PREDICTIVE ABILITY OF ARTIFICIAL NEURAL NETWORK (ANN) AND AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) ON COVID-19 CASES IN NIGERIA.

ABSTRACT

Over the last two years, several cases of severe acute respiratory syndrome with unknown etiology were reported in December 2019 in Wuhan City, China. The coronavirus, part of a large family of viruses, was the main cause of this outbreak. Two notable types of coronaviruses are SARS-CoV-1 and MERS-CoV, which caused outbreaks in 2003 and 2012, respectively. The novel coronavirus is the third type in this family that resulted in a massive pandemic and was designated as COVID-19 by the World Health Organization. To help mitigate the impact, an increasing number of prediction methods have been developed. Despite the growing sophistication of these methods, our literature review found that research exploring the impact of Artificial Neural Networks and Autoregressive Integrated Moving Average predictive methods is limited. Consequently, this work combines ARIMA and ANN models. To achieve this objective, we began by analyzing the ability of the two models (Artificial Neural Network and Autoregressive Integrated Moving Average) to predict COVID-19 cases for the next 30 days. The results of both approaches are compared for predictive accuracy and variability. We evaluate the results using mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The findings indicate that for prediction purposes, neural networks should be considered, and for efficiency with large samples and significant training data, the neural network should also be taken into account.

Keywords: COVID-19, ARIMA, ANN, pandemic

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 MATERIALS AND METHODS

The results reported in this research work 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;

$$y'_t = I + \varphi_1 y'_{t-1} + \varphi_2 y'_{t-2} + \cdots + \varphi_p y'_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \quad (2.0)$$

The AR part of ARIMA shows that the time series is regressed on its own 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 Gorge 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.

Box-Jenkins forecasting models consist of a four-step iterative procedure as follows;

a) Model Identification,

b) Model Estimation,

c) Model Checking (Goodness of fit) and

d) Model Forecasting.

Model Identification

Model identification involves determining the orders (p, d and q) of the AR and MA components of the model. Basically it seeks the answers for whether data is stationary or non- stationary? What is the order of differentiation (d), which makes the time series data stationary?

First stage of ARIMA model building is to identify whether the variable, which is being forecasted, is stationary in time series or not. By stationary we mean, the values of variable over time varies around a constant mean and variance. The ARIMA model cannot be built until we make this series stationary, one whose values vary more or less uniformly about a fixed level over time. This can be achieved by applying a technique of "regular differencing", a process of computing the difference between every two successive values, computing a differenced series which has all the overall trend removed.

Suppose a single differencing does not achieve stationarity, to have an ARIMA (p, d, q) model with “d” as the order of differencing used. In that case, it may be repeated, although rarely, if ever, are more than two regular differencing required. Caution is to be taken in differencing as over-differencing will tend to increase in the standard deviation, rather than a reduction. The best idea is to start with differencing with lowest order (of first order, d=1) and test the data for unit root problems.

The major tools used in the identification stage are plots of the series, correlogram of autocorrelation (ACF), and partial autocorrelation (PACF). The decision is not straight forward and in less typical requires not only experience but also a good idea of experimentation with alternative models (as well as technical parameters of ARIMA). However, a majority of empirical time series patterns can be sufficiently approximated using one of the 5 basic models that can be identified based on the shape of Autocorrelation (ACF) and partial autocorrelation (PACF).

The Autocorrelation Function (ACF)

For a covariance stationary time series $\{Y_t\}$ the autocorrelation function ρ_k is given by

$$\rho_k = Corr\,(Y_t, Y_{t-k}) for\ k = 1, 2, 3, \ldots \qquad (2.1)$$

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

The Partial Autocorrelation Function (PACF)

If $\{Y_t\}$ is normally distributed time series, then the PACF at lag k is given by

$$\emptyset_{kk} = Corr(Y_k, Y_{t-k} |\ Y_{t-1}, Y_{t-2}, \ldots, Y_{t-k+1}) \qquad (2.2)$$

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

The Extended Autocorrelation Function (EACF)

For a mixed ARMA model, ACF and PACF have infinitely many nonzero values, making it difficult to identify mixed models from the sample ACF and PACF. The extended autocorrelation function (EACF) (Tsay, R. and Tiao, G., 1984) is a graphical tool used to identify the ARMA orders.

According to (Cryer, J. and Chan, K., 2008) Let

$$W_{t,k,j} = Y_t - \Phi_1 Y_{t-1} - \ldots - \Phi_k Y_{t-k} \quad (2.3)$$

be the autoregressive residuals are defined with the AR coefficients estimated iteratively, assuming the AR order is *k* and the MA order is *j*. The sample autocorrelations of $W_{t,k,j}$ are referred to as the EACFs. Tsay, R. and Tiao, G. (1984) suggested summarizing the information in the sample EACF by a table with the element in the *k*th row and *j*th column equal to the symbol X if the lag *j* + 1 sample correlation of $W_{t,k,j}$ is significantly different from 0. In such a table, an ARMA (*p*,*q*) process will have a theoretical pattern of a triangle of zeroes, with the upper left-hand vertex corresponding to the ARMA orders.

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

$$y_t = c \; \emptyset_1 y_{t-1} + e_t \quad (2.4)$$

y_t is stationary when $|\emptyset_1| < 1$, with constant mean $\mu = \frac{c}{1-\emptyset_1}$ and a constant variance $\gamma_0 = \frac{\sigma^2}{1-{\emptyset^2}_1}$

Augmented Dickey-Fuller (ADF) Test of Unit Root

In Dickey-Fuller's test, null hypothesis (H_0) is that the series has unit root / not stationary while alternative hypothesis (H_1) 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 immediate previous one i.e $Y_t = X_t - X_{t-1}$ (2.5)

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 a model is identified, the next stage for the Box-Jenkins approach is to estimate the parameters. There are several methods for estimating the parameters. All of them should produce very similar estimates, but may be more or less efficient for any given model. In general, during the parameter estimation phase, a function maximization algorithm is used (i.e. the so-called Quasi-Newton method; refers to the description of the Nonlinear estimation method) to maximize the likelihood (probability) of the observed series, given the parameter values.
The estimate statement is used to specify the ARIMA model and to estimate the parameters of that model. Now the question may arise, how do we know whether the identified model is appropriate or not? One simple way to answer is diagnostic checking on residual term obtained from ARIMA model applying the same ACF and PACF functions. Obtain ACF and PACF of residual term up to certain lags of the estimated ARIMA model and then check whether the coefficients are statistically significant or not with either Box-Pierce Q or LJung-Box (LB) statistics.

If the result obtains from the model is purely random, then estimated ARIMA model is correct or else we have to look for alternative specification of the model. Similarly, diagnostic checking can also be done through Adjusted , minimum of Akaike Information Criterion (AIC), minimum of Bayesian information Criterion (BIC), Root Mean Square Error (RMSE), Schwarz Bayesian Criterion (SBC) and lowest mean absolute percent error (MAPE) values. However, for this project work AIC, BIC and RMSE will be used to choose the best fitted model.

