Journal of Computing and Applied Informatics (JoCAI) Vol. 5, No. 2, 2021 | 85 – 101

DATA SCIENCE
Journal of Computing and Applied Informatics

Time Series Forecasting of Global Price of Soybeans Using a Hybrid SARIMA and NARNN Model

Yeong Nain Chi

Department of Agriculture, Food and Resource Sciences, University of Maryland Eastern Shore, MD, USA

Abstract. The primary purpose of this study was to demonstrate the role of time series models in predicting process applying advanced techniques using the time series data of monthly global price of soybeans from January 1990 to January 2021. The univariate SARIMA, NARNN-LM and Hybrid-LM models were compared with mixed conclusion in terms of the superiority in forecasting performance for the target time series. The mean square error (MSE) was used to compare the forecasting performance among the three models. The comparative results revealed that the Hybrid-LM model with 8 neurons in the hidden layer and 3 time delays (MSE = 186.43259) yielded higher accuracy than the NARNN-LM model with 8 neurons in the hidden layer and 3 time delays (MSE = 222.42221), and the SARIMA, ARIMA$(0,1,3)(0,0,2)_{12}$, model (MSE = 284.966473) in this study. The results of this study showed that the Hybrid-LM model, a combination of SARIMA and NARNN models, has both linear and nonlinear modelling capabilities which can be a better choice for modelling the target time series. This study could provide an integrated modelling approach as a decision-making supportive method for formulating price forecast of soybeans for the global soybean market.

Keyword: Global Price, Soybeans, Time series forecasting, SARIMA, NARNN, Hybrid.

Received 24 Februari 2021 | Revised 31 July 2021 | Accepted 31 July 2021

1 Introduction

Globally, there is a huge demand for soybeans, that are used widely in livestock feed, food, fuel and industrial products. China imported approximately 93.5 metric tons of soybeans in 2016, which accounted for 65% of the world total soybean imports [1]. China is the world's largest importer of soybeans, while the U.S. and Brazil account for about 80% of global exports of soybeans [2]. According to Trading Economics, the U.S., Brazil, Argentina and Paraguay are the biggest producers and exporters of soybeans in the world, concentrating more than 80% of total production and 90% of total exports. China is the biggest importer of soybeans (60% of total

*Corresponding author at: 11868 College Backbone Road, 1102 Trigg Hall, Department of Agriculture, Food and Resource Sciences, University of Maryland Eastern Shore, Princess Anne, MD 21853, USA

E-mail address: ychi@umes.edu

ISSN: 2580-6769 | e-ISSN: 2580-829X | https://doi.org/10.32734/jocai.v5.i2-5674
Journal Homepage: https://jocai.usu.ac.id

imports) followed by the European Union, Mexico, Japan and Taiwan. (https://tradingeconomics.com/commodity/soybeans),

Due to the availability of vast agricultural land, the U.S. has managed to become the highest producer of soybeans globally with over 117 million tons of grain being harvested as of 2016. The growth of soybean production in the U.S. is attributed to shifting land from crops such as wheat and corn to soybeans. In 2018-19, soybean harvest was recorded at 35.6 million hectares, despite the increase of Chinese tariffs against U.S. soybeans [2]. However, despite strong market potential, soybeans remain a marginally attractive commercial crop due to a high cost base, poor transport infrastructure and uncertain trade policies. It is also still not very attractive crop for smallholders and small enterprises as they lack appropriate inputs, expertise and a sure market.

Recently, the increase in tariffs for soybeans has presented unique challenges for the U.S. and China with both markets seeing significant challenges and risks. Thus, China gradually increased its imports of soybeans from Brazil and Argentina, in a bid to counter the tariff imposed by the U.S. In the long-term, the negative impact of the U.S.-China trade war on the U.S. soybean industry will be mitigated by the increasing demand for U.S. soybeans of other importing countries. However, if trade agreements cannot be reached between the U.S. and China, the soybean industry will continue to face a very difficult time in the future.

However, price forecast is vital to facilitate efficient decisions and will play a major role in coordinating the supply and demand of soybeans globally. Not only in terms of demand and supply framework, but also associated with imports and exports competition of soybeans in the global soybeans market, price forecast still plays an important role for the future trends of soybeans consumption and production in the world. Time series forecasting is the use of a model to predict future values based on previously observed values. Furthermore, neural network models have become one of the most popular trends for time series modeling and forecasting.

Autoregressive Integrated Moving Average (ARIMA) is one of the most popular linear models in time series forecasting. Neural network models could be a potential alternative to the traditional linear time series models. Recently, many studies have integrated time series analysis and neural network framework together, a combination of Seasonal Autoregressive Integrated Moving Average (SARIMA) and Nonlinear Autoregressive Neural Network (NARNN) model, in medical sciences [3] [4] [5] [6], business [7], tourism [8]. From these studies reported, this hybrid model can explore the reliable model to forecast the time series for a better performance. Specifically, it can take advantage of the unique strength of SARIMA and NARNN models in linear and nonlinear modeling, and can be an effective way to improve forecasting accuracy achieved by either of the models used separately. Thus, the primary purpose of this study was to demonstrate

the role of time series models in predicting process applying advanced techniques using the time series data of monthly global price of soybeans from January 1990 to January 2021.

2 Materials

The long-term records of monthly global price of soybeans **(**units: U.S. dollars per metric ton, not seasonally adjusted) from January 1990 to January 2021 (Figure 1), is available to the public from International Monetary Fund, Global Price of Soybeans [PSOYBUSDM], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/PSOYBUSDM. Average monthly global price of soybeans from January 1990 to January 2021 was $303.37 U.S. dollars per metric ton with the standard deviation of $109.73 (Minimum: $158.31, Maximum: $622.91, and Median: $278.04).

