International Journal of Data Science
Vol. 2, No. 1, June 2021, pp. 19-37

Application of Nonlinear Autoregressive Neural Network to Model and Forecast Time Series Global Price of Bananas

Yeong Nain Chi [a,1,*], Orson Chi [b,2]

[a] Department of Agriculture, Food, and Resource Sciences, University of Maryland Eastern Shore, Princess Anne, MD 21853, United States
[b] Cybersecurity Engineering Technology Program, University of Maryland Eastern Shore, Princess Anne, MD 21853, United States
[1] ychi@umes.edu; [2]ochi@umes.edu
* corresponding author

ARTICLE INFO

Article history
Received February 21, 2021
Revised March 7, 2021
Accepted March 10, 2021

Keywords
bananas
global price
time series
modeling
forecasting
nonlinear autoregressive neural network
Levenberg-Marquardt algorithm
Bayesian regularization algorithm
scaled conjugate gradient algorithm

ABSTRACT

The primary purpose of this study was to apply the nonlinear autoregressive neural network to model the long-term records of monthly global price of bananas from January 1990 to November 2020. The development of the optimal architecture for the nonlinear autoregressive neural network requires determination of time delays, the number of hidden neurons, and an efficient training algorithm. Through training of the nonlinear autoregressive neural network models, the prediction performance of the models was evaluated by its mean squared error value, the average squared difference between the observed and predicted values. In this study, the empirical results revealed that the NAR-BR model with 13 neurons in the hidden layer and 6 time delays provided the best performance at its smaller mean squared error value and yielded higher accuracy than the NAR-LM model with 12 neurons in the hidden layer and 4 time delays and NAR-SCG model with 12 neurons in the hidden layer and 6 time delays. Understanding past global price of bananas is important for the analyses of current and future global price of bananas changes. In order to sustain these observations, research programs utilizing the resulting data should be able to improve our understanding and narrow projections of future global price of bananas significantly.

1. Introduction

Bananas are one of the world's most commonly eaten fruit, and are the world's favorite fruit in terms of consumption quantity globally. The U.S. Food and Drug Administration has approved the following nutrient content descriptors for bananas: fat-free, sodium-free, cholesterol-free, a good source of fiber, a good source of vitamin C, a good source of vitamin B-6, and a good source of potassium. Bananas are the fourth most important food crop after wheat, rice, and maize in terms of production, and are one of the biggest profit makers in supermarkets, making them critical for economic and global food security.

Asia-Pacific is the largest banana-producing region, while India and China were the two leading banana-producing countries. Asia-Pacific led the market with a 61% share of global consumption, while India was the world's leading producer of banana 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).

Central America and the Caribbean is the largest exporting region, responsible for approximately 80% of global exports. In 2018, the global banana exports were estimated at 23.3 million tons, and Ecuador being the largest banana exporter accounted for 24.7% of the global exports. Belgium, Costa Rica and Columbia were the other top banana exporters in the world, whereas, the United States was the leading importer of banana with 18% share in the world imports (Voora and Bermudez, 2020).

Bananas are among the most traded fruits in the world. In 2017, 22.7 million tons of bananas, excluding plantains, were traded, representing almost 20% of global production that year. The value of this trade was worth $11 billion, which is higher than the export value of any other exported fruit (Voora and Bermudez, 2020). The Market Reports World (2019) forecasted that the global banana sector would experience a compound annual growth of 1.21% in consumption from 2019 to 2024, reaching a global consumption volume of 136 million tons by 2025, compared to 116.2 million tons in 2017.

Bananas have become such a common, inexpensive grocery item, thus, that people often forgot where they come from and how they got here. However, the way bananas are produced and exported gives an insight into a number of global issues. Although bananas may only look like a fruit, actually they bring a wide variety of environmental, economic, social, and political questions.

As one of the first tropical fruit to be exported, bananas are a cheap way to bring the tropics to North America and Europe. Furthermore, banana production has negative effects on the environment and society, particularly external environmental and social costs that still have not captured in prices currently (Ruiz et al., 2017).

The banana industry has been the subject of numerous studies, debates and presentations at all scales, from the farmer to the international trade system. The focus of most of these works have been on the economics of the banana industry, whether it be conditions for trade, competition among producers, or the role of globalization in the banana industry. Key in these works has been the influences of the global political economy on the operation of the banana industry, the levels of social dependence on the banana industry, micro and macro-economic policies as well as how the banana industry influences decisions made by the political structure of many countries and regions.

However, the primary purpose of this study was to apply the nonlinear autoregressive neural network to model the long-term records of monthly global price of bananas from January 1990 to November 2020. According to literature review of global banana market, there were several articles using time series techniques to forecast banana production (Hamjah, 2014; Hossain et al., 2016; Eyduran et al., 2020), while quite a few studies focusing on banana price forecasting (Omar et al., 2014; Fatin et al., 2020).

Neural networks have become one of the most popular trends in 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; Rebortera and Fajardo, 2019). Despite the importance of banana demand and supply in the global banana market, there is a lack of studies in the technical literature available on global price of bananas forecasting schemes. Thus, the contributions of this study not only can provided information which are important in decision making process related to the future global price of bananas change impacts, but also can be employed in forecasting the future performance for global price of bananas change outcomes.

2. Materials

The long-term records of monthly global price of bananas (units: U.S. dollars per metric ton, not seasonally adjusted) from January 1990 to November 2020 (Figure 1), is available to the public from International Monetary Fund, retrieved from FRED, Federal Reserve Bank of St. Louis (https://fred.stlouisfed.org/series/PBANSOPUSDM). Average monthly global price of bananas was $710.59 U.S. dollars per metric ton with the standard deviation of $272.47 (Minimum: $250.51, Maximum: $1,298.34, and Median: $648.51).

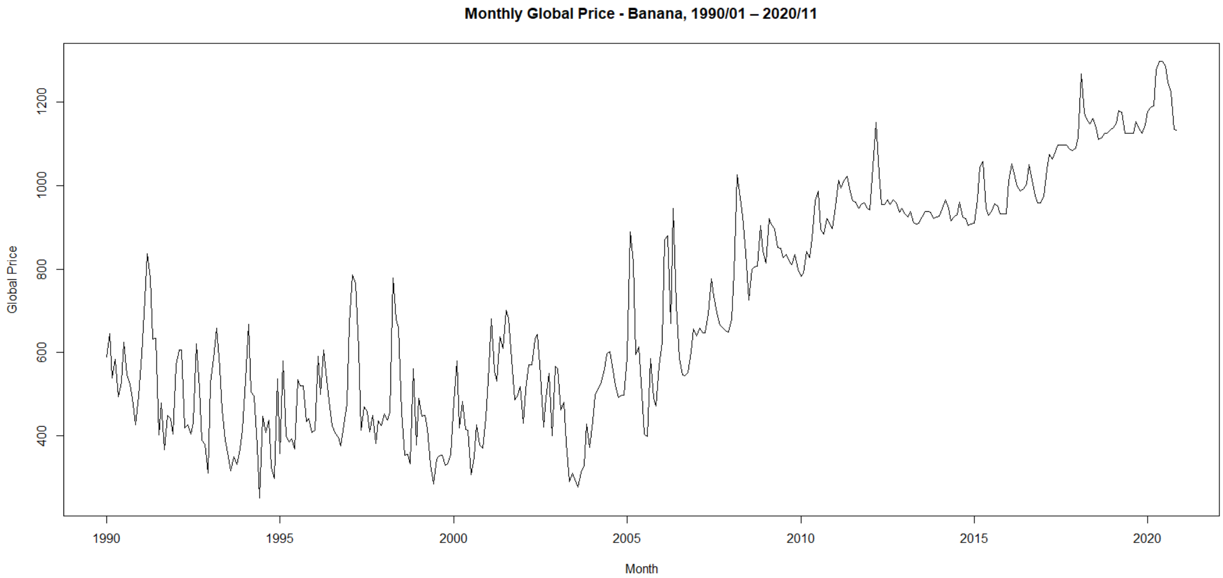

Fig. 1 Time Series Plot of Monthly Global Price of Bananas, January 1990 ~ November 2020 (Source: own work)

3. Methods

A time series is a set of observations, each one being recorded at a specific time t. The sequence of random variables $\{y_t: t = 1, 2, \cdots, T\}$ is called a stochastic process and serves as a model for an observed time series. The idea behind the autoregressive (AR) model is to explain the present value of the time series, y_t, by a function of p past values, $(y_{t-1}, y_{t-2}, \cdots, y_{t-p})$. Thus, the AR process of order p, AR(p), is defined by the equation:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t$$

where $\phi = (\phi_1, \phi_2, \cdots, \phi_p)$ is the vector of model coefficients and p is a non-negative integer, and ε_t is white noise, i.e., $\varepsilon_t \sim N(0, \sigma^2)$.