AIC Criterion

Akaike's (1973) information criterion (AIC) plays a major role for selecting the best order of the ARIMA (p, d, q) model when we have several models that all adequately represent a given set of time series.

Suppose $\{Y_t\}$ is a Gaussian autoregressive ARMA (p,q) process with coefficient vector $\psi = (\phi, \theta)$. For a zero- mean causal invertible ARMA (p, q) process, In the general case, the AIC is calculated as;

$$AIC\ (\psi) = \ -2\ InL_x\left(\psi, S_x \frac{\psi}{n}\right) + 2k \qquad (2.6)$$

Where;

$L_x\left(\psi, S_x \frac{\psi}{n}\right)$ is the likelihood function

n is the sample size

k is the total number of parameters I. e k = p + q + 1

For fitting autoregressive models, Jones, R. (1975) and Shibata, R. (1976) suggested that AIC has a tendency to overestimate p. The AIC is a biased estimator. Hurvich and Tsai (1989) showed that the bias can be approximately eliminated by adding another nonstochastic penalty term to the AIC, resulting in the corrected AIC, denoted by AICc and defined by the formula

$$AIC_c = AIC + \frac{2(K+1)(K+2)}{N-K\ -2} \qquad (2.7)$$

BIC Criterion

Schwarz's Bayesian information criterion (1978), known as (BIC) is another criterion that attempts to correct the over fitting nature of the AIC. For a zero-mean causal invertible ARMA (p, q) process, the BIC is given by:

$$BIC\ (\psi) = \ -2\ In\ L_x\left(\psi, S_x \frac{\psi}{n}\right) + k\log(n) \qquad (2.8)$$

As a rule of thumb, we would expect as small value as possible for all of these criteria to select the most appropriate autoregressive model.

Model Checking (Goodness of Fit)

In this step, the model must be checked for adequacy by considering the properties of the residuals, specifically whether the residuals from an ARIMA model follow a normal distribution or are random. An overall assessment of the model's adequacy is provided using the Ljung-Box statistic.

The AIC and BIC are measures of the goodness of fit for an estimated statistical model. Given a data set, several competing models may be ranked according to their AIC or BIC, with the model having the lowest information criterion value being the best. After estimating the parameters of the ARIMA model, the next step in the Box-Jenkins approach is to check the adequacy of that model, commonly referred to as model diagnostics. Ideally, a model should extract all systematic information from the data. The diagnostic check is employed to determine the adequacy of the chosen model. One assumption of the ARIMA model is that the residuals should constitute white noise. A series εt is said to be white noise if εt is a sequence of independent and identically distributed random variables with a finite mean and variance. Additionally, if εt is normally distributed with a mean of zero and a variance of $\sigma 2$, then the series is termed Gaussian White Noise. For a white noise series, all ACF values should be zero. In practice, if the model's residuals are white noise, the ACF of the residuals will be approximately zero. If the assumptions are not met, a different model for the series must be explored. A statistical tool such as the Ljung-Box Q statistic can be used to determine whether the series is independent.

The partial autocorrelation function (PACF) identifies the appropriate lag p in an extended ARIMA (p,d,q) model. Both ACF and PACF are also used to verify whether the model selected based on AIC, BIC, and RMSE criteria is suitable. This involves testing whether the estimated model conforms to the specifications of a stationary univariate process. Specifically, the residuals should be independent of one another and constant in mean and variance over time. The unexplained part of the data (i.e., the residuals) should be minimal. Plotting the mean and variance of residuals over time, performing a Ljung-Box test, or plotting the autocorrelation of residuals are helpful methods for identifying misspecification. For this project work, the Ljung-Box Q statistic will be applied for the diagnostic checking of the selected model for the series. The Ljung-Box statistic is defined as:

$LB=n\ \ n+2\ \ k-1m\rho 2kn-k \sim \chi 2m$ (2.9)

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

Ljung-Box Q statistic:

The Box-Ljung test (1978) is a diagnostic tool used to test the lack of fit of a time series model. The test is applied to the residuals of a time series after fitting an ARMA (p,q) model to the data. The test examines m autocorrelations of the residuals. If the autocorrelations are very small, we conclude that the model does not exhibit a significant lack of fit.

In general, the Box-Ljung test is defined as:

H_0: The model does not exhibit lack of fit

H_1: The model exhibits lack of fit

The test statistic is given by

$$Q = n\,(n+2)\sum_{k=1}^{m}\frac{\hat{r}^2{}_k}{n-k} \qquad (2.10)$$

Where $\hat{r}_k$ is the estimated autocorrelation of the series at lag k, and m is the number of lags being tested. The Ljung Box test rejects the null hypothesis, indicating that the model has significant lack of fit, if $Q > \chi^2{}_{1-\alpha,h}$

Where $\chi^2{}_{1-\alpha,h}$ is the chi – square distribution table value with h degrees of freedom and significance level α. Because the test is applied to residuals, the degrees of freedom must account for the estimated model parameters so that $h = m - p - q$, where p and q indicate the number of parameters from the ARMA (p,q) model fit to the data.

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 which is inspired by the structure and the 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 1 & 2., the interconnected structure of a neural network is showed by indicating its simple internal processors.