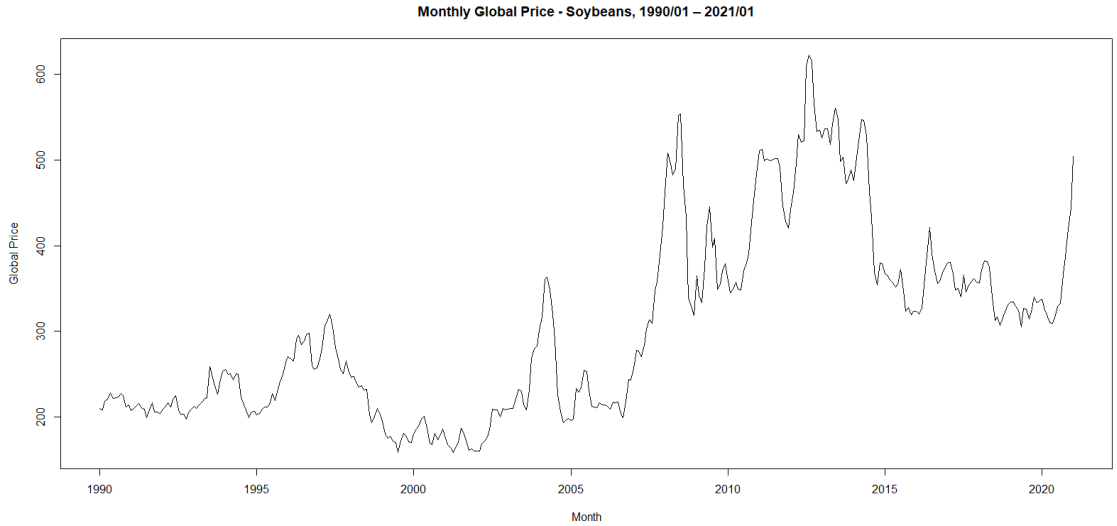

Figure 1. Time Series Plot of Monthly Global Price of Soybeans, January 1990 ~ January 2021 (Source: own work)

3 Methods

3.1 Seasonal ARIMA (SARIMA) Model

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. For the Autoregressive Integrated Moving Average (ARIMA) model, the ARIMA(p, d, q) model can be expressed as:

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

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), $\phi = (\phi_1, \phi_2, \cdots, \phi_p)$ is the vector of model coefficients for

the autoregressive process, $\theta = (\theta_1, \theta_2, \cdots, \theta_q)$ is the vector of model coefficients for the moving average process, and e_t is the residual error (i.e., white noise) [9]. The purpose of each of these parts is to make the model better fit to predict future points in a time series [9].

The SARIMA model is an extension of the ARIMA model that explicitly supports univariate time series with a seasonal component. Statistically, ARIMA(p, d, q)(P, D, Q)$_S$ is used to represent the SARIMA model, where P = the order of the seasonal autoregressive process, D = the number of seasonal differences applied to the time series, Q = the order of the seasonal moving average process, and S = the seasonality of the model, i.e., the number of time steps for a single seasonal period.

In time series analysis, the Box-Jenkins methodology [10] refers to a systematic method of identifying, estimating, checking, and forecasting ARIMA models [11], that can be applied to find the best fit of a time series. The Box-Jenkins methodology also can be used as the process for estimating the SARIMA 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 of the SARIMA model.

In order to figure out good parameters for the model, Akaike's Information Criterion (AIC) or Bayesian Information Criterion (BIC) can be used to determine the orders of a SARIMA model that is obtained by minimizing the AIC or BIC value. In this study, 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 estimate the model parameters to fit the SARIMA to achieve the purpose of this study.

3.2 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, $(y_{t-1}, y_{t-2}, \cdots, y_{t-p})$. Thus, the AR process of order p, AR(p), is defined by the equation:

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

where $\phi = (\phi_1, \phi_2, \cdots, \phi_p)$ is the vector of model coefficients for the autoregressive process, and e_t is white noise, i.e., $e_t \sim N(0, \sigma^2)$ [9]. The NARNN is a natural generalization of the classic linear AR(p) process. The NARNN of order p can be expressed as:

$$y_t = \Phi(y_{t-1}, y_{t-2}, \cdots, y_{t-p}, w) + \varepsilon_t \quad (3)$$

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, $(y_{t-1}, y_{t-2}, \cdots, y_{t-p})$, to the future value, y_t, which is equivalent to a nonlinear autoregressive model [12].

With the time series data, lagged values of the time series can be used as inputs to a neural network, so-called this the NARNN model. Mathematically, the NARNN model [13] can be written by the equation of the form as:

$$y_t = a_0 + \sum_{j=1}^{k} w_j \, \Phi(b_{0j} + \sum_{i=1}^{d} w_{ij} y_{t-i}) + \varepsilon_t \quad (4)$$

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

The most common learning rules for the NARNN model are the Levenberg-Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient training algorithms. In this study, the Levenberg-Marquardt (LM) algorithm was considered, because 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 [14].

The LM algorithm, first published by Levenberg [15] and then rediscovered by Marquardt [16], has become a standard technique for nonlinear least-squares problems. It can be thought of as a combination of the steepest descent and the Gauss-Newton methods. 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 [17],

$$F(x) = (1/2) \, \Sigma_{i=1}^{n} [f_i(x)]^2 \quad (5)$$

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

$$(J_i^T J_i + \lambda_i I) \, \varphi_i = - J_i^T f_i \quad (6)$$

where J_i is the Jacobian of f_i, I is the identity matrix, and λ_i are the non-negative scalars, called combination coefficient. In the Levenberg-Marquardt algorithm, for some scalar $\Delta > 0$ related to λ_i, the vector φ_i is the solution of the constrained subproblem of minimizing $(1/2) \, \| J_i \, \varphi + f_i \|_2^2$ subject to $\| \varphi \|_2 \leq \Delta$ [18]. In this study, MATLAB (2019a) was used as the tool to estimate the NARNN model using the LM training algorithm for the target time series monthly global price of soybeans.

