

Forecasting of Ghana's Interest Rates Using Artificial Neural Networks

Philip Danso, Priscilla Awuah Mante, Susana Nelson Gyimah, and Vicentia Kwakyewaa Yeboah

Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

DOI: <https://doi.org/10.51244/IJRSI.2026.1308000313>

Received: 05 September 2026; Accepted: 15 September 2026; Published: 24 September 2026

ABSTRACT

Accurate interest-rate forecasts are important for monetary-policy analysis, financial planning, investment decisions, and risk management. This study develops an artificial neural network (ANN) model for forecasting Ghana's interest rate from selected macroeconomic indicators. A total of 264 monthly observations covering 2000–2021 were obtained from the Ghana Statistical Service and the Bank of Ghana. Gross domestic product, inflation, unemployment, and money supply were used as explanatory inputs, while interest rate was specified as the target variable. Because the variables were measured on different scales, min–max normalization was applied before model estimation. A two-layer feed-forward neural network with a 4–10–1 architecture was implemented in MATLAB, comprising four input variables, 10 hidden neurons, and one output neuron. A tan-sigmoid transfer function was used in the hidden layer, a linear transfer function in the output layer, and the Levenberg–Marquardt backpropagation algorithm was used for training. The data were partitioned into training (70%), validation (15%), and testing (15%) subsets. Model performance was assessed using mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), weighted absolute percentage error (WAPE), and the correlation coefficient (R). For the full dataset, the model produced MSE = 0.00108, RMSE = 0.03286, MAE = 0.3000, WAPE = 4.24%, and R = 0.99297. The test subset produced R = 0.98374, indicating a strong association between predicted and observed values. These findings suggest that a feed-forward ANN can model nonlinear relationships between selected macroeconomic indicators and Ghanaian interest rates. However, the results should be interpreted as evidence of predictive performance within the study design rather than as proof of superiority over alternative forecasting models.

Keywords: interest rate forecasting; artificial neural network; Ghana; macroeconomic indicators; Levenberg–Marquardt; machine learning

INTRODUCTION

Interest rates are an important component of economic and financial systems because they influence borrowing, saving, investment, consumption, and the allocation of financial resources. For households and businesses, changes in interest rates affect the cost of obtaining credit and the returns associated with saving and investment. Interest rates are also important to financial institutions because they influence lending decisions, asset valuation, portfolio management, and exposure to financial risk. At the macroeconomic level, interest rates constitute an important channel through which monetary policy can influence economic activity and price stability. Consequently, understanding and anticipating interest-rate movements is important for policymakers, investors, financial institutions, businesses, and other economic agents.

In Ghana, interest rates are particularly relevant to the conduct and transmission of monetary policy. The Bank of Ghana operates an inflation-targeting monetary policy framework in which the Monetary Policy Rate (MPR) serves as the main policy instrument for signaling the stance of monetary policy and anchoring short-term market interest rates (Bank of Ghana, n.d.). Changes in monetary conditions can therefore influence market interest rates and, ultimately, borrowing and lending conditions within the economy. However, the transmission of monetary policy through interest rates in Ghana is not necessarily immediate or complete. Kovanen (2011), for example, found that the adjustment of wholesale and retail market interest rates to

changes in Ghana's policy rate can be gradual and incomplete. These characteristics make the behavior and prediction of interest rates an important area of economic and financial analysis.

Despite their importance, interest rates are difficult to forecast accurately because their movements may be associated with several interacting economic conditions. Inflation, economic activity, liquidity conditions, labour-market developments, expectations, and monetary and fiscal policy may contribute to changes in interest rates. Furthermore, relationships among these factors may not remain constant over time. Changes in economic conditions, policy regimes, financial markets, and external shocks can alter the relationships observed in historical data. Consequently, forecasting methods must be capable of extracting useful information from multiple economic variables while accommodating potentially complex relationships among them.

Traditional approaches to financial and economic forecasting have relied extensively on statistical, time-series, and econometric models. These methods remain valuable because they provide established frameworks for representing temporal and economic relationships. Nevertheless, some conventional model specifications require assumptions about the functional relationships among variables. When the underlying relationships are highly nonlinear or involve complicated interactions, a predetermined functional form may not adequately represent all patterns contained in the data. The increasing availability of economic data and advances in computational methods have therefore encouraged the application of machine-learning techniques to economic and financial forecasting.

Artificial neural networks (ANNs) are among the machine-learning approaches that have attracted attention for forecasting applications. ANNs are computational models consisting of interconnected processing units that learn relationships between inputs and outputs by adjusting network weights during training. An important characteristic of multilayer feed-forward neural networks is their ability to approximate complex functional relationships. Hornik et al. (1989) demonstrated that multilayer feed-forward networks possess universal approximation capabilities under appropriate conditions. This flexibility makes ANNs potentially useful for economic and financial applications in which relationships between predictor and target variables may be nonlinear.

The application of neural networks to interest-rate-related forecasting has continued to develop. Neural-network models have been investigated for predicting interest-rate spreads, government bond yields, and yield curves, among other financial variables. For example, Nunes et al. (2019) applied artificial neural networks to yield-curve forecasting and found that multilayer perceptron models using appropriately selected features produced encouraging forecasting results. Similarly, Wegener et al. (2016) investigated neural-network approaches to forecasting government bond yields and showed that incorporating information from cointegration relationships could improve neural-network forecasts. Such studies indicate that ANN-based methods can provide useful tools for modeling interest-rate-related financial data. However, their performance depends on the structure of the problem, the information supplied to the model, and the methods used for training and evaluation.

The potential usefulness of ANNs does not imply that they will always outperform conventional forecasting approaches. Neural networks are sensitive to factors such as data quality, input selection, network architecture, model complexity, training procedures, and validation strategies. A model may fit training observations closely without maintaining the same level of performance when applied to unseen data. Appropriate separation of observations into training, validation, and testing sets is therefore necessary when assessing predictive performance. Furthermore, comparisons with alternative forecasting methods are required before conclusions can be drawn about the relative superiority of one modeling approach over another.

Although neural networks have been increasingly applied to financial forecasting internationally, their usefulness must be evaluated within the specific economic environment and dataset under consideration. Forecasting results obtained for one country or financial market cannot automatically be assumed to apply to another because countries differ in monetary-policy frameworks, financial-market structures, macroeconomic conditions, and the behavior of economic variables. This provides a rationale for examining ANN-based interest-rate forecasting using Ghana-specific macroeconomic data.

Against this background, the present study evaluates the predictive performance of a feed-forward artificial neural network for Ghana's interest rate using selected macroeconomic indicators. The model uses gross domestic product (GDP), inflation, unemployment, and money supply as input variables, with interest rate specified as the target variable. Historical observations covering the period 2000–2021 are used to train, validate, and test the network. Model performance is evaluated using mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), weighted absolute percentage error (WAPE), and the correlation coefficient (R) between observed and predicted values.