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

$$y_t = \Phi(y_{t-1}, y_{t-2}, \cdots, y_{t-p}, w) + \varepsilon_t$$

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

In the neural network frameworks, single hidden layer feedforward neural network is the most widely used model for time series modeling and forecasting (Zhang et al., 1998). A single hidden layer feedforward neural network consists of neurons that are ordered into layers. The first layer is called the input layer, the last layer is called the output layer, and the layer between is the hidden layer (Figure 2).

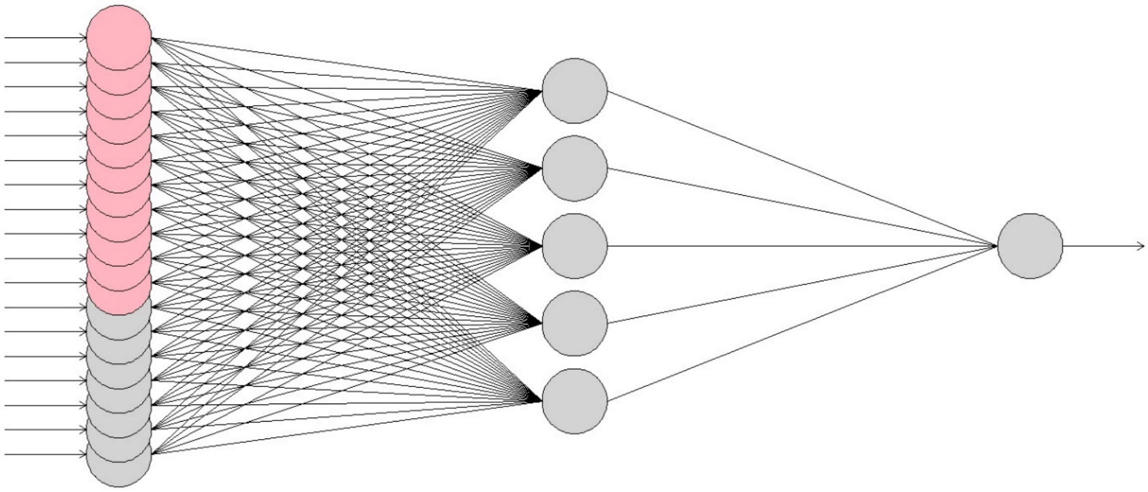

Fig. 2 Single Hidden Layer Feedforward Neural Network (Source: own work)

With the time series data, lagged values of the time series can be used as inputs to a neural network, so-called this the nonlinear autoregressive (NAR) neural network model. Mathematically, the NAR neural network model (Benrhmach et al., 2020) can be written by the equation of the form as:

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

where d is 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, $\Phi(y) = 1 / (1 + \exp(-y))$.

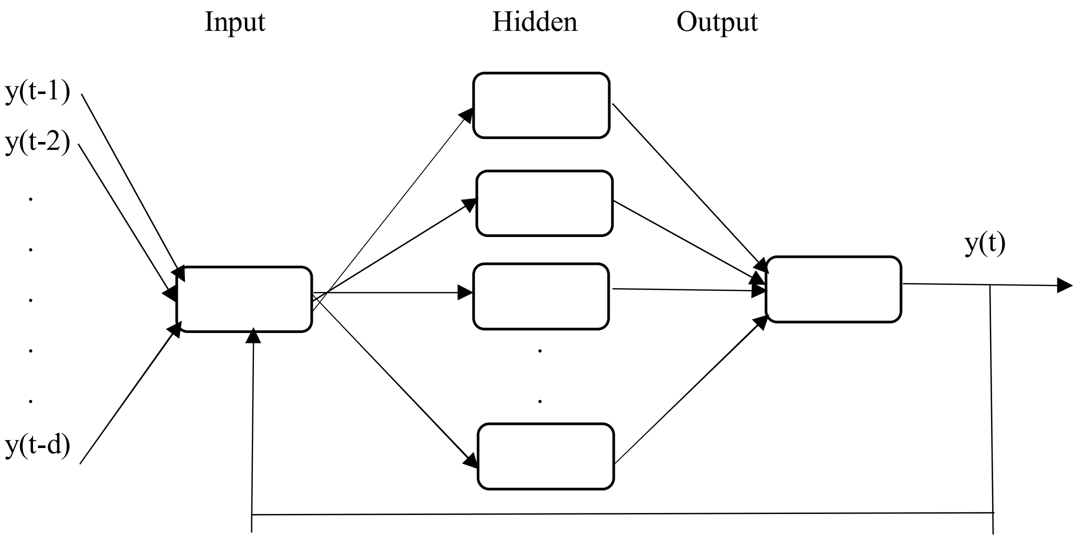

Fig. 3 NAR Neural Network (Source: own work)

In MATLAB, the NAR neural network (Figure 3) applied to time series prediction using its past values of a univariate time series can be expressed as follows:

$$y(t) = \Phi(y(t-1), y(t-2), \cdots, y(t-d)) + e(t)$$

where y(t) is the value of the time series 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 NAR neural network 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), ⋯, y(t-d)). The function $\Phi(\cdot)$ is a 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 and predicted output values (Beale et al., 2019).

In order to train the NAR neural network, open loop architecture (Figure 4) shows a block diagram of the NAR neural network generated during MATLAB processing in the MATLAB (2019a) Neural Network Toolbox. In Figure 4, the block y(t) is the input series consisting of monthly global price of bananas observations. The number "1" at the bottom of the block indicates univariate time series.

The hidden layer of the network is illustrated in the second block, namely "Hidden Layer with Delays". The inner boxes "w" and "b" represent input-hidden weights and bias respectively for a single neuron in the hidden layer. The term "1:2" denotes the number of delays used. The larger box after the summation sign indicates the sigmoid transfer function of each neuron. The number "10" at the bottom of the "Hidden Layer with Delays" block denotes the number of hidden neurons.

The "Output Layer" block represents the output layer of the network. The inner boxes "w" and "b" represent the hidden-output weights and biases respectively. The transfer function of the output layer is linear. There is only one output neuron, which is denoted below the "Output Layer" block. The last block y(t) represents the predicted output. This output y(t) is different from the input y(t). Since the output of the network is a prediction of the input time series, MATLAB signifies both with the same variable (Beale et al., 2019).

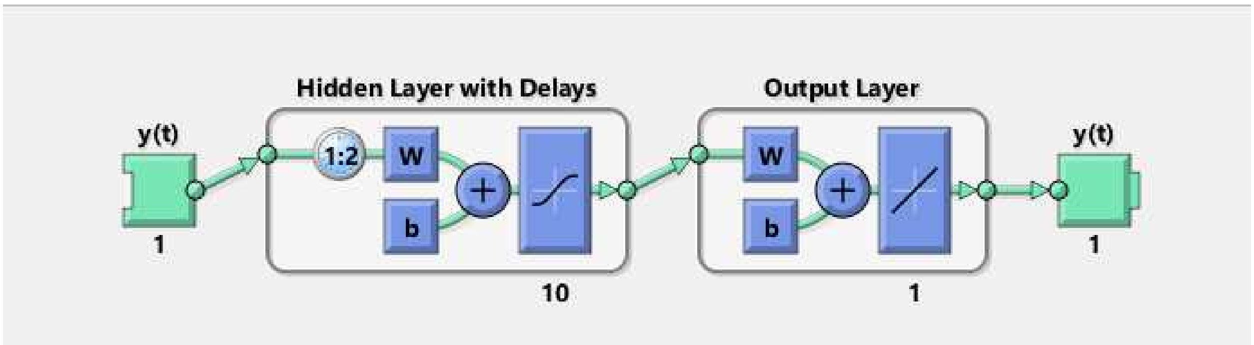

Fig.4 NAR Neural Network Architecture (Source: own work)

The most common learning rules for the NAR neural network are the Levenberg-Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient training algorithms. Training is the process of determining the optimal network weights and bias points of the single hidden layer feedforward neural network. This is done by defining the total error function between the network's output and the desired target and then minimizing it with respect to the weights.

3.1 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm, first published by Levenberg (1944) and then rediscovered by Marquardt (1963), is a commonly used iterative algorithm to solve non-linear minimization problems. These minimization problems arise especially in least squares curve fitting. This curve-fitting method is a combination of the gradient descent and the Gauss-Newton. It works without computing the exact Hessian matrix. Instead, it works with the gradient vector and the Jacobian matrix, therefore increasing the training speed and has stable convergence (Gavin, 2020).

3.2 Bayesian Regularization Algorithm

The Bayesian Regularization (BR) algorithm was introduced by MacKay (1992), which automatically sets the best possible performance function to accomplish the excellent generalization on the basis of Bayesian inference approach. The BR algorithm is based on the probabilistic interpretation of network parameters. Bayesian optimization of regularization parameters depends upon the calculation of the Hessian matrix at the minimum point. Therefore, the BR algorithm includes a probability distribution of network weights and the network architecture can be identified as a probabilistic framework (Sariev & Germano, 2020).