3.3 Hybrid SARIMA and NARNN (Hybrid) Model

The SARIMA and NARNN models are good at modelling linear and nonlinear problems for the time series, respectively. However, using the hybrid model, a combination of SARIMA and NARNN model 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, which can be expressed as follows:

$$y_t = f(L_t, N_t) \tag{7}$$

where linear component is represented by L_t, and nonlinear component is shown by N_t. Assuming that the linear and nonlinear components in the time series have simply additive relationships. Zhang [12] states that the time series can be considered as a combination of a linear and nonlinear components as follows:

$$y_t = L_t + N_t \tag{8}$$

These two components should be estimated from the time series. First, the linear component will be modelled by the SARIMA model in this study. Then, the residuals from the SARIMA model will have only the nonlinear relationship, which can be obtained by taking difference of actual values and predicted values as follows:

$$e_t = y_t - \hat{L}_t \tag{9}$$

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:

$$\hat{N}_t = e_t = f(e_{t-1}, e_{t-2}, \cdots, e_{t-n}) + \varepsilon_t \tag{10}$$

where *f* is the transformation function modelled by the NARNN model, and ε_t is the random error. The forecast from the SARIMA and NARNN models are combined to obtain the forecast of the time series $\hat{y}_t$ which is denoted by

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \tag{11}$$

In order to find the results for the Hybrid model, MATLAB (2019a) was used as the tool using the LM training algorithm to analyze time series monthly global price of soybeans to achieve the purpose of this study.

4 Results

4.1 Seasonal ARIMA (SARIMA) Model

R 4.0.2 for Windows is 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 monthly global price of soybeans from January 1990 to January 2021 in this study. The function “decompose()”

in R was applied to estimate the seasonal component, trend component and irregular component of a seasonal time series (Figure 2). The estimated seasonal component definitely displayed seasonality with a pattern recurrence occurring once every 12 months.

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.

Since the ACF of the time series, seasonal adjusted monthly global price of soybeans from January 1990 to January 2021, showed strong positive statistically significant correlations at up to 26 lags that never decay to zero, and suggested that the time series was non-stationary. In terms of non-stationary time series, differencing can be used to transform a non-stationary time series into a stationary one. When both trend and seasonality are present, thus, both a non-seasonal first difference and a seasonal difference need to apply. The first difference of a time series is the time series of changes from one period to the next. Notice that the graph of the first difference of the time series looked approximately stationary. According to the Augmented Dickey-Fuller Test, Dickey-Fuller = -7.5714 with lag order = 7 and the p-value of the test was smaller than 0.01. It rejected the null hypothesis that is non-stationary, and also suggested that the first difference of the time series was stationary.

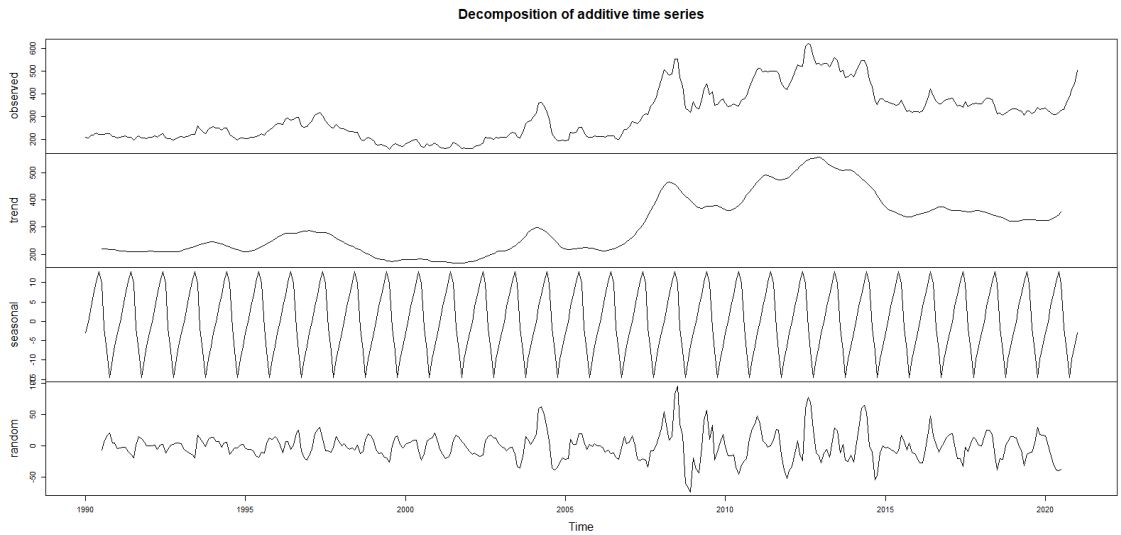

Figure 2. Decomposition of Monthly Global Price of Soybeans, January 1990 ~ January 2021 (Source: own work)

The ACF of first difference shown in Figure 3 showed a steady decay after the first few lags and bounce around between being positive and negative statistically significant. The corresponding PACF of first difference in Figure 4 showed a significant positive spike at the first lag followed by correlations that were statistically significant.

