**Comparative Predictive Performance of Artificial Neural Networks and ARIMA Models for COVID-19 Case Forecasting in Nigeria.**

# **ABSTRACT**

# 

COVID-19, a novel strain of coronavirus first identified in December 2019 in Wuhan, China, quickly escalated into a global pandemic. In response, various predictive models have been employed to monitor and forecast its spread. This study compares the predictive performance of two such models Autoregressive Integrated Moving Average (ARIMA) and Artificial Neural Network (ANN) using COVID-19 daily case data in Nigeria from March 23, 2020, to April 23, 2021, sourced from the Nigeria Centre for Disease Control (NCDC) and Our World in Data. The dataset was partitioned into 80:20 and 60:40 training-testing ratios. Results showed that ANN significantly outperformed ARIMA in both scenarios. For 80% training and 20% testing, ANN achieved RMSE = 0.00164 and MAE = 0.00024, while ARIMA recorded RMSE = 0.85758 and MAE = 98.52. Similarly, with 60% training and 40% testing, ANN achieved RMSE = 0.00049 and MAE = 0.00157, compared to ARIMA's RMSE = 0.82258 and MAE = 57.29. The findings indicated that for prediction purposes, neural networks should be considered, and for efficiency with large samples and significant training data, neural networks should also be taken into account.

**Keywords**: COVID-19, ARIMA, Artificial Neural Network, Forecasting, Predictive Modeling

**1.0 Introduction**

Several cases of severe acute respiratory syndrome with unknown etiology were reported in December 2019 in Wuhan City, China (Tang K, McCall B, Song PX, 2020). The coronavirus, a large family of viruses, was the main cause of this outbreak (Zhao S, Wan H, 2020). Two notable types of coronaviruses, SARS-CoV-1 and MERS-CoV, caused outbreaks in 2003 and 2012, respectively (Al-qaness, 2020). The novel coronavirus is the third type in this family, which has led to a massive pandemic known as COVID-19, as designated by the World Health Organization. The origins of this virus are still unknown, but it is most likely related to bats (Du Z, Anastassopoulu, 2020). This disease has an incubation period of over 14 days, a mortality rate between 2% and 3% (Ahmadi, 2020), and is transmitted through respiratory droplets and contact with contaminated surfaces (Al-qaness, 2020). COVID-19 spread rapidly in China (Song PX, 2020) and across the globe (Nishiura H, 2020). As of February 12, 2021, the total confirmed cases and deaths from this virus were 107,686,655 and 2,368,571, respectively, affecting over 223 countries (WHO, 2020). The first case of COVID-19 in Nigeria was discovered in Lagos on February 17, 2020, after which the disease spread rapidly throughout the country (Muniz, 2020). The total number of confirmed cases and deaths in Nigeria reached 167,200 and 2,127, respectively, on June 1, 2021 (NCDC, 2020). Awareness of disease trends is crucial for making decisions about preventive interventions; modeling to predict the number of new cases in the coming days offers valuable insight into these trends. Numerous studies have confirmed the superior performance of machine learning algorithms compared to more traditional models (Mozhgan, 2017). However, neither ARIMA nor ANN has been definitively proven to be more accurate than the other across different medical fields; thus, studies continue to compare them (Hue H, 2018). This comparison will also be part of this study to determine the most accurate model for forecasting the spread of the coronavirus in Nigeria. Chen (2015) compared the performance of a Probabilistic Neural Network with a GMM-Kalman Filter and a random walk approach for predicting the direction of return on the market index of the Taiwan Stock Exchange. They concluded that PNN has stronger forecasting power than both the GMM–Kalman filter and random walk models due to PNN's superior ability to identify erroneous data and outliers, as well as its independence from prior information about the underlying probability density functions of the data. Tansel et al. (2010) compared the effectiveness of linear optimization, ANNs, and genetic algorithms (GAs) in modeling time series data based on accuracy, convenience, and computational time. The study showed that linear optimization techniques provided the best estimates, while GAs yielded similar results when the parameter boundaries and resolution were carefully selected, with NNs providing the least accurate estimates. The study by Sehwan et al. (2007) also compared the forecasting performance of ARIMA and ANN models in predicting the Korean Stock Price Index, showing that the ARIMA model generally produced more accurate forecasts than the back-propagation neural network (BPNN) model, particularly for mid-range forecasting horizons. Stergiou (2017) used the ARIMA model on a 17-year time series data set (from 1964 to 1980, comprising 204 observations) of monthly pilchard catches (Sardina pilchardus) from Greek waters to forecast up to 12 months ahead, comparing the forecasts with actual catch data from 1981, which was not used to estimate parameters. Since the outbreak of COVID-19 in Nigeria, no researcher has compared the performance of machine learning algorithms to traditional models, thus, this study aims to evaluate the predictive ability of the Artificial Neural Network and ARIMA model using COVID-19 data. In particular, the specific objectives are to fit ARIMA and ANN models, use each of the two methods for forecasting, and finally determine the better model.

**2.0 METHODOLOGY**

The results reported in this research are based on secondary data obtained from the Our World and NCDC websites, which span from April 1st, 2020, to February 28th, 2021.

## **2.1 Auto-regressive integration Moving Average (ARIMA)**

The Autoregressive Integrated Moving Average (ARIMA) model is a combination of the differenced autoregressive model with the moving average model. Which is expressed as;

The AR part of ARIMA shows that the time series is regressed on its past data. The MA part of ARIMA indicates that the forecast error is a linear combination of past respective errors. The *I* part of ARIMA shows that the data values have been replaced with differenced values of *d* order to obtain stationary data, which is the requirement of the ARIMA model approach.

### **2.2.1 Box-Jenkins ARIMA process of Model Analysis**

ARIMA models are a class of models that have capabilities to represent stationary as well as non-stationary time series and to produce accurate forecasts based on a description of historical data of single variable. Since it does not assume any particular pattern in the historical data of the time series that is to be forecast, this model is very different from other models used for forecasting. In time series analysis, the Box-Jenkins method named after the statisticians George Box and Gwilym Jenkins, applies Autoregressive Moving Average ARMA or Autoregressive Integrated Moving Average ARIMA models to find the best-fitted model to past values of a time series.

**Model Identification**

Model identification in ARIMA involves determining the appropriate orders (p, d, q) for autoregression, differencing, and moving average. The first step is to assess if the time series is stationary i.e., its values fluctuate around a constant mean and variance. Non-stationary data must be transformed using regular differencing, typically of first order (d = 1), to achieve stationarity. If needed, a second differencing may be applied, though over-differencing should be avoided as it can inflate variability. Tools such as time series plots, autocorrelation (ACF), and partial autocorrelation (PACF) plots guide model selection. Most empirical time series can be modeled using a few basic ARIMA structures identified through ACF and PACF patterns.