3.3 Scaled Conjugate Gradient Algorithm

The Scaled Conjugate Gradient (SCG) algorithm, developed by Moller (1993), is based on the Conjugate Gradient Method, but this algorithm does not perform a line search at each iteration. Unlike many other standard backward propagation algorithms, the SCG algorithm is fully automated, includes no critical user-specific parameters, and avoids a time consuming line search.

4. Results

In time series analysis, the lag plot can be used to check whether a time series is random or not. Meanwhile, the shape of the lag plot can provide clues about the underlying structure of the time series. The tighter the data is clustered around the diagonal, the more autocorrelation is present. In this case, Figure 5 showed a positive linear pattern suggesting that the time series was non-random and autocorrelation was present relatively.

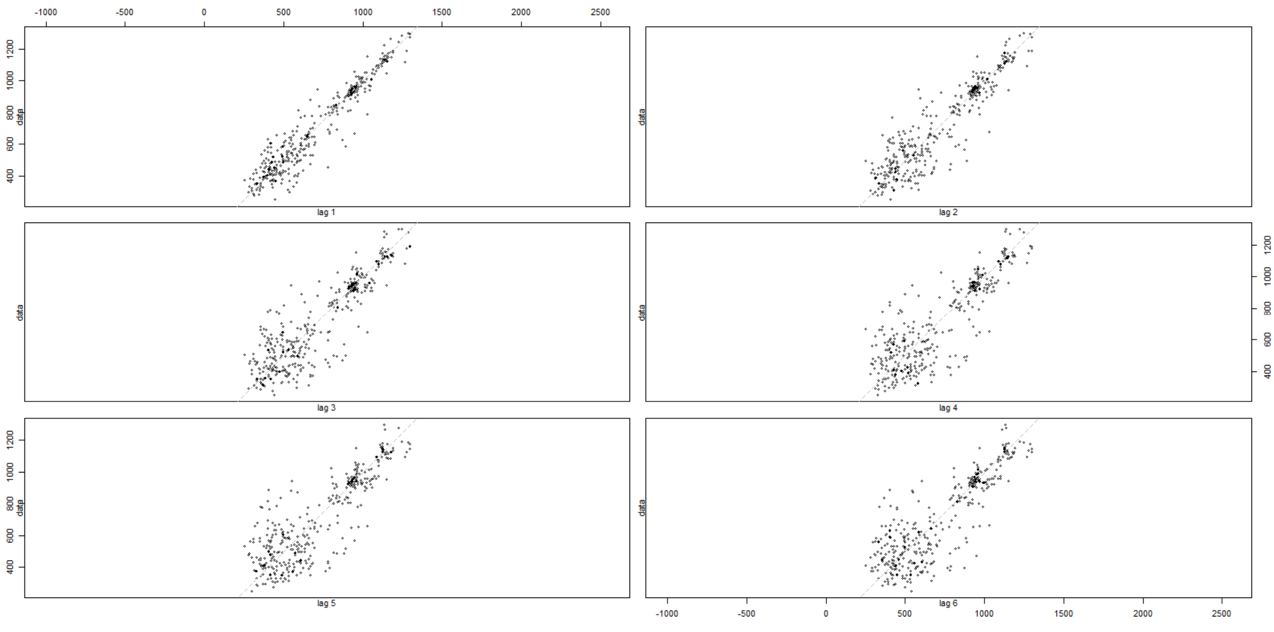

Fig. 5 Lag Plots of Monthly Global Price of Bananas, January 1990 ~ November 2020 (Source: own work)

In this study, the NAR neural network was applied to model time series monthly global price of bananas. Furthermore, the logistic sigmoid and linear transfer functions at the hidden and output layers was 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, BR, and SCG training algorithms respectively for the target time series in the MATLAB (2019a) Neural Network Toolbox: 371 time steps of one element, monthly global price of bananas from January 1990 to November 2020.

Three kinds of target time steps were set aside for the training, validation and testing phases in this case study. The training target time steps are presented to the network during training, and the network is adjusted according to its error. The validation target time steps are used to measure network generalization, and to halt training when generalization stops improving. The testing target time steps 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, 371 data samples were divided into 259 data for the training, 56 data for the validation, and 56 data for the testing. The development of the optimal architecture for the NAR neural network 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, BR, and SCG algorithms were employed for training of the NAR neural network and their performance were evaluated under the optimal neural network structure. The prediction performance of the models was evaluated by mean squared error (MSE), the average squared difference between the observed (y_i) and predicted ($\hat{y}_i$) values*. The error analysis showed that the NAR neural network with 12 neurons in the hidden layer and 4 time delays provided the best performance (MSE = 3225.23531) using the LM algorithm (NAR-LM) (Table 1). By using the BR algorithm, the NAR neural network with 13 neurons in the hidden layer and 6 time delays provided the best performance (MSE = 2422.92329) (NAR-BR) (Table 2). While the NAR neural network with 12 neurons in the hidden layer and 6 time delays provided the best performance (MSE = 4813.51336) with the SCG algorithm (NAR-SCG) (Table 3).

Table 1. NAR Model Selection Using the Levenberg-Marquardt Algorithm

Number of Hidden Neurons	Number of Delays (d)				
	2	3	4	5	6
3	5571.65946	6443.00570	5203.74953	5287.86699	5802.21986
4	6036.26473	5623.93575	5589.21637	5435.62360	4944.16317
5	6344.33775	5833.59727	4748.46805	4169.26025	5211.84250
6	7369.91708	5034.24024	4605.21220	4191.99443	4239.25794
7	6160.75564	5778.53304	5186.37906	5066.52972	4519.05082
8	5444.98374	4599.75204	4471.76037	4951.11190	4311.84726
9	5252.34058	6191.74787	4451.63980	4577.79683	4794.93680
10	5031.01176	4201.61277	5507.53525	6600.77050	3873.57360
11	4993.81295	5823.85883	5360.72426	4395.36528	5086.20796
12	6368.56630	4280.21322	3225.23531	4403.06781	3339.54790
13	4347.74883	3926.77755	4746.08029	4332.26641	3869.40767
14	5699.00720	4534.86434	4690.43483	4695.66237	5299.91103
15	5415.89208	5026.15132	4552.56480	4265.97320	4162.98781

* (MSE) = $(1/n) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (Source: own work)

Table 2. NAR Model Selection Using the Bayesian Regularization Algorithm

Number of Hidden Neurons	Number of Delays (d)				
	2	3	4	5	6
3	5778.49399	6209.78745	5032.43830	4760.70621	5168.95386
4	5863.37004	5199.74526	5945.25706	4932.39329	4424.61122
5	6196.10208	5323.18588	4833.34419	5181.05544	3965.32897
6	6441.72891	5030.57146	4317.18315	3949.29480	3580.90862
7	6377.76878	5016.47057	3970.77884	4022.91651	3483.05353
8	6233.75258	4743.71319	5221.09453	3273.99239	3190.07888
9	5534.76289	5465.76416	3840.21004	3694.83649	3100.09269
10	6201.08851	5554.21348	4736.58169	3840.83131	3284.26150
11	5457.84188	4902.52157	3686.45929	5097.96367	3693.93363
12	6355.52381	5398.50228	4248.69577	3426.01142	3312.22220
13	5410.39075	4322.74916	4437.74000	4072.21156	2422.92329
14	6171.54572	5821.56211	4049.09912	2922.18648	5125.63906
15	6089.83804	5172.81161	3519.81452	4689.56671	2479.67486

* (MSE) = $(1/n) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (Source: own work)

Table 3. NAR Model Selection Using the Scaled Conjugate Gradient Algorithm

Number of Hidden Neurons	Number of Delays (d)				
	2	3	4	5	6
3	7388.09939	6759.95014	10894.21161	5544.30538	5831.82648
4	5313.81622	8332.44462	6228.86238	6472.37397	5753.20714
5	7586.97806	9115.54613	9784.73189	6000.41319	5953.97660
6	6688.58883	7692.22647	5879.03070	5921.23140	5167.93009
7	7229.13532	6492.79564	8070.23782	7785.70990	5405.35857
8	7499.38122	5581.29073	5985.89982	7087.08284	6681.72434
9	8785.02160	6920.43810	5204.50872	7082.72056	5716.81388
10	7167.03510	7054.60220	10368.04597	5626.93984	5633.45762
11	6603.32834	6725.00332	6981.35164	7355.55987	5868.59331
12	7107.25794	5254.40217	5719.51383	6378.96397	4813.51336
13	6370.79363	5247.37116	7275.51741	6299.47651	5659.03838
14	6584.07460	5614.02738	8161.39697	5863.40080	6258.69102
15	7751.24044	7994.44528	5815.33628	9744.88082	9679.14565

* (MSE) = $(1/n) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (Source: own work)

4.1 NAR Neural Network Training Output

The LM algorithm typically requires more memory but less time. Training automatically stop when generalization stop improving, as indicated by an increase in the mean square error of the validation samples (Beale et al., 2019). Figure 6 displayed the training progress using the LM algorithm, stopped when the validation error increased for six iterations with Performance = 2720, Gradient = 2581.7597, and Mu = 1.00 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 BR algorithm typically requires more time but can result in good generalization for difficult, small or noise datasets (Beale et al., 2019). Figure 7 displayed the training progress using the BR algorithm, training stopped according to adaptive weight minimization (regularization) with Performance = 2420, Gradient = 175.59747, Mu = 5.00e+10, Effective Number Parameters = 81.4746, and Sum Squared Parameters = 96.386 at epoch 570. In terms of processing time, the BR algorithm took 0:00:04 during training. The term mu is the control parameter for the algorithm used to train the network. Choice of mu directly affect the error convergence. The SCG algorithm requires less memory. 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). The training for the network, in this case, was done using the SCG algorithm. Figure 8 displayed the training progress using the SCG algorithm, stopped when the validation error increased for six iterations with Performance = 4770, and Gradient = 2562.042 at epoch 57. In terms of processing time, the SCG algorithm took 0:00:00 during training.

