

Research article

Time Series forecasting global price of bananas using Hybrid ARIMA-NARNN model

Yeong Nain Chi^{1,*} and Orson Chi²

¹ Department of Agriculture, Food and Resource Sciences, University of Maryland Eastern Shore, Princess Anne, MD 21853, USA

² Cybersecurity Engineering Technology Program, University of Maryland Eastern Shore, Princess Anne, MD 21853, USA

* **Correspondence:** Email: ychi@umes.edu.

Abstract: This study tried to demonstrate the role of Time Series models in a modeling and forecasting process using publicly available long-term records of monthly global price of bananas during the period of January 1990 to November 2021 reported in the International Monetary Fund. Following the Box–Jenkins methodology, an ARIMA (2,1,4) with a drift model was selected as the best-fit model for the Time Series, according to its lowest AIC value. Using the Levenberg-Marquardt algorithm, the results revealed that the NARNN model with 12 neurons in the hidden layer and 6 times delays provided the best performance in the nonlinear autoregressive neural network models at its smaller MSE value. The ARIMA and NARNN models are good at modelling linear and nonlinear problems for the Time Series, respectively. However, using the HYBRID model, a combination of the ARIMA and NARNN models that has both linear and nonlinear modeling capabilities can be a better choice for modeling the Time Series. The comparative results revealed that the HYBRID model with 11 neurons in the hidden layer and 3 times delays yielded higher accuracy than the NARNN model with 12 neurons in the hidden layer and 6 times delays, and the ARIMA (2,1,4) with a drift model, according to its lowest MSE in this study.

Keywords: Time Series; global price of bananas; forecasting; ARIMA; NARNN; HYBRID ARIMA-NARNN; Levenberg-Marquardt algorithm

JEL Codes: C22, C45, C53

1. Introduction

Like COVID-19, Tropical Race 4 (TR4), also known as Panama Disease, has adversely affected the global banana industry. Recently, TR4 has suddenly accelerated, spreading from Asia to Australia, the Middle East, Africa and Latin America, where the majority of the bananas are shipped globally. Currently, TR4 is now in more than 20 countries, prompting fears of a banana pandemic and shortages of the world's favorite fruit bananas (Altendorf, 2019).

Bananas, the world's most popular fruit with different tastes, sizes, and colors, are the fourth most important food crop after wheat, rice, and maize in terms of production, and they are the world's favorite fruit in terms of consumption quantity (Ruiz et al., 2017). Excluding plantains, 22.7 million tons of bananas were traded, which was almost 20% of global production in 2017. The value of this trading was \$11 billion, which is higher than the export value of any other exported fruit (Voora and Bermudez, 2020).

Asia is the largest banana-producing region, while India and China are the two leading banana-producing countries. The Asia-Pacific is the market leader with a 61% share of global consumption, while India is the world's leading producer of bananas by accounting for nearly 25.7% of the total output. The Philippines consolidated its position as the second largest exporter of bananas behind Ecuador (Voora and Bermudez, 2020).

Latin America and the Caribbean are the largest exporting regions, responsible for approximately 80% of global exports. The global banana exports were estimated at 23.3 million tons in 2018, and Ecuador accounted for 24.7% of these global exports, as the largest exporter of bananas. Belgium, Costa Rica, and Columbia were the other top banana exporters in the world, while the United States was the leading importer of bananas with an 18% share of the world's imports (Voora and Bermudez, 2020).

The retail value of the banana sector was estimated to be between \$20 billion and \$25 billion in 2016 (Voora and Bermudez, 2020). According to the Market Reports World (2019), the global banana sector should experience a compound annual growth of 1.21% in consumption from 2019 to 2024, reaching a global consumption volume of 136.0 million tons by 2025, compared to 116.2 million tons in 2017.

According to the U.S. Bureau of Labor Statistics, the average consumer price of bananas (all fresh, traditional, first-quality organic and non-organic yellow bananas) was \$0.53 (per pound on average in U.S.) with a standard deviation of \$0.06 (minimum: \$0.40, maximum: \$0.64, and median: \$0.51). Furthermore, the average consumer price for bananas was 39.91% higher in 2020 versus 1990 (a \$13.29 difference in value). Between 1990 and 2020, bananas costing \$33.30 in 1990 cost \$46.59 in 2020 for an equivalent purchase. Compared to the overall inflation rate of 2.30% during this same period, inflation for bananas was lower.

Time Series forecasting a model to predict future values based on previously observed values is one of the most applied data science techniques in business. This model is used extensively in finance, supply chain-management, production, and inventory planning. It has a well-established theoretical grounding in statistics. Several articles have used Time Series techniques to forecast banana production (Hamjah, 2014; Hossain et al., 2016; Eyduran et al., 2020), but only a few studies have focused on banana price forecasting (Omar et al., 2014; Fatin et al., 2020).

Neural networks have become one of the most popular trends in machine learning for Time Series modeling and forecasting. Recently, there is increasing interest in using neural networks to model and forecast banana harvest yields (Rathod et al., 2017; Rathod and Mishra, 2018; Rebordera and Fajardo, 2019). Despite the importance of banana demand and supply in the global markets, there is still a lack

of studies in the technical literature available on global banana price forecasting schemes. Hence, the primary purpose of this study was to apply an autoregressive integrated moving average (ARIMA) and nonlinear autoregressive neural network (NARNN) hybrid (HYBRID) model to forecast global price of bananas during the period of January 1990 to November 2021. The findings in this study may be able to bridge an important gap in Time Series forecasting by combining the best statistical and machine learning methods. The rest of the paper is made up of the following sections: Section 2 introduces Time Series data and ARIMA, NARNN, and HYBRID models for this study, Section 3 presents and analyzes the empirical results, and finally Section 4 concludes the study.

2. Materials and methods

2.1. Materials

The long-term records of the monthly global price of bananas (units: U.S. dollars per metric ton, not seasonally adjusted) from January 1990 to November 2021 (Figure 1) are available to the public from the International Monetary Fund, retrieved from FRED, Federal Reserve Bank of St. Louis (<https://fred.stlouisfed.org/series/PBANSOPUSDM>). The average monthly global price of bananas was \$726.15 per metric ton with a standard deviation of \$281.88 (minimum: \$250.51, maximum: \$1,298.34, and median: \$659.98).

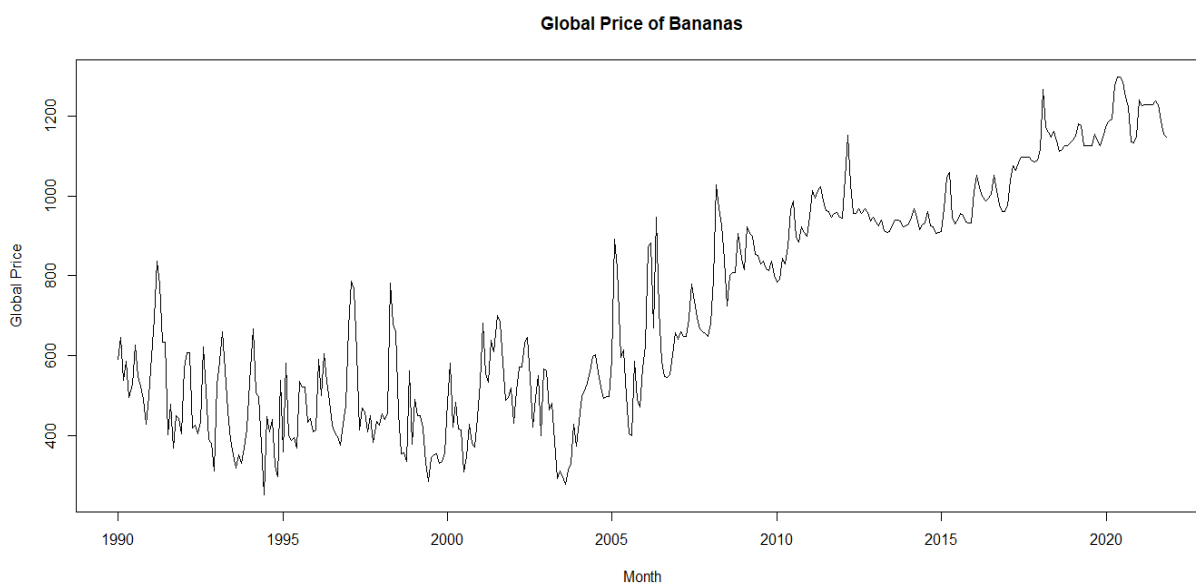


Figure 1. Time Series plot of global price of bananas, January 1990–November 2021 (R output).