**The Autocorrelation Function (ACF)**

For a covariance stationary time series the autocorrelation function is given by

ACF is a good indicator of the order of the MA (q) model since it cuts off after lag q (i.e. = 0 for k > q). On the other hand the ACF tails off for AR (p) model.

**The Partial Autocorrelation Function (PACF)**

If is normally distributed time series, then the PACF at lag k is given by

) (2.2)

PACF is a good indicator of the order of the AR (p) model since it cuts off after lag p (i.e. = 0 for k > p). On the other hand the PACF tails off for MA (q) model.

**Stationarity Analysis**

Hipel and McLeod mentioned a simple algorithm (developed by Schur and Pagano) for determining the stationarity of an AR process. For AR (1) model

is stationary when with constant mean and a constant variance

**Augmented Dickey-Fuller (ADF) Test of Unit Root**

In Dickey-Fuller's test, null hypothesis () is that the series has unit root / not stationary while alternative hypothesis ( ) is that the series has no unit root / stationary. The hypothesis is then tested by performing appropriate differencing of the data in order and applying ADF test to differenced time series data. First order differencing (d=1) means we generate a table of differenced data of current and the immediate previous one

i.e

This test will enables us to go further in steps for ARIMA model development i.e. to find suitable values of p in AR and q in MA in our model. For that, we need to examine the correlogram and partial correlogram of the stationary (order of differenced) time series.

**Model Estimation**

Once the ARIMA model structure is identified, the next phase involves estimating its parameters. This is typically done using maximum likelihood estimation methods, which aim to find the parameter values that make the observed data most probable. Optimization techniques such as the Quasi-Newton method are often used for this purpose. To validate the estimated model, residual analysis is essential. This includes examining autocorrelation plots of the residuals and applying statistical tests like the Ljung-Box Q-test to check if the residuals behave like random noise. If they do, the model is considered adequate; otherwise, re-specification may be necessary. Model selection is further guided by statistical criteria, including the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Root Mean Square Error (RMSE). In this study, these metrics are used to determine the most suitable model for forecasting Box et al., (2015).

**Model Checking (Goodness of Fit)**

After estimating ARIMA model parameters, it is essential to evaluate the model’s adequacy through diagnostic checks. A well-fitted model should have residuals that exhibit white noise characteristics, uncorrelated, normally distributed errors with constant variance and zero mean. If these assumptions hold, it suggests the model has captured all relevant patterns in the data. Key tools for this evaluation include the Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), and the Ljung-Box Q statistic. The Ljung-Box test specifically assesses whether residual autocorrelations are significantly different from zero, thereby identifying model misspecification. Additionally, model selection criteria such as the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are employed to compare competing models. The model with the lowest AIC or BIC is generally preferred for its balance between fit and parsimony.

In this study, residual analysis, ACF/PACF plots, and the Ljung-Box test are applied to confirm that the selected ARIMA model meets the statistical assumptions of a stationary and well-specified process. The Ljung-Box statistic is defined as: (Box et al., (2015).

𝐿𝐵 = +2 𝑘−1𝑚𝜌2𝑘𝑛−𝑘 ~ 𝜒2𝑚 (2.5)

Where n= sample size and m is the lag length.

**Model Forecasting**

Once the model has been selected, the estimated residual of the model is carefully examined to follow a white noise process (a random process of random variables that are uncorrelated, having mean zero and finite variance). The parameters of the model are tested for significance and the final model estimated; then forecasting is done. Forecasting with this system is straight forward; the forecast is the expected values, evaluated at a particular point in time.

## **2.3 Artificial Neural Network (ANN)**

Artificial Neural Network (ANN) is an extension of Generalized Linear Models (GLM). This data mining algorithm is so popular for modeling nonlinear associations. ANN is comprised of three layers: an input, output and hidden layer(s). Each layer is formed from Neurons and Synapses. The neurons in the input layer are previous observations used for forecasting future values in the output layer. Other layers within input and output are called hidden layers. An artificial neural network (ANN), usually called neural network (NN) is a computational model that is inspired by the structure and functionality of biological neurons. They are used as statistical data modeling tools in order to model complex relationships between inputs and outputs.

Their high performance in modeling relationships between inputs and outputs makes NNs reliable tools, which can also be used in the development of forecasting models. In this project, in addition to models developed by using linear regression, intelligent models were created by using NNs. In Figure 3.0 & 3.1., the interconnected structure of a neural network is showed by indicating its simple internal processors.

**Figure 2.0:** The framework of ANN showing input layer, Hidden layer and output layer

Each processor in the NN receives information from an upper level and each processor in the NN transfers output to a lower level. Information (inputs) can be received from other neurons or directly from the environment. The pattern of information given to the input processing units gives an indication of the problem being presented to the NN. The output can be transferred to other neurons or directly to the environment. The pattern of outputs transferred by the output processing units represents the result of the computations performed by the NN. The neurons in the input buffer of the NN work as the dendrites of a biological neuron which is responsible of receiving information from environment or other neurons. The neurons in the hidden layers connect input buffer and output layer like cell body of the biological neuron which is responsible from carrying processed information to other neurons. The neurons in the output layer works as the axon part of the biological neuron by carrying processed information to other neurons or directly environment.

The understanding of the hidden layer requires knowledge of weights, bias, and activation functions. Weights in an ANN are the most important factor in converting an input to impact the output. This is similar to slope in linear regression, where a weight is multiplied to the input to add up to form the output. Weights are numerical parameters which determine how strongly each of the neurons affects the other. For a typical neuron, if the inputs are x1, x2, and x3, then the synaptic weights to be applied to them are denoted as w1, w2, and w3.

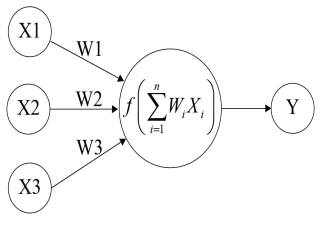
Output is

Simply, this is a matrix multiplication to arrive at the weighted sum.

Bias is like the intercept added in a linear equation. It is an additional parameter which is used to adjust the output along with the weighted sum of the inputs to the neuron.

The processing done by a neuron is thus denoted as: *output = sum (weights \* inputs) + bias*

A function is applied on this output and is called an activation function. The input of the next layer is the output of the neurons in the previous layer, as shown in figure 3.2:



**Figure 2.2:** How the neuron in Artificial Neural Network works

The direction of information flow in a NN, starts from the input buffer, goes through the hidden layer(s) and finishes in the output layer. A neural network performs computations by feeding inputs through connections with weights. The transfer function (activation function) of a neuron converts the input to output, which will be transferred to other neurons or the environment.

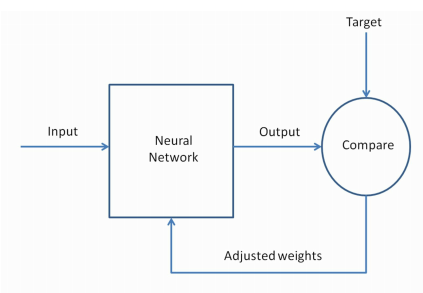
The number of hidden layers in an ANN can be none, one or more. There is no strict definition for the number of hidden layers, but it is known that one hidden layer is sufficient for most of the applications.