The objective of the study is therefore to determine the extent to which a feed-forward ANN can model the relationship between the selected macroeconomic indicators and Ghana's interest rate and generate predictions that correspond closely with observed values. Rather than assuming that the ANN is superior to existing forecasting techniques, the study evaluates the performance of the specified ANN within the available Ghanaian dataset. The findings provide evidence regarding the potential application of neural-network methods to interest-rate forecasting in Ghana and establish a basis for future research that can directly compare ANN performance with conventional econometric, time-series, and other machine-learning forecasting approaches.

LITERATURE REVIEW

Interest-rate forecasting has attracted considerable attention in economic and financial research because interest rates influence borrowing, saving, investment, asset valuation, and monetary-policy decisions. Accurate forecasts can therefore provide useful information to policymakers, financial institutions, investors, and businesses. However, forecasting interest rates remains challenging because their movements are associated with several economic and financial factors whose relationships may change over time. Changes in inflation, economic activity, liquidity, monetary policy, and market expectations can influence interest-rate behavior, making the forecasting process complex.

Various statistical and econometric approaches have traditionally been used to forecast interest rates and related financial variables. These include autoregressive models, vector autoregressive models, term-structure models, and other time-series techniques. Oh and Han (2000), for example, noted the extensive application of statistical approaches such as vector autoregressive (VAR), autoregressive conditional heteroskedasticity (ARCH), generalized autoregressive conditional heteroskedasticity (GARCH), and other time-series models to interest-rate analysis. Although these approaches remain important, the complexity of financial relationships has encouraged researchers to investigate alternative methods capable of accommodating nonlinear patterns in financial data.

Artificial neural networks (ANNs) have consequently received increasing attention as forecasting tools. ANNs are computational models consisting of interconnected processing units that learn relationships between input and output variables through the adjustment of network parameters during training. An important theoretical justification for using neural networks is their ability to approximate complex functional relationships. Hornik et al. (1989) demonstrated that multilayer feed-forward neural networks can approximate a broad class of functions to a desired degree of accuracy under appropriate conditions. This approximation capability makes neural networks attractive for forecasting problems in which the relationship between predictors and the variable being predicted may not follow a simple linear form.

The application of neural networks to forecasting does not, however, imply that conventional forecasting methods are ineffective. Instead, ANN methods provide an alternative framework for representing relationships that may be difficult to specify using predetermined functional forms. Their flexibility allows several predictor variables to be incorporated simultaneously, while the network learns relationships among the predictors and target variable from the available data. This characteristic has encouraged the application of ANNs to several areas of financial forecasting, including stock prices, exchange rates, bond yields, yield curves, and interest rates.

Early research specifically examining ANN-based interest-rate forecasting provides evidence of the potential of this approach. Oh and Han (2000) proposed an interest-rate forecasting method that combined change-point

detection with a backpropagation neural network. Their approach was based on the observation that interest-rate series may experience change points associated with shifts in monetary-policy conditions. The proposed procedure first identified change points in the interest-rate series, classified observations into change-point groups, and subsequently incorporated this information into the neural-network forecasting process. Their work demonstrates that structural changes in financial series may contain useful information for ANN forecasting and that preprocessing the information supplied to a network can be important for predictive performance.

Neural networks have also been applied to government bond-yield forecasting. Wegener et al. (2016) investigated government bond yields in Germany and France using a hybrid approach that combined cointegration analysis with neural-network forecasting. Their results showed that incorporating an error-correction term obtained from the cointegrating relationship into the neural-network model produced superior forecasts to the neural-network model used alone. The authors suggested that differencing cointegrated variables may result in the loss of useful long-run information, whereas incorporating the error-correction term allows some of this information to be retained. This finding is important because it illustrates that neural networks and conventional econometric methods do not necessarily need to be viewed as competing approaches. Instead, information obtained through econometric analysis can complement the nonlinear modeling capability of neural networks.

Nunes et al. (2019) extended the application of artificial neural networks to yield-curve forecasting. They compared multivariate linear regression with multilayer perceptron models and investigated single-task and multitask neural-network approaches for forecasting the European yield curve at several forecasting horizons. Their findings showed that the multilayer perceptron using the most relevant features achieved the strongest overall results across the forecasting horizons considered. They further found that the variables identified as relevant differed according to the target yield and forecasting horizon. These findings highlight the importance of feature selection and suggest that ANN forecasting performance depends not simply on the quantity of information supplied to the model but on the relevance of that information to the particular forecasting problem.

Evidence from other financial applications also demonstrates that ANN performance should not be assumed to be universally superior to traditional forecasting methods. Foster et al. (1992), for example, examined neural-network forecasting using hundreds of short and noisy economic and demographic time series. They found that neural networks used directly as function approximators were generally less accurate than linear regression for the series examined, although neural networks used to combine conventional forecasts produced some improvements. Such evidence is important because it demonstrates that the flexibility of neural networks does not guarantee better forecasting performance in every setting. Model performance depends on the nature of the data and how the neural network is specified, trained, and evaluated.

The relevance of ANN methods to African monetary and financial applications has also begun to receive attention. Ogundele et al. (2020), for example, examined the viability of an artificial neural-network approach for analyzing monetary data in relation to monetary-policy-rate validation in Nigeria. Although their objective differed from direct interest-rate forecasting, the study demonstrates the potential application of neural-network techniques to monetary-policy analysis within an African economy. In Ghana, Nti (2021) developed ANN models for predicting the Ghana cedi exchange rate against the US dollar and British pound using several macroeconomic variables, including inflation, monetary policy, interest rate, trade balance, broad money, and international reserves. The study compared backpropagation, radial-basis-function, and generalized-regression neural networks and reported differences in predictive performance across the three architectures. While the target variable in that study was the exchange rate rather than the interest rate, it provides Ghana-specific evidence that ANN methods can be applied to relationships involving multiple macroeconomic and financial indicators.

Despite their flexibility, neural networks present methodological challenges that must be considered when evaluating forecasting performance. One important problem is overfitting. Overfitting occurs when a network learns characteristics of the training observations, including noise or sample-specific patterns, so closely that its performance deteriorates when it is presented with unseen observations. The flexibility that enables neural

networks to approximate complicated relationships can therefore become a disadvantage when the network is excessively complex relative to the available data. Hsieh (2004) identified overfitting as an important problem in neural-network modeling and emphasized the use of observations not employed in model fitting to evaluate model performance.

The separation of data into training, validation, and testing sets is therefore important in ANN development. Training observations are used to adjust network weights and biases, whereas validation data can provide information about how well the network generalizes during the training process. A separate testing sample can then be used to evaluate the final model on observations that were not used in parameter estimation. This distinction is particularly important because strong performance on training observations alone does not demonstrate that a neural network will maintain the same predictive ability when applied to new data.

Network architecture and training procedures can also influence forecasting results. Decisions concerning the number of hidden layers, number of neurons, activation functions, input variables, training algorithm, and stopping criteria may affect the final model. Furthermore, different initial network parameters can sometimes produce different solutions during nonlinear optimization. Consequently, the reported performance of an ANN should be interpreted in relation to the specific network architecture, training procedure, data, and evaluation criteria used in the study.