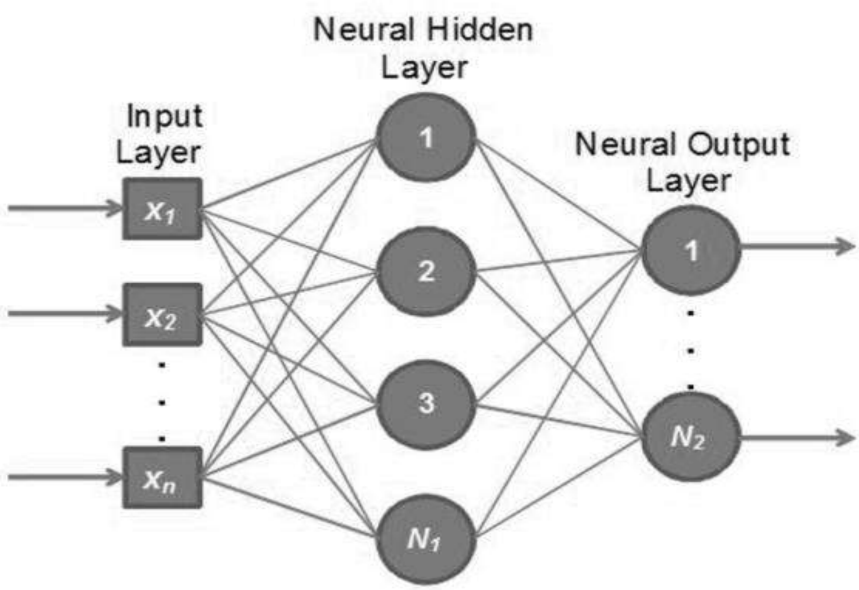


Figure 1: The network plot showing input layer, Hidden layer and output layer

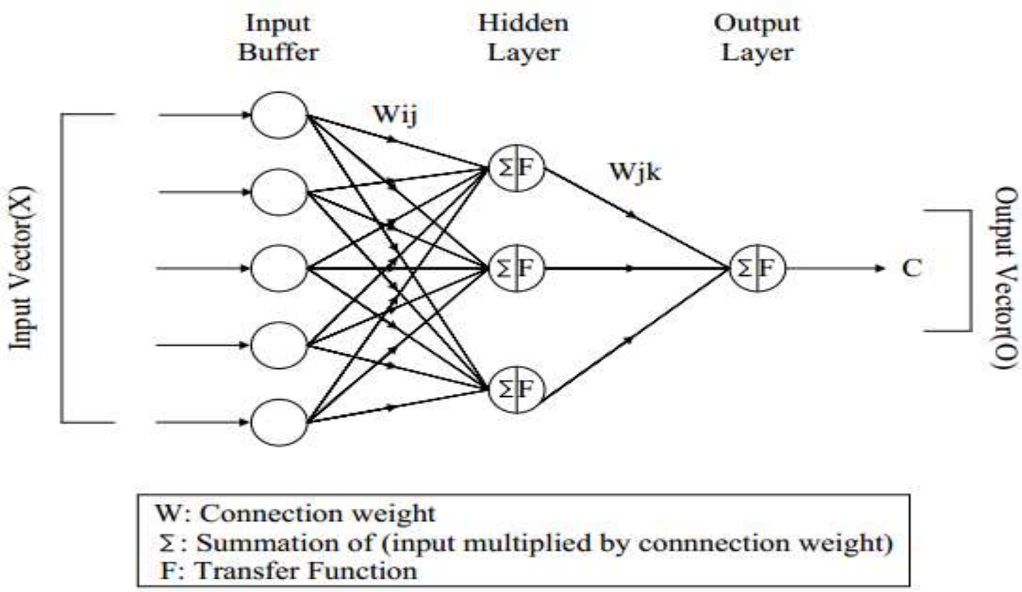


Figure 2: Neural Network Model

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 x_1, x_2, and x_3, then the synaptic weights to be applied to them are denoted as w_1, w_2, and w_3.

Output is

$$y = f(x) = \sum_{i=1}^{n} x_i w_i \ \textit{where n is the number of inputs} \qquad (2.11)$$

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:

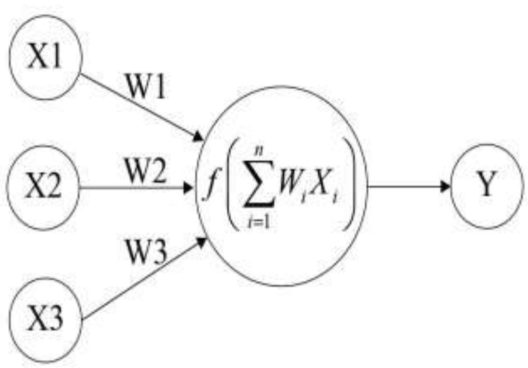


Figure 3: How the neuron in Artificial Neural Network works

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;

$$f(x) = \frac{e^x}{(1 + e^x)}$$

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 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 shows a forward and reverse pass with a single hidden layer:

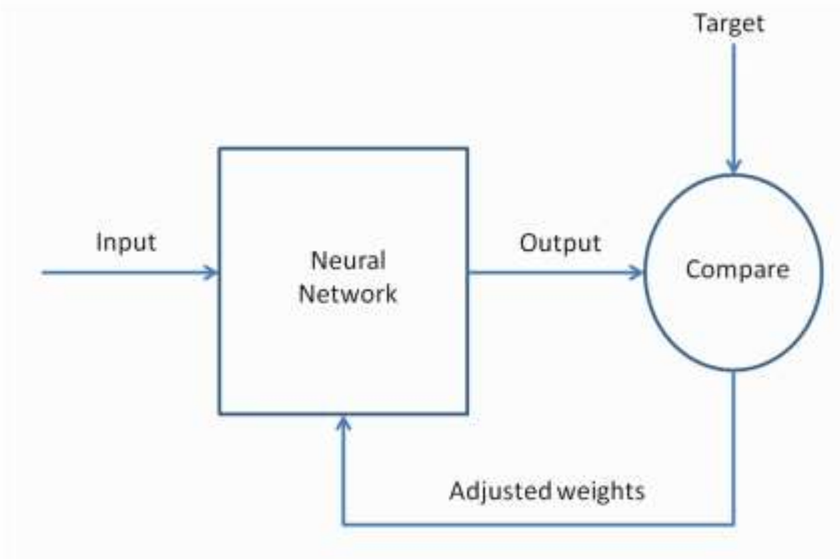


Figure 4: forward and reverse pass of neural network with a single hidden layer

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.

2.3.2 Details of Development of Neural Network

The data, details of which are explained earlier, were used in NN models development. Feed forward neural networks were used to develop ANN models for the prediction of the outcome variable. We divide the data into training and test set. Training set is used to find the relationship between dependent and independent variables while the test set assesses the performance of the model. For the purpose of this work, we use 60%, 80% and 100% of each of the dataset as training set and the remaining 40%, 20% as test set. The assignment of the data to training and test set is done using Block Sampling.

Before fitting a neural network, some preparation need to be done. As a first step, we address data preprocessing. It is extremely important to normalize the data before training a neural network. There are different methods to scale the data (z-normalization, min-max scale e. t. c). For the purpose of this research, we chose to use the min-max method and scale the data in the interval [0, 1] because scaling in the intervals [0, 1] tends to give better results. Without normalization, it is not possible to get accurate estimates by using NN models. All neural networks have one hidden layer including different numbers of hidden units resulting in one Architectures. The architecture is (7:2:1). The input layer has 7 inputs, the hidden layer has 2 neurons and the output layer has a single output.

2.4 Performance measure

We use three measures to evaluate the performance of each method. The first measure used is the Root mean squared error (RMSE), the mean absolute percentage error (MAPE), mean absolute error (MAE).

2.4.1 Mean Square Error (MSE)

It measures the average squared distance between the response, y_i and its predicted value $\hat{y}_i$ The MSE is popular, but is sensitive to large prediction errors (that is, large values of $y_i - \hat{y}_i$) since they are squared.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (2.12)$$

2.4.2 Mean Absolute Error (MAE)

The mean absolute error is a measure of difference between two continuous variables. Consider a scatter of n points, where point i has coordinates (x_i, y_i) plot. It is the average vertical distance between each point and the (Y = X) line, which is known as one-to-one line.

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \qquad (2.13)$$

2.4.3 Mean Absolute Percentage Error

The mean absolute percentage error also known as mean absolute deviation is a measure of prediction accuracy of forecasting methods in statistics. It is usually expressed in of percentage.

$$MAPE = \frac{1}{n}\sum_{i-1}^{n}\frac{A_t - F_t}{A_t} \qquad (3.14)$$

Where A_t is the actual value and F_t is forecast value.