2.2. ARIMA model

Time Series involves data collected sequentially in time. In a univariate Time Series, it is a set of single (scalar) observations recorded at a specific time t . The sequence of random variables $\{y_t: t = 1, 2, \dots, T\}$, $y_t \in \mathbb{R}$ where $t \in T$ is the time indexing when the data was observed, is called a stochastic

process. This stochastic process is often used in modeling Time Series data to find the parameters of the Time Series, and then use them as a model in predicting future values of the Time Series.

Autoregressive Integrated Moving Average (ARIMA) is a statistical analysis model for modeling and forecasting Time Series data to predict future trends. The purpose of each of these parts is to make the model fit better to predict future points in the Time Series (Montgomery et al., 2008). Statistically, ARIMA (p, d, q) model can be expressed as:

$$\begin{aligned} y_t &= \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \\ &= \sum_{i=1}^p \phi_i y_{t-i} - \sum_{j=1}^q \theta_j e_{t-j} + e_t \end{aligned} \quad (1)$$

where p = the order of the autoregressive process (the number of lagged terms),
 d = the number of differences required to make the Time Series stationary,
 q = the order of the moving average process (the number of lagged terms),
 $\phi = (\phi_1, \phi_2, \dots, \phi_p)$ is the vector of model coefficients for the autoregressive process,
 $\theta = (\theta_1, \theta_2, \dots, \theta_q)$ is the vector of model coefficients for the moving average process, and
 e_t = the residual error (i.e., white noise).

In backshift notation B , “backshift operator” or “lag operator”, is a useful notational device when working with Time Series lags: $By_t = y_{t-1}$ (means “back up by one time unit”) and $B^k y_t = y_{t-k}$ (means “backshift k times”). Thus, the ARIMA (p, d, q) model can be expressed in backshift notation as:

$$\phi_p(B) (1 - B)^d y_t = c + \theta_q(B) e_t \quad (2)$$

where $\phi_p(B) = (1 - \phi_1 B - \dots - \phi_p B^p) = (1 - \sum_{i=1}^p \phi_i B^i)$, $\theta_q(B) = (1 - \theta_1 B - \dots - \theta_q B^q) = (1 - \sum_{j=1}^q \theta_j B^j)$, c is a constant, and e_t is the error term.

In Time Series forecasting, the Box–Jenkins methodology (Box & Jenkins, 1970) refers to a systematic method of identifying, estimating, checking, and forecasting ARIMA models (Box et al., 2016). It is used as the process to forecast the ARIMA model in this study based on its autocorrelation function (ACF) and partial autocorrelation function (PACF) as a means of determining the stationarity of the univariate Time Series and the lag lengths.

In a Time Series, a common assumption is that the Time Series is stationary, which means that the statistical properties (i.e., mean and variance) of the process do not change over time. Therefore, the Box-Jenkins methodology starts with the assumption that the Time Series should be as on stationary status. Differencing can be used if the Time Series is not stationary. Empirically, plots and summary statistics can be used to identify trends and autoregressive elements to get an idea of the amount of differencing and the size of the lag that will be required for model identification.

In order to figure out good parameters for the Time Series model, the Akaike’s Information Criterion (AIC) or the Bayesian Information Criterion (BIC) can be used to determine the orders of an ARIMA model when its minimized value is attempted. In the diagnostic checking step, plots and statistical tests of the residual errors can also be used to determine the model fitting, to evaluate the model fitting in the context of the Time Series data, and to check where the model can be improved.

2.3. Nonlinear autoregressive neural network (NARNN) model

The idea behind the autoregressive (AR) process is to explain the present value of the Time Series, y_t , by a function of p past values, $(y_{t-1}, y_{t-2}, \dots, y_{t-p})$. Thus, the AR process of order p , $AR(p)$, is defined by the equation:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t = \sum_{i=1}^p \phi_i y_{t-i} + \epsilon_t \quad (3)$$

where $\phi = (\phi_1, \phi_2, \dots, \phi_p)$ is the vector of model coefficients for the autoregressive process, and ϵ_t is white noise, $\epsilon_t \sim N(0, \sigma^2)$.

The NARNN is a natural generalization of the classic linear AR(p) process. The NARNN of order p can be expressed as:

$$y_t = \Phi(y_{t-1}, y_{t-2}, \dots, y_{t-p}, w) + \epsilon_t \quad (4)$$

where $\Phi(\cdot)$ is an unknown function determined by the neural network structure and connection weights, w is a vector of all parameters (weights), and ϵ_t is the error term. Thus, it performs a nonlinear functional mapping from the past observations, $(y_{t-1}, y_{t-2}, \dots, y_{t-p})$, to the future value, y_t , which is equivalent to a nonlinear autoregressive model (Zhang, 2003).

With the Time Series data, lagged values of the Time Series can be used as inputs to a neural network, which is called the NARNN model. Mathematically, the NARNN model (Benrhmach et al., 2020) can be written by the equation of the form as:

$$y_t = a_0 + \sum_{j=1}^k w_j \Phi(b_{0j} + \sum_{i=1}^d w_{ij} y_{t-i}) + \epsilon_t \quad (5)$$

where d = the number of input units,

k is the number of hidden units,

a_0 is the constant corresponding to the output unit,

b_{0j} is the constant corresponding to the hidden unit j ,

w_j is the weight of the connection between the hidden unit j and the output unit,

w_{ij} is the parameter corresponding to the weight of the connection between the input unit i and the hidden unit j , and $\Phi(\cdot)$ is a nonlinear function, so-called this the transfer (activation) function. The logistic function (i.e., sigmoid) is commonly used as the hidden layer transfer function, that is, $\Phi(y) = 1 / (1 + \exp(-y))$.

The most common learning rules for the NARNN model are the Levenberg-Marquardt (LM), Bayesian Regularization, and Scaled Conjugate Gradient training algorithms. In this study, the LM algorithm was considered, because it works without computing the exact Hessian matrix, which increases the training speed, and it has stable convergence (Gavin, 2020). The LM algorithm, first published by Levenberg (1944) and then rediscovered by Marquardt (1963), has become a standard technique for nonlinear least-squares problems. The LM algorithm is an iterative technique that locates the minimum of an objective function $F(x)$ that is expressed as the sum of squares of nonlinear functions (Madsen et al., 2004):

$$F(x) = (1/2) \sum_{i=1}^n [f_i(x)]^2 \quad (6)$$

Furthermore, the LM algorithm steps to search the direction of the iteration given by the solution φ_i to the equations,

$$(J_i^T J_i + \lambda_i I) \varphi_i = -J_i^T f_i \quad (7)$$

where J_i is the Jacobian of f_i , I is the identity matrix, and λ_i are the non-negative scalars, called the combination coefficient.

In the LM algorithm, for some scalar $\Delta > 0$ related to λ_i , the vector φ_i is the solution of the constrained sub-problem of minimizing $(1/2) \|J_i \varphi + f_i\|_2^2$ subject to $\|\varphi\|_2 \leq \Delta$ (Gill et al., 1981, p. 136).

2.4. ARIMA-NARNN Hybrid (HYBRID) model

The ARIMA and NARNN models are good at modeling linear and nonlinear problems for the Time Series, respectively. The Hybrid model, a combination of the ARIMA and NARNN models has both linear and nonlinear modelling capabilities, can be a better choice for modelling the Time Series. Assuming that an unknown function can be used to demonstrate the relationship between linear and nonlinear components in the Time Series, that relationship can be illustrated as follows:

$$y_t = f(L_t, N_t) \quad (8)$$

where the linear component is represented by L_t , and the nonlinear component is shown by N_t . Assuming that the linear and nonlinear components in the Time Series have simple additive relationships. Zhang (2003) states that the Time Series can be considered as a combination of the linear and nonlinear components as follows:

$$y_t = L_t + N_t \quad (9)$$

First, the linear component will be modelled by the ARIMA model. Then, the residuals from the ARIMA model will have only the nonlinear relationship, which can be obtained by taking the difference of the observed values and the predicted values as follows:

$$e_t = y_t - \hat{L}_t \quad (10)$$

where e_t is the residual of the linear model at time t , and \hat{L}_t is the predicted value for time t . To find the nonlinear relationship, residuals can be modelled by the NARNN model in this study as follows:

$$\hat{N}_t = e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (11)$$

where f is the transformation function modeled by the NARNN model, and ε_t is the random error. The forecasts from the ARIMA and NARNN models are combined to obtain the forecast of the Time Series \hat{y}_t , which is denoted by:

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \quad (12)$$

3. Results

3.1. ARIMA model

R 4.0.2 for Windows, an open source for statistical computing and graphics supported by the R Foundation for Statistical Computing, was used as the tool to model and forecast global price of bananas during the period of January 1990 to November 2021 for this study. The function “decompose” in R can be applied to estimate the seasonal, trend and irregular components of a seasonal Time Series. The results revealed that the estimated trend component showed a steady increase over time, and the estimated seasonal component definitely displayed seasonality, with a pattern recurrence occurring once every 12 months (yearly).