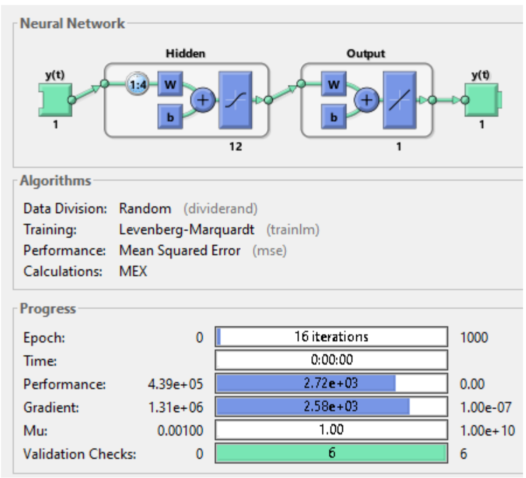

Fig. 6 NAR-LM Neural Network Training Output (Source: own work)

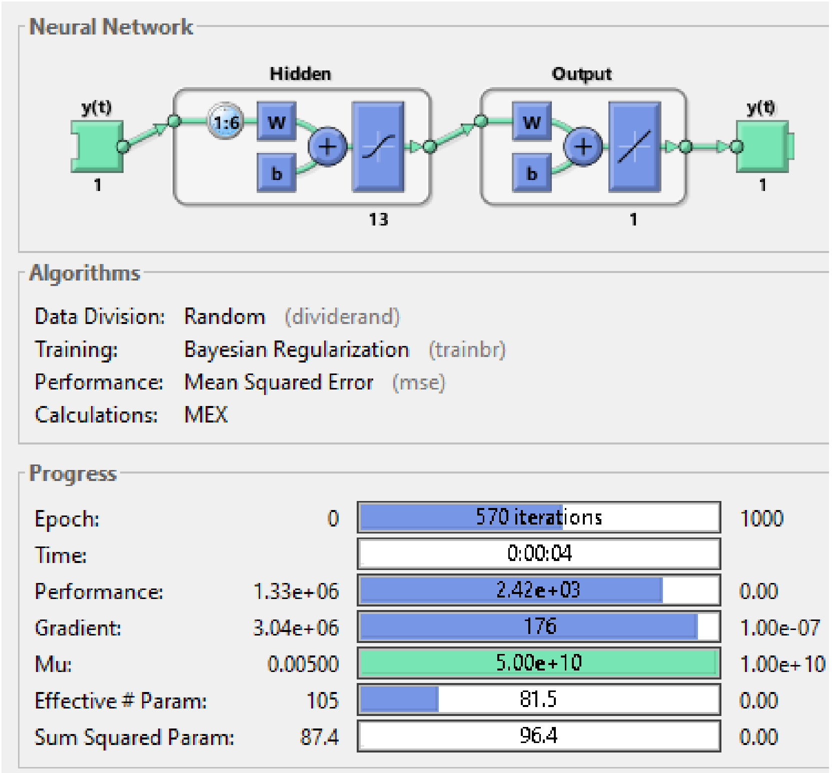

Fig. 7 NAR-BR Neural Network Training Output (Source: own work)

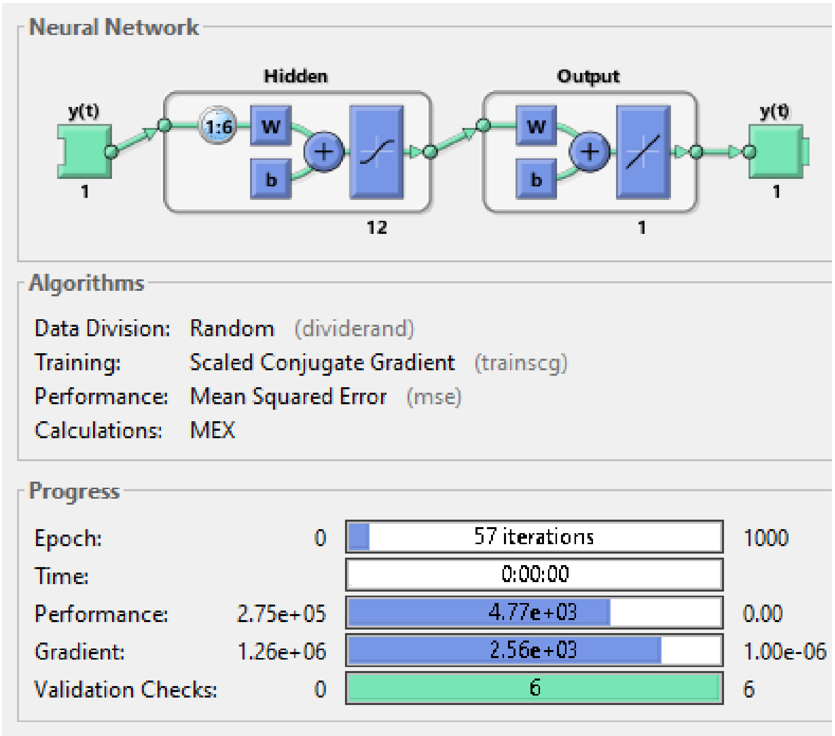

Fig. 8 NAR-SCG Neural Network Training Output (Source: own work)

4.2 NAR Neural Network Best Performance

The performance plot illustrated the relationship between the training, validation, and testing phases in forecasting monthly 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. As illustrated in Figure 9, the best performance for the validation phase was 5770.609 at epoch 10 for the NAR-LM model, while the NAR-BR model revealed the best performance (MSE = 2422.9233) for the training phase at epoch 569 (Figure 10). For the NAR-SCG model, the plot (Figure 11) showed the best performance for the validation phase was 7404.3825 at epoch 51. 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. 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 has occurred.

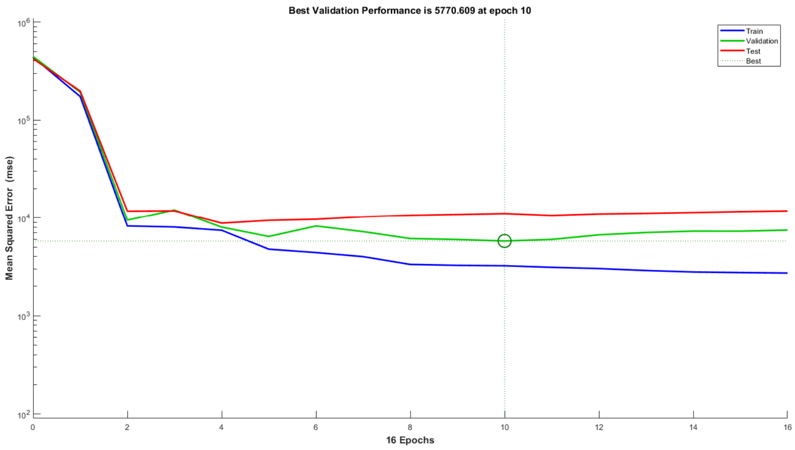

Fig. 9 Performance Plot of the NAR-LM Model (Source: own work)

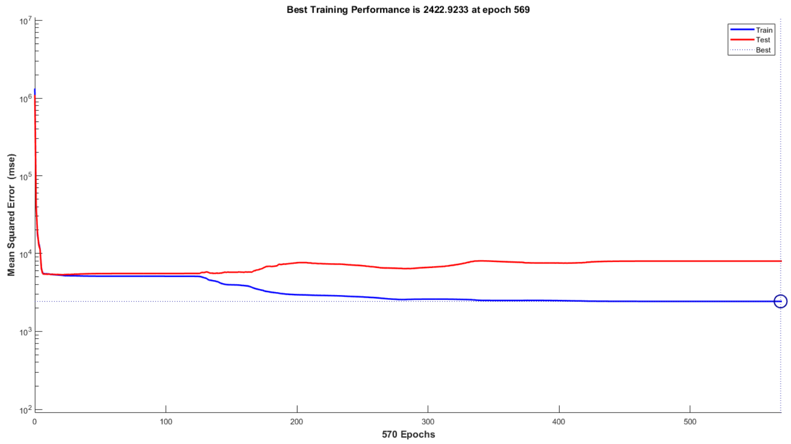

Fig. 10 Performance plot of the NAR-BR Model (Source: own work)