2.5. Design of data

The data used for this study is a secondary data downloaded from Our-word 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

This section gives the result of the analysis based on the proposed method reported in chapter three. The Autoregressive integrated moving average (ARIMA), artificial neural network (ANN). After dividing the dataset into train (100%, 80% and 60%) and test data (20%, 40%), ARIMA model and ANN was fitted to each of the train data, the performance of the best model for ARIMA and ANN was tested using the test data. The overall performance of the two model was justified by considering their error measurement, mean square error (MSE), mean absolute error (MAE) and mean absolute percentage error (MAPE).

3.1 Estimation of Auto regressive Moving Average (ARIMA)

Table 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*

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, the plots in Fig. 5 shows the graphical visualization of the non-stationary data using time plot and qqplot. The ACF and PACF of the raw data can be plotted.

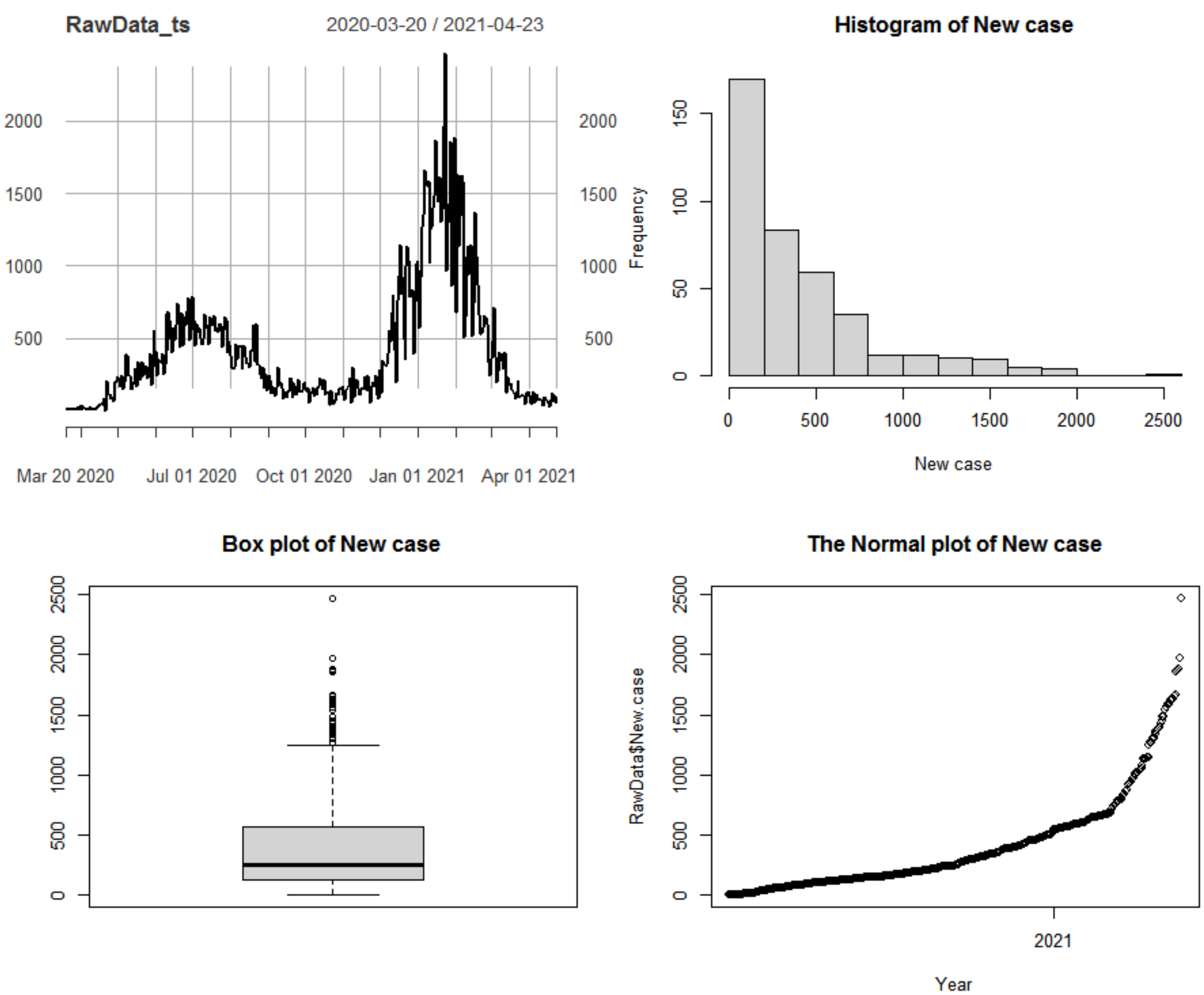


Figure 5: Graph of New case showing time plot, histogram, box plot and qqplot.

From the graph above, it is evident that the dataset is not normal and stationary.

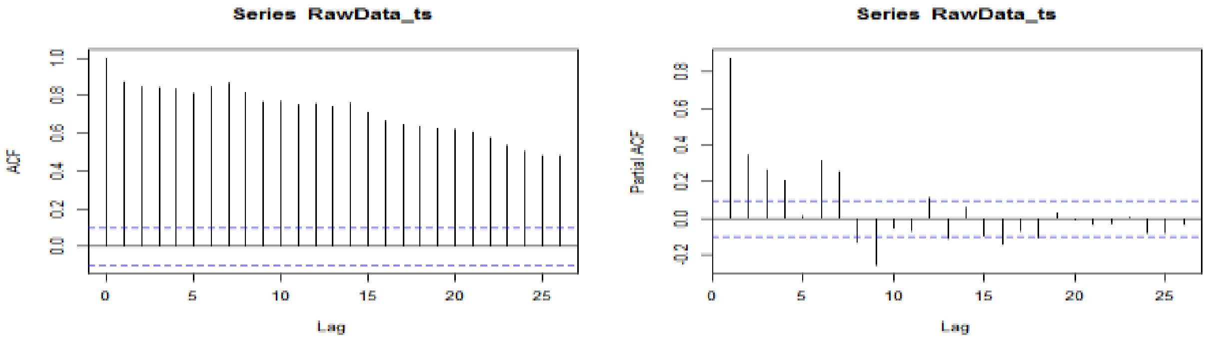


Figure 6: The graph of ACF and PACF of the raw dataset

From Figure 6, the ACF decay rapidly which shows evident of non-stationary.

