. As shown in Figure 6, they prepared two sets of input data: first, the input data $x(t)$, and then the target data, which defines the desired output y(t).

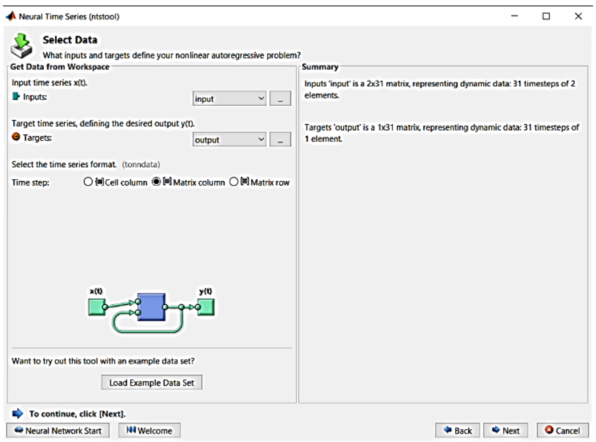

Figure 6 – Data selection

They then converted the time series format into a matrix column.

Validation and Test Data. The percentage of validation and testing was varied as shown in Figure 7.

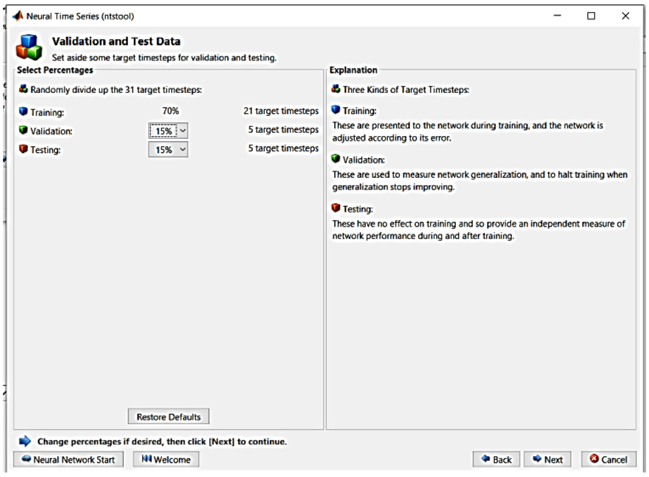

Figure 7 – Validation and testing data

The timesteps for training were fixed, but when the validation and testing timesteps were varied, the percentages were normalised to ensure that everything summed up to 100%.

Network Architecture. The researchers chose the number of neurons and input/feedback delays, as shown in Figure 8.

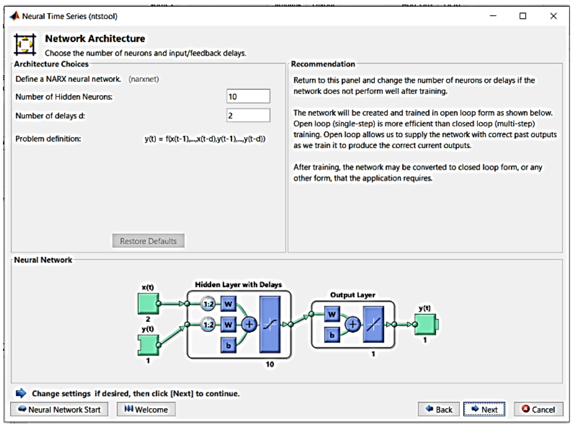

Figure 8 – Network Architecture of the Neural Network

However, if the network did not perform well after training, they could adjust the number of neurons or delays. In this study, they varied the number of neurons and delays to determine the optimal values that produced highly accurate results.

Training the Network. The researchers trained the network to fit the input and produce the target data. To do this, they selected a training algorithm. For this study, three training algorithms were employed: Levenberg-Marquardt, Bayesian Regularisation, and Scaled Conjugate Gradient (Figure 9).