The temporal nature of economic and financial observations creates an additional challenge. Interest rates and macroeconomic variables are observed over time, meaning that the chronological ordering of observations contains important information. A practical forecasting exercise typically involves using information available in earlier periods to predict observations in subsequent periods. Therefore, evaluation procedures that preserve temporal ordering can provide a stronger assessment of genuine forecasting ability. This consideration becomes especially important when machine-learning techniques are applied to economic time series because an evaluation procedure that allows information from later periods to influence model development may produce an overly favorable estimate of predictive performance.

The literature therefore presents a balanced view of artificial neural networks in financial and interest-rate forecasting. Studies such as Oh and Han (2000), Wegener et al. (2016), and Nunes et al. (2019) demonstrate that neural networks can provide useful approaches to interest-rate and fixed-income forecasting. At the same time, these studies show that forecasting performance is affected by structural changes, predictor selection, model specification, and the information incorporated into the network. Evidence from broader forecasting research also indicates that neural networks do not necessarily outperform conventional approaches in every application. The effectiveness of an ANN must therefore be evaluated empirically within the particular dataset and economic environment being studied.

This consideration is particularly relevant to Ghana. The country's interest rates operate within a monetary-policy and macroeconomic environment that may differ substantially from those represented in much of the international ANN forecasting literature. The Bank of Ghana uses the Monetary Policy Rate as an important instrument for signaling the monetary-policy stance and guiding short-term market interest rates. Relationships among interest rates, inflation, money supply, economic activity, and other macroeconomic indicators may therefore reflect conditions specific to the Ghanaian economy. Findings obtained from neural-network models developed for European, Asian, or other financial markets cannot automatically be assumed to apply to Ghana.

The reviewed literature shows applications of neural networks to interest rates, government bond yields, yield curves, monetary-policy analysis, and other financial variables. Ghana-specific evidence also demonstrates the application of ANN techniques to financial variables such as exchange rates. However, the literature identified for the present study provides comparatively limited evidence on the use of a feed-forward ANN to model Ghana's interest rate directly from a combination of domestic macroeconomic indicators. This provides the motivation for the present study. GDP, inflation, unemployment, and money supply are used as input variables, while interest rate is specified as the target variable.

The study evaluates whether a feed-forward artificial neural network can learn the relationships contained in these Ghanaian macroeconomic data and generate predictions that correspond closely with observed interest-rate values. Rather than presuming that ANN is superior to established forecasting methods, the study evaluates the predictive performance of the specified network and provides Ghana-specific evidence that can serve as a basis for future comparisons with econometric, time-series, and alternative machine-learning models.

METHODOLOGY

This study employed a quantitative predictive modeling approach using an artificial neural network (ANN) to model interest rates in Ghana based on selected macroeconomic indicators. A two-layer feed-forward neural network with a 4–10–1 architecture was developed using the Neural Network Fitting application in the MATLAB Deep Learning Toolbox. Gross domestic product (GDP), inflation, unemployment rate, and money supply were specified as the input variables, while interest rate was specified as the target variable. The dataset consisted of 264 monthly observations obtained from the Ghana Statistical Service and the Bank of Ghana, covering the period from 2000 to 2021.

Data Preprocessing and Normalization

The variables included in the analysis were measured on different numerical scales. Consequently, min-max normalization was applied before neural-network training to transform the variables to a common range between 0 and 1. For a variable X , normalization was performed as

$$X_i^* = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}, \quad (1)$$

where X_i represents the original observation, $X_{\min}$ and $X_{\max}$ denote the minimum and maximum observed values of the corresponding variable, respectively, and X_i^* represents the normalized observation. Scaling the variables to a common numerical range facilitated numerical optimization during network training and reduced the influence of differences in variable magnitude.

Following preprocessing, the dataset was divided into training, validation, and testing subsets. The training set comprised 70% of the observations, while the validation and testing sets each comprised 15%. The training subset was used to estimate the network parameters, the validation subset was used to monitor model generalization during training, and the testing subset provided an additional evaluation of the trained network on observations not directly used in parameter estimation.

Artificial Neural Network Architecture

A two-layer feed-forward neural network was employed to model the relationship between the four macroeconomic predictors and interest rate. The network consisted of 10 neurons in the hidden layer and one neuron in the output layer, giving a 4–10–1 network architecture. The hidden layer used the hyperbolic tangent sigmoid (TANSIG) transfer function, while the output layer used the linear (PURELIN) transfer function.

The four input variables can be represented for observation t by the vector

$$x_t = \begin{bmatrix} x_{1t} \\ x_{2t} \\ x_{3t} \\ x_{4t} \end{bmatrix}, \quad (2)$$

where x_{1t} , x_{2t} , x_{3t} , and x_{4t} denote GDP, inflation, unemployment rate, and money supply, respectively.

For hidden neuron j , the net input is given by

$$z_{jt} = \sum_{i=1}^4 w_{ji} x_{it} + b_j, \quad (3)$$

where w_{ji} represents the weight connecting input variable i to hidden neuron j , and b_j represents the corresponding hidden-layer bias.

The hyperbolic tangent sigmoid (TANSIG) transfer function was used in the hidden layer, whereas the linear transfer function (PURELIN) was used in the output layer, consistent with the architecture implemented in the original model. The activation of hidden neuron j was therefore determined as

$$h_{jt} = \text{tansig}(z_{jt}) = \frac{2}{1 + e^{-2z_{jt}}} - 1. \quad (4)$$

The nonlinear TANSIG function maps the hidden-layer activation to the interval $(-1, 1)$, allowing the network to represent nonlinear relationships between the macroeconomic predictors and interest rate.

The output of the network was calculated as

$$\hat{y}_t = \sum_{j=1}^H v_j h_{jt} + b_o, \quad (5)$$

where $\hat{y}_t$ represents the predicted interest rate, H denotes the number of hidden neurons, v_j represents the weight connecting hidden neuron j to the output neuron, and b_o represents the output-layer bias. Because PURELIN is an identity function,

$$f(a) = a, \quad (6)$$

the network generates a continuous output appropriate for predicting interest rates.

Combining the hidden- and output-layer operations, the network can be represented as

$$\hat{y}_t = \sum_{j=1}^H v_j \left[\frac{2}{1 + \exp \left\{ -2 \left(\sum_{i=1}^4 w_{ji} x_{it} + b_j \right) \right\}} - 1 \right] + b_o. \quad (7)$$

Equation (7) summarizes the nonlinear relationship learned by the ANN between the four macroeconomic input variables and the target interest rate.