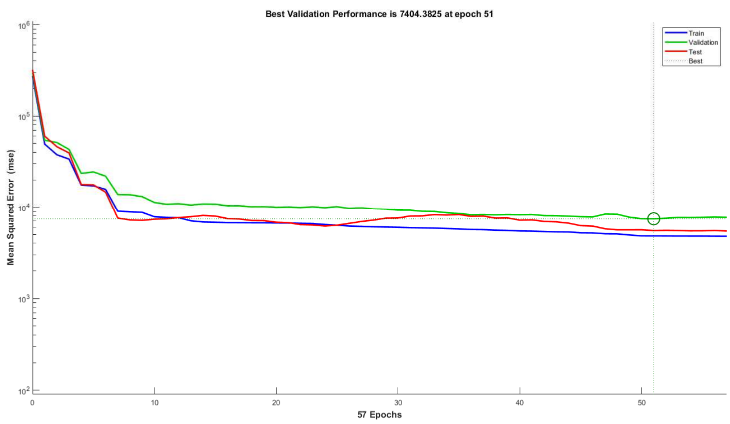

Fig. 11 Performance Plot of the NAR-SCG Model (Source: own work)

4.3 NAR Neural Network 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 = 1, 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 12, the regression R value for the training phase was 0.97803, for the validation phase was 0.97063, for the testing phase was 0.91543, and for the all samples was 0.96755, respectively for the NAR-LM model. For the NAR-BR model (Figure 13), the regression R value for the training phase was 0.98362, for the testing phase was 0.94369, and for the all samples was 0.97801 respectively. The regression R values for the NAR-SCG model (Figure 14) showed that the training phase was 0.966923, the validation phase was 0.95153, the testing phase was 0.96527, and the all samples was 0.96399 respectively.

In the training phase, the all R values were above 0.9, this can be seen in Figure 12, Figure 13, and Figure 14, indicating that the all three models fit equally well statistically. Similarly, the all R values were above 0.9 that displayed acceptable fit from the testing phase perspective, which indicated good predictive abilities of the all three models for new datasets.

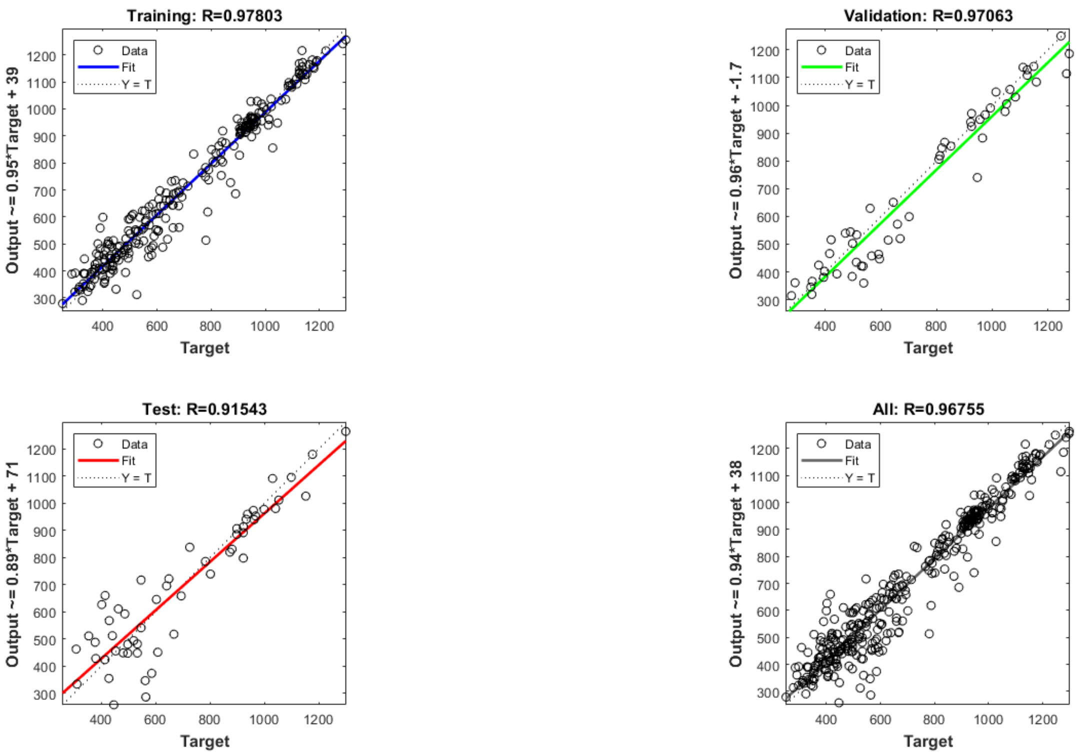

Fig. 12 Regression Plots of the NAR-LM Model (Source: own work)

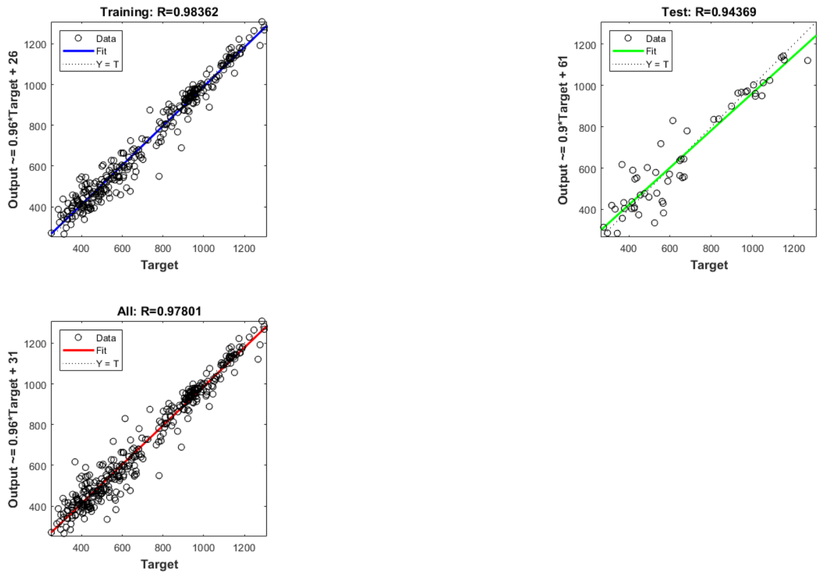

Fig. 13 Regression Plots of the NAR-BR Model (Source: own work)

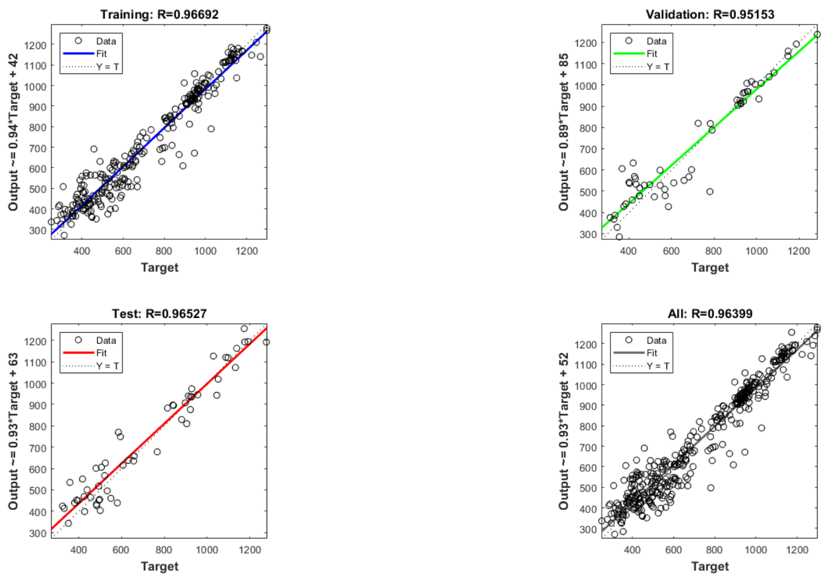

Fig. 14 Regression Plots of the NAR-SCG Model (Source: own work)

4.4 NAR Neural Network Error Histogram

The error histogram can give 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 15 and Figure 17), the blue bars represent the training data, the green bars represent the validation data, and the red bars represent the testing data, while in the Figure 16, it only has the blue bars representing the training data, and the red bars representing the testing data. The results showed that there had a few training points and some testing points outside of the range. These outliers were also visible on the training and testing regression plots (Figures 15,16,17). If the outliers are valid data points but are unlike the rest of the data, then the network is extrapolating for these points. It means more data similar to the outlier points should be considered in training analysis and that the network should be retrained.