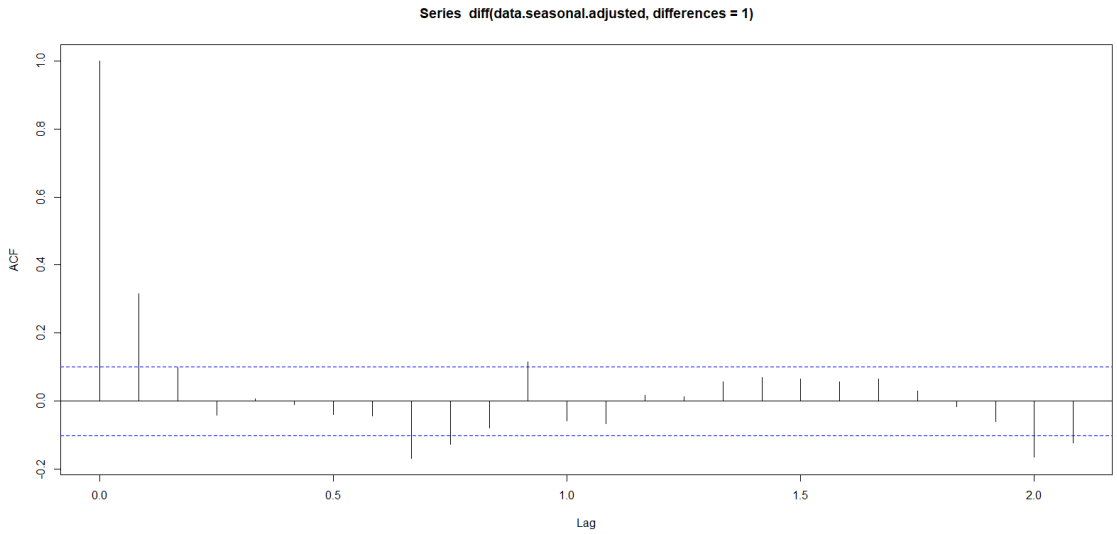

Figure 3. ACF Plot of First Difference of Seasonal Adjusted Monthly Global Price of Soybeans, January 1990 ~ January 2021 (Source: own work)

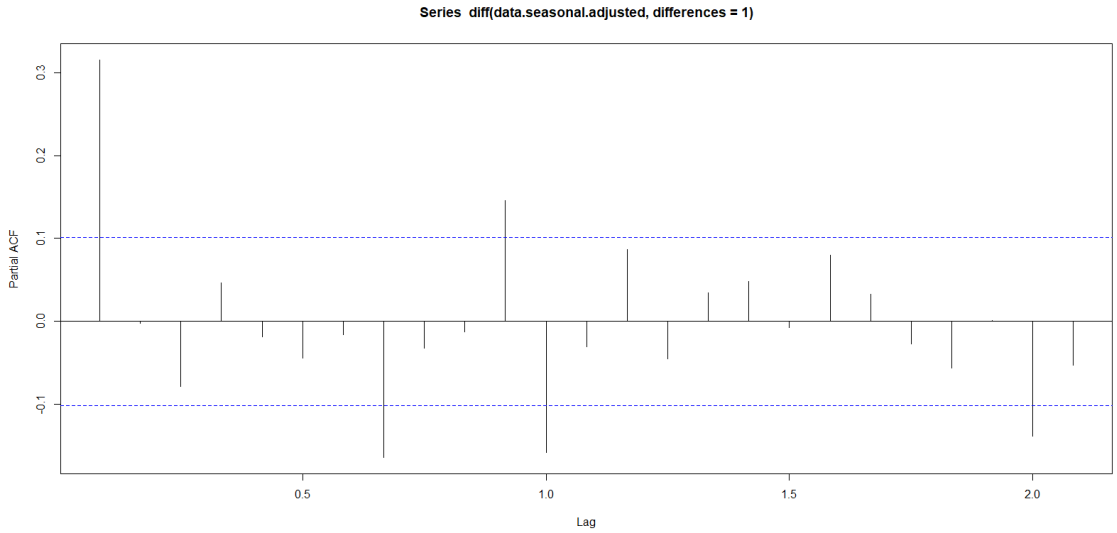

Figure 4. PACF Plot of First Difference of Seasonal Adjusted Monthly Global Price of Soybeans, January 1990 ~ January 2021 (Source: own work)

Seasonal differencing is defined as a difference between a value and a value with lag that is a multiple of seasonality (S). In this case, S = 12 (months per year) is the span of the periodic seasonal behavior. The graph of the 12^{th} difference of the time series looked approximately stationary. Meanwhile, the test statistic of the Augmented Dickey-Fuller Test was Dickey-Fuller = -26.789 with lag order = 7 and the p-value of the test was smaller than 0.01. It rejected the null hypothesis that is non-stationary, and also suggested that the 12^{th} first difference of the time series was stationary. Figure 5 showed that ACF most likely a steady decay after the first few lags and bounce around between being positive and negative statistically significant. Meanwhile, Figure 6 showed what PACF mostly looks like a steady negative decay in the partial correlations toward zero.

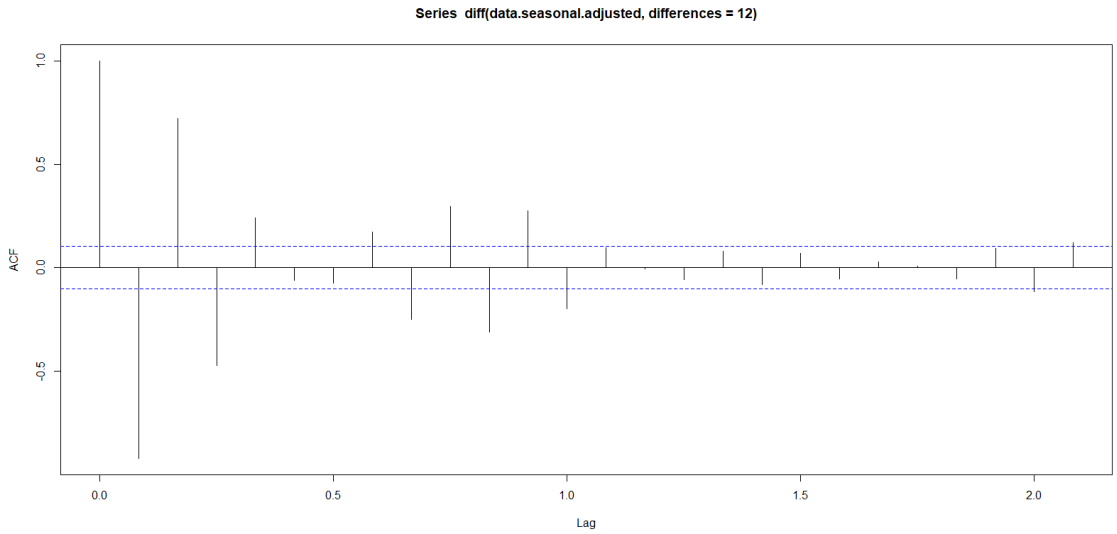

Figure 5. ACF Plot of 12th Difference of Seasonal Adjusted Monthly Global Price of Soybeans, January 1990 ~ January 2021 (Source: own work)