There are many choices for the type of transfer function (activation function) that can be used. Linear, sigmoid or step type transfer functions (activation function) are used in various applications of NN models but the sigmoid function is the most popular one. By using sigmoid-type transfer function, NN models can learn and capture the relation between input and output parameters. The sigmoid function is a mathematical function that produces a sigmoidal curve; a characteristic curve for its S shape. This is the earliest and often used activation function. This squashes the input to any value between 0 and 1, and makes the model logistic in nature. The sigmoid function is defined by the formula;

After the building of NN model, the next step is training. The Resilient Back Propagation is a common method of teaching ANNs. The resilient back propagation method is a local adaptive learning scheme, performing supervised batch learning in a multilayer perceptron. The basic principle of the resilient back propagation method is to eliminate the harmful influence of size of the partial derivative on the weight step.

### **2.3.1 Step by step Illustration of a neural network**

The step-by-step approach to understand the forward and reverse pass with a single hidden layer will be taken. The input layer has 7 neuron (That lags 7) and the output will solve the COVID-19 new cases for the next one month. Figure 3.2 shows a forward and reverse pass with a single hidden layer:



**Figure 2.3:** forward and reverse pass of neural network with a single hidden layer

Next, is the step by step operations to be done for network training;

1. The input will be taken as lags of the output ( that is 7 lags)
2. The hidden layers will be chosen as (4,2)
3. The dataset will be normalized to a binary number [0, 1] where necessary.
4. Date with missing values are removed.
5. We use the output error to compute error signals for previous layers. The partial derivative of the activation function is used to compute the error signals.
6. Then we build the neural network plots (i.e, Network plot and model) for each of the training set
7. Apply the weight adjustments.
8. Then we use the model to predict the remaining percentage (Model testing)

The complete pass back and forth is called a **training cycle.** The updated weights and biases are used in the next cycle. We keep recursively training until the error is very minimal.

*R*-software was used for fitting an ANN model for the time series. Some commands and functions with input and output variables have been used. The *R* library ‘*neuralnet’* is used to train and build the neural network. The *nnet* function is used to fit neural networks. The arguments are: *size* which determines the number of units in the hidden layer, and *maxit* determines the maximum number of iterations. The objects are: *fitted values* is used for the fitted values for the training data and *residuals* is used to show the residuals for the training data (Venables, W. N. and Ripley, B. D., 2002).

## **2.4 Design of data**

The data used for this study is secondary data downloaded from Our World website and NCDC website, the data span from March, 23, 2020 to April, 23, 2021 of new cases of COVID-19 in Nigeria. The two models will predict the possible cases of COVID – 19 for the next month (April 24 – May 24, 2021) and also the dataset will be divided into (80% train and 20% test), (60% train and 40% test), the train data will be used to predict the remaining percentage of the data using the two model above.

# **3.0 RESULTS AND DISCUSSIONS**

### **3.1 Estimation of Auto Regressive Moving Average (ARIMA)**

The analysis was carried out using the proposed methods: Autoregressive Integrated Moving Average (ARIMA) and Artificial Neural Network (ANN). To evaluate model performance, the dataset was split into training and testing sets at proportions of 100%, 80%, and 60% for training, with the corresponding 20% and 40% for testing. Both ARIMA and ANN models were trained on each training subset, and their predictive accuracy was assessed using the test sets. Model evaluation was conducted using standard error metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

The descriptive statistics of the original data without transformation revealed non-normality and non-stationarity. The Shapiro Wilk test returned a significant value (p < 0.05), which implies that the data are not normally distributed and also the ADF test returned a non-significant value, which implies that the data is not stationary (Table 3.1), the plots in Fig. 3.1 shows the graphical visualization of the non-stationary data using time plot and qqplot. To confirm the non-stationarity, the ACF and PACF graph presented in Figure 3.2 shows that ACF decay rapidly which shows evident of non-stationary.

Differencing the dataset once resulted in a stationary series. The ADF test on the differenced data returned a significant statistic (D = -7.7679, p < 0.05), confirming stationarity. The time plot of the differenced data showed stable mean and variance over time, while the histogram indicated an approximately normal distribution centered around zero. The ACF and PACF plots after differencing also supported stationarity (Figure 3.3).

**Table 3.1:** Preliminary Test (Normality and Stationarity)

|  |  |  |
| --- | --- | --- |
| **Statistic** | **W** | **P** |
| Shapiro Wilk | **0.80602\*\*** | **< 0.05** |
| Augmented Dickey-Fuller test | **-0.75574** | **> 0.05** |

*\*\* Significant at 0.05*

#### **3.1.1 Model identification and Estimation**

Fitting the ARIMA requires setting the order for the model called the parameter p, d, q for which the Autoregressive (AR) part of the model takes the parameter p, the “I” part which is the integrated differencing order takes parameter d and the Moving Average (MA) part takes the parameter q. For this analysis, the parameter d equals to 1 since only one differencing was taken, and also, the ACF AND PACF plot from the previous section suggested the models in (Table 3.2).

**Table 3.2: Different combination of ARIMA models**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **RMSE** | **AIC** | **AICc** | **BIC** |
| ARIMA (2 , 1, 2) | 170.3025 | 5244.75 | 5245.21 | 5280.63 |
| ARIMA (3, 1, 2) | 169.733 | 5246.14 | 5246.82 | 5289.99 |
| ARIMA (5, 1, 2) | 162.4016 | 5219.02 | 5220.28 | 5278.82 |
| ARIMA (6, 1, 2) | 159.7842 | 5209.97 | 5211.58 | 5277.74 |
| **ARIMA (6, 1, 1)** | **159.7835** | **5205.97** | **5207.23** | **5265.77** |
| ARIMA (3, 1, 1) | 170.9642 | 5247.84 | 5248.3 | 5283.71 |
| ARIMA (5, 1, 1) | 162.7369 | 5216.78 | 5217.73 | 5268.6 |
| ARIMA ( 3, 1, 4) | 166.4121 | 5231.06 | 5231.42 | 5262.97 |
| ARIMA (2, 1, 3) | 168.8116 | 5241.84 | 5242.52 | 5285.69 |
| ARIMA (2, 1, 8) | 157.3634 | 5206.5 | 5208.96 | 5290.21 |

As stated in the model identification, the model suggested by the ACF and PACF was fitted, AIC and RSME were shown in the table 4.2. By comparison the RMSE, AIC criterion indicates that ARIMA (6, 1, 1) model should be fitted for the COVID-19 cases which support the model fitting based on the criteria.

Maximum likelihood estimation was used and shows the results obtained from the R statistical software in Table 4.3. Here we see that,. We also see that the estimated noise variance is 25.628. Noting the p-values, the estimates of all autoregressive and moving average coefficients are significantly different from zero statistically, as is the intercept term.