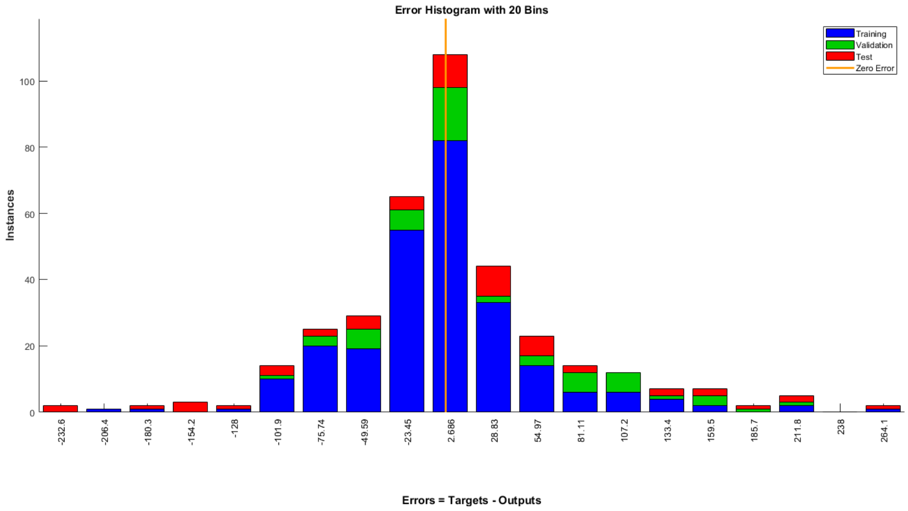

Fig. 15 Error Histogram of the NAR-LM Model (Source: own work)

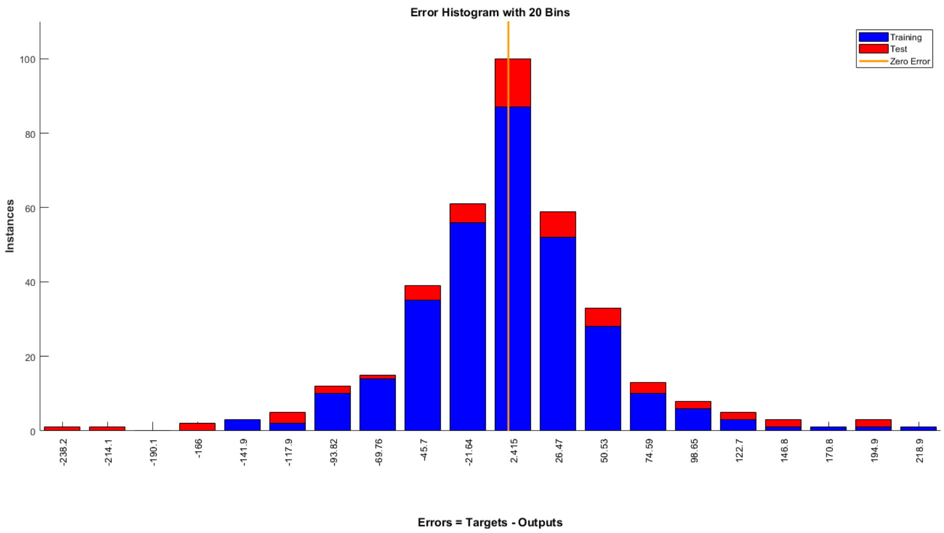

Fig. 16 Error Histogram of the NAR-BR Model (Source: own work)

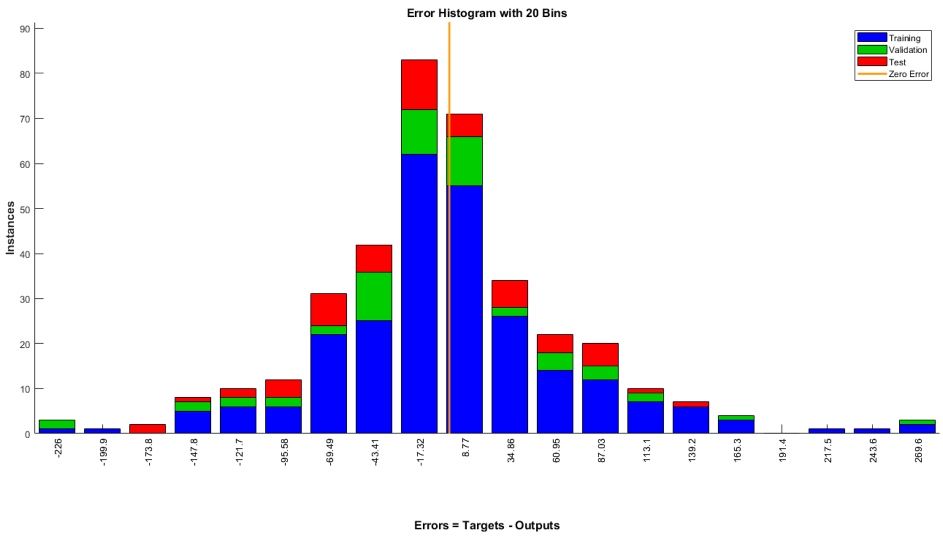

Fig. 17 Error Histogram of the NAR-SCG Model (Source: own work)

4.5 NAR Neural Network Time-Series Response

The dynamic network time-series response plots were displayed in Figure 18 for the NAR-LM model, Figure 19 for the NAR-BR model, Figure 20 for the NAR-SCG model, respectively, 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.

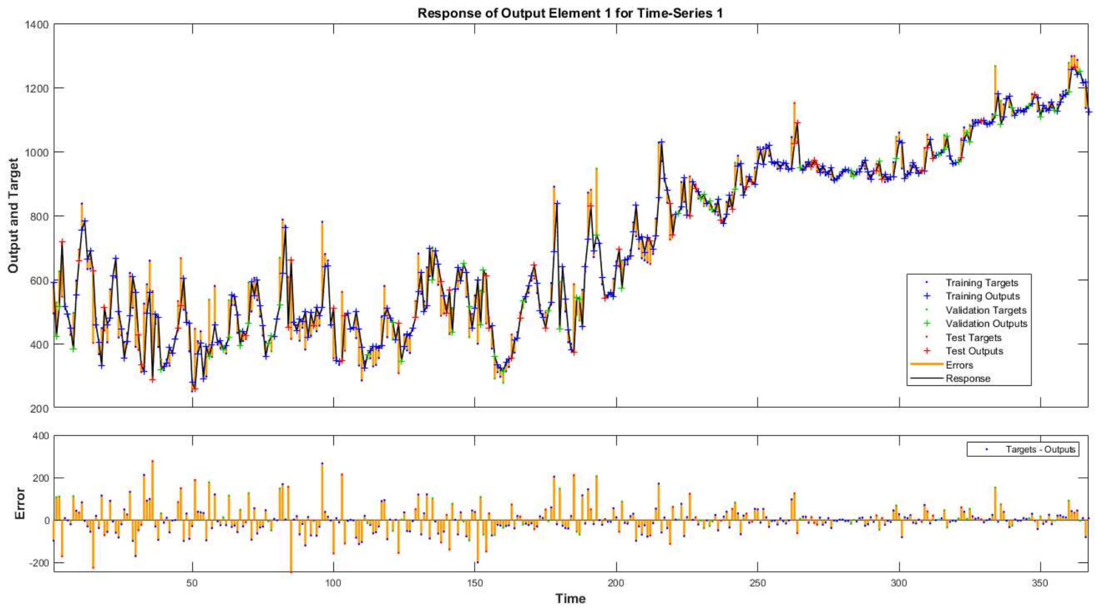

Fig. 18 Network Time-Series Response of the NAR-LM Model (Source: own work)