Forward Propagation and Error Computation

During forward propagation, the input observations were transmitted through the network to generate predicted interest rates. For each observation t , the weighted input to each hidden neuron was first computed using Equation (3), followed by application of the TANSIG activation function:

$$h_{jt} = \frac{2}{1 + e^{-2z_{jt}}} - 1. \quad (8)$$

The resulting hidden-layer outputs were subsequently transmitted to the output neuron:

$$a_t = \sum_{j=1}^H v_j h_{jt} + b_o. \quad (9)$$

Since PURELIN was used as the output transfer function,

$$\hat{y}_t = a_t. \quad (10)$$

The prediction error associated with observation t was then defined as

$$e_t = y_t - \hat{y}_t, \quad (11)$$

where y_t denotes the observed interest rate and $\hat{y}_t$ denotes the corresponding ANN prediction.

The network training objective was based on minimizing the squared prediction errors. The error function can be represented as

$$E = \frac{1}{2} \sum_{t=1}^n (y_t - \hat{y}_t)^2. \quad (12)$$

Backpropagation and Network Training

The neural network was trained using backpropagation, whereby the prediction errors obtained during forward propagation were propagated backward through the network to determine the sensitivity of the error function to the network weights and biases.

For an output-layer weight v_j , the chain rule gives

$$\frac{\partial E_t}{\partial v_j} = \frac{\partial E_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial v_j}. \quad (13)$$

Since

$$\frac{\partial E_t}{\partial \hat{y}_t} = \hat{y}_t - y_t \quad (14)$$

and

$$\frac{\partial \hat{y}_t}{\partial v_j} = h_{jt}, \quad (15)$$

the gradient associated with the output-layer weight is

$$\frac{\partial E_t}{\partial v_j} = (\hat{y}_t - y_t)h_{jt}. \quad (16)$$

For a weight w_{ji} connecting input variable i to hidden neuron j , the chain rule becomes

$$\frac{\partial E_t}{\partial w_{ji}} = \frac{\partial E_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_{jt}} \frac{\partial h_{jt}}{\partial z_{jt}} \frac{\partial z_{jt}}{\partial w_{ji}}. \quad (17)$$

Given that

$$\frac{\partial \hat{y}_t}{\partial h_{jt}} = v_j, \quad (18)$$

the derivative of the TANSIG activation function is

$$\frac{\partial h_{jt}}{\partial z_{jt}} = 1 - h_{jt}^2, \quad (19)$$

and

$$\frac{\partial z_{jt}}{\partial w_{ji}} = x_{it}, \quad (20)$$

the resulting gradient is

$$\frac{\partial E_t}{\partial w_{ji}} = (\hat{y}_t - y_t)v_j(1 - h_{jt}^2)x_{it}. \quad (21)$$

Thus, backpropagation determines the sensitivity of the network error to each weight and bias. These derivatives are subsequently used by the optimization algorithm to update the network parameters. The original study employed the Levenberg–Marquardt backpropagation algorithm for this optimization process.

Levenberg–Marquardt Training Algorithm

The Levenberg–Marquardt (LM) algorithm was used to optimize the weights and biases of the feed-forward neural network. The method is designed for nonlinear least-squares optimization and provides an interpolation between the Gauss–Newton method and gradient-descent behavior (Levenberg, 1944; Marquardt, 1963). Its application to feed-forward neural-network training has been described by Hagan and Menhaj (1994).

Let the complete vector of adjustable network parameters be represented by

$$\theta = [\theta_1, \theta_2, \dots, \theta_p]^T, \quad (22)$$

where the elements of θ consist of the weights and biases of the network.

For n observations, the vector of prediction errors is

$$e(\theta) = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}. \quad (23)$$

The corresponding sum-of-squares objective function can be expressed as

$$E(\theta) = \frac{1}{2} e(\theta)^T e(\theta) = \frac{1}{2} \sum_{t=1}^n e_t^2. \quad (24)$$

The Jacobian matrix J contains the first-order partial derivatives of the individual prediction errors with respect to each network parameter:

$$J(\theta) = \begin{bmatrix} \frac{\partial e_1}{\partial \theta_1} & \frac{\partial e_1}{\partial \theta_2} & \cdots & \frac{\partial e_1}{\partial \theta_p} \\ \frac{\partial e_2}{\partial \theta_1} & \frac{\partial e_2}{\partial \theta_2} & \cdots & \frac{\partial e_2}{\partial \theta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_n}{\partial \theta_1} & \frac{\partial e_n}{\partial \theta_2} & \cdots & \frac{\partial e_n}{\partial \theta_p} \end{bmatrix}. \quad (25)$$

Each row of J therefore corresponds to an observation, while each column represents the sensitivity of the prediction errors to one of the network parameters. The Jacobian provides information about how small changes in the network's weights and biases affect the prediction errors. In neural-network training, these first-order derivatives are efficiently obtained through the backpropagation procedure.

For a nonlinear least-squares problem, the gradient of the objective function can be approximated using

$$g = J^T e, \quad (26)$$

while the Hessian matrix can be approximated by

$$H \approx J^T J. \quad (27)$$

This approximation forms the basis of the Gauss–Newton approach. The corresponding parameter update can be expressed as

$$\theta_{k+1} = \theta_k - (J^T J)^{-1} J^T e, \quad (28)$$

where k denotes the current training iteration.

The Levenberg–Marquardt algorithm modifies the Gauss–Newton approximation by introducing a damping parameter, μ , and the identity matrix, I . The resulting parameter update is

$$\theta_{k+1} = \theta_k - (J^T J + \mu I)^{-1} J^T e, \quad (29)$$

where θ_k represents the current vector of weights and biases, J is the Jacobian matrix, J^T is its transpose, e is the vector of network errors, I is the identity matrix, and μ is the damping parameter.

The damping parameter controls the behavior of the optimization procedure. When μ is small, Equation (29) approaches the Gauss–Newton update,

$$\theta_{k+1} \approx \theta_k - (J^T J)^{-1} J^T e. \quad (30)$$

When μ becomes relatively large,

$$J^T J + \mu I \approx \mu I,$$

so that

$$\theta_{k+1} \approx \theta_k - \frac{1}{\mu} J^T e, \quad (31)$$

which has the form of a gradient-descent update with an effective step size of $1/\mu$. Thus, the LM algorithm combines the rapid local convergence associated with the Gauss–Newton method with the greater stability of gradient descent when the current parameter estimates are farther from a minimum.

During network training, the input observations were first propagated forward through the network to obtain predicted interest rates. The differences between the observed and predicted values were then used to construct the error vector. Backpropagation was used to calculate the derivatives required to construct the Jacobian matrix, after which the LM update in Equation (29) was used to adjust the weights and biases. This forward- and backward-propagation procedure was repeated iteratively until the applicable stopping criterion was reached.

The computational sequence can therefore be summarized mathematically as

$$x_t \rightarrow \hat{y}_t \rightarrow e_t \rightarrow J \rightarrow \Delta \theta \rightarrow \theta_{k+1}. \quad (32)$$

In other words, the input variables generate a predicted interest rate through forward propagation; the prediction is compared with the observed interest rate to obtain the error; backpropagation determines the derivatives forming the Jacobian; and the Levenberg–Marquardt algorithm uses this information to update the weights and biases so as to reduce the prediction error. This iterative procedure is consistent with the optimization approach reported in the original study.