Seasonal adjustment is the estimation and removal of seasonal effects that are not explainable by the dynamics of trends or cycles from a Time Series to reveal certain non-seasonal features. This can be done by subtracting the estimated seasonal component from the original Time Series. After removed

the seasonal variation, the seasonally adjusted Time Series only contained the trend component and an irregular component.

The ACF of the Time Series, seasonal adjusted global price of bananas from January 1990 to November 2021, showed a strongly positive correlations at up to 26 lags that never decay to zero. Meanwhile, the test statistic of the Augmented Dickey-Fuller Test was Dickey-Fuller = -3.1414 with lag order = 7, and the p-value of the test was 0.09806. This suggested that the Time Series was non-stationary and needed to be differenced. This was also illustrated by the single spike at the first lag followed by small apparently random values after the first lag for the PACF. Since the PACF cut off after the first lag, it seems that the Time Series followed the autoregressive (AR) process.

In Time Series analysis, differencing can be used to transform a non-stationary Time Series into a stationary one. When both trend and seasonality are present, both a non-seasonal first difference and a seasonal difference should be applied separately. According to the Augmented Dickey-Fuller Test, Dickey-Fuller = -10.352 with lag order = 7, and the p-value of the test was smaller than 0.01. It rejected the null hypothesis that was non-stationary, and suggested that the first difference of the Time Series was stationary. The ACF of the first differenced Time Series showed a significant positive spike at the first lag, followed by correlations that were statistically significant. The corresponding PACF of the first differenced Time Series showed a most likely gradual decrease after the first few negative lags. Since the ACF cut off after the first lag and the PACF decreased gradually, a reasonable conclusion was that the first differenced Time Series followed the moving average (MA) process.

In order to find a solution, the function “`auto.arima()`” from the “`forecast`” package in R 4.0.2 for Windows was employed to identify both the structure of the Time Series (stationary or not) and type (seasonal or not), and sets the model's parameters, that takes into account the AIC, AICc, or BIC values generated to determine the best fit ARIMA model. Consequently, the ARIMA (2,1,4) with a drift model was selected to be the best fit model for the Time Series, seasonal adjusted global price of banana from January 1990 to November 2021, according to the lowest AIC value (= 4313.15) in this study, and the parameters of the ARIMA (2,1,4) with a drift model were presented in Table 1.

Table 1. Parameters of ARIMA (2,1,4) with Drift Model.

Parameter	Estimate	Standard Error
Constant	1.7817	0.8797
AR Lag 1	-0.3986	0.1539
AR Lag 2	0.1309	0.1281
Difference		
MA Lag 1	0.1042	0.1484
MA Lag 2	-0.3869	0.1159
MA Lag 3	-0.1057	0.0822
MA Lag 4	-0.2921	0.0564
Sigma ² estimated as 4751, Log Likelihood = -2148.58		
AIC = 4313.15, AICc = 4313.54, BIC = 4344.72		
Training Set Error Measures:		
RMSE = 66.89923, MSE = 4475.50697, MAE = 47.63762, MAPE = 8.500749		

Source: own work

A common task when building a forecasting model is to check that the residuals satisfy some assumptions that they are uncorrelated, normally distributed, etc. The top figure of Figure 2 shows that the residuals from the ARIMA (2,1,4) with a drift model did not violate the assumption of constant location and scale. The bottom right figure of Figure 2 shows that the residual histogram did not reveal a series deviation from normality in this case. The ACF plot of the residuals (the bottom left figure of Figure 2) also shows that all sample autocorrelations were within the threshold limits, indicating that the residuals appeared to be random.

The Ljung-Box Q-test (Ljung and Box, 1978) is a diagnostic tool used to test the lack of fit of a Time Series model. In this case, the test statistic of the Ljung-Box Q-test was $Q = 20.292$ with 17 degrees of freedom, and the p-value of the test was 0.2596 with model degrees of freedom: 7 and total lags used: 24, indicating that the residuals were random and that the model provided an adequate fit to the Time Series. Combined with the Ljung-Box Q-test statistic, this suggested that the ARIMA (2,1,4) with a drift model appropriately modeled the dynamics for this Time Series. Forecasting process with the ARIMA (2,1,4) with a drift model also indicated a good fit of the model for forecasting at a constant increasing rate in the short-term (Figure 3).

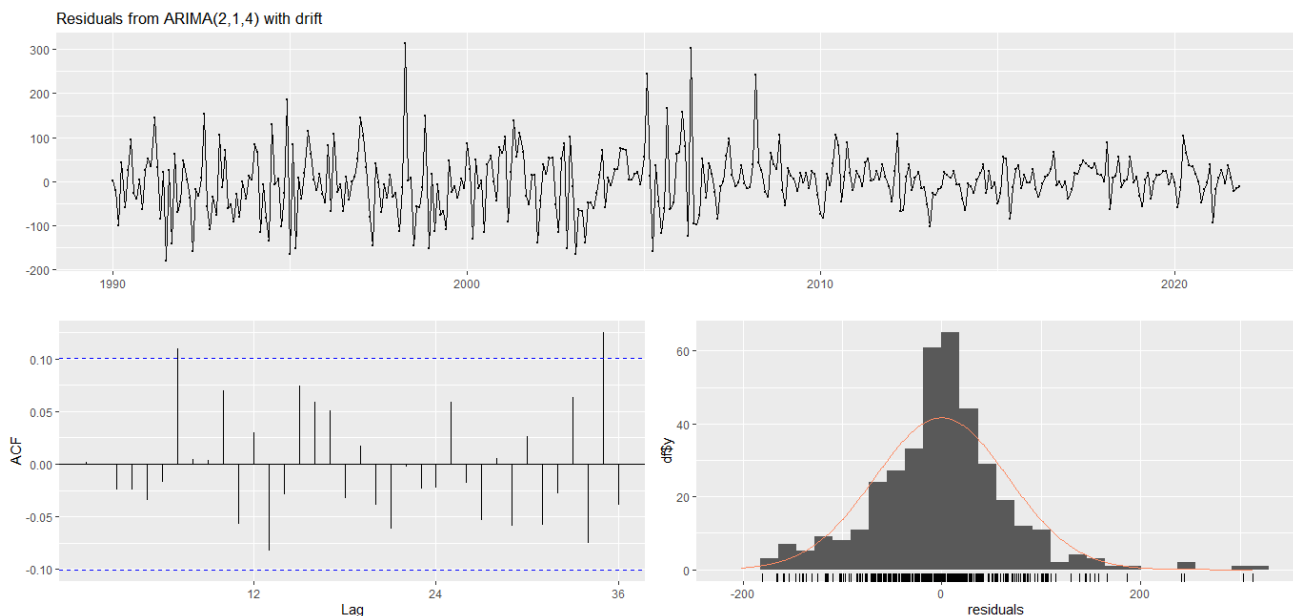


Figure 2. Residuals from ARIMA (2,1,4) with Drift Model (R output).