3.1.1 Differencing

The Augmented Dickey Fuller statistic return a significant vale (D = -7.7679, $p < 0.05$)

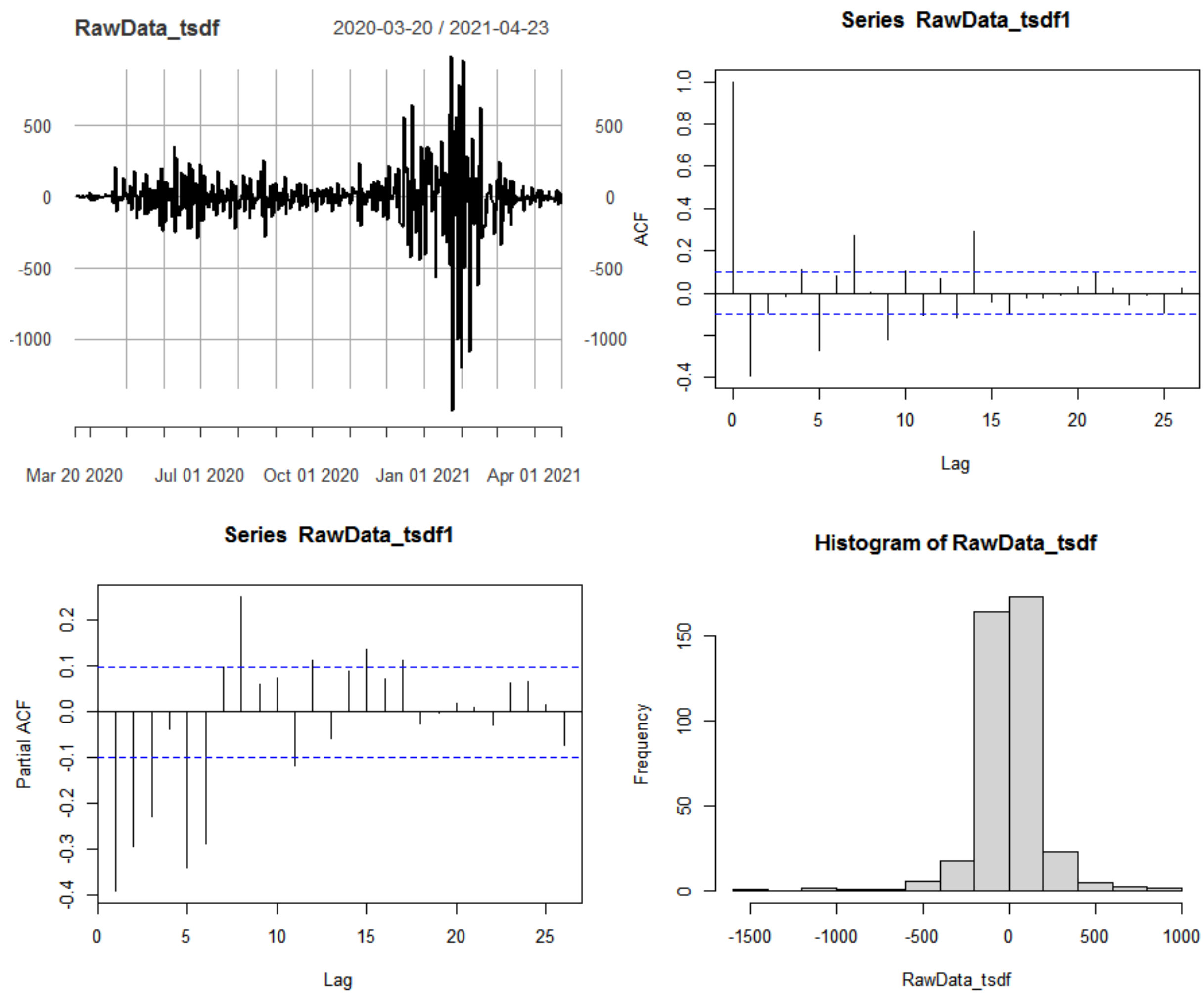


Figure 7: The graph of normalizing and differenced dataset.

From the graph above it can be show that, the time plot of the differenced dataset has a constant mean and variance over time, by differencing the data to be stationary, it can be shown by the histogram that the data are now normally distributed with mean zero and a constant variance. The ACF and PACF plot shows an evident of a stationary dataset.

3.1.2 Model identification

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 2

3.1.3 Model Estimation

Table 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 were fitted, AIC and RSME were shown in the table 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 show the results obtained from the R statistical software in Table 3. Here we see that $\phi\hat{}_1 = -0.835, \hat{\phi}_2 = -0.6070, \hat{\phi}_3 = -0.5361, \hat{\phi}_4 = -0.4034, \hat{\phi}_5 = -0.4911, \hat{\phi}_6 = -0.3432, \hat{\theta}_1 = 0.1324$. 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: Maximum Likelihood Estimates from R Software: COVID 19 cases

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

Sigma^2 estimated as 25.628: log likelihood = -2066.06 AIC = 5205.97 AICc = 5207.23 BIC = 5265.77

The estimated model would be written as

$$\begin{aligned} W_t - 0.312 = {} & -0.8357\,(W_{t-1} - 0.312) - 0.6070(W_{t-2} - 0.312) - 0.5361(W_{t-3} - 0.312) \\ & - 0.4034(W_{t-4} - 0.312) - 0.4911(W_{t-5} - 0.312) - 0.3432(W_{t-6} - 0.312) \\ & + e_t + 0.1324e_{t-1} \end{aligned}$$

Where $W_t = Y_t - Y_{t-1}$ *and the intercept of ARIMA is* $\theta_0 = \mu\,(1 - \phi)$,

3.1.4 Model adequacy

Test of Randomness using LJUNG BOX

H_0: The model does not show lack of fit

H_1: The model *does* show a lack of fit

Table 4: 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

In order to test the adequacy of ARIMA (6, 1, 1) the hypothesis above was tested. The decision rule is given as: do not reject Ho, if and only if the p-values of box-pierce test at non seasonal lag 7 and 8 are greater than 0.05, otherwise the null hypothesis will be rejected.

The p-values are all greater than 0.05 (95% confidence interval) therefore we fail to reject the null hypothesis. Hence, the model is good to forecast the future cases of COVID-19 in Nigeria.

The ACF residuals, PACF residuals, standardize residuals, and Box Ljung plot for the various p value are shown in Fig. 8