**Table 3.3: Maximum Likelihood Estimates of the selected Model**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Coefficients | AR (1) | AR (2) | AR (3) | AR (4) | AR (5) | AR (6) | MA (1) | Intercept |
|  | -0.8357 | -0.6070 | -0.5361 | -0.4034 | -0.4911 | -0.3432 | 0.1324 | 0.3124 |
| SE | 0.1433 | 0.1106 | 0.0956 | 0.0882 | 0.0800 | 0.0691 | 0.1455 | 0.0030 |
| P - value | 0.0001 | 0.0232 | 0.0007 | 0.0001 | 0.0140 | 0.0005 | <0.0001 | <0.0001 |

*Estimated as 25.628: log likelihood = -2066.06 AIC = 5205.97 AICc = 5207.23 BIC = 5265.77*

The estimated model would be written as

Where

To evaluate the adequacy of the **ARIMA (6,1,1) model**, the Box-Pierce (Ljung-Box) test was applied. The decision rule stated that the null hypothesis (H₀), which assumes no lack of fit, would not be rejected if the p-values at non-seasonal lags 7 and 8 were greater than 0.05 **(Table 3.3).** As the resulting p-values exceeded this threshold at the 95% confidence level, H₀ was not rejected. This indicates that the residuals are independently distributed and supports the model’s adequacy for forecasting future COVID-19 cases in Nigeria.

Residual diagnostics were further assessed using ACF and PACF plots, standardized residual plots, and the Ljung-Box p-value plot as shown in Figure 3.4. According to ARIMA model assumptions, residuals should behave like white noise having a zero mean, constant variance, and no autocorrelation. The ACF of the standardized residuals (Figure 3.5) confirmed these conditions, showing no significant autocorrelation and constant variance around a zero mean.

In addition, the Shapiro-Wilk test for normality applied to the residuals yielded a non-significant p-value, indicating that the residuals are approximately normally distributed, as expected for white noise. These findings confirm that the ARIMA(6,1,1) model meets all adequacy criteria and is statistically reliable. Having verified both the model's significance and its diagnostic validity, the dataset was subsequently divided into training and testing subsets to compare ARIMA's predictive performance with that of the ANN model.

**Table 3.3**: Box – Ljung test

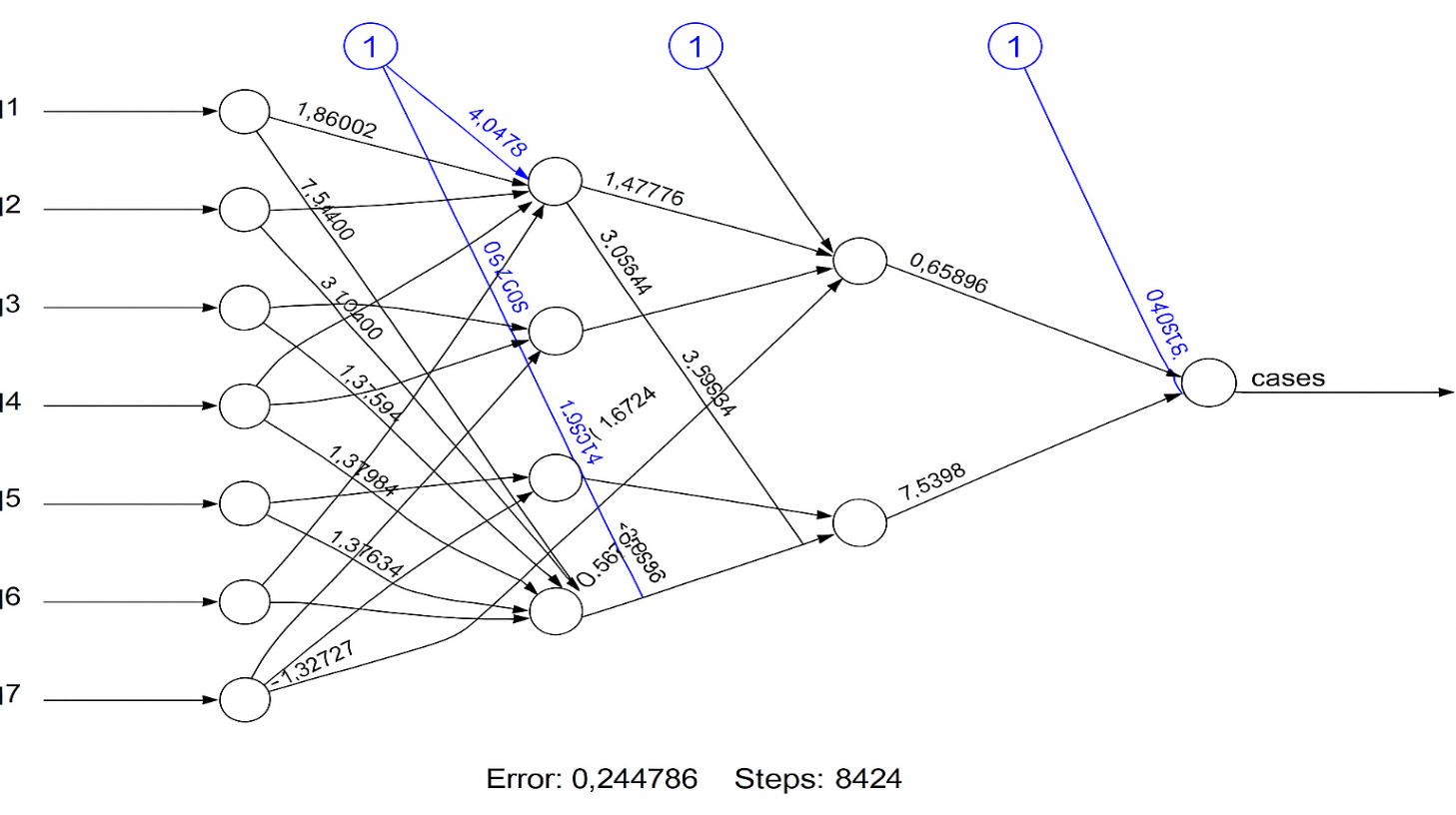
|  |  |  |  |
| --- | --- | --- | --- |
| Model | Chi-Square | Lag | P value |
| ARIMA (6,1,1) | 7.9195 | 7 | 0.3397 |
| ARIMA (6,1,1) | 56.206 | 8 | 0.2191 |