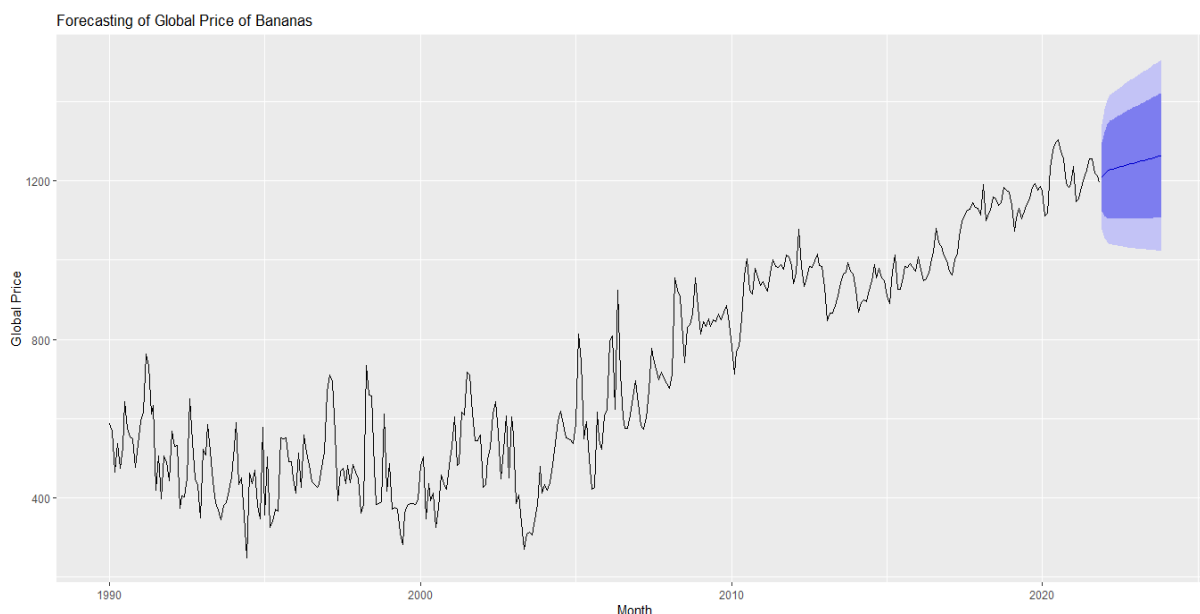


Figure 3. Observed and Forecasted Global Price of Bananas (R output).

3.2. Nonlinear autoregressive neural network (NARNN) model

In MATLAB, the NARNN model applied to Time Series prediction using its past values of a univariate Time Series can be expressed as follows:

$$y(t) = \Phi(y(t-1), y(t-2), \dots, y(t-d)) + e(t) \quad (13)$$

where $y(t)$ is the Time Series value at time t , d is the time delay, and $e(t)$ is the error of the approximation of the Time Series at time t . This equation describes how the NARNN model is used to predict the future value of a Time Series, $y(t)$, using the past values of the Time Series, $(y(t-1), y(t-2), \dots, y(t-d))$. The function $\Phi(\cdot)$ is an unknown nonlinear function, and the training of the neural network aims to approximate the function by means of the optimization of the network weights and neuron bias. This tends to minimize the sum of the squared differences between the observed (y_i) and predicted (\hat{y}_i) values (i.e., $MSE = (1/n) \sum_{i=1}^n (y_i - \hat{y}_i)^2$) (Beale et al., 2019).

In this study, the NARNN model was applied to model and forecast the Time Series, global price of bananas from January 1990 to November 2021. Furthermore, the logistic sigmoid and linear transfer functions at the hidden and output layers were used respectively. The number of hidden neurons and the number of delays was set experimentally after a data pre-processing and analysis stage. The extracted features were trained using the LM training algorithm for the target Time Series in the MATLAB (2019a) Neural Network Toolbox: 383 timesteps of one element, global price of bananas from January 1990 to November 2021.

The training target timesteps are presented to the network during training, and the network is adjusted according to its error. The validation target timesteps are used to measure network generalization and to halt training when generalization stops improving. The testing target timesteps have no effect on training and so provide an independent measure of network performance during and after training (Beale et al., 2019). The division of the Time Series in this analytical work was 70% for

the training, 15% for the validation, and 15% for the testing. Randomly, 383 data samples were divided into 269 data for the training, 57 data for the validation, and 57 data for the testing

The development of the optimal architecture for the NARNN model requires determination of time delays, the number of hidden neurons, and an efficient training algorithm. The optimum number of time delays and hidden neurons were obtained through a trial-and-error procedure. Furthermore, the LM algorithms were employed for training of the NARNN model and its performance was evaluated under the optimal neural network structure. The prediction performance of the models was evaluated by its MSE. The error analysis showed that the NARNN model with 12 neurons in the hidden layer and 6 times delays provided the best performance (MSE = 3058.11185) using the LM algorithm (Table 2).

Table 2. NARNN model selection using Levenberg-Marquardt algorithm.

Number of hidden neurons	Number of delays (d)				
	2	3	4	5	6
8	5495.27669	4894.51856	5619.10430	4526.92313	3947.92018
9	4957.78393	4824.05006	5716.89865	3964.32643	3212.59409
10	5179.43173	5084.66744	4904.24470	4626.09369	6040.92292
11	6407.52867	4092.28144	4162.03297	4210.05822	3305.64168
12	5142.34650	4757.30244	4276.64426	3266.47802	3058.11185

Source: Author's work

3.2.1. NARNN training output

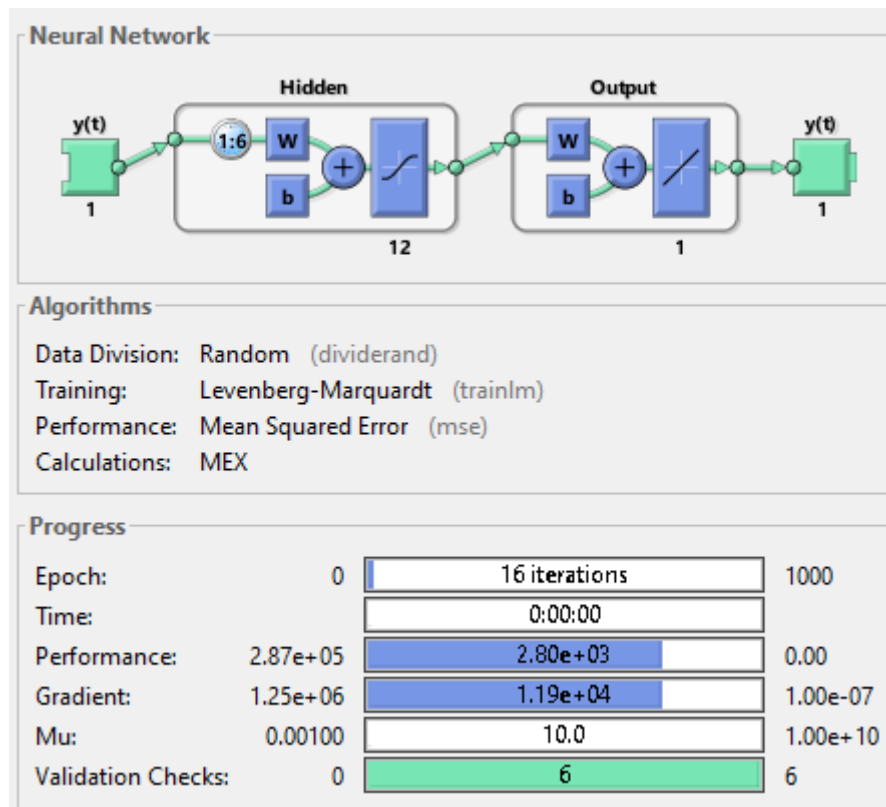


Figure 4. NARNN training output (MATLAB output).

In order to train the NARNN, open loop architecture (Figure 4) shows a block diagram of the NARNN generated during MATLAB processing in the MATLAB (2019a) Neural Network Toolbox. Figure 3 displays the training progress using the LM algorithm stopped when the validation error increased for six iterations with Performance = 2800, Gradient = 11862.397, and Mu = 10.00 at epoch 16. In terms of processing time, the LM algorithm took 0:00:00 during training. The term epoch represents the number of iterations during training in which it is attempted to minimize the error function. The LM algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples (Beale et al., 2019).

3.2.2. NARNN best performance

The performance plot illustrated the relationship between the training, validation, and testing phases in forecasting global price of bananas, in terms of MSE versus the number of epochs. The performance was evaluated by taking MSE and epochs after the training was completed, and then the values were generated. The performance plot is a useful diagnostic tool to plot the training, validation, and testing errors to check the progress of training. It also illustrated that the training stopped when the validation error increased at the circled epoch. As illustrated in Figure 5, the best performance for the validation phase was 7321.0222 at epoch 10 for the NARNN model. The results showed a good network performance because the validation error and testing error have similar characteristics, and it did not appear that any significant overfitting had occurred.