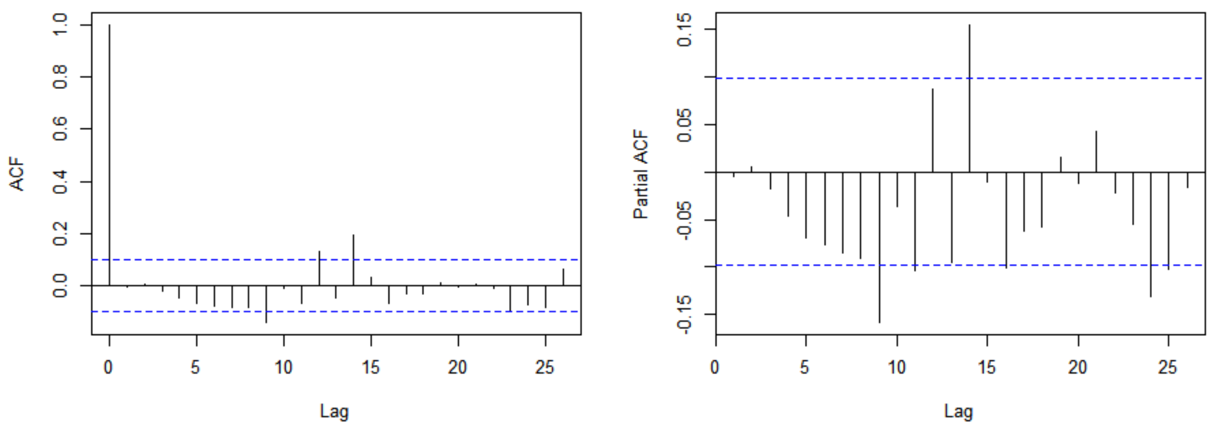


Figure 8: The residuals plot of ACF and PACF of the best model

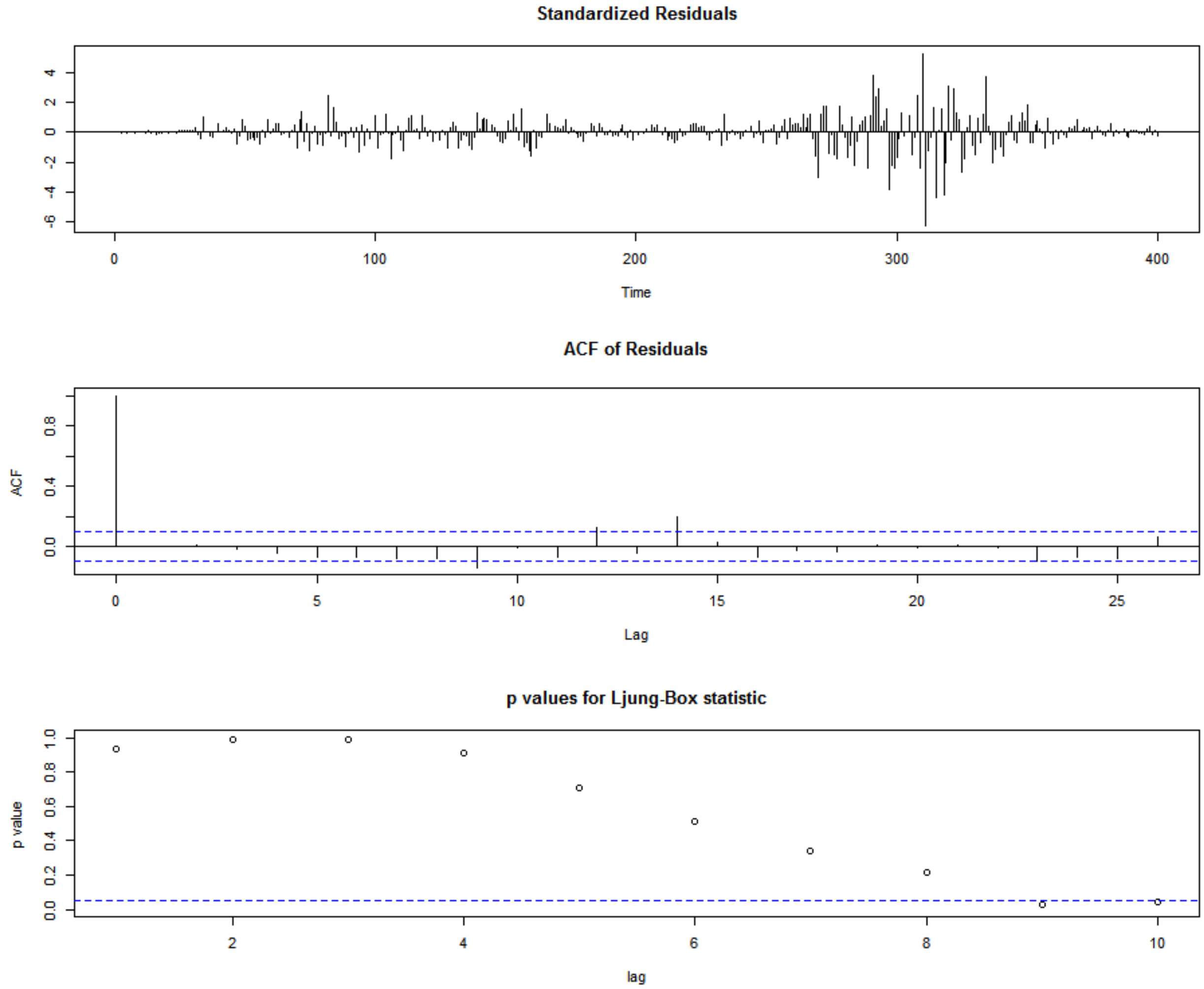


Figure 9: The standardized residual plot, p values for Ljung – Box statistic plot and the ACF residuals plot

In accordance with ARIMA model assumptions of time series: white noise assumes that the residuals have zero mean, constant variance and the autocorrelation of any two observations of such sequence is always zero (uncorrelated). The Figure 9 of ACF residual plot of the standardized residuals shows no significant autocorrelation and it also shows a zero mean and constants variance. Also the p-values of the Box-Pierce Statistics for each lag up to 8 are all significant indicating that the model is adequate and reliable enough. The normality test using Shapiro Wilk test of the residual return a non- significant value, which ascertain the assumption that the White noise are normally distributed. The above supporting proves gave reasons to use the ARIMA (6, 1, 1) as the best model. Since the model ARIMA (6, 1, 1) has been confirmed to be adequate and its coefficient is significant, we then now proceed to divide the dataset into train and test data.

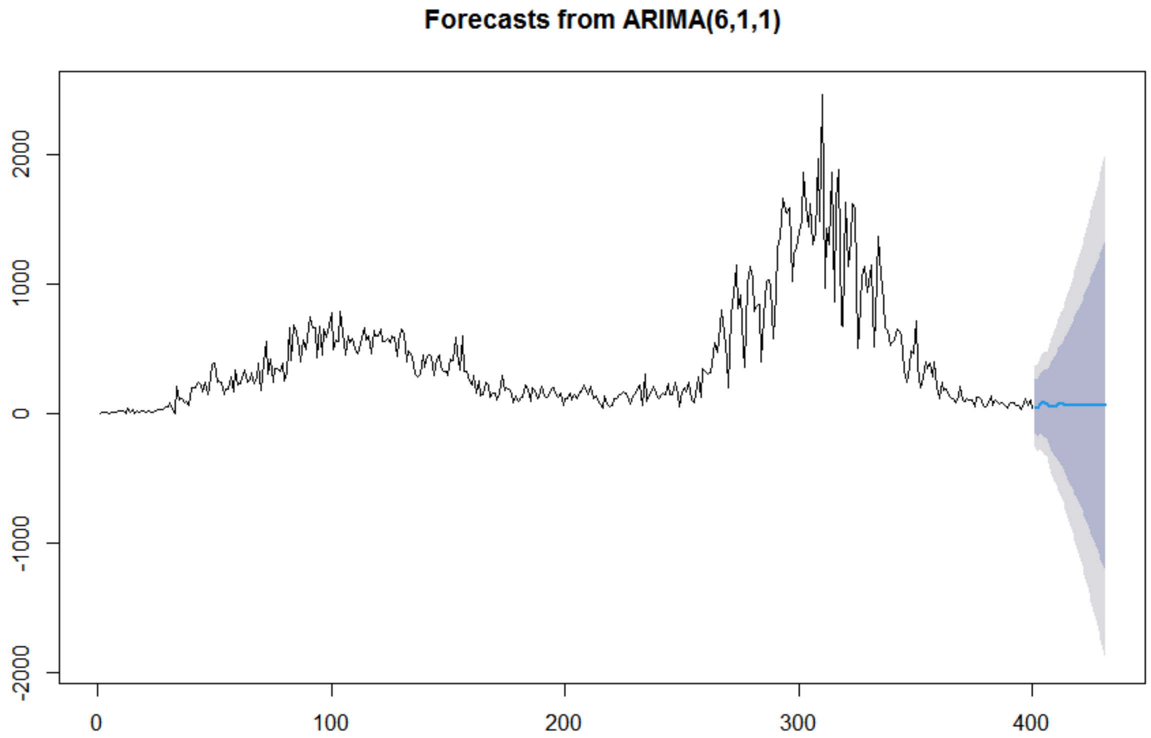


Figure 10 : Data and forecasted results of ARIMA (6, 1, 1) models for COVID 19 cases in Nigeria