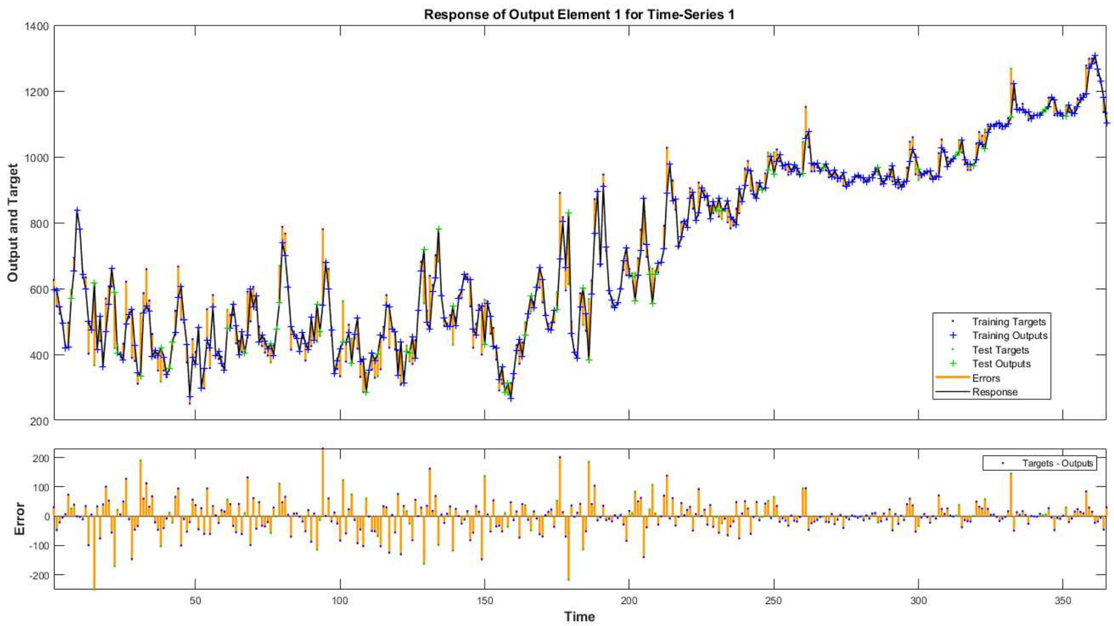

Fig. 19 Time-Series Response of the NAR-BR Model (Source: own work)

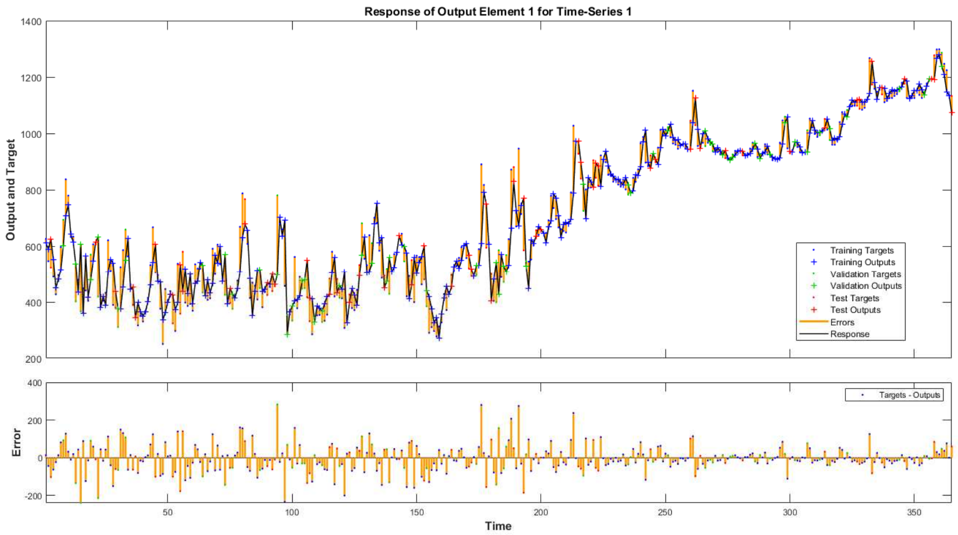

Fig. 20 Time-Series Response of the NAR-SCG Model (Source: own work)

4.6 NAR Neural Network 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 NAR-LM model (Figure 21), the NAR-BR model (Figure 22), and the NAR-SCG model (Figure 23) respectively, 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 suggest 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, retrain the network will change the initial weights and biases of the network, and may produce an improved network after retraining.

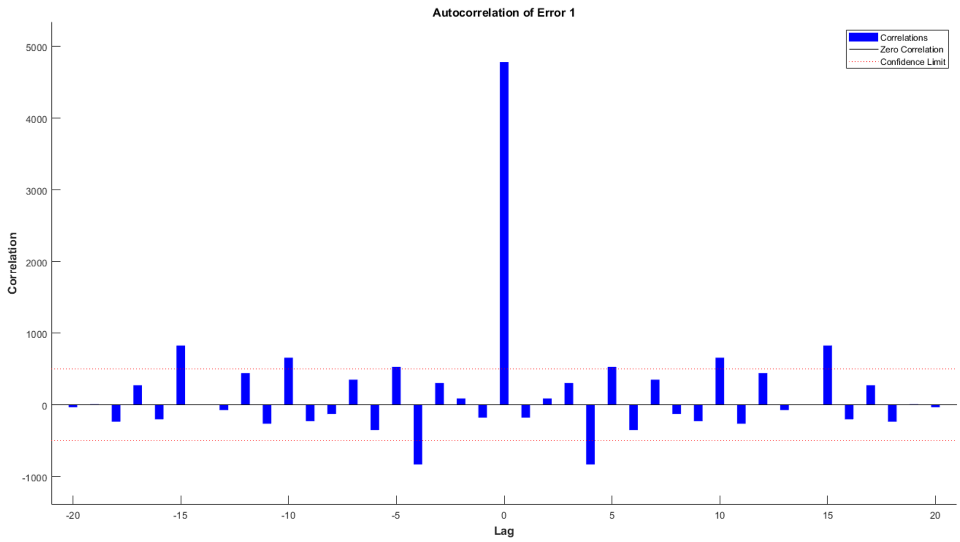

Fig. 21 Error Autocorrelation of the NAR-LM Model (Source: own work)

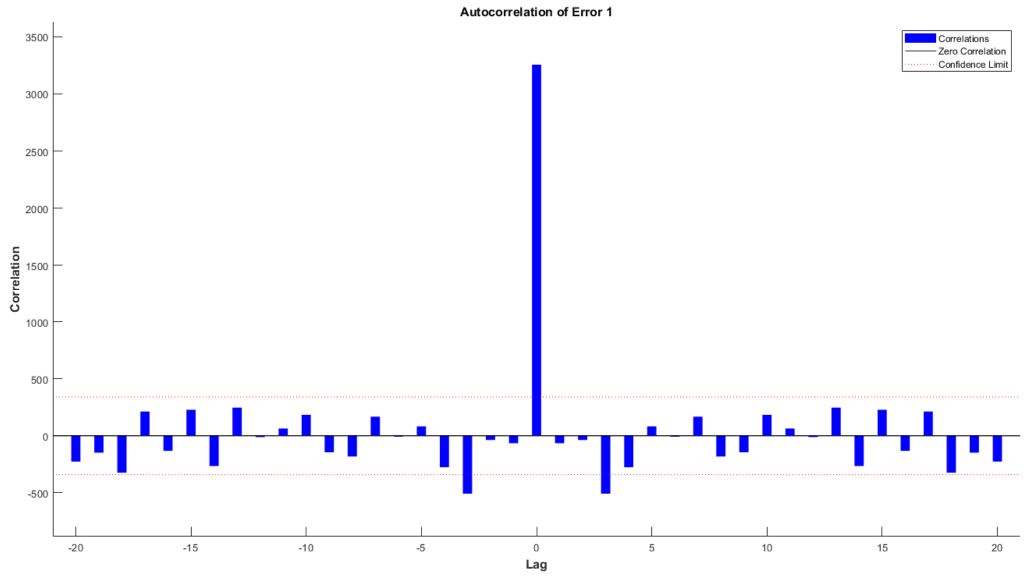

Fig. 22 Error Autocorrelation of the NAR-BR Model (Source: own work)

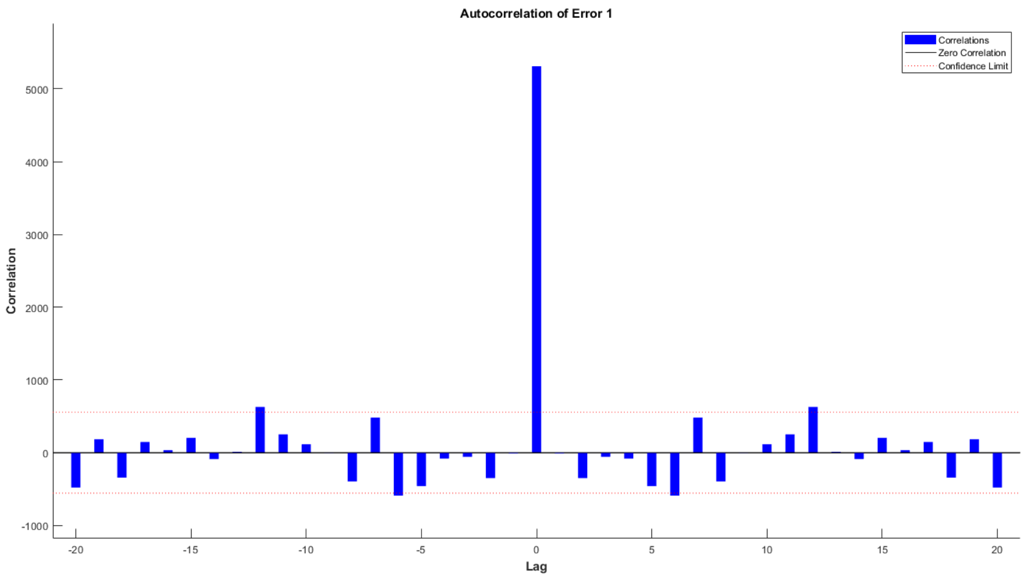

Fig. 23 Error Autocorrelation of the NAR-SCG Model (Source: own work)

4.7 Comparison of the NAR-LM, NAR-BR, and NAR-SCG Models

The purpose of this study was to evaluate the accuracy of the NAR-LM, NAR-BR, and NAR-SCG models used in monthly global price of bananas forecast. The performance was tested with the MSE and regression R values. Results and comparisons were summarized in Table 4. In this analytical work, the division of the time series dataset was 70% for the training data, 15% for the validation data, and 15% for the testing data. Randomly, 371 data samples were divided into 259 data for the training, 56 data for validation, and 56 data for testing.