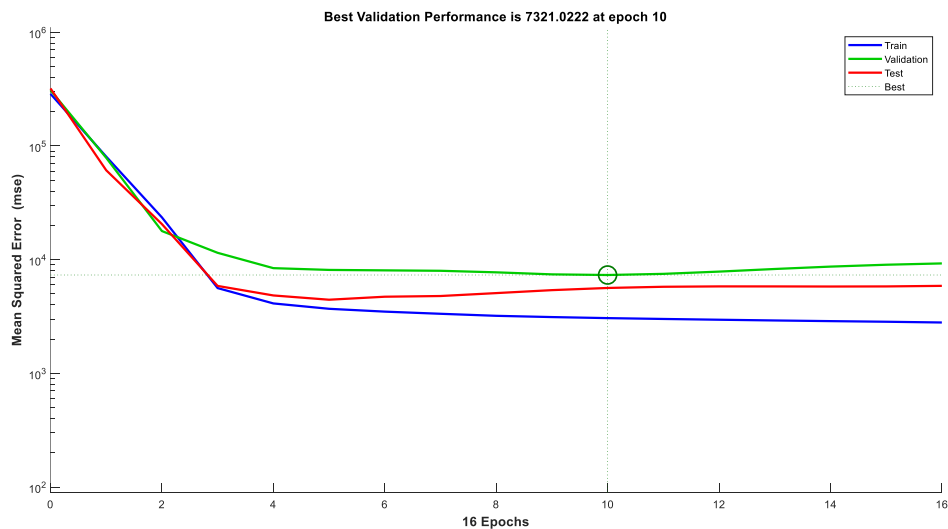


Figure 5. Performance plot of the NARNN model (MATLAB output).

3.2.3. NAR NN regression

In the regression plots, the dashed line in each plot represents the perfect result outputs = targets, which can be seen on the regression plots. The solid line in each plot represents the best-fit linear regression line between outputs and targets. On top of each plot, the regression R value measures the

correlation between the outputs and the targets. If R equals one, this indicates that there is an exact linear relationship between the outputs and the targets. If R is close to zero, then there is no linear relationship between the outputs and the targets.

As illustrated in Figure 5, the regression R value for the training phase was 0.98054; for the validation phase, 0.95846; for the testing phase, 0.9628; and for all the samples, 0.97416 for the NARNN model. In the training phase, the all R values were above 0.9. This can be seen in Figure 6, indicating that all three models fit equally well statistically. Similarly, all the R values were above 0.9 thus displaying acceptable fit from the testing phase perspective, which indicated good predictive abilities for all three models for the new datasets.

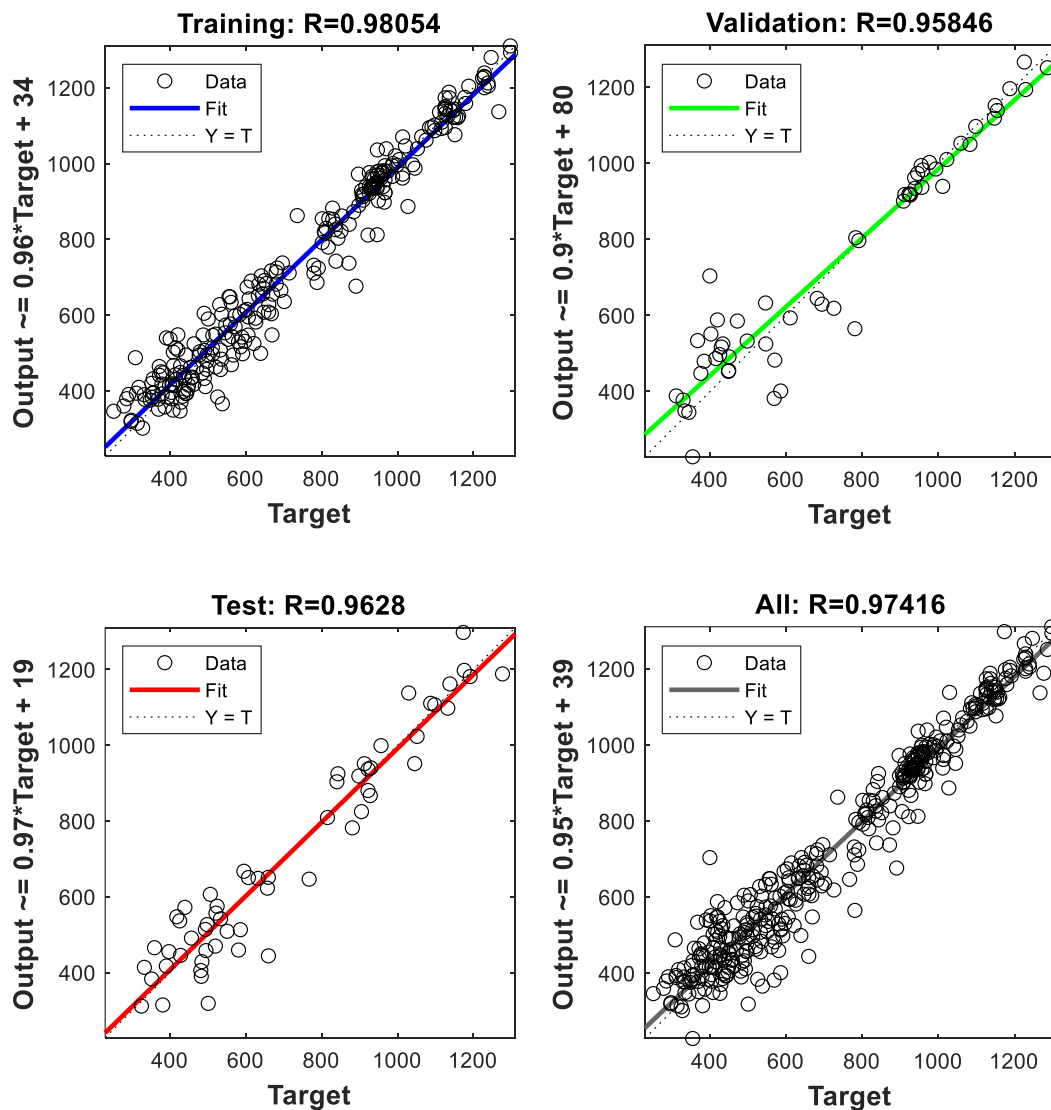


Figure 6. Regression plots of the NARNN model (MATLAB output).

3.2.4. NAR neural network error histogram

The error histogram gives an indication of outliers, which are data points where the fit is significantly worse than that of most of the data. In the error histograms (Figure 7), the blue bars represent the training data, the green bars represent the validation data, and the red bars represent the testing data. The results showed that there were a few training points and some testing points outside of the range. These outliers were also visible on the training and testing regression plots (Figure 6). If the outliers are valid data points but are unlike the rest of the data, then the network is extrapolating for these points. This means more data similar to the outlier points should be considered in training analysis and that the network should be retrained.

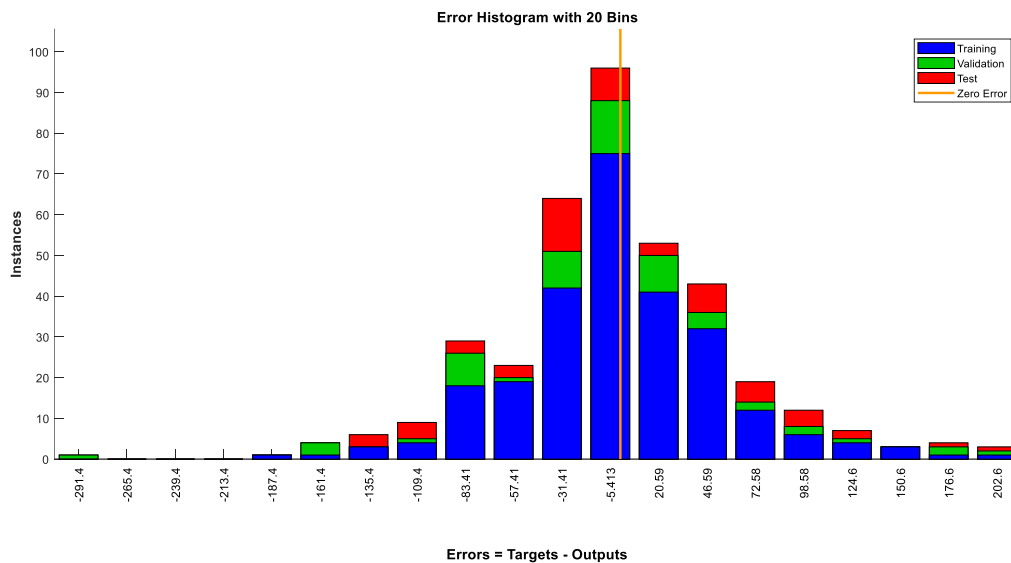


Figure 7. Error histogram of the NARNN model (MATLAB output).