Model Performance Evaluation

The predictive performance of the ANN was evaluated using mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), weighted absolute percentage error (WAPE), and the Pearson correlation coefficient (R). These measures were specified in the original methodology for evaluating network performance.

MSE was calculated as

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (33)$$

while RMSE was calculated as

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (34)$$

MAE was calculated as

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (35)$$

and WAPE was calculated as

$$WAPE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i|} \times 100. \quad (36)$$

Lower MSE, RMSE, MAE, and WAPE values indicate smaller discrepancies between the observed and predicted interest rates.

The Pearson correlation coefficient was additionally used to evaluate the strength of the linear association between observed and predicted interest-rate values:

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}. \quad (37)$$

Values of R approaching +1 indicate a strong positive association between observed and predicted values. In this study, R was interpreted strictly as the Pearson correlation coefficient and not as the coefficient of determination (R^2).

Model performance was examined separately for the training, validation, and testing subsets. The training results describe model fit to observations used during parameter estimation, while validation and test performance provide information about the network's ability to generalize to observations not directly used to estimate its parameters.

RESULTS AND DISCUSSION

The performance of the artificial neural network (ANN) was evaluated using the training, validation, and testing datasets and by examining the overall agreement between the observed and predicted interest-rate values. The evaluation considered the correlation coefficient (R), mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and weighted absolute percentage error (WAPE). In addition to these numerical measures, regression plots, training-state diagnostics, the performance curve, and the comparison between target and predicted values were examined to provide a more complete assessment of network performance.

Regression Performance of the ANN

Figure 1 presents the regression plots for the training, validation, testing, and combined datasets. The plots compare the network outputs with the corresponding target values. The dotted line $Y = T$ represents perfect agreement between the predicted and observed values, while the solid fitted lines show the empirical relationship obtained from the ANN outputs.

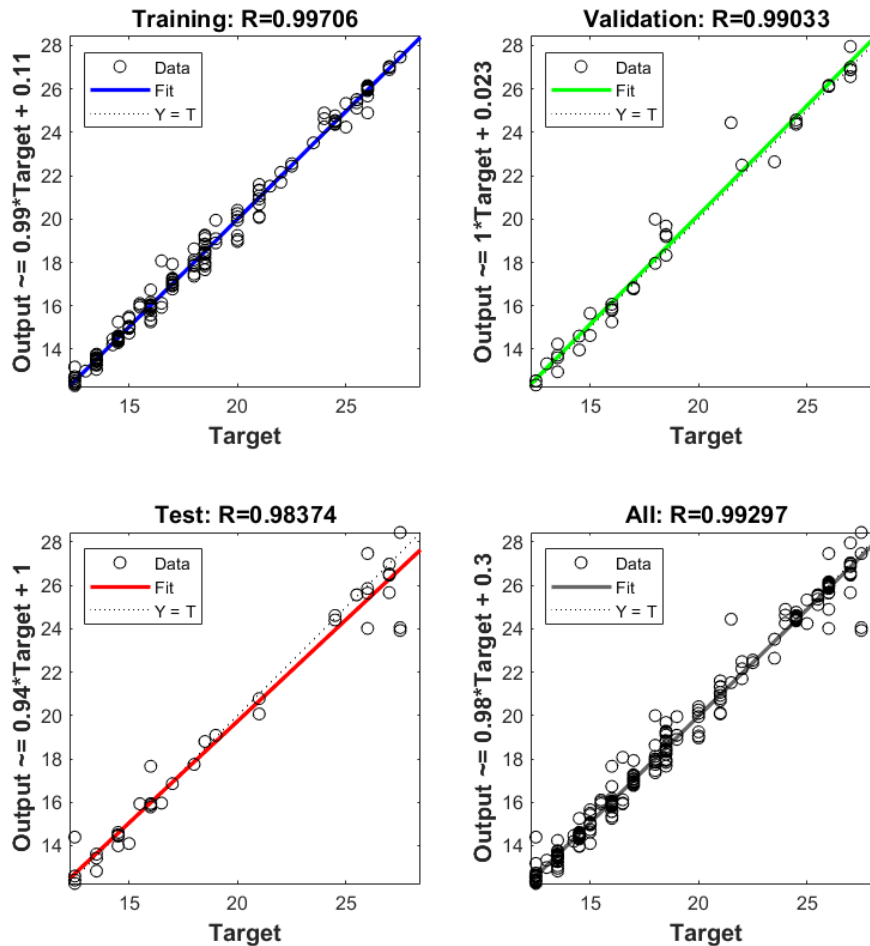


Figure 1. Regression plots for the ANN model: (a) training dataset, $R = 0.99706$; (b) validation dataset, $R = 0.99033$; (c) testing dataset, $R = 0.98374$; and (d) overall dataset, $R = 0.99297$.

Training Performance

The training dataset produced a correlation coefficient of $R = 0.99706$, indicating a very strong positive association between the ANN-predicted interest rates and the corresponding target values. The training MSE was 0.00151351, while the RMSE was 0.03890. Most observations in Figure 1a are concentrated close to the fitted regression line and the ideal $Y = T$ line, showing a high level of agreement between predicted and observed values within the training sample.

The fitted regression relationship for the training dataset is approximately

$$\text{Output} \approx 0.99(\text{Target}) + 0.11$$

The slope being close to 1 and the intercept being relatively small further indicate that the network outputs closely follow the target values across the range represented in the training data. However, the high training correlation alone is not sufficient evidence of generalization because these observations were used to estimate

the network weights and biases. The validation and testing results are therefore important for determining whether the fitted relationship extends beyond the training sample.

Validation Performance

The validation dataset produced a correlation coefficient of $R = 0.99033$, an MSE of 0.00155369, and an RMSE of 0.03941. As shown in Figure 1b, most validation observations remain close to the fitted line, although a few observations show greater deviation from the ideal $Y = T$ relationship than those observed in the training dataset.

The fitted validation relationship is approximately

$$\text{Output} \approx 1.00(\text{Target}) + 0.023$$

The slope is approximately 1, while the intercept is close to zero, indicating strong agreement between network predictions and the corresponding target values. The validation MSE increased only slightly from 0.00151351 for training to 0.00155369 for validation. Similarly, the validation RMSE of 0.03941 remained close to the training RMSE of 0.03890.

The similarity between the training and validation performances is important because a considerably lower training error accompanied by a substantially larger validation error would provide stronger evidence of overfitting.

In the present analysis, the relatively small difference between the two subsets suggests that the network retained strong predictive agreement when applied to observations that were not directly used to update the model parameters.

Test Performance

The testing dataset represented 15% of the observations and provided an additional assessment of the trained network on held-out data. The test dataset produced $R = 0.98374$, an MSE of 0.00185058, and an RMSE of 0.04301.

As illustrated in Figure 1c, most test observations remain reasonably close to both the fitted regression line and the ideal $Y = T$ line. Nevertheless, greater dispersion is visible at some of the higher target values compared with the training and validation plots. This is reflected in the slightly lower correlation coefficient and higher error values for the test subset.