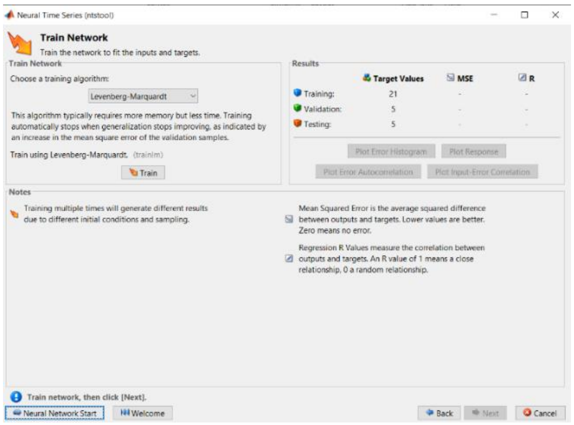

Figure 9 – Training the Network

After training the network, they calculated the mean square error and regression value. They defined the mean square error as the average of the squared differences between the outputs and the targets. The lower the value of the mean square error, the better the model developed. The regression (R) value was used to measure the correlation between the outputs and the targets. The closer the R to 1, the better the developed model.

Deploying Solution. Generation of deployable versions of the neural network, such as application deployment, code generation, Simulink deployment, and graphical representations, can be carried out using the toolbox on this page (Figure 10).

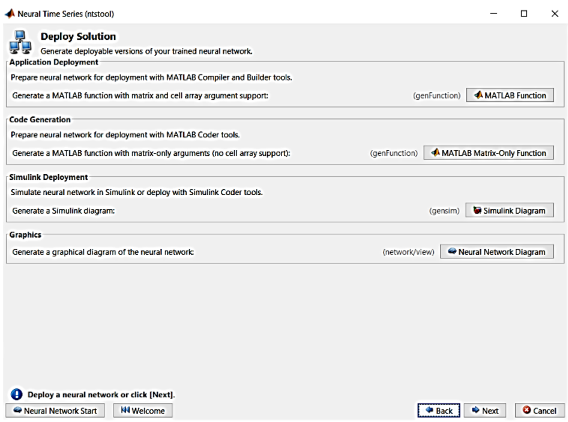

Figure 10 – Deploying the solution

Saving Results. The researchers generated the MATLAB script for the developed model, which is presented on this page (Figure 11).

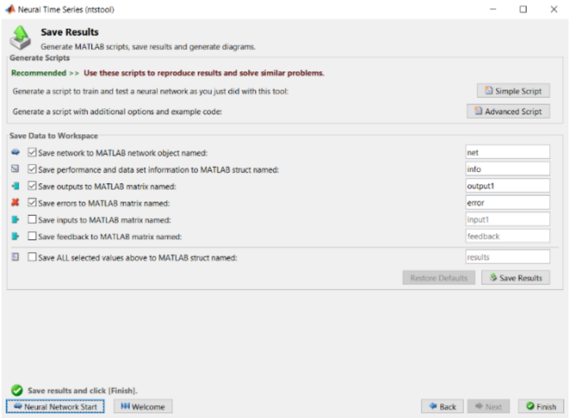

Figure 11 – Saving results

Additionally, the results, along with all data from the model, were saved to the workspace.

The diagrams of the different neural network models considered in this study are presented in Figures 12–19, as indicated.