3.2.5. NARNN time-series response

The dynamic network time-series response plots were displayed in Figure 8 for the NARNN model, showing that the outputs were distributed evenly on both sides of the response curve, and the errors versus time were small in the training, validation, and testing phases. The results indicated that the all three models were able to predict the Time Series over the simulation period efficiently.

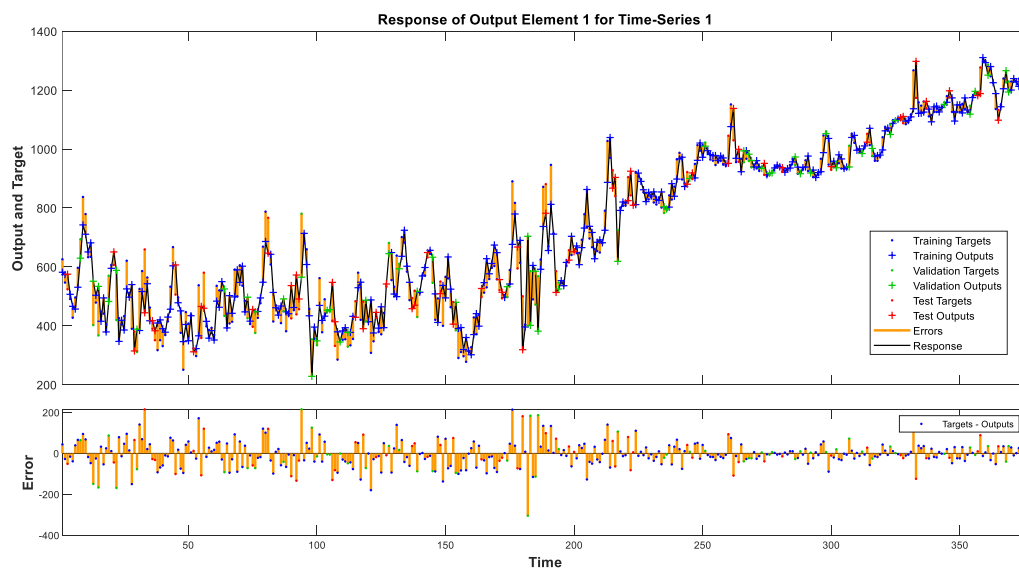


Figure 8. Network time-series response of the NARNN model (MATLAB output).

3.2.6. NARNN error autocorrelation

The error autocorrelation function describes how the prediction errors are related in time. For a perfect prediction model, there should only be one nonzero value of the autocorrelation function, and it should occur at zero lag (this is the MSE). This would mean that the prediction errors are completely uncorrelated with each other (white noise). If there is significant correlation in the prediction errors, then it should be possible to improve the prediction perhaps by increasing the number of delays in the tapped delay lines.

The correlations for the NARNN model (Figure 9), except for the one at zero lag, all fell approximately within the 95% confidence limits around zero, so the models seemed to be adequate. There are, however, some exceptions which suggests that the created network can be improved by retraining it or by increasing the number of neurons in the hidden layer. If even more accurate results are required, retraining the network will change the initial weights and biases of the network, and may produce an improved network after retraining.

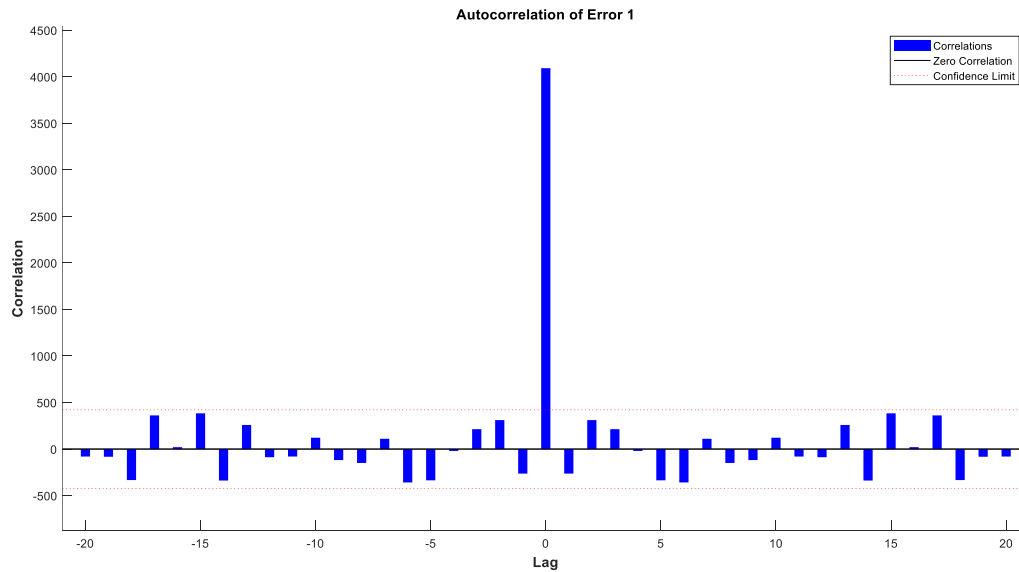


Figure 9. Error autocorrelation of the NARNN model (MATLAB output).

3.3. ARMA-NARNN hybrid (HYBRID) model

In MATLAB, the HYBRID model applied to Time Series prediction using its past residuals from the SARIMA model can be expressed as follows:

$$e(t) = \Phi(e(t-1), e(t-2), \dots, e(t-d)) + \varepsilon(t) \quad (14)$$

where $e(t)$ is the residual of the Time Series at time t , d is the time delay, and $\varepsilon(t)$ is the error term. This equation describes how the HYBRID model is used to predict the future residual of a Time Series, $e(t)$, using the past residuals of the Time Series, $(e(t-1), e(t-2), \dots, e(t-d))$.

Similarly, the development of the optimal architecture for the HYBRID model requires the determination of time delays, the number of hidden neurons, and an efficient training algorithm. The results of the error analysis showed that the HYBRID model with 11 neurons in the hidden layer and 3 time delays also provided the best performance (MSE = 2982.66655) with the LM algorithm (HYBRID) (Table 3). At the same time, the training progress using the LM algorithm for the Mixed-LM model stopped when the validation error increased for six iterations with Performance = 2700, Gradient = 175.1257, and Mu = 10.00 at epoch 13 (Figure 10).

Table 3. HYBRID model selection using Levenberg-Marquardt algorithm.

Number of hidden neurons	Number of delays (d)				
	2	3	4	5	6
8	3920.93665	3859.27889	3302.37490	4348.93180	4955.52066
9	4341.29026	3843.05508	3645.42461	3062.75879	4927.80700
10	3675.49724	3739.71316	3456.30464	3914.27581	5361.73173
11	5313.30679	2982.66655	4256.62635	4250.12751	3101.86040
12	3679.04563	3952.80522	3311.36531	3860.98596	4491.00417

Source: Author's work

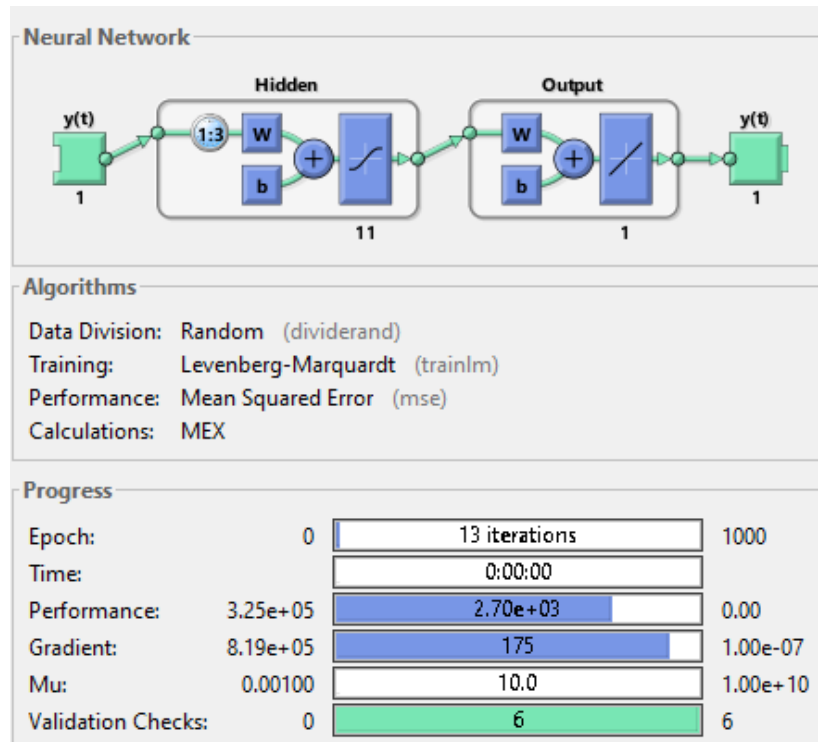


Figure 10. HYBRID model training using Levenberg-Marquardt algorithm (MATLAB output).