The fitted relationship for the test data is approximately

$$\text{Output} \approx 0.94(\text{Target}) + 1.00$$

Although the slope deviates somewhat more from unity than those of the training and validation datasets, the test correlation of 0.98374 remains very high. The RMSE increased from 0.03890 in the training sample to 0.04301 in the test sample. This modest increase indicates some deterioration in predictive agreement when the network was applied to held-out observations, but the difference is relatively small.

Overall Regression Performance

When the observations were considered collectively, the ANN produced an overall correlation coefficient of $R = 0.99297$. The overall MSE and RMSE were 0.00108 and 0.03286, respectively. Figure 1d shows that the majority of observations are located close to the fitted regression line, although some departures are visible, particularly toward the higher end of the target range.

The overall fitted relationship is approximately

$$\text{Output} \approx 0.98(\text{Target}) + 0.30$$

The slope of 0.98 is close to unity, indicating that the predicted values generally changed in close correspondence with the targets. The overall value of $R = 0.99297$ therefore indicates a very strong positive linear association between observed and predicted values.

It is important, however, to distinguish R from the coefficient of determination, R^2 . The MATLAB regression plot reports a correlation coefficient. Consequently, $R = 0.99297$ should not be interpreted directly as meaning that the ANN explained 99.3% of the variance in interest rates. If an R^2 statistic is required, it should be calculated explicitly from the observed and predicted values. In the present study, the reported statistic is therefore interpreted strictly as evidence of a very strong positive correlation between targets and network outputs.

Table 1. ANN performance across the training, validation, testing, and overall datasets

Dataset	R	MSE	RMSE
Training	0.99706	0.00151351	0.03890
Validation	0.99033	0.00155369	0.03941
Testing	0.98374	0.00185058	0.04301
Overall	0.99297	0.00108	0.03286

Overall, the relatively small differences among the training, validation, and testing statistics suggest that the strong relationship observed during model fitting was not restricted exclusively to the training observations.

Training State of the Neural Network

The training-state plots provide additional information on how the network parameters changed during optimization. Figure 2 presents the gradient, the Levenberg-Marquardt damping parameter (μ), and the validation-check sequence throughout the 58 training epochs.

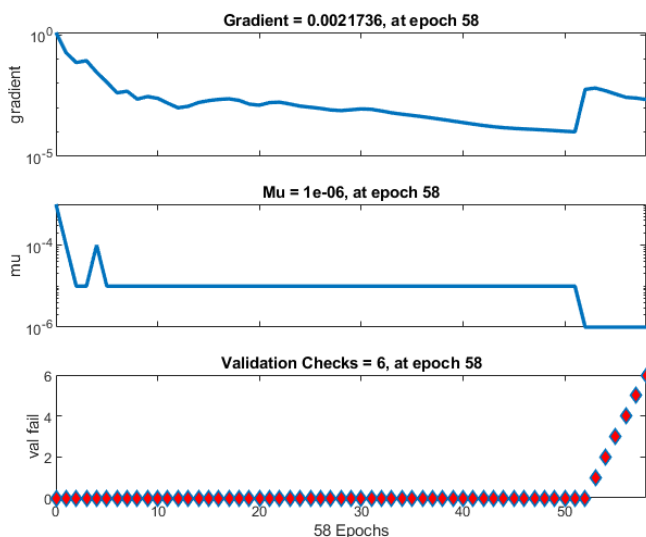


Figure 2. ANN training-state plots showing (a) gradient, (b) Levenberg-Marquardt damping parameter (μ), and (c) validation checks across 58 epochs.

Gradient

The gradient decreased substantially during the early stages of training and continued to decline as training progressed. At epoch 58, the reported gradient was 0.0021736. The gradient reflects the sensitivity of the

network error to changes in its weights and biases. A reduction in gradient magnitude generally indicates that the parameter adjustments required to reduce the error are becoming smaller. Figure 2a also shows a temporary increase in the gradient near the end of training before it declined again. This indicates that the optimization path was not perfectly monotonic. Such behavior may occur because the Levenberg-Marquardt algorithm dynamically adjusts its parameter-update behavior according to changes in the error surface. Therefore, the final gradient should be interpreted as an indicator of the state of optimization at termination rather than as proof that the network reached a global optimum.

Levenberg-Marquardt Damping Parameter

Figure 2b shows the behavior of the Levenberg-Marquardt damping parameter, μ , during training. At epoch 58,

$$\mu = 1 \times 10^{-6}$$

As explained in the Methodology, μ determines the balance between the Gauss-Newton and gradient-descent-like behaviors of the Levenberg-Marquardt algorithm. The parameter-update equation is

$$\theta_{k+1} = \theta_k - (J^T J + \mu I)^{-1} J^T e$$

When μ is relatively small, the parameter update approaches the Gauss-Newton method, whereas larger values produce more gradient-descent-like behavior. The small value of μ recorded near the end of training therefore indicates that the optimization was operating closer to the Gauss-Newton region of the algorithm. The plot also shows that μ changed during the initial training epochs before remaining relatively stable for much of the optimization and subsequently decreasing to 10^{-6} . This adaptive behavior is a defining feature of the Levenberg-Marquardt procedure.

Validation Checks

The validation-check plot in Figure 2c shows that no consecutive validation failures were recorded during most of the training process. After the network reached its best validation performance, however, validation performance failed to improve for six successive checks. The validation-failure count therefore reached six at epoch 58. The six consecutive validation failures constituted the stopping condition for the training procedure. This early-stopping mechanism is useful for limiting continued optimization of the training error after validation performance has ceased to improve. Consequently, the termination of training at epoch 58 was determined by the validation criterion rather than solely by continued reductions in training error.

Performance of the Network Across Training Epochs

The evolution of the training, validation, and testing MSE values is shown in Figure 3.

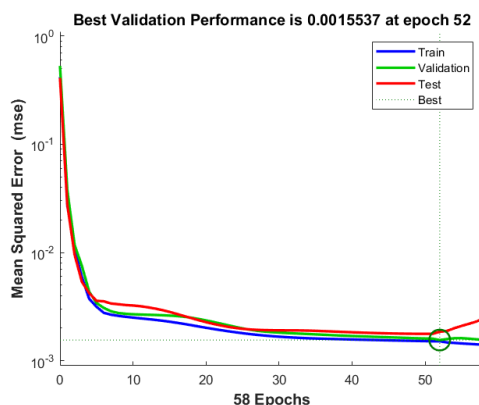


Figure 3. Mean squared error for the training, validation, and testing datasets across 58 epochs, showing the best validation performance of 0.0015537 at epoch 52.

The performance curves indicate that MSE decreased rapidly during the first several epochs. The steep initial decline demonstrates that the network learned a substantial part of the input-output relationship during the early stages of optimization. Thereafter, reductions in error became progressively smaller, with the three curves approaching relatively stable levels.