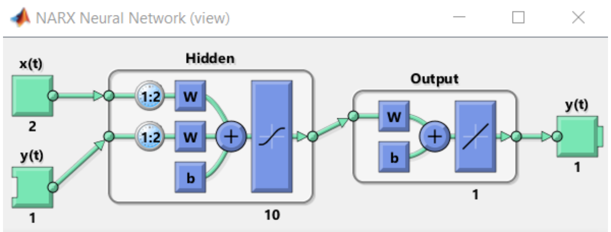

Figure 12 – NARX Neural Network

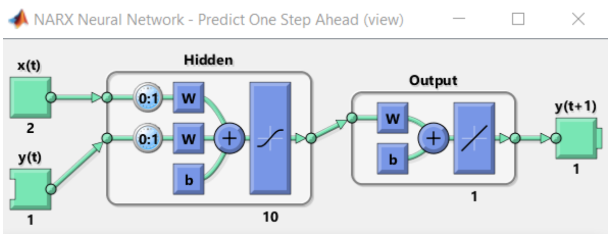

Figure 13 – NARX Neural Network - Predict One Step Ahead

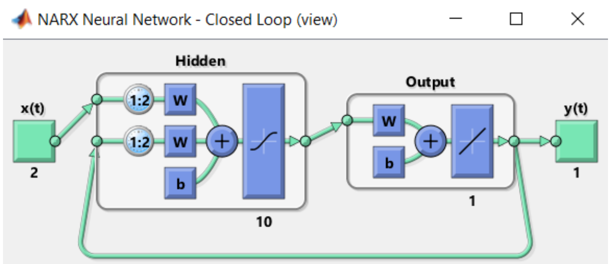

Figure 14 – NARX Neural Network - Closed Loop

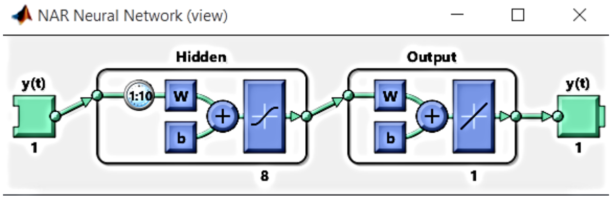

Figure 15 – NAR Neural Network

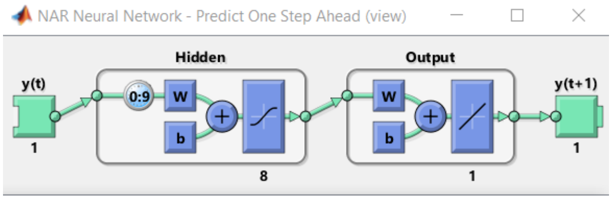

Figure 16 – NAR Neural Network - Predict One Step Ahead

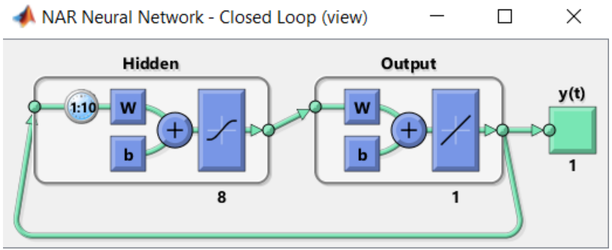

Figure 17 – NAR Neural Network - Closed Loop

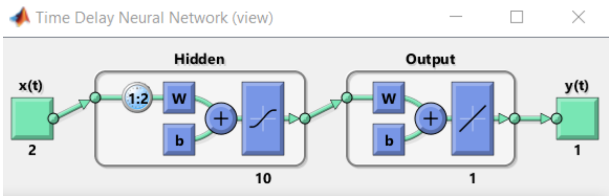

Figure 18 – Time Delay Neural Network

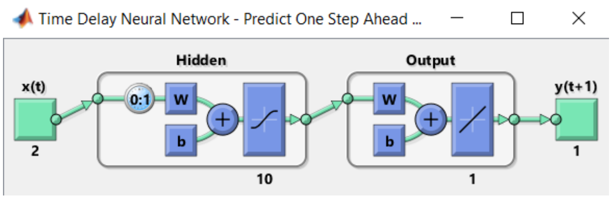

Figure 19 – Time Delay Neural Network - Predict One Step Ahead

Simulation of the Neural Network Model. The simulation of the network was carried out using various dynamic time series methods, with the number of neurons in the hidden layer, the number of delays, and the training algorithms being varied. The dynamic time series methods, training algorithm, number of neurons in the hidden layer used and number of delays applied were as given in Table 1.

Table 1 – Varied neural network parameters

Method	Training Algorithm	No. of Neurons in the Hidden Layer	No. of Delays
NARX	Levenberg	10	2
NARX	Bayesian	10	2
NARX	Scaled Conjugate	10	2
NARX	Levenberg	8	2
NARX	Levenberg	5	5
NARX	Bayesian	8	2
NARX	Bayesian	5	5
NARX	Bayesian	2	8
NARX	Scaled Conjugate	8	2
NARX	Scaled Conjugate	2	8
NARX	Scaled Conjugate	3	10
NAR	Levenberg	10	2
NAR	Levenberg	8	2
NAR	Levenberg	5	5
NAR	Levenberg	2	8
NAR	Bayesian	10	2
NAR	Bayesian	8	2
NAR	Bayesian	5	5
NAR	Bayesian	2	8
NAR	Bayesian	3	10
NAR	Scaled Conjugate	10	2
NAR	Scaled Conjugate	8	2
NAR	Scaled Conjugate	5	5
Non-linear	Levenberg	10	2
Non-linear	Levenberg	5	5
Non-linear	Levenberg	2	8
Non-linear	Levenberg	3	10
Non-linear	Bayesian	10	2
Non-linear	Bayesian	8	2
Non-linear	Bayesian	3	10
Non-linear	Scaled Conjugate	10	3
Non-linear	Scaled Conjugate	8	2
Non-linear	Scaled Conjugate	5	5