### **3.2 Fitting Artificial Neural model For COVID 19 cases in Nigeria**

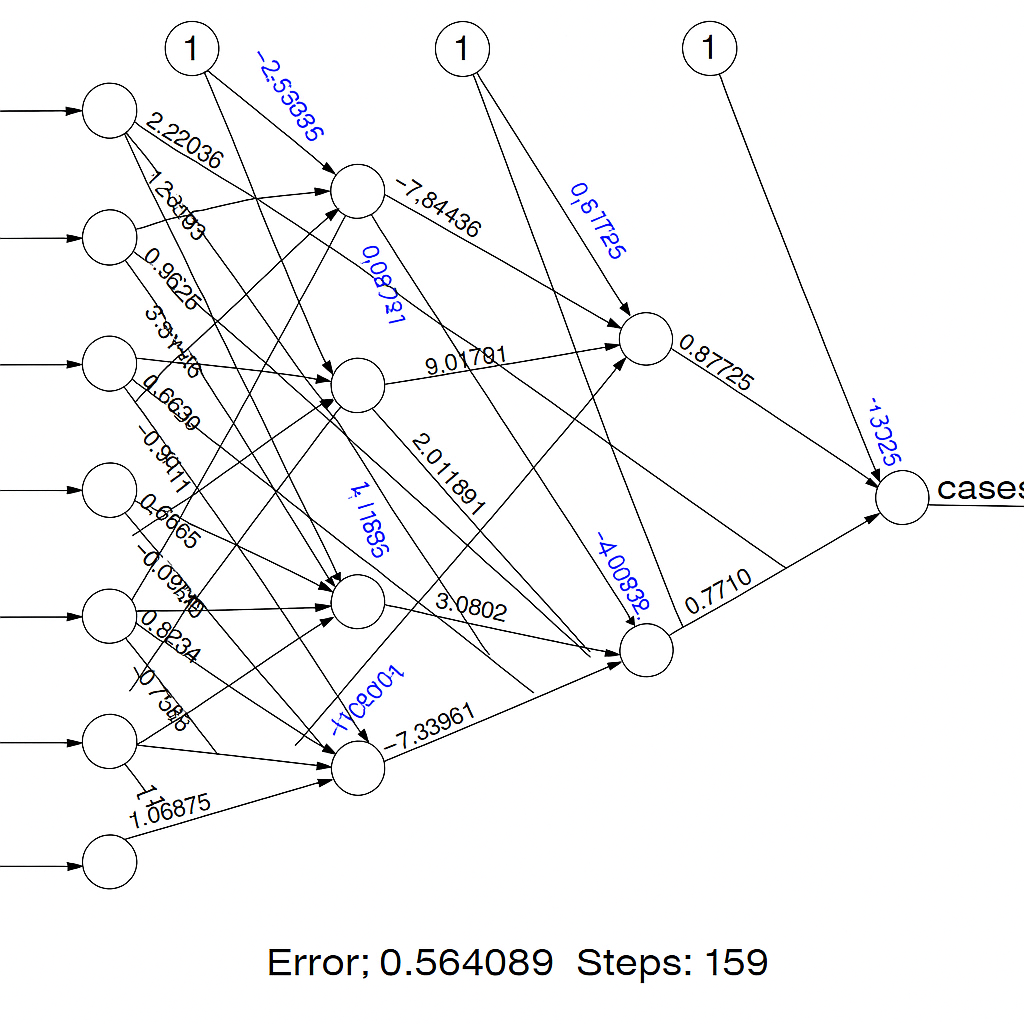
Applying the artificial neural network (ANN) model, the training dataset was matched to the same number of observations used in the ARIMA model (400 observations), representing 80% of the dataset, while the remaining 20% was reserved for model evaluation. The ANN architecture consisted of an input layer corresponding to the lag variables, followed by two hidden layers and an output layer (Cases of COVID-19). The input layer received the lagged observations as predictors, while the hidden layers processed the data to capture nonlinear patterns. The final output layer produced the forecasted COVID-19 case values. (Figure 3.6 and 3.7).

In configuring the network structure, consideration was given to both training feasibility and model complexity. It was acknowledged that although deeper networks can capture complex relationships, they are more difficult to train and prone to overfitting. Therefore, a two-layer network architecture was selected, which is often sufficient for modeling non-linear decision boundaries in time series data. The first hidden layer contained four neurons, while the second had two neurons.

The number of input units corresponded to the number of lagged variables used as features, while a single output unit was used to predict the daily number of COVID-19 cases. The learning rate of the algorithm was also carefully tuned, as a rate that is too high could lead to instability during training, while a rate that is too low may result in slow convergence. The training process was completed successfully, and the performance of the ANN model was visually represented in the network diagram, as shown in Figure 3.6, with an error rate of 0.244786 after 8,424 iterations.

****

**Figure 3.7:** The net plotfor 80% training and 20% testing data

****

**Figure 3.8:** The net plotfor 60% training and 40% testing data

MSE is used as stopping criteria in the network. Smaller values of RMSE indicate higher accuracy in forecasting. The Neural network result shows that the minimum **MSE equals 0.0016430** for considering the model with fifteen units in the hidden layer, 7 lags and the learning rate equals to 0.01. The RMSE for ANN and ARIMA were shown in Table 4.4 and 4.5 respectively.

**Table 3.4:** Model Testing and Comparison for ARIMA and ANN model. (80% Training and 20% testing)

|  |  |  |  |
| --- | --- | --- | --- |
| ARIMA (6, 1, 1) | | ANN | |
| RMSE | MAE | RMSE | MAE |
| 0.857577 | 98.51822 | **0.0016430** | 0.000236677 |

**Table 3.5:** Model Testing and Comparison **for** ARIMA and ANN model. (60% Training and 40% testing)

|  |  |  |  |
| --- | --- | --- | --- |
| ARIMA (6, 1, 1) | | ANN | |
| RMSE | MAE | RMSE | MAE |
| 0.822577 | 57.29007 | **0.00048898** | 0.00156544 |

**Table 3.6:** Actual and predicted results of ANN and ARIMA (6, 1, 1) models for COVID 19 cases for the first 15 testing data

|  |  |  |  |
| --- | --- | --- | --- |
| **80% Training and 20% testing** | | | |
| *Date* | *Actual data* | *Predicted Values* | |
| ANN | ARIMA |
| 2/1/2021 | 676 | 677 | 701 |
| 2/2/2021 | 1634 | 1644 | 1272 |
| 2/3/2021 | 1138 | 1140 | 997 |
| 2/4/2021 | 1340 | 1339 | 1414 |
| 2/5/2021 | 1624 | 1623 | 1398 |
| 2/6/2021 | 1588 | 1589 | 1022 |
| 2/7/2021 | 504 | 500 | 512 |
| 2/8/2021 | 643 | 641 | 677 |
| 2/9/2021 | 1056 | 1060 | 1148 |
| 2/10/2021 | 1131 | 1129 | 1100 |
| 2/11/2021 | 938 | 937 | 939 |
| 2/12/2021 | 1005 | 1009 | 1242 |
| 2/13/2021 | 1143 | 1141 | 1120 |
| 2/14/2021 | 520 | 521 | 600 |
| 2/15/2021 | 744 | 749 | 749 |