The best validation performance was

$$\text{MSE}_{\text{validation}} = 0.0015537$$

at epoch 52. The vertical marker in Figure 3 identifies the epoch at which this minimum validation error occurred. Although the training error continued to decrease slightly after epoch 52, the validation error did not improve further. The test error also began to increase toward the final epochs.

This divergence near the end of training illustrates the importance of validation-based stopping. Continued optimization of the training dataset beyond the best validation point could improve training fit while failing to improve, or potentially reducing, generalization performance. The six subsequent validation failures caused training to terminate at epoch 58.

The relationship between Figures 2 and 3 is therefore internally consistent. The best validation performance occurred at epoch 52, followed by six consecutive validation failures:

$$52 + 6 = 58$$

which corresponds to the reported stopping epoch. The performance curve therefore provides evidence that validation performance was actively monitored during network training rather than selecting a final model solely on the basis of training error.

Overall Error Measures

In addition to the subset-specific performance measures, the overall ANN predictions were evaluated using MSE, RMSE, MAE, WAPE, and the correlation coefficient. The results are summarized in Table 2.

Table 2. Overall ANN performance measures

Performance measure	Value
MSE	0.00108
RMSE	0.03286
MAE	0.3000
WAPE	4.24%
R	0.99297

The MSE of 0.00108 represents the average squared prediction error on the scale used to calculate network performance. Similarly, the RMSE of 0.03286 represents the square root of the mean squared error. Because normalization was applied during ANN modeling, error statistics derived from the normalized network outputs should be interpreted with respect to the scale on which they were calculated.

The reported MAE was 0.3000. MAE measures the average magnitude of prediction errors without considering their direction and therefore provides an intuitive measure of the typical absolute discrepancy between observed and predicted values.

The WAPE was 4.24%, calculated as

$$\text{WAPE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i|} \times 100.$$

The value of 4.24% indicates that the total absolute prediction error was equivalent to approximately 4.24% of the aggregate magnitude of the observed values. This interpretation is preferable to describing WAPE as the average percentage error of each individual prediction because WAPE is calculated from aggregated absolute errors.

The complementary percentage,

$$100\% - 4.24\% = 95.76\%$$

explains the description of approximately 95.76% predictive accuracy in the original analysis. However, this value is simply the complement of WAPE and should not be presented as an independently estimated accuracy measure. Accordingly, the present article reports WAPE directly as the defined forecasting error statistic.

Comparison of Target and Predicted Values

Figure 4 provides a direct comparison of the normalized target values and corresponding ANN predictions across the sequence of observations.

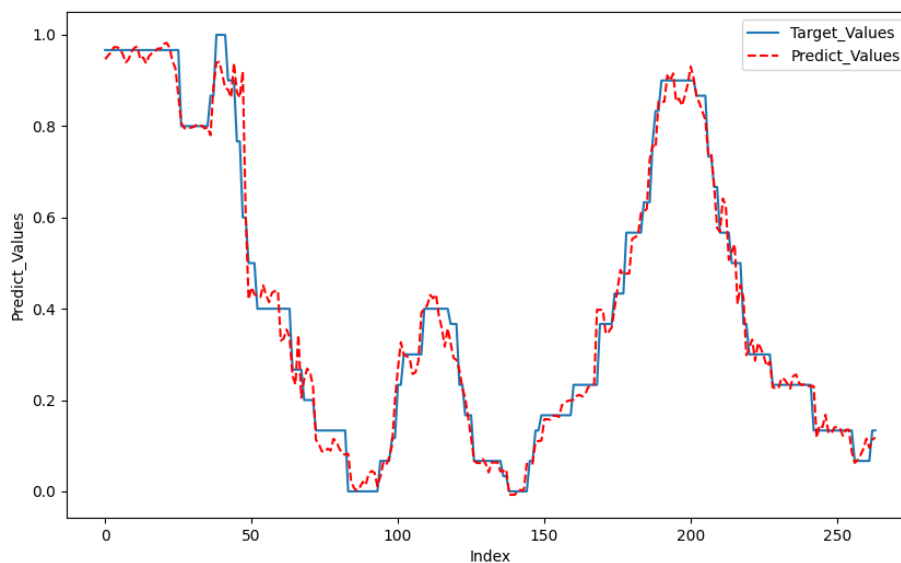


Figure 4. Comparison of normalized target and ANN-predicted interest-rate values across the observation sequence.

The predicted series closely follows the general movement of the target series. Major increases, declines, turning points, and relatively stable periods in the target values are reproduced by the ANN predictions. The close visual correspondence between the two series supports the numerical results obtained from the regression analysis.

At several points, small differences are visible between the target and predicted values, particularly around sharp transitions and local peaks or troughs. These differences indicate that the ANN does not reproduce every individual observation exactly. Nevertheless, the predicted series captures the broad structure and direction of the observed pattern across the dataset.

The ability of the ANN to follow these changes is consistent with its use of a nonlinear TANSIG hidden-layer activation function. Unlike a model constrained to a predetermined linear relationship, the ANN can represent nonlinear mappings between the selected macroeconomic predictors and the target interest rate. The strong correspondence observed in Figure 4 therefore suggests that GDP, inflation, unemployment, and money supply collectively contained information associated with the interest-rate patterns represented in the study dataset.

The result is consistent with previous applications of neural networks to interest-rate and yield forecasting. Oh and Han (2000), for example, demonstrated the usefulness of neural-network-based methods in interest-rate forecasting, while Wegener et al. (2016) applied neural networks to government bond-yield forecasting. Nunes et al. (2019) similarly demonstrated the applicability of artificial neural networks to yield-curve forecasting. These studies support the broader proposition that nonlinear learning methods can be useful when modeling complex financial relationships.

The present results should not, however, be interpreted as evidence that the ANN is necessarily superior to conventional forecasting methods. Because no competing ARIMA, VAR, linear regression, or other benchmark model was estimated using the same Ghanaian dataset, the results do not establish comparative superiority over alternative forecasting approaches.

This qualification is consistent with the wider forecasting literature. Neural networks can approximate complex nonlinear relationships, but their forecasting performance depends on the structure of the data, model specification, training procedure, and evaluation design. Foster et al. (1992), for example, showed that neural networks do not necessarily outperform simpler methods when dealing with short and noisy time series. The strong performance obtained in the present study should therefore be viewed as evidence that the selected ANN architecture was capable of learning a close relationship between the selected predictors and Ghanaian interest rates rather than as evidence that ANN models are universally superior.

Overall Interpretation and Study Limitations

Taken together, the regression plots, training-state diagnostics, performance curve, error measures, and target-versus-prediction plot provide consistent evidence that the fitted ANN achieved strong predictive agreement within the dataset used in this study. Correlation coefficients remained high across the training ($R = 0.99706$), validation ($R = 0.99033$), and testing ($R = 0.98374$) datasets, while the overall correlation was $R = 0.99297$. The relatively small difference between training and testing performance provides evidence that the fitted relationship was not limited exclusively to the training observations.