3.2 Fitting Artificial Neural model For COVID 19 cases in Nigeria

Applying ANN, the percentage of observations for training, which must have the same number of observations, 400, as we have in ARIMA for training is determined, we have analyzed 80% for training, and 20% for comparison in the prediction. The layers may be described as: *Input* layer: accepts the data vector or pattern; *Hidden* layers: one or more layers. *Output* layer: takes the output from the final hidden layer to produce the target values.

In choosing the number of layers, the following considerations are made. Multi-layer networks are harder to train than single layer networks. A two layer network (one hidden) can model any decision boundary. Two layer networks are most commonly used in pattern recognition.

The number of output units is determined by the number of output classes. The number of inputs is determined by the number of input dimensions. The network will not model complex decision boundaries for few hidden units and it will have poor generalization for too many number of hidden units. We started with one hidden layer and end with two layers (first layer with 4 neurons and second layer with two neurons). The performance of the algorithm is influence with choosing different learning rates. The algorithm may become unstable for high learning rate and might take longer time to converge.

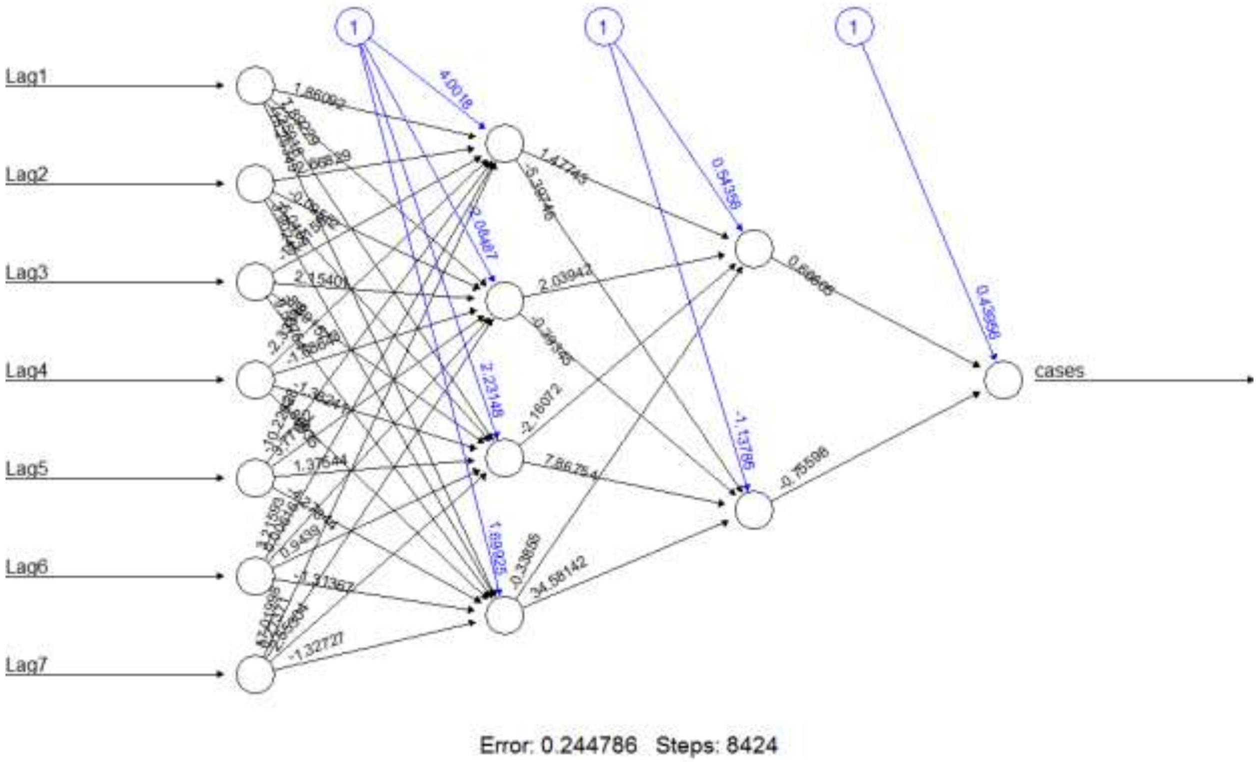


Figure 11: The net plot for 80% training and 20% testing data

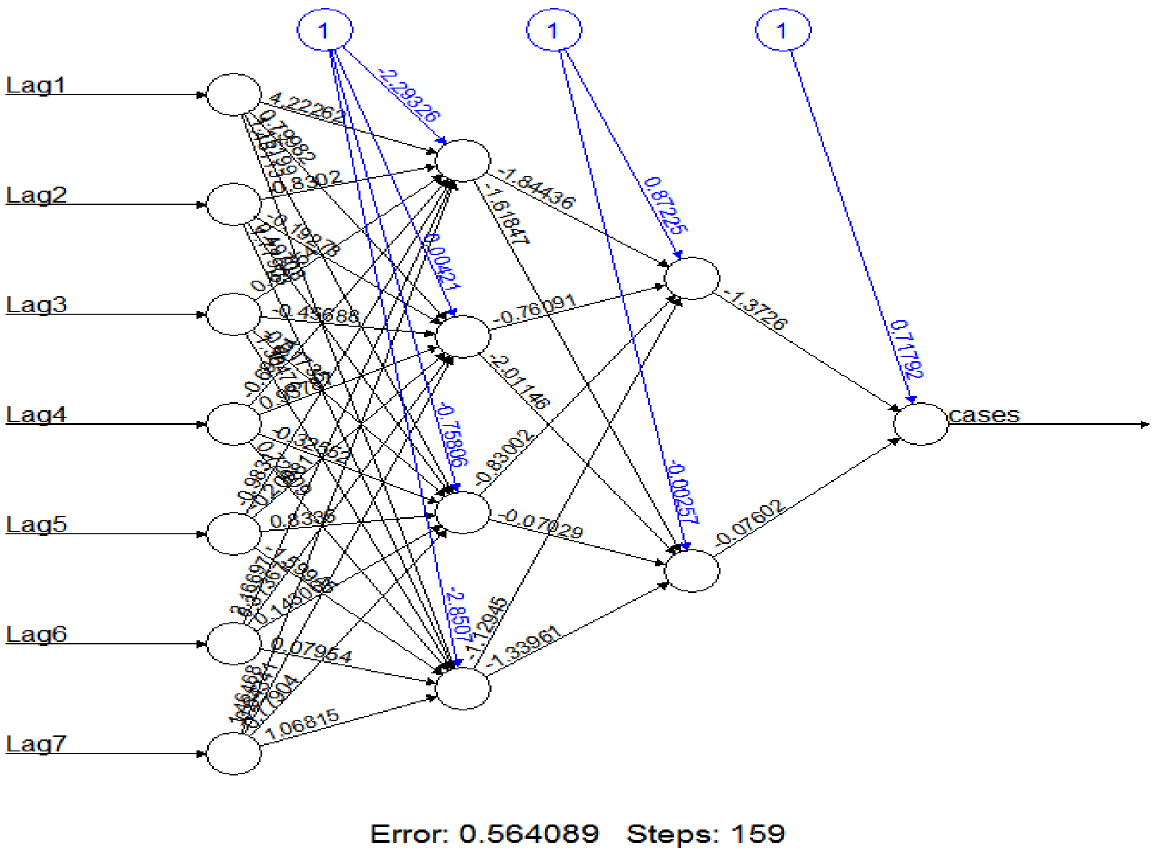


Figure 12: The net plot for 60% training and 40% testing data

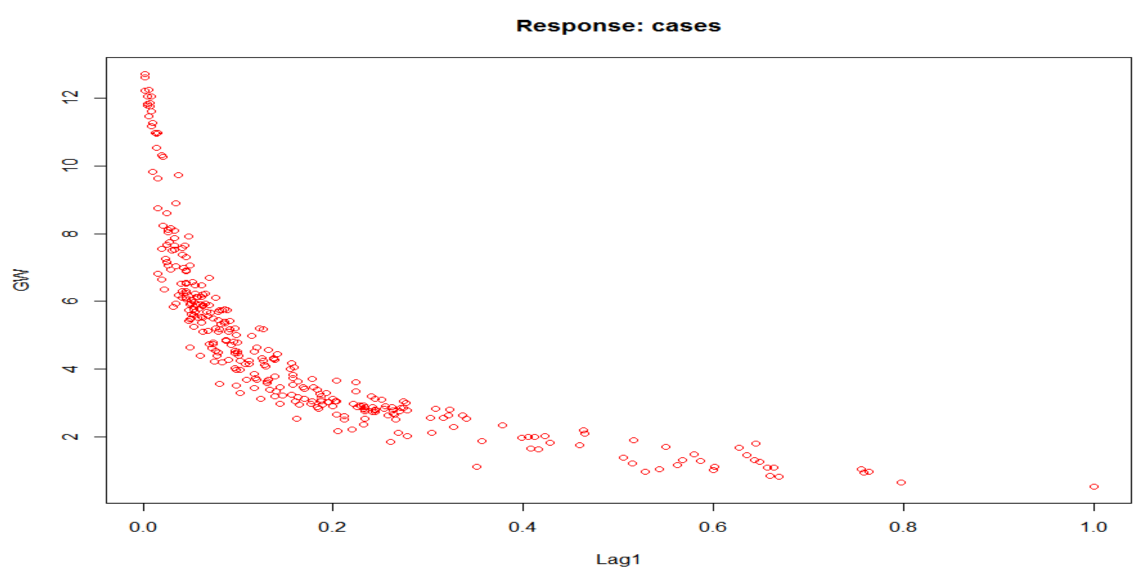


Figure 13: The weight plot of the normalized inputs unit

R-software is used for fitting 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).

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 5 and 6 respectively.

Table 5: 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 6: 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 7: 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	Forecast	
		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 8: 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 obs			