RESULTS AND DISCUSSION

Table 2 presents the training simulation results for the gas production at the gas flow station. Generally, several studies have applied ANN for predicting gas production in a reservoir.

Table 2 – Outputs of the model evaluation

Method	Training Algorithm	No. of Neurons in the Hidden Layer	No. of Delays	MSE	R
NARX	Levenberg	10	2	2147483647	0.685
NARX	Bayesian	10	2	2147483647	-0.712
NARX	Scaled Conjugate	10	2	2147483647	0.9324
NARX	Levenberg	8	2	2147483647	-0.274
NARX	Levenberg	5	5	2147483647	0.427
NARX	Bayesian	8	2	2147483647	0.955
NARX	Bayesian	5	5	2147483647	0.98
NARX	Bayesian	2	8	2147483647	0.997
NARX	Scaled Conjugate	8	2	2147483647	0.837
NARX	Scaled Conjugate	2	8	2147483647	0.952
NARX	Scaled Conjugate	3	10	2147483647	0.988
NAR	Levenberg	10	2	2147483647	-0.345
NAR	Levenberg	8	2	2147483647	0.177
NAR	Levenberg	5	5	2147483647	-0.552
NAR	Levenberg	2	8	2147483647	-0.327
NAR	Bayesian	10	2	2147483647	0.804
NAR	Bayesian	8	2	2147483647	0.985
NAR	Bayesian	5	5	2147483647	-0.49
NAR	Bayesian	2	8	2147483647	0.972
NAR	Bayesian	3	10	2147483647	0.699
NAR	Scaled Conjugate	10	2	2147483647	0.562
NAR	Scaled Conjugate	8	2	2147483647	0.913

Method	Training Algorithm	No. of Neurons in the Hidden Layer	No. of Delays	MSE	R
NAR	Scaled Conjugate	5	5	2147483647	-0.256
Non-Linear	Levenberg	10	2	2147483647	-0.7098
Non-Linear	Levenberg	5	5	2147483647	0.244
Non-Linear	Levenberg	2	8	2147483647	-0.807
Non-Linear	Levenberg	3	10	2147483647	0.732
Nonlinear	Bayesian	10	2	2147483647	-0.277
Nonlinear	Bayesian	8	2	2147483647	0.58
Nonlinear	Bayesian	3	10	2147483647	0.85
Nonlinear	Scaled Conjugate	10	3	2147483647	0.89
Nonlinear	Scaled Conjugate	8	2	2147483647	0.75
Nonlinear	Scaled Conjugate	5	5	2147483647	0.87

In this study, the researchers clarified the importance of the number of neurons and the number of hidden layers. They varied the number of hidden layers, adjusted the number of neurons in each layer, and the number of delays until the model achieved optimal performance. They also varied other parameters during the neural-network simulations and presented the results in Table 2. The results presented in Table 2 indicate that certain parameter combinations can yield negative values of the regression value (R). As such, it is essential to find a suitable combination of parameters to apply in developing the neural network model.

The result of the training performance, as measured by the mean square error and the number of epochs during training, is depicted in Figure 20. Successful training was completed, resulting in the lowest errors in the verification and testing curves, which were nearly identical based on the epoch numbers. As shown in Figure 20, the result of the gas production converged to a mean square of 116985161372386.2 at the 21st iteration.