*Obs = Observations, Std. = Standard*

**Table 3.7:** Actual and predicted results of ANN and ARIMA (6, 1, 1) models for COVID-19 cases for the first 15 testing data

|  |  |  |  |
| --- | --- | --- | --- |
| **60% Training and 40% testing the first 15 observations** | | | |
| *Date* | *Actual data* | *Predicted Values* | |
| ANN | ARIMA |
| 11/13/2020 | 156 | 156 | 162 |
| 11/14/2020 | 112 | 111 | 187 |
| 11/15/2020 | 152 | 151 | 158 |
| 11/16/2020 | 157 | 158 | 171 |
| 11/17/2020 | 152 | 152 | 172 |
| 11/18/2020 | 236 | 234 | 174 |
| 11/19/2020 | 146 | 145 | 170 |
| 11/20/2020 | 143 | 145 | 171 |
| 11/21/2020 | 246 | 244 | 173 |
| 11/22/2020 | 155 | 158 | 169 |
| 11/23/2020 | 56 | 51 | 172 |
| 11/24/2020 | 168 | 171 | 171 |
| 11/25/2020 | 198 | 190 | 172 |
| 11/26/2020 | 169 | 167 | 171 |
| 11/27/2020 | 246 | 243 | 245 |

*Obs = Observations, Std. = Standard*

The RMSE for ARIMA and ANN equal 0.857577 and 0.0016430 for 80% training and 20% testing, an also 0.822577 and 0.00048898 for 60% training and 40% testing,respectively (Tables 3.4 and 3.5). This result shows that the RMSE of ANN is 1.54% of RMSE for ARIMA. In other words, the RMSE of ARIMA model is 521.958 times RMSE of the ANN model. This means ANN model outperformed ARIMA model and the model is much more accurate and efficient than the ARIMA forecasting model. The predicted values of the remaining percentage (20% and 40%) were shown in Table 3.6 and 3.7. it can be seen that, the predicted values for ANN model were very close to the actual value than that of ARIMA model predicted values. Table 3.8 shows the predicted values for the next 30 days.

## **4.0 CONCLUSION**

To the best of our knowledge, this project has proposed two efficient approaches forecasting models used in the medical field for the prediction of disease. In the first model artificial neural network using a multilayer, is trained by minimizing RMSE and the second model consists of using ARIMA model on COVID-19 daily cases in Nigeria. The results of both models reveal that ANNs outperform and offer consistent prediction performance compared to ARIMA model and hence preferable as a robust prediction model for COVID-19 daily cases in Nigeria.. We hereby recommend that; For the purpose of prediction Artificial Neural Network should be considered over the conventional Autoregressive Integrated Moving Average.

# **ACKNOWLEDGEMENT**

All praise is due to Almighty Allah for the gift of life and knowledge. I sincerely thank my advisor for his invaluable guidance, encouragement, and mentorship throughout this research. His support greatly contributed to the successful completion of this work.

**COMPETING INTERESTS**

Authors have declared that no competing interests exisit.

**CONSENT**

All authors declare that written informed consent was obtained from the patient (or other approved parties) for publication of this case report and accompanying images.

# 

# **Reference**

Ahmadi A, Shirani M, Rahmani F (2020). Modeling and Forecasting Trend of COVID-19 Epidemic in Iran. *medRxiv,* doi: https://doi.org/10.1101/2020.03.17.20037671*.*

Anastassopoulou C, Russo L, Tsakris A, Siettos C (2020). Data-Based Analysis, Modelling and Forecasting of the novel Coronavirus (2019-nCoV) outbreak. *medRxiv,* doi:https://doi.org/ 10.1101/2020.02.11.20022186

Al-qaness MA, Ewees AA, Fan H, Abd El Aziz M (2020). Optimization Method for Forecasting Confirmed Cases of COVID-19 in China. *J Clin Med,* 9(3): 674.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: Forecasting and control (5th ed.). Wiley.

Chen, Y.B. Yang, J. Dong, and A. Abraham, “Time-series forecasting using flexible neural tree model,” *Information Sciences*, vol. 174, no. 3-4, pp. 219–235, 2005.

Du Z, Xu X, Wu Y, Wang L, Cowling BJ, Meyers LA (2020). The serial interval of COVID-19 from publicly reported confirmed cases. *medRxiv,* doi: https://doi.org/10.1101/ 2020.02.19.20025452.

Hu Z, Ge Q, Jin L, Xiong M (2020). Artificial intelligence forecasting of covid-19 in china. *arXiv preprint arXiv:*2002.07112.

McCall B (2020). COVID-19 and artificial intelligence: protecting health-care workers and curbing the spread. *Lancet Digit Health,* 2(4): 166-7. doi.org/10.1016/S2589-7500 (20)30054-6.

Meyler, A., Kenny, G., and Quinn, T. (2008). Forecasting Irish Inflation Using ARIMA Models, Technical Paper 3/RT/1998, Central Bank of Ireland Research Department.

Muniz-Rodriguez K, Fung IC-H, Ferdosi SR et al (2020). Transmission potential of COVID-19 in Iran. *medRxiv,* doi: https://doi.org/10.1101/ 2020.03.08.20030643.

Mozhgan S, Faradmal J, Poorolajal J, Mahjub H (2017). Model-based Recursive Partitioning for Survival of Iranian Female Breast Cancer Patients: Comparing with Parametric SurvivalModels. *Iran J Public Health,* 46(1): 35-43.

Nishiura H, Linton NM, Akhmetzhanov AR (2020). Serial interval of novel coronavirus (COVID-19) infections.*IntJInfectDis,* 93:284-6. doi: https://doi.org/10.1016/j.ijid.2020.02.060

Organization WH. Novel coronavirus (2019-nCoV); 2020; Available from URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports

Stergiou, K. I. (1989), *Modeling and forecasting the fishery for pilchard (Sardina pilchardus) in Greek waters using ARIMA time-series models*, ICES Journal of Marine Science,Volume 46, No. 1, Pp. 16-23.