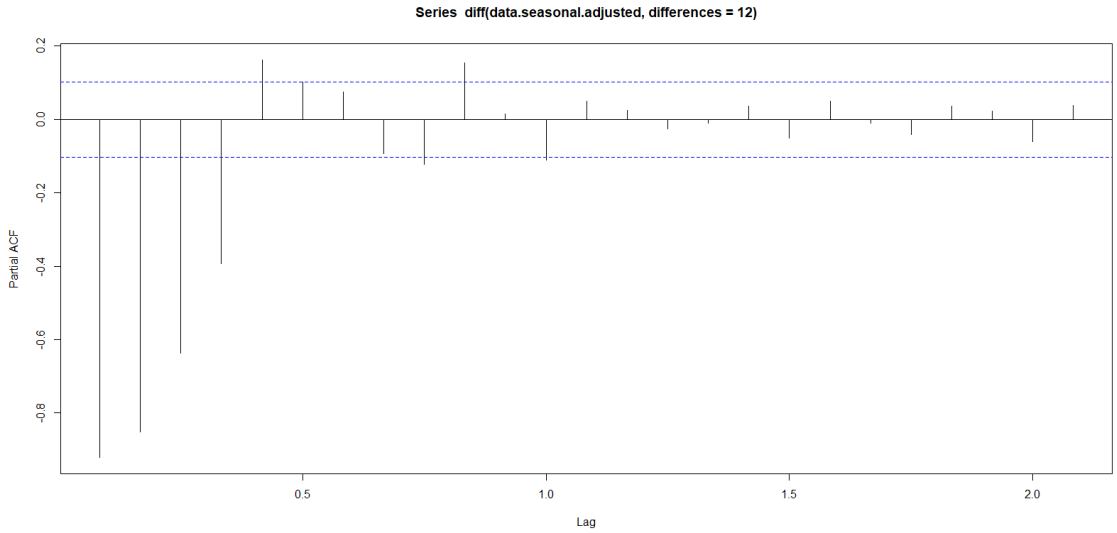

Figure 6. PACF Plot of 12th Difference of Seasonal Adjusted Monthly Global Price of Soybeans, January 1990 ~ January 2021 (Source: own work)

Empirically, the choice of the model order is somewhat arbitrary. In this study, the auto.arima() function from the "forecast" package in R 4.0.2 for Windows was employed to identify both the structure of the 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 SARIMA model. Consequently, the ARIMA(0,1,3)(0,0,2)$_{12}$ model was selected to be the best fit model for the time series, according to the lowest AIC value (= 3172.57) in this study. Given this option, the ARIMA(0,1,3)(0,0,2)$_{12}$ model was chosen for further forecasting process, and the parameters of the ARIMA(0,1,3)(0,0,2)$_{12}$ model were presented in Table 1.

Table 1. Parameters of the ARIMA(0,1,3)(0,0,2)$_{12}$ Model

Parameter	Estimate	Standard Error
Difference	1	
MA Lag 1	0.3458	0.0529
MA Lag 2	0.1165	0.0544
MA Lag 3	-0.0785	0.0545
SMA1	-0.1388	0.0539
SMA2	-0.1696	0.0568
Sigma^2 estimated as 289.6, Log Likelihood = -1580.29		
AIC = 3172.57, AICc = 3172.81, BIC = 3196.09		
RMSE = 16.88095, MSE = 284.966473, MAE = 11.61128, MAPE = 3.720898		

Source: own work

The Ljung-Box Q-test [19] is a diagnostic tool used to test the lack of fit of a time series model. In this example, the test statistic of the Ljung-Box Q-test was Q = 31.958 with 19 degrees of freedom and the p-value of the test was 0.03159 (model degrees of freedom: 5, total lags used: 24), indicating that the residuals were random and that the model provided an adequate fit to the data relatively. Figure 7 illustrated that the black line represented the visuals of monthly global price of soybeans dataset without forecasting and the red line represented the visuals of monthly global price of soybeans dataset with forecasted values. Forecasting process with the ARIMA(0,1,3)(0,0,2)$_{12}$ model indicated a good fit of the SARIMA model for forecasting in this study.

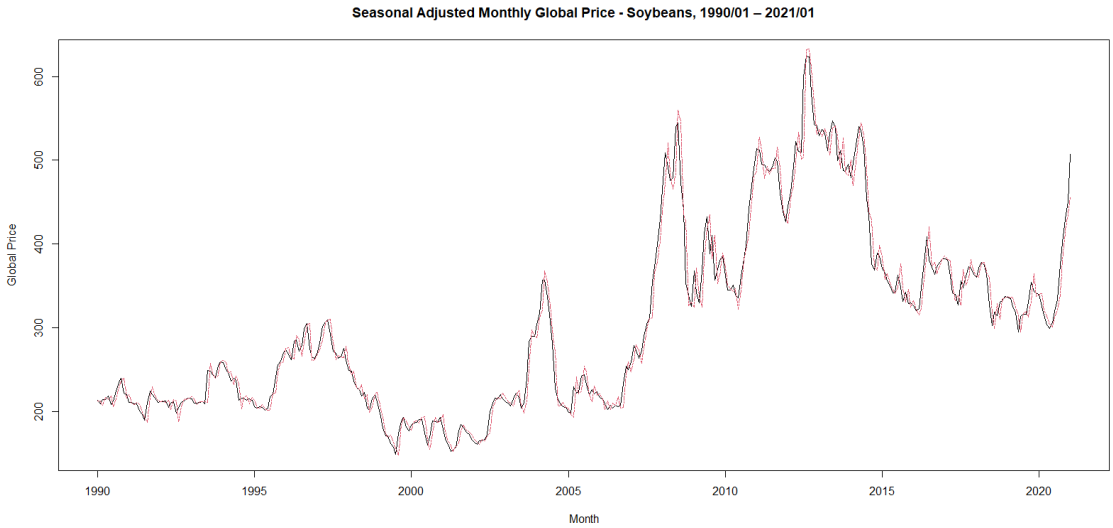

Figure 7. Observed and Forecasted Monthly Global Price of Soybeans (Source: own work)