As illustrated in Figure 11, the best performance for the validation phase was 5943.2994 at epoch 7 for the HYBRID model. The results also showed a good network performance because the validation and testing errors have similar characteristics, and did not appear that any significant overfitting has occurred. In the error histograms (Figure 12), the blue bars represent the training data, the green bars represent the validation data, and the red bars represent the testing data. The results showed that there had a few testing and validation points outside of the range.

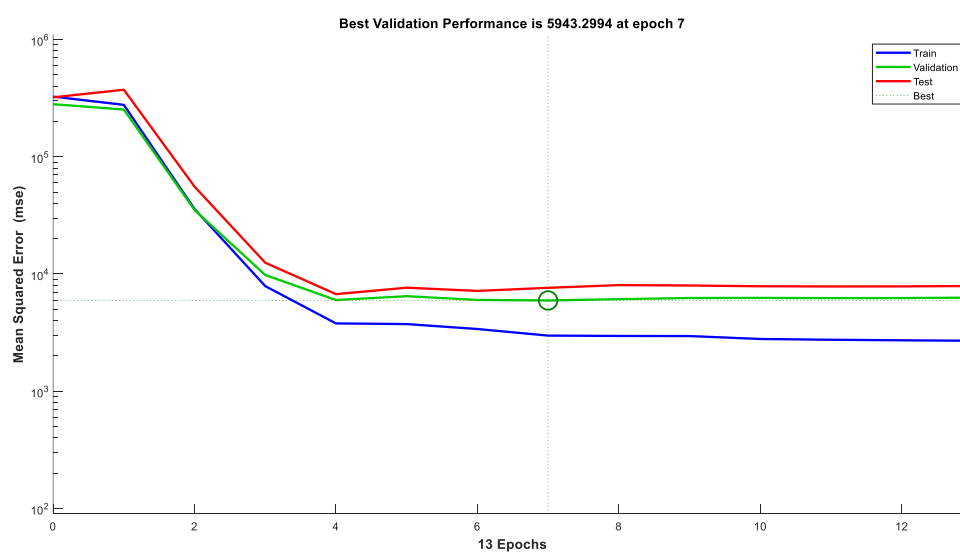


Figure 11. Performance plot of the HYBRID model (MATLAB output).

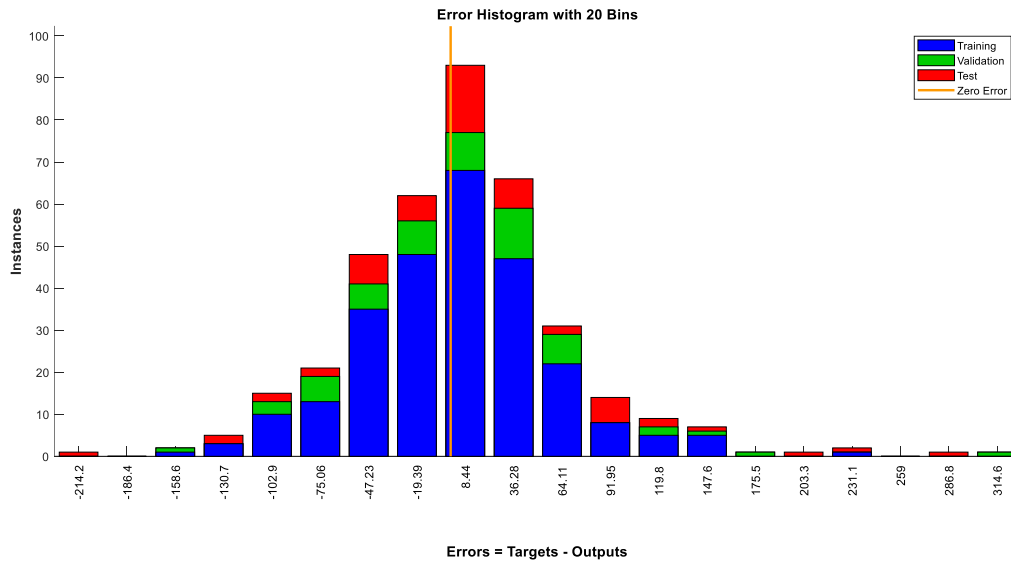


Figure 12. Error histogram of the HYBRID model (MATLAB output).



Figure 13. Network Time-Series response of the HYBRID model (MATLAB Output).

The dynamic network time-series response plots were displayed in Figure 13 for the HYBRID model, showed that the outputs were distributed evenly on both sides of the response curve, and the errors versus time were small in the training, validation, and testing phases. The results indicated that the HYBRID model was able to predict the Time Series over the simulation period efficiently as well. For the HYBRID model, the correlations except for the one at zero lag, all fell approximately within the 95% confidence limits around zero, so the model was adequate (Figure 14).

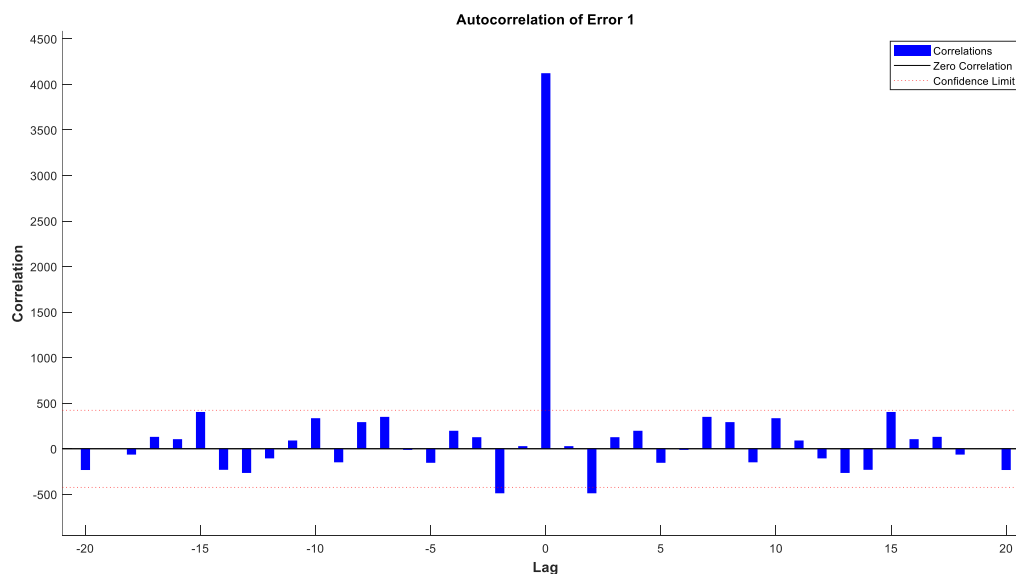


Figure 14. Error autocorrelation of the HYBRID model (MATLAB Output).

4. Discussion and conclusions

Assuming normal weather conditions and no further spread of banana plant diseases, banana production and trade volumes are rapidly increasing in response to fast population growth in producing countries as well as expanding global import demand. Obviously, the future will provide more intensive knowledge, with a better understanding of the natural complexities of living systems. Bringing together a wide variety of perspectives and concepts requires holistic solutions that involve working across disciplines, principles, and methods to support interdisciplinarity and transdisciplinarity, to explore and formalize systems concepts, and to develop systemic methods for learning and change.