Song PX, Wang L, Zhou Y et al (2020). An epidemiological forecast model and software assessing interventions on COVID-19 epidemic in China. *medRxiv,* doi.org/10.1101/2020.02.29.20029421

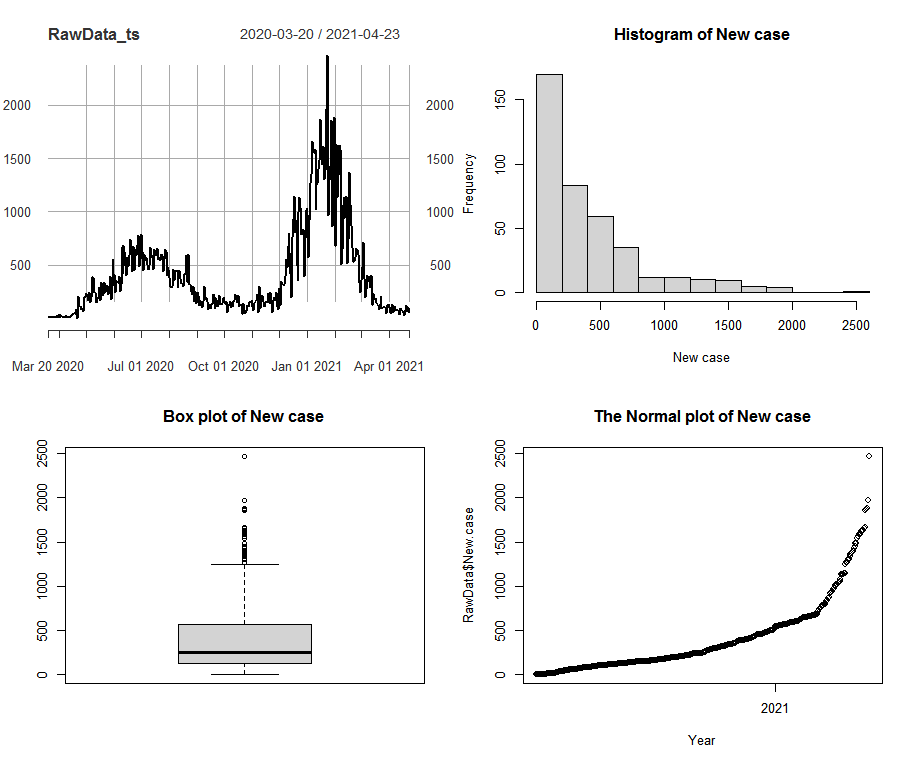
Sehwan S, Gao D, Zhuang Z et al (2020). Estimating the serial interval of the novel coronavirus disease (COVID-19): A statistical analysis using the public data in Hong Kong from January 16 to February 15, 2020. *medRxiv,* doi:https://doi.org/10.1101/2020.02.21.20026559

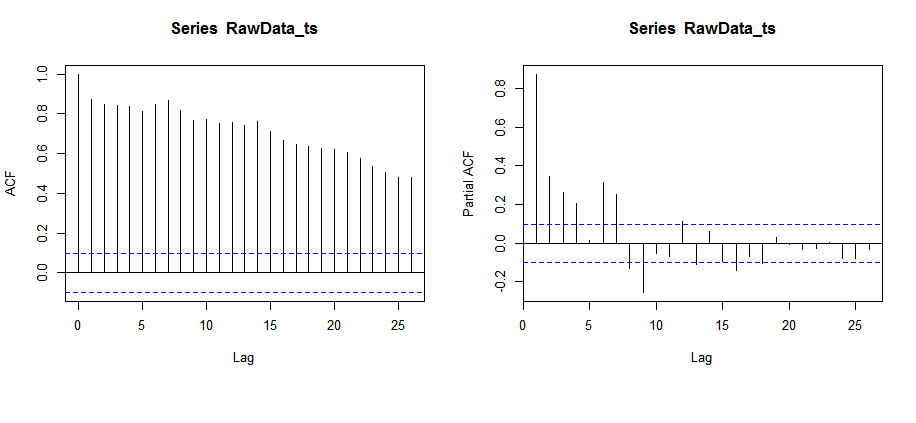
Tansel, I. N.,Yang, S. Y., Venkataraman, G., Sasirathsiri, A., Bao, W, Y., and Mahendrakar, N. (2010). Modeling time series data by using neural networks and genetic algorithms, in *Smart Engineering* *System Design: Neural Networks, Fuzzy Logic, Evolutionary* *Programming, Data Mining, and Complex Systems: Proceedings* *of the Intelligent Engineering Systems Through Artificial Neural* *Networks*, C. H. Dagli, A. L. Buczak, J.Ghosh,M. J. Embrechts, and O. Erosy, Eds., vol.9,pp. 1055–1060, ASMEPress,NewYork, NY, USA.

Tang K, Huang Y, Chen M (2020). Novel Coronavirus 2019 (Covid-19) epidemic scale estimation: topological network-based infection dynamic model. *medRxiv,* doi: 10.1101/2020.02.20.20023572.

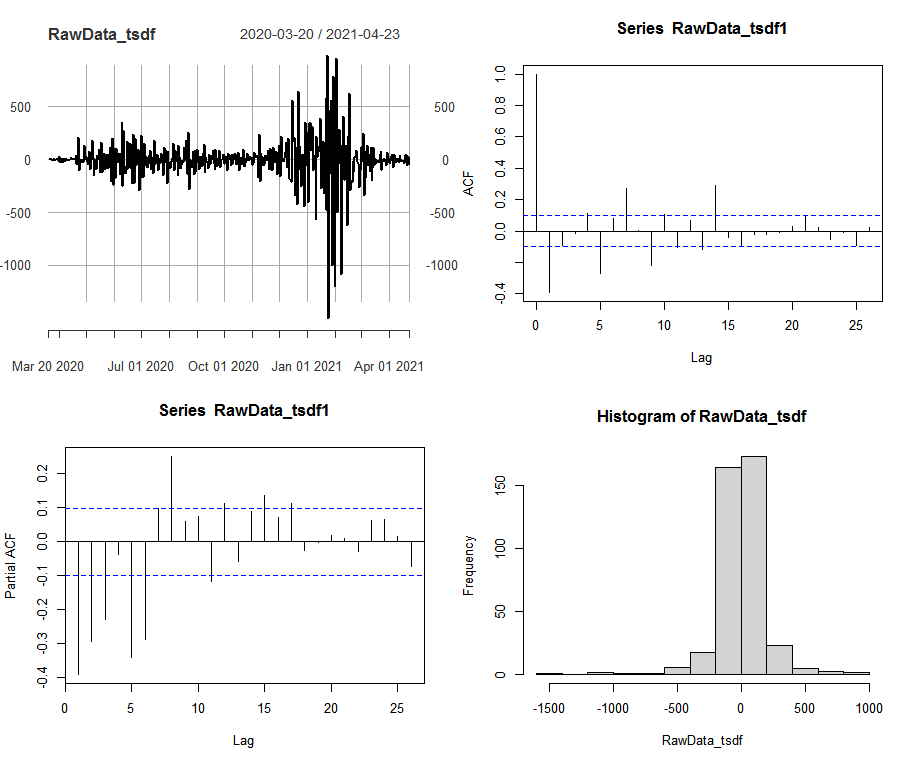
**APENDIX 1A**

**Graph and Tables of ARIMA Models**

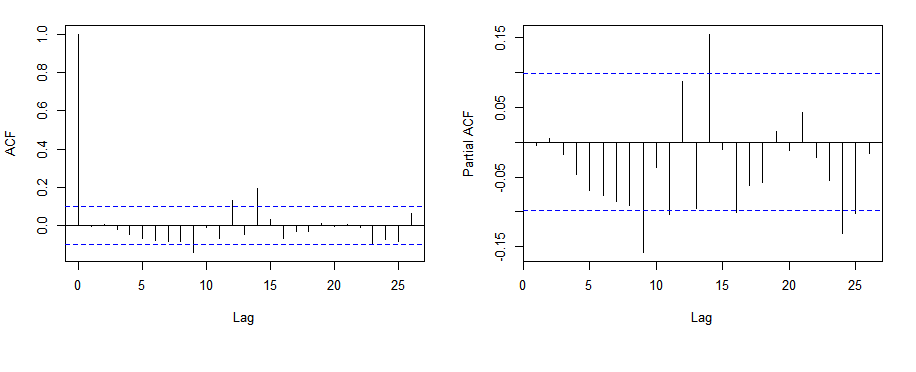
 **Figure 3.1:** Graph of New case showing time plot, histogram, box plot and qqplot.