4.2 Nonlinear Autoregressive Neural Network (NARNN) Model

In MATLAB (2019a), the NARNN model applied to time series prediction using its past values of a univariate time series can be expressed as follows:

$$y(t) = \Phi(y(t-1), y(t-2), \cdots, y(t-d)) + e(t) \quad (12)$$

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), ⋯, y(t-d)) [1]. The function Φ(·) 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 and predicted output values (i.e., MSE) [20].

In this study, the NARNN model was applied to model time series monthly global price of soybeans. 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: 373 timesteps of one element, monthly global price of soybeans from January 1990 to January 2021.

Three kinds of target timesteps were set aside for the training, validation and testing phases in this case study. 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 [20]. 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, 373 data samples were divided into 261 data for the training, 56 data for the validation, and 56 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 their performance were evaluated under the optimal neural network structure. The prediction performance of the models was evaluated by its 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 NARNN model with 8 neurons in the hidden layer and 3 time delays provided the best performance (MSE = 222.42221) using the LM algorithm (NARNN-LM).

The LM algorithm typically requires more memory but less time. Training automatically stopped when generalization stop improving, as indicated by an increase in the MSE of the validation samples [1]. The results revealed the training progress using the LM algorithm stopped when the validation error increased for six iterations with Performance = 210, Gradient = 137, and Mu = 1.00 at epoch 16. In terms of the processing time, the LM algorithm took 0:00:00 during training.

The performance plot is a useful diagnostic tool to plot the training, validation, and testing errors to check the progress of training. It also illustrates that the training stops when the validation error increased at the circled epoch. The performance was evaluated by taking MSE and epochs after the training was completed, and then the values were generated. As illustrated, the best performance for the validation phase was 429.9167 at epoch 6 for the NARNN-LM model. The results 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 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. Otherwise, there is no linear relationship between the outputs and the targets. As illustrated in Figure 8, the regression R value for the training phase was 0.9902, for the validation phase was 0.98367, for the testing phase was 0.984063, and for the all samples was 0.98719, respectively, indicated good predictive abilities of the NARNN-LM model for new datasets.

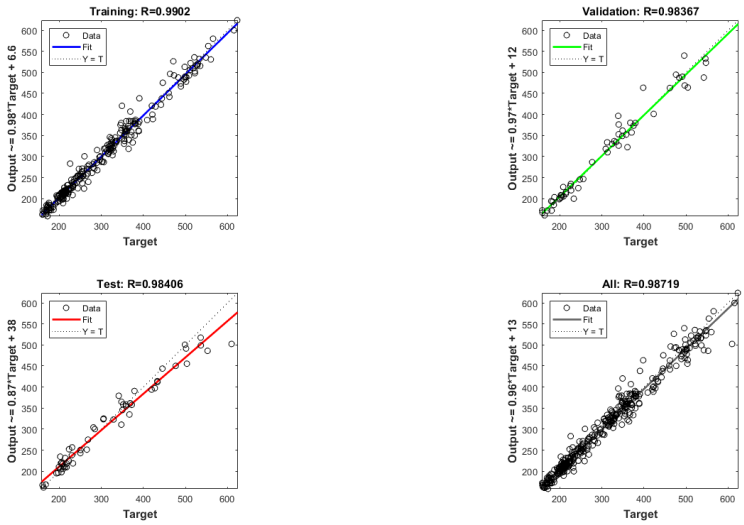

Figure 8. Regression Plots of the NARNN-LM Model (Source: own work)

The dynamic network time-series response plots were displayed in Figure 9 for the NARNN-LM 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 NARNN-LM model was able to predict the time series over the simulation period efficiently.

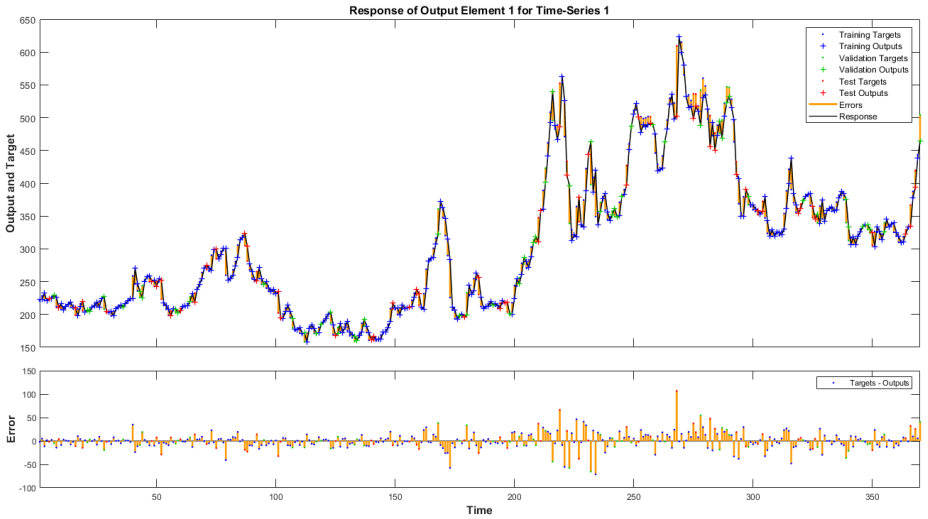

Figure 9. Network Time-Series Response of the NARNN-LM Model (Source: own work)

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 (i.e., 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-LM model (Figure 10) except for the one at zero lag, almost all fell approximately within the 95% confidence limits around zero, so the model 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.