The training in the NAR-LM model yielded 96.755% accuracy for the all samples, 97.803% for the training, 97.063% for the validation, and 91.543% for the testing respectively. The training in the NAR-BR model yielded 97.801% accuracy for the all samples, 98.362% for the training, and 94.369% for the testing respectively. The training in the NAR-SCG model yielded 96.399%

accuracy for the all samples, 96.692% for the training, 95.153% for the validation, and 96.527% for the testing.

In the training phase, the NAR-BR model yielded the lowest MSE compared to the NAR-LM and NAR-SCG models. The regression R value computed by the NAR-BR model was also higher compared to the NAR-LM and NAR-SCG models for monthly global price of bananas forecast. These comparative results showed that the NAR-BR model with 13 neurons in the hidden layer and 6 time delays yielded higher accuracy than the NAR-LM model with 12 neurons in the hidden layer and 4 time delays and NAR-SCG model with 12 neurons in the hidden layer and 6 time delays.

Table 4. Comparisons of the NAR-LM, NAR-BR, and NAR-SCG Models

	Target Values	MSE			Regression R		
		NAR-LM	NAR-BR	NAR-SCG	NAR-LM	NAR-BR	NAR-SCG
Training	259	3225.23531	2422.92329	4813.51336	0.97803	0.98362	0.96692
Validation	56	5770.60999	------	7404.38253	0.97063	------	0.95153
Testing	56	11059.28376	7952.95818	5512.11972	0.91543	0.94369	0.96527
All	371				0.96755	0.97801	0.96399

(Source: own work)

5. Conclusion

Bananas rank as a leading crop in world agricultural production and trade. Assuming normal weather conditions and no further spread of banana plant diseases, bananas have seen rapidly increasing production and trade volumes in response to fast population growth in producing countries as well as expanding global import demand. Obviously, the future is more intensive in knowledge, more understanding of the natural complexities of living systems. In order to bring together a wide variety of perspectives and concepts, it requires holistic solutions that involve working across disciplines, principles and methods to support interdisciplinary and transdisciplinary tasks, 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 banana markets, there is a lack of studies in the technical literature available on global price forecasting schemes. Prediction is a kind of dynamic filtering, in which past values of one or more-time series are used to predict future values. Currently, neural network is one of the most popular machine learning methods, which is able to do prediction tasks in a more reliable manner. With time series data, lagged values of the time series can be used as inputs to a neural network, the NAR neural network was applied to time series prediction using its past values of a univariate time series in this study.

Empirically, the results revealed that the NAR-BR model with 13 neurons in the hidden layer and 6 time delays yielded higher accuracy than the NAR-LM model with 12 neurons in the hidden layer and 4 time delays and NAR-SCG model with 12 neurons in the hidden layer and 6 time delays. Hence, the NAR-BR neural network model not only can provide information which are important in decision making process related to the future global price of bananas change impacts, but also can be employed in forecasting the future performance for global price of bananas change outcomes.

In order to sustain these observations, research programs utilizing the resulting data should be able to improve significantly our understanding and narrow projections of future global price of banana. For the further research tasks, the nonlinear autoregressive exogenous (NARX) neural network can be considered not only past information of the same time series (monthly global price of bananas), but also current and past information of the externally determined time series that influences the time series of interest (i.e., bananas production, weather information, etc.) in terms of forecasting accuracy consideration.

References

[1] 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, pp. 13-20. Retrieved from: http://www.fao.org/3/ca6911en/CA6911EN_TR4EN.pdf

[2] Beale, Mark Hudson, Hagan, Martin T., & Demuth, Howard B. (2019). Deep Learning Toolbox™: Getting Started Guide. Natick, MA: The MathWorks, Inc.

[3] Benrhmach, G., Namir, K., Namir, A., and Bouyaghroumni, J. (2020). Nonlinear autoregressive neural network and extended Kalman filters do prediction of financial time series. Journal of Applied Mathematics, Vol. 2020, Article ID 5057801, 1-6.

[4] Eyduran, S. P., Akın, M., Eyduran, E., Çelik, S., Ertürk, Y. E., and Ercişli, S. (2020). Forecasting banana harvest area and production in Turkey using time series analysis. *Erwerbs-Obstbau*, 62, 281–291.

[5] Fatin, Z. N., Titik, E., and Mulyatno, S. B. (2020). The analysis of price and market integration of banana commodities in Lampung, Indonesia. *Russian Journal of Agricultural and Socio-Economic Sciences*, 3(99), 2020, 61-68.

[6] Gavin, Henri P. (2020). The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering Duke University. 19 pages. Retrieved from: http://people.duke.edu/~hpgavin/ce281/lm.pdf

[7] Hamjah, M. A. (2014). Forecasting major fruit crops productions in Bangladesh using Box-Jenkins ARIMA Model. *Journal of Economics and Sustainable Development*, 5(7), 96-107.

[8] Hossain, M. M., Abdulla, F., and Majumder, A. K. (2016). Forecasting of banana production in Bangladesh. American Journal of Agricultural and Biological Sciences, 11(2), 93-99.

[9] Levenberg, Kenneth *(1944).* A Method for the Solution of Certain Non-Linear Problems in Least Squares*. Quarterly of Applied Mathematics, 2(2): 164–168.* doi*:*10.1090/qam/10666

[10] MacKay, David J. C. (1992). Bayesian Interpolation. Neural Computation, Vol. 4, No. 3, pp. 415–447. https://doi.org/10.1162/neco.1992.4.3.415

[11] 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 pp. Retrieved from: https://www.marketreportsworld.com/banana-market-13487750

[12] Marquardt, Donald W. (1963). An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics, Vol. 11, No. 2, pp. 431-441.

[13] Moller, Martin Fodslette. (1993). A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, Volume 6, Issue 4, pp. 525-533. https://doi.org/10.1016/S0893-6080(05)80056-5

[14] Omar, M. I., Dewan, M. F., and Hoq, M. S. (2014). Analysis of price forecasting and spatial co-integration of banana in Bangladesh. *European Journal of Business and Management*, 6(7), 244-255.

[15] Rathod, S., Mishra, G. C., and Singh, K. N. (2017). Hybrid time series models for forecasting banana production in Karnataka State, India. *Journal of the Indian Society of Agricultural Statistics*, 71(3), 193–200.

[16] Rathod, S., and Mishra, G. C. (2018). Statistical models for forecasting mango and banana yield of Karnataka, India. *Journal of Agricultural Science and Technology*, 20, 803-816.

[17] Rebortera, M. A., and Fajardo, A. C. (2019). An enhanced deep learning approach in forecasting banana harvest yields. *International Journal of Advanced Computer Science and Applications*, 10(9), 275-280.

[18] Ruiz, A., Fobelets, V., Grosscurt, C., Galgani, P., Lord, R., Hardwicke, R., Tarin, M., P, G., McNeil, D., and Aird, S. (2017). The external costs of banana production: A global study. Research Report Prepared for Fairtrade International, True Price & Trucost. Retrieved from: http://makefruitfair.org/wp-content/uploads/2017/07/170224_Research_Report_External_Cost_of_Bananas_-_final.pd

[19] Voora, V., Larrea, C., and Bermudez, S. (2020). Global market report: Bananas. Sustainable Commodities Marketplace Series 2019, The International Institute for Sustainable Development, 12

pp. Retrieved from: https://www.iisd.org/system/files/publications/ssi-global-market-report-banana.pdf

[20] Sariev, Eduard, & Germano, Guido. (2020). Bayesian regularized artificial neural networks for the estimation of the probability of default. Quantitative Finance, Vol. 20, No. 2, pp. 311–328. https://doi.org/10.1080/14697688.2019.1633014

[21] Zhang, G. Peter, Patuwo, B. Eddy, and Hu, Michael Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14, 35-62.

[22] Zhang, G. Peter. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175.