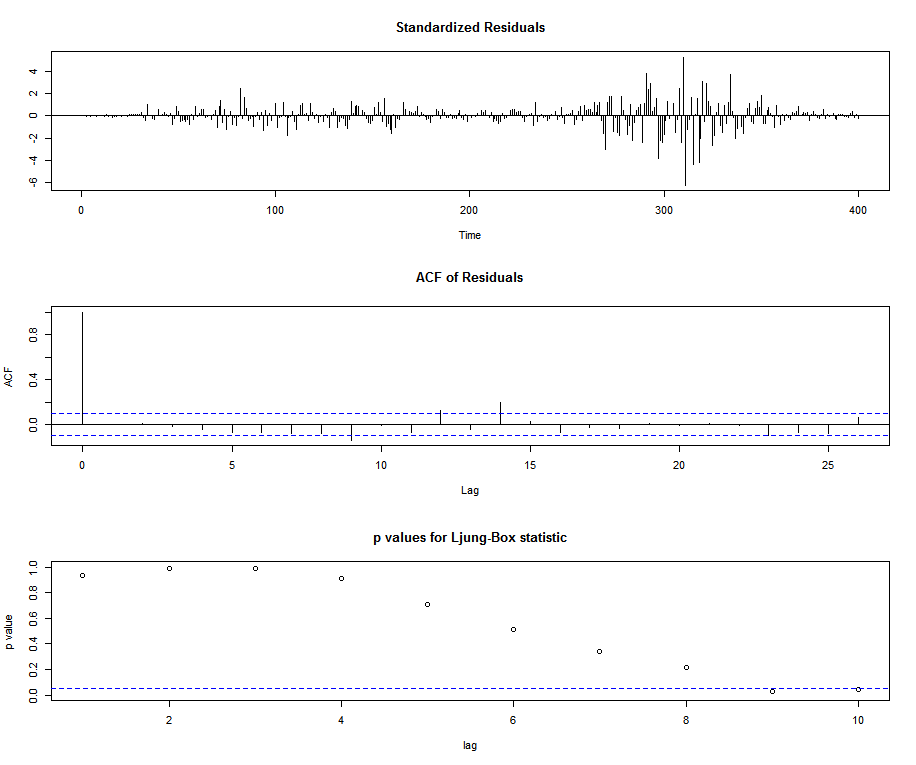
**Figure 3.2:** The graph of ACF and PACF of the dataset before differencing.

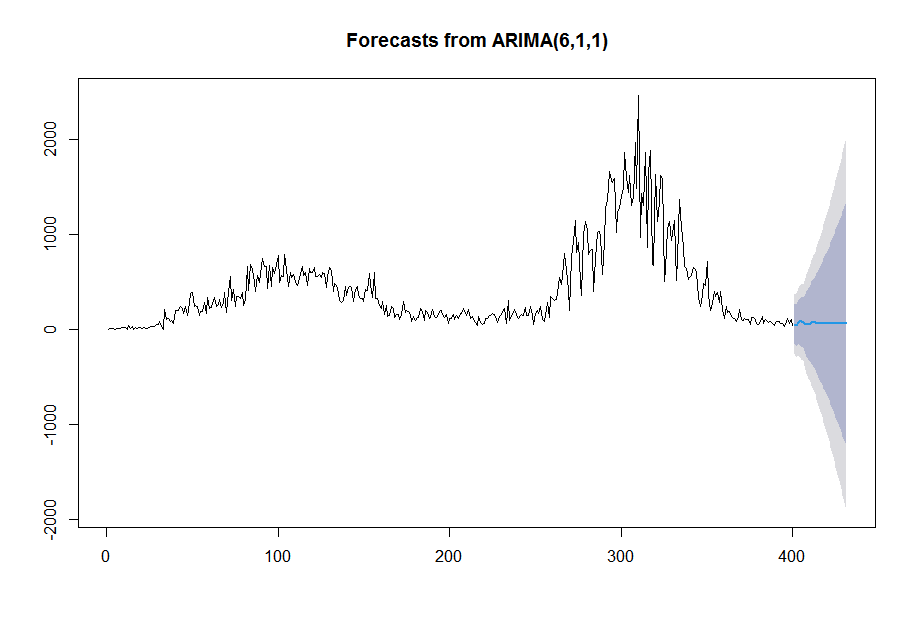


**Figure 3.3:** The graph of normalizing and differenced dataset.



**Figure 3.4:** ACF and PACF of the residuals terms of the selected model.



**Figure 3.5:** The standardized residual plot, p values for the Ljung–Box statistic plot and the ACF residuals plot****

**Figure 3.6:** Data and forecasted results of ARIMA (6, 1, 1) models for COVID 19 cases in Nigeria

**APENDIX 1B:**

**ANN and ARIMA Out-of-sample forecast**

**Table 3.8:** The predicted values from ANN and ARIMA (6, 1, 1) model for the Next 31 days.

|  |  |  |
| --- | --- | --- |
| **Date** | **ANN** | **ARIMA** |
| April, 24 | 57 | 56 |
| April 25 | 61 | 42 |
| April 26 | 81 | 76 |
| April 27 | 75 | 79 |
| April 28 | 69 | 79 |
| April 29 | 80 | 75 |
| April 30 | 91 | 62 |
| May 1 | 62 | 56 |
| May 2 | 58 | 57 |
| May 3 | 64 | 66 |
| May 4 | 70 | 69 |
| May 5 | 69 | 71 |
| May 6 | 75 | 68 |
| May 7 | 78 | 64 |
| May 8 | 65 | 60 |
| May 9 | 62 | 60 |
| May 10 | 60 | 63 |
| May 11 | 70 | 64 |
| May 12 | 55 | 65 |
| May 13 | 62 | 64 |
| May 14 | 63 | 62 |
| May 15 | 66 | 61 |
| May 16 | 61 | 60 |
| May 17 | 67 | 61 |
| May 18 | 60 | 61 |
| May 19 | 69 | 61 |
| May 20 | 62 | 61 |
| May 21 | 65 | 60 |
| May 22 | 60 | 59 |
| May 23 | 64 | 59 |
| May 24 | 58 | 59 |