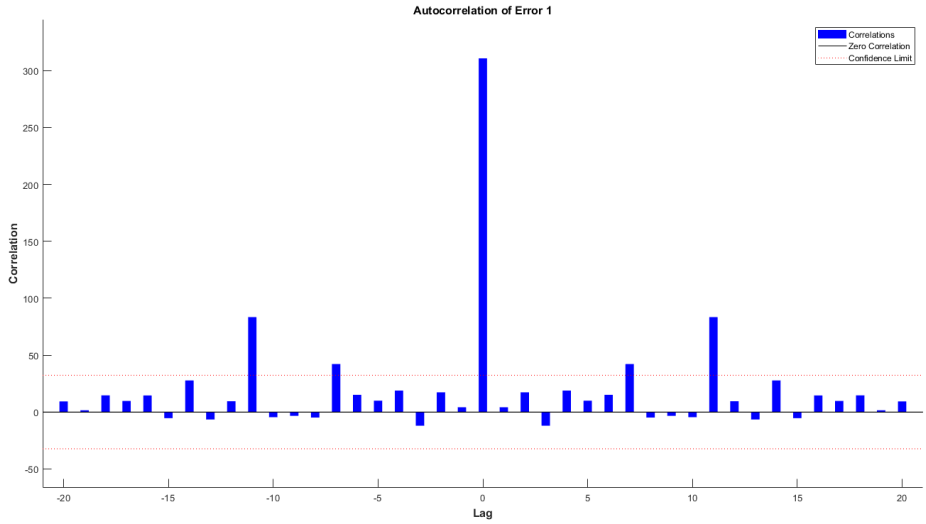

Figure 10. Error Autocorrelation of the NARNN-LM Model (Source: own work)

4.3 Hybrid SARIMA and NARNN (Hybrid) Model

In MATLAB (2019a), the Hybrid model applied to time series prediction using its past residuals from the SARIMA model can be expressed as follows:

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

where e(t) is the residual of the time series at time t, d is the time delay, and ε(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), ⋯, e(t-d)) [20].

Similarly, the development of the optimal architecture for the Hybrid model requires determination of time delays, the number of hidden neurons, and an efficient training algorithm. According to the results of the error analysis using the MATLAB (2019a) Neural Network Toolbox, it showed that the Hybrid model with 8 neurons in the hidden layer and 3 time delays also provided the best performance (MSE = 186.43259) with the LM algorithm (Hybrid-LM). At the same time, the training progress using the LM algorithm for the Hybrid-LM model stopped when the validation error increased for six iterations with Performance = 180, Gradient = 169, and Mu = 1.00 at epoch 19.

The performance plot illustrates that the training stops when the validation error increased at the circled epoch, and evaluated by taking MSE and epochs after the training was completed, and then the values were generated. As illustrated, the best performance for the validation phase was 350.6018 at epoch 13 for the Hybrid-LM 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.

The dynamic network time-series response plots were displayed in Figure 11 for the Hybrid-LM 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-LM model was able to predict the time series over the simulation period efficiently as well. For the Hybrid-LM model, the correlations except for the one at zero lag, all fell approximately within the 95% confidence limits around zero, so the model was to be adequate (Figure 12).

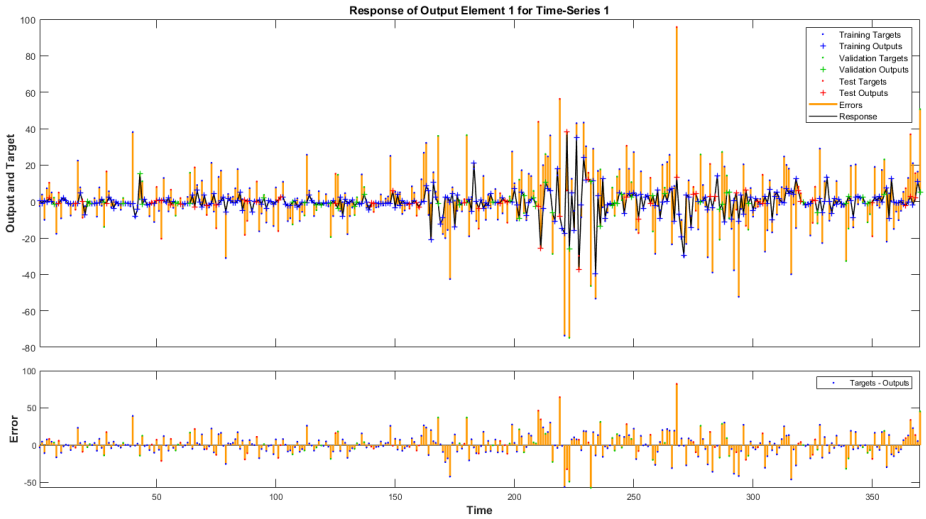

Figure 11. Network Time-Series Response of the Hybrid-LM Model (Source: own work)

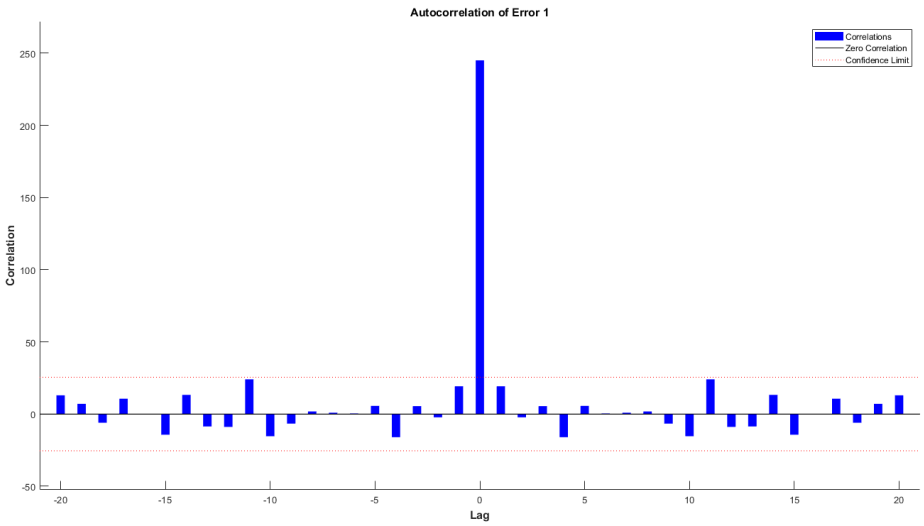

Figure 12. Error Autocorrelation of the Hybrid-LM Model (Source: own work)