The training-state analysis further showed that optimization proceeded until the validation-based stopping criterion was reached. The best validation MSE of 0.0015537 occurred at epoch 52, followed by six consecutive validation failures and termination at epoch 58. At the final epoch, the gradient was 0.0021736 and the Levenberg-Marquardt damping parameter was 1×10^{-6} .

Despite these strong results, several limitations should be considered when interpreting the findings. First, the study reports observations covering 2000–2021 and describes the dataset as monthly; however, the documentation does not clearly specify how macroeconomic variables that may originally be available at different frequencies, particularly GDP and unemployment, were transformed or aligned to the monthly series. A complete reproducibility record should therefore document the frequency-harmonization procedure.

Second, although the observations were divided into training, validation, and testing subsets using a 70:15:15 allocation, the available documentation does not clearly establish whether the split was performed randomly or chronologically. This distinction is particularly important for time-dependent economic data. A random holdout assesses performance on unseen observations drawn from the historical dataset, whereas a chronological holdout more closely represents genuine forecasting in which earlier information is used to predict later periods.

Third, normalization was performed before network training, but the available methodological record does not clearly establish whether the minimum and maximum values used for scaling were obtained exclusively from the training dataset. For a strict out-of-sample forecasting design, preprocessing parameters should be estimated from the training data and then applied unchanged to validation and testing observations.

Finally, the study evaluated a single ANN framework without estimating benchmark forecasting models using the same observations. Future work should therefore compare the ANN against conventional time-series and econometric approaches, such as ARIMA, vector autoregression, and regression-based models, as well as alternative machine-learning architectures. Chronological holdout evaluation or rolling-origin forecasting could also provide a stronger assessment of out-of-sample predictive performance.

Subject to these limitations, the combined numerical and graphical evidence indicates that the feed-forward ANN successfully learned a strong relationship between the selected macroeconomic indicators and Ghanaian interest rates within the study dataset. The findings therefore support the usefulness of ANN-based methods as a potential tool for interest-rate modeling while also highlighting the importance of rigorous out-of-sample validation and direct benchmark comparisons in future forecasting research.

CONCLUSION

This study evaluated the application of an artificial neural network (ANN) for modeling Ghana's interest rates using four macroeconomic indicators: gross domestic product (GDP), inflation, money supply, and unemployment. Historical data covering the period from 2000 to 2021 were used to develop a two-layer feed-forward neural network trained with the Levenberg–Marquardt backpropagation algorithm. The model was evaluated using training, validation, and testing datasets together with several performance measures, including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), weighted absolute percentage error (WAPE), and the correlation coefficient (R).

The results demonstrated strong agreement between the observed and ANN-predicted interest-rate values. Correlation coefficients of 0.99706, 0.99033, and 0.98374 were obtained for the training, validation, and testing datasets, respectively, while the overall correlation coefficient was 0.99297. The overall MSE and RMSE were 0.00108 and 0.03286, respectively, while an MAE of 0.3000 and WAPE of 4.24% were recorded. The high correlation coefficients across the three data subsets, together with the relatively small differences between the training, validation, and testing errors, indicate that the network learned a strong relationship between the selected macroeconomic variables and interest rates within the dataset examined.

The training diagnostics further showed that the best validation performance occurred at epoch 52, with an MSE of 0.0015537. Training subsequently terminated at epoch 58 after six consecutive validation failures. At termination, the gradient was 0.0021736 and the Levenberg–Marquardt damping parameter was 1×10^{-6} . These results, together with the close correspondence between the target and predicted series, demonstrate the ability of the fitted feed-forward ANN to represent the relationships present in the study data.

Nevertheless, the findings should be interpreted within the limitations of the modeling design. Although the dataset was described as monthly, the procedure used to harmonize macroeconomic variables that may originally have been available at different frequencies was not fully documented. In addition, the available methodological record does not clearly establish whether the 70:15:15 training-validation-test division was chronological or random, or whether normalization parameters were estimated exclusively from the training data. These considerations are particularly important when assessing genuine out-of-sample forecasting performance for economic time-series data. Furthermore, because no alternative forecasting model was estimated on the same dataset, the present findings do not establish that the ANN is superior to conventional econometric or time-series forecasting approaches.

Future research should therefore evaluate the model using chronologically ordered holdout samples or rolling-origin forecasting and explicitly document the treatment of variables reported at different frequencies. ANN performance should also be compared directly with benchmark approaches such as ARIMA, vector autoregression (VAR), and regression-based models using identical data and evaluation periods. Further

studies could investigate alternative architectures, including recurrent neural networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, as well as systematic hyperparameter optimization and additional theoretically relevant macroeconomic predictors. Such extensions would provide stronger evidence regarding the robustness and practical forecasting value of ANN-based approaches for Ghanaian interest-rate modeling.

REFERENCES

1. Bank of Ghana. (n.d.). Our monetary policy framework. Bank of Ghana.
2. Foster, W. R., Collopy, F., & Ungar, L. H. (1992). Neural network forecasting of short, noisy time series. *Computers & Chemical Engineering*, 16(4), 293–297. [https://doi.org/10.1016/0098-1354\(92\)80049-F](https://doi.org/10.1016/0098-1354(92)80049-F)
3. Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989–993. <https://doi.org/10.1109/72.329697>
4. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
5. Hsieh, W. W. (2004). Nonlinear multivariate and time series analysis by neural network methods. *Reviews of Geophysics*, 42(1), RG1003. <https://doi.org/10.1029/2002RG000112>
6. Kovanen, A. (2011). Monetary policy transmission in Ghana: Does the interest rate channel work? (IMF Working Paper No. 11/275). International Monetary Fund. <https://doi.org/10.5089/9781463925314.001>
7. Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2), 164–168. <https://doi.org/10.1090/qam/10666>
8. Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441. <https://doi.org/10.1137/0111030>
9. Nti, A. E. (2021). Artificial neural network model for predicting exchange rate in Ghana [MPhil thesis, University of Mines and Technology]. University of Mines and Technology Institutional Repository.
10. Nunes, M., Gerding, E., McGroarty, F., & Niranjana, M. (2019). A comparison of multitask and single task learning with artificial neural networks for yield curve forecasting. *Expert Systems with Applications*, 119, 362–375. <https://doi.org/10.1016/j.eswa.2018.11.012>
11. Ogundele, O. S., Ujunwa, A., & Mohammed, A. A. (2020). A neural network scheme for monetary policy rate validation in Nigeria. *American Journal of Artificial Intelligence*, 4(2), 62–72. <https://doi.org/10.11648/j.ajai.20200402.13>
12. Oh, K. J., & Han, I. (2000). Using change-point detection to support artificial neural networks for interest rates forecasting. *Expert Systems with Applications*, 19(2), 105–115. [https://doi.org/10.1016/S0957-4174\(00\)00025-7](https://doi.org/10.1016/S0957-4174(00)00025-7)
13. Wegener, C., von Spreckelsen, C., Basse, T., & von Mettenheim, H.-J. (2016). Forecasting government bond yields with neural networks considering cointegration. *Journal of Forecasting*, 35(1), 86–92. <https://doi.org/10.1002/for.2385>