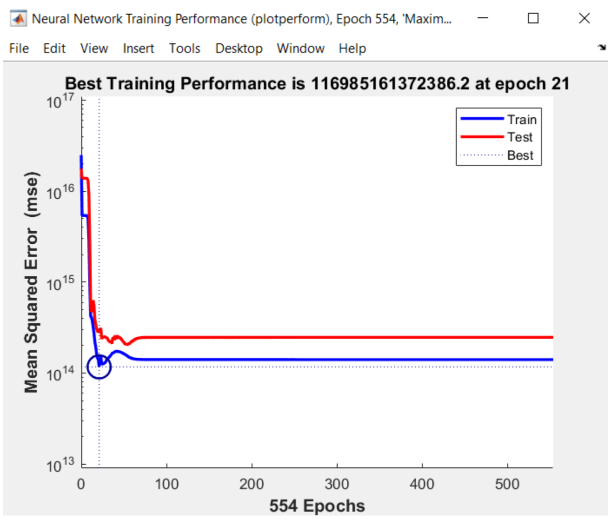

Figure 20 – Plot of mean squared error

Figure 21 shows the correlation between the actual gas production and the predicted gas production obtained from the developed neural network.

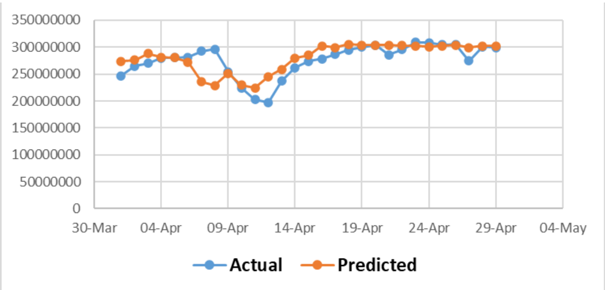

Figure 21 – Graph of actual and predicted gas production

As can be seen, there was a similarity in the trend of the gas production values for the gas flow station, indicating that the developed neural network was a valid one. The predictive neural model yielded a range of production data between 200 MMSCF and 300 MMSCF.

The regression values obtained for the training set, testing set, and the desired output, as shown in Figure 22, were 0.8567, 0.9389, and 0.86248, respectively. Since the values were close to 1, it indicated a strong relationship between the output and the target data.

The results obtained also suggested that there was a need to develop ANN predictive tools for evaluating the performance of a gas flow station. By comparing its results with the reported data from the gas flow station, it was demonstrated that the ANN model achieved high accuracy in its predictions. The reason for this excellent predictive performance was due to the careful selection of specific parameters for training the ANN model. The researchers found that the difference between the predictive results and the field report data was less than 2%.

The weekly average gas production values were calculated and compared with actual and predicted values from the developed neural network, as shown in Figure 23.

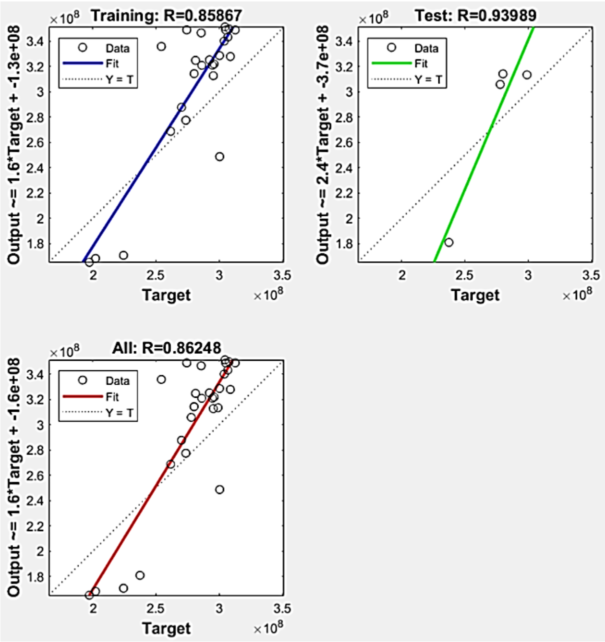

Figure 22 – Plot of regression values

According to the figure, there was a good agreement between the actual and the predicted values. The researchers identified this as another indication of the validity of the developed neural network model.