Date	Actual data	Forecast	
		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

Table 9: 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

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 5 and 6). This result shows that 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 4.6 and 4.7. 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 9 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 thanks, adorations, appreciation, glorification and honour are due to none except Almighty Allah for sparing my life and for giving me knowledge, wisdom and understanding throughout the course of struggle in FUTA. I owe a debt of gratitude and appreciation to my supervisor Prof. Fasorobanku Olusoga Akin, whose incredible patience, inspiration, hard work and support helped me complete this project. You always took your time to listen to my ideas and answer any questions that I had. Your patience, simplicity and leadership skills will never be forgotten. Thank you very much! Many heartfelt thanks to statistics department's lecturers who have helped to build me into the statistician that I am today, as well as the one that I will become. I am indebted to you all for the knowledge you have passed unto me. God Bless you all.

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.

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.

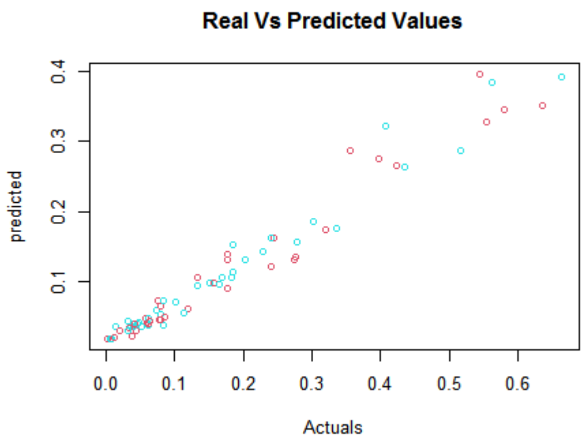
Song PX, Wang L, Zhou Y et al (2020). An epidemiological forecast model and software assessing interventions on COVID-19 epidemic in China. *medRxiv,* doi.org/10.1101/2020.02.29.20029421

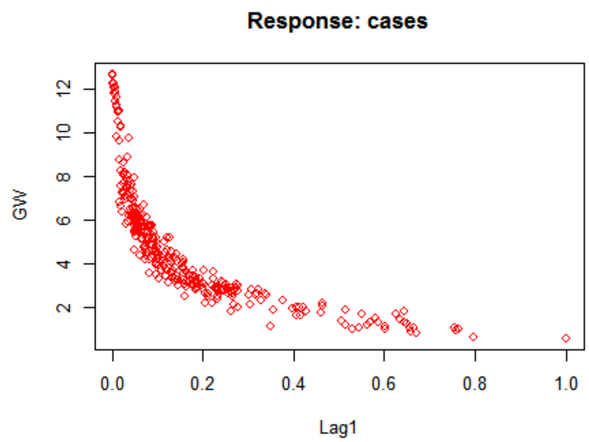
Sehwan S, Gao D, Zhuang Z et al (2020). Estimating the serial interval of the novel coronavirus disease (COVID-19): A statistical analysis using the public data in Hong Kong from January 16 to February 15, 2020. *medRxiv,* doi:https://doi.org/10.1101/2020.02.21.20026559

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.

Appendix


Real Vs Predicted Values
predicted
Actuals
0.1
0.2
0.3
0.4
0.0
0.1
0.2
0.3
0.4
0.5
0.6


Response: cases
GW
Lag1
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0