5 Conclusion

Prices forecast aids farmers and industries to plan for future farming activities and budgeting which is largely depending upon expected future prices. Therefore, forecasting future price of soybeans has become a crucial component in price policy. However, price forecast is vital to facilitate efficient decisions and will play a major role in coordinating the supply and demand of soybeans globally.

Forecast is a kind of dynamic filtering, in which past values of the time series are used to predict future values. Empirically, the SARIMA 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 SARIMA and NARNN models has both linear and nonlinear modelling capabilities, can be a better choice for modelling the time series.

The comparative results revealed that the Hybrid-LM model with 8 neurons in the hidden layer and 3 time delays (MSE = 186.43259) yielded higher accuracy than the NARNN-LM model with 8 neurons in the hidden layer and 3 time delays (MSE = 222.42221), and the SARIMA, ARIMA$(0,1,3)(0,0,2)_{12}$, model (MSE = 284.966473) in this study.

This study contributed to the current understanding of how to obtain a better performance to forecast the target time series in agribusiness applying advanced techniques. The significance of this study provided a hands-on tool to educate the students learning the time series analysis in advance. Futhermore, this Hybrid SARIMA and NARNN model not only can provided richer information which are important in decision making process related to the future global price of soybeans impacts, but also can be employed in forecasting the future performance for global price of soybeans change outcomes.

REFERENCES

[1] F. Taheripour, and W. E. Tyner, "Impacts of possible Chinese 25% tariff on U.S. soybeans and other agricultural commodities", *Choices*, 33(2), pp. 1-7, 2018.

[2] F. Gale, C. Valdes, and M. Ash, "The interdependence of China, United States, and Brazil in soybean trade", OCS-19F-01, Economic Research Service, United States Department of Agriculture, 2019. [Online] Available: https://www.ers.usda.gov/publications/pub-details/?pubid=93389 [Accessed: February 22, 2021].

[3] L. Yu, L. Zhou, L. Tan, H. Jiang, Y. Wang, S. Wei, and S. Nie, "Application of a new hybrid model with seasonal auto-regressive integrated moving average (ARIMA) and nonlinear auto-regressive neural network (NARNN) in forecasting incidence cases of HFMD in Shenzhen, China", *PLoS ONE*, 9(6), pp. 1-9, 2014.

[4] L. Zhou, P. Zhau, D. Wu, C. Cheng, and H. Huang, "Time series model for forecasting the number of new admission inpatients", *BMC Medical Informatics and Decision Making*, 18:39, pp. 1-11, 2018.

[5] Y. Wang, C. Xu, S. Zhang, Z. Wang, L. Yang, Y. Zhu, and J. Yuan, "Temporal trends analysis of tuberculosis morbidity in mainland China from 1997 to 2025 using a new SARIMA-NARNNX hybrid model", *BMJ Open*, 9, e024409, pp. 1-11, 2019.

[6] Y. Zheng, L. Zhang, C. Wang, K. Wang, G. Guo, X. Zhang, and J. Wang, "Predictive analysis of the number of human brucellosis cases in Xinjiang, China", Scientific Reports, 11:1513, pp. 1-10, 2021.

[7] M. Li, S. Ji, and G. Liu, "Forecasting of Chinese e-commerce sales: an empirical comparison of ARIMA, nonlinear autoregressive neural network, and a combined ARIMA-NARNN model". *Mathematical Problems in Engineering*, 2018, Article ID 6924960, pp. 1-12, 2018.

[8] M. Yollanda, and D. Devianto, "Hybrid model of seasonal ARIMA-ANN to forecast tourist arrivals through Minangkabau International Airport", Proceedings of the 1st International Conference on Statistics and Analytics, Bogor, Indonesia, 2019.

[9] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*, Hoboken, N.J.: John Wiley & Sons. Inc., 2008.

[10] G. E. P. Box, and G. M. Jenkins, *Time series analysis: forecasting and control*, Holden-Day, San Francisco, 1970.

[11] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control* (5th ed.), Hoboken, N.J.: John Wiley and Sons Inc., 2016.

[12] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model", *Neurocomputing*, 50, pp. 159-175, 2003.

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

[14] H. P. Gavin, "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems", Department of Civil and Environmental Engineering, Duke University, 19 pages, 2020. [Online] Available: http://people.duke.edu/~hpgavin/ce281/lm.pdf [Accessed: February 22, 2021].

[15] K. Levenberg, "A method for the solution of certain non-linear problems in least squares", *Quarterly of Applied Mathematics*, 2(2), pp. 164–168, 1944.

[16] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics*, 11(2), pp. 431-441, 1963.

[17] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for non-linear least squares problems*, Lecture Notes, Technical University of Denmark, 2004. [Online] Available: http://www.imm.dtu.dk/courses/02611/nllsq.pdf. [Accessed: February 22, 2021]

[18] P. R. Gill, W. Murray, and M. H. Wright, "The Levenberg-Marquardt Method", §4.7.3 in *Practical Optimization*, London: Academic Press, pp. 136-137, 1981.

[19] G. M. L jung, and G. E. O. Box, "On a measure of lack of fit in time series models", *Biometrika*, 65(2), pp. 297-303, 1978.

[20] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Deep learning Toolbox™: getting started guide*, Natick, MA: The MathWorks, Inc., 2019.