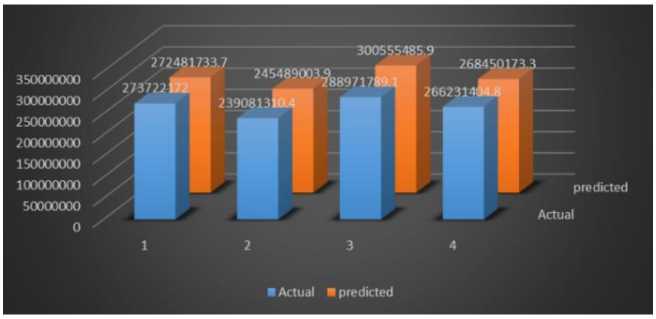

Figure 23 – Statistical comparison of the actual and the predicted data sets

CONCLUSIONS

The researchers found that MATLAB successfully developed a neural network model to predict gas production. They discovered that the neural network model could accurately forecast gas production for the gas flow station, thereby increasing efficiency and output. The results of the developed predictive model demonstrated the versatility of the artificial neural network in successfully predicting gas production for a gas flow station. The developed neural network analysis compared the actual amount of gas produced at the gas flow station to the expected values, and it was discovered that the developed neural network could accurately predict the production of more gas. The results provided by the predictive model indicated its validity, as it yielded values close to one for both the regression values of the training and testing datasets. Furthermore, a value difference of less than 2% between the predictive results and the field report data indicated that the field production data and the predicted data were closely related.

REFERENCES

1. Balan, B., Mohaghegh, S., & Ameri, S. (1995). State-Of-The-Art in Permeability Determination from Well Log Data: Part 1- A Comparative Study, Model development. *SPE Eastern Regional Meeting*. doi: 10.2118/30978-ms

2. Mohaghegh, S. (2000). Virtual-Intelligence Applications in Petroleum Engineering: Part 1 – Artificial neural networks. *Journal of Petroleum Technology, 52*(09), 64–73. doi: 10.2118/58046-jpt

3. Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing, Volume 1: Explorations in the Microstructure of Cognition: Foundations*. Cambridge: The MIT Press

4. Stubbs, D. A. (1989). Three applications of neurocomputing in biomedical research. *International 1989 Joint Conference on Neural Networks*

5. Alkinani, H. H., Al-Hameedi, A. T., Dunn-Norman, S., Flori, R. E., Alsaba, M. T., & Amer, A. S. (2019). Applications of artificial neural networks in the petroleum industry: a review. *SPE Middle East Oil and Gas Show and Conference*. doi: 10.2118/195072-ms

6. Mohaghegh, S., Balan, B., & Ameri, S. (1995). State-Of-The-Art in Permeability Determination from Well Log Data: Part 2-Verifiable, Accurate Permeability Predictions, The Touch-Stone of all Models. *SPE Eastern Regional Meeting*. doi: 10.2118/30979-ms

7. Al-Fattah, S. M., & Startzman, R. A. (2001). Predicting natural gas production using an artificial neural network. *SPE Hydrocarbon Economics and Evaluation Symposium.* doi: 10.2118/68593-ms

8. Adeyemi, B. J., & Sulaimon, A. A. (2012). Predicting wax formation using an artificial neural network. *Nigeria Annual International Conference and Exhibition*. doi: 10.2118/163026-ms

9. Costa, L. a. N., Maschio, C., & Schiozer, D. J. (2014). Application of Artificial Neural Networks in a History Matching Process. *Journal of Petroleum Science and Engineering, 123*, 30–45. doi: 10.1016/j.petrol.2014.06.004

10. Al-Naser, M., Elshafei, M., & Al-Sarkhi, A. (2016). Artificial neural network application for multiphase flow patterns detection: A new approach. *Journal of Petroleum Science and Engineering, 145*, 548–564. doi: 10.1016/j.petrol.2016.06.029

11. Ghahfarokhi, P. K., Carr, T., Bhattacharya, S., Elliott, J., Shahkarami, A., & Martin, K. (2018). A Fibre-optic Assisted Multilayer Perceptron Reservoir Production Modelling: A Machine Learning Approach in Prediction of Gas Production from the Marcellus Shale. *Proceedings of the 9th Unconventional Resources Technology Conference*. doi: 10.15530/urtec-2018-2902641

12. Marfo, S. A., & Kporxah, C. (2020). Predicting Oil Production Rate Using Artificial Neural Network and Decline Curve Analytical Methods. *Proceedings of 6th UMaT Biennial International Mining and Mineral Conference*, pp. 43-50