Despite the importance of banana demand and supply in the global markets, there is a lack of studies in the technical literature available on global price of bananas forecasting schemes. Forecasting is a kind of dynamic filtering, in which past values of the Time Series can be used to predict future values. In Time Series forecasting, how to increase the prediction accuracy as much as possible with the non-stationary and noise data is a significant challenge. Experimentally, we hope to explore a hybrid model that can effectively improve the performance of global price of bananas throughput forecasting. In the proposed HYBRID model, a combination of statistical model in linear modeling and machine learning model to capture nonlinear pattern from the residuals of linear model, can be a better choice for modeling and forecasting the Time Series for this study. The fundamental idea is that this combination supports for the limitations of one model with the strengths of the other.

Empirically, the ARIMA and NARNN models are good at modelling linear and nonlinear problems for the Time Series, respectively. In this study, the best choice Time Series model was an ARIMA (2,1,4) with a drift model as its lowest AIC values among other models. It was noticed that this ARIMA (2,1,4) with a drift model gave evidence about future global price of bananas would increase over time. Furthermore, the NARNN model with 12 neurons in the hidden layer and 6 times delays was evaluated as the optimal neural network structure using the LM algorithm.

This study not only wanted to find the best fit model for the Time Series, global price of bananas from January 1990 to November 2021, but also tried to evaluate the accuracy of the ARIMA, NARNN, and HYBRID models used in the forecasting of global price of bananas. The comparative results revealed that the HYBRID model with 11 neurons in the hidden layer and 3 times delays (MSE = 2982.66655) yielded higher accuracy than the NARNN model with 12 neurons in the hidden layer and 6 times delays (MSE = 3058.11185), and the ARIMA (2,1,4) with a drift model (MSE = 4475.50697) in this study (Table 4).

Table 4. Time-Series model selection using mean squared error.

	ARIMA (2,1,4)	NARNN (Hidden Layer = 12, d = 6)	HYBRID (Hidden Layer = 11, d = 3)
MSE	4475.50697	3058.11185	2982.66655

Source: Author's work

According to the results of this study, this HYBRID ARIMA-NARNN model can provide richer information which is important in the decision-making process related to the future global price of bananas, it can be employed in forecasting the future performance for global price of bananas change outcomes. Thus, this study may provide an integrated modeling approach as a decision-making supportive method for forecasting global price of bananas in advance. Understanding past global price of bananas is important for the analysis of current and future global price changes for bananas.

In order to sustain these observations, research programs utilizing the resulting data should be able to improve significantly our understanding and narrow our projections of the future changes and variabilities in the global price of bananas. For further research tasks, the nonlinear autoregressive exogenous (NARX) neural network can consider past information of the same Time Series (global price of bananas), current and past information of the externally determined Time Series that influences the Time Series of interest (i.e., banana production, banana sales, weather information, disease impact information, etc.) in terms of forecasting accurately.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

- Altendorf S (2019) Banana Fusarium Wilt Tropical Race 4: A mounting threat to global banana markets? In 2019 Food Outlook - Biannual Report on Global Food Markets, FAO, Rome, 13–20. Available from: http://www.fao.org/3/ca6911en/CA6911EN_TR4EN.pdf.
- Beale MH, Hagan MT, Demuth HB (2019) Deep Learning Toolbox™: Getting Started Guide. *Deep Learn* Natick, MA: The MathWorks, Inc.
- Benrhmach G, Namir K, Namir A, et al. (2020) Nonlinear autoregressive neural network and extended Kalman filters do prediction of financial Time Series. *J Appl Math* 2020: 1–6. <https://doi.org/10.1155/2020/5057801>
- Box GEP, Jenkins GM, Reinsel GC, et al. (2015) *Time Series Analysis: Forecasting and Control* (5th ed.). John Wiley and Sons Inc.

- Eyduran SP, Akin M, Eyduran E, et al. (2020) Forecasting banana harvest area and production in Turkey using Time Series analysis. *Erwerbs-Obstbau* 62: 281–291. <https://doi.org/10.1007/s10341-020-00490-1>
- Fatin ZN, Titik E, Mulyatno SB (2020) The analysis of price and market integration of banana commodities in Lampung, Indonesia. *Russ J Agric Socio-Econ Sci* 3: 61–68. <https://doi.org/10.18551/rjoas.2020-03.07>
- Gardner M, Dorling SR (1998) Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. *Atmos Environ* 32: 2627–2636. [https://doi.org/10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0)
- Gavin HP (2013) The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering Duke University, 19. Available from: <http://people.duke.edu/~hpgavin/ce281/lm.pdf>.
- Hamjah MA (2014) Forecasting major fruit crops productions in Bangladesh using Box-Jenkins ARIMA Model. *J Econ Sustainable Dev* 5: 96–107. Available from: <https://core.ac.uk/download/pdf/234646336.pdf>.
- Hossain MM, Abdulla F, Majumder AK (2016) Forecasting of banana production in Bangladesh. *Am J Agric Biol Sci* 11: 93–99. <https://doi.org/10.3844/ajabssp.2016.93.99>
- Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. *Q Appl Math* 2: 164–168. <https://doi.org/10.1090/qam/10666>
- Ljung GM, Box GEP (1978) On a measure of lack of fit in Time Series models. *Biometrika* 65: 297–303. <https://doi.org/10.2307/2335207>
- MacKay DJC (1992) Bayesian Interpolation. *Neural Comput* 4: 415–447. <https://doi.org/10.1162/neco.1992.4.3.415>
- Market Reports World (2019) *Banana market size, share, analysis - segmented by geography - growth, trends, and forecast (2019–2024)*. Global Banana Market Research Report, Market Reports World, 94. Available from: <https://www.marketreportsworld.com/banana-market-13487750>.
- Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. *J Soc Ind and Appl Math* 11: 431–441. <https://doi.org/10.1137/0111030>
- Moller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6: 525–533. [https://doi.org/10.1016/S0893-6080\(05\)80056-5](https://doi.org/10.1016/S0893-6080(05)80056-5)
- Montgomery DC, Jennings CL, Kulahci M (2008) *Introduction to Time Series Analysis and Forecasting*. Hoboken, John Wiley & Sons. Inc.
- Omar MI, Dewan MF, Hoq MS (2014) Analysis of price forecasting and spatial co-integration of banana in Bangladesh. *Eur J Bus Manage* 6: 244–255. Available from: <https://www.iiste.org/Journals/index.php/EJBM/article/view/11463>.
- Rathod S, Mishra GC, Singh KN (2017) Hybrid Time Series models for forecasting banana production in Karnataka State, India. *J Indian Soc Agric Stat* 71: 193–200. Available from: https://www.researchgate.net/publication/322165651_Hybrid_Time_Series_Models_for_Forecasting_Banana_Production_in_Karnataka_State_India.
- Rathod S, Mishra GC (2018) Statistical models for forecasting mango and banana yield of Karnataka, India. *J Agric Sci Technol* 20: 803–816. Available from: <http://jast.modares.ac.ir/article-23-19768-en.html>.
- Rebortera MA, Fajardo AC (2019) An enhanced deep learning approach in forecasting banana harvest yields. *Int J Adv Comput Sci Appl* 10: 275–280. <https://doi.org/10.14569/IJACSA.2019.0100935>

- Ruiz A, Fobelets V, Grosscurt C, et al. (2017) The external costs of banana production: A global study. Research Report Prepared for Fairtrade International, True Price Trucost. Available from: http://makefruitfair.org/wp-content/uploads/2017/07/170224_Research_Report_External_Cost_of_Bananas_-_final.pdf.
- Voora V, Larrea C, Bermudez S (2020) Sustainable Commodities Marketplace Series 2019, The International Institute for Sustainable Development, *Global market report: Bananas* 12. Available from: <https://www.iisd.org/system/files/publications/ssi-global-market-report-banana.pdf>.
- Sariev E, Germano G (2020) Bayesian regularized artificial neural networks for the estimation of the probability of default. *Quant Financ* 20: 311–328. <https://doi.org/10.1080/14697688.2019.1633014>
- Young WL (1977) The Box-Jenkins approach to Time Series analysis and forecasting: principles and applications. *RAIRO. Rech opérationnelle*, 11: 129–143. Available from: http://www.numdam.org/article/RO_1977__11_2_129_0.pdf.
- Zhang GP, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: The state of the art. *Int J Forecast* 14: 35–62. [https://doi.org/10.1016/S0169-2070\(97\)00044-7](https://doi.org/10.1016/S0169-2070(97)00044-7)
- Zhang GP (2003) Time Series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50: 159–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)



AIMS Press

© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)