https://ijfscfrtjournal.isrra.org/index.php/Formal_Sciences_Journal/index

Modeling of the Microsoft Stock Prices Using Machine Learning and Classical Models: Identification of Optimal Model for Application

Chrysogonus Chinagorom Nwaigwe^a, Desmond Chekwube Bartholomew^{b*}, Emmanuel Chigozie Umeh^c, Godwin Onyeka Nwafor^d, Ibrahim Adamu^e Simplicius Chidiebere Oguguo^f

^{a,b,c,d,e}Department of Statistics, Federal University of Technology Owerri, Imo State, Nigeria ^fDepartment of Mathematics, Imo State College of Education, Ihitte Uboma, Imo State, Nigeria ^aEmail: chrysogonus.nwaigwe@futo.edu.ng ^bEmail: desmond.bartholomew@futo.edu.ng ^cEmail: emmanuelchigozie60@gmail.com ^dEmail: godwin.nwafor@futo.edu.ng ^eEmail: Ibrahim.adamu@futo.edu.ng ^fEmail: Simpliciusoguguo@gmail.com

Abstract

In this study, Microsoft stock price was modeled using two traditional time series models and two machine learning models for reliable predictions of the future behavior of the stock prices and more gainful investment. Model metrics such as AIC, BIC, Log-likelihood, RMSE, and confidence set test were the basis for comparison of the models. The results showed that the GARCH model outperformed the ARIMA and Support Vector Regression models while the Long Short-Time Memory –Recurrent model outperformed the GARCH model. Forecasts from the Long Short-Time Memory were made and found to be highly reliable. The results of the forecast also showed an uptrend movement up to a price of around \$275 from November 2023 to January 2024. In conclusion, the LSTM-RNN is capable of accurately tracking and forecasting movements of volatile stock prices and is preferred over the other models considered in this study.

Keywords: Machine learning models; Traditional time series models; Stock prices; Support zone; Resistance zone.

Received: 10/12/2023 Accepted: 11/16/2023 Published: 11/26/2023

* Corresponding author.

1. Introduction

The stock market is an essential aspect of any economy, as it facilitates the allocation of resources and raises capital for businesses. As such, the ability to predict stock prices has garnered significant attention from investors, financial analysts, and researchers alike. Nigeria, as the largest economy in Africa, has a thriving stock market, and accurate predictions of stock prices could be valuable to investors and policymakers. Suppose a reliable and efficient model could be built to anticipate the short-term price of an individual stock, as well as the pace at which individuals invest in the stock market.

In that case, the stock market's business potential may expand. Investment decisions must be based on accurate forecasts because of the tremendous volatility of stock prices.

Modeling a system and anticipating its future behavior has been widely researched and studied for many years, particularly the topic of trend analysis on time series or index series, which can both be represented as a sequence [1]. Unfortunately, in many forecasting situations, the most traditional statistical model cannot produce the results expected. This is because a standard statistical model must evaluate whether the system is a linear or nonlinear model, what the right order of function for prediction is, and how to test the forecasting model's fitness [2]. Over the years, many technologies, including machine learning and statistical methodologies, have been employed to predict stock prices. However, due to the increased amount of data and the expectation of more accurate model prediction, machine learning models are now being utilized, which have an advantage in terms of accuracy and speed of prediction over classic statistical time series methods. Therefore, an alternative is to seek kinds of intelligent methods as prediction tools like the Long Short-Time Memory Recurrent Neural Network (LSTM-RNN) and Support Vector Machine (SVM) in which the crucial problem mentioned in the traditional statistical model can be avoided.

Several authors have studied stock market prices and the findings are reviewed thus:

Reference [3] forecasted stock market indices on the New York Stock Exchange using linear regression and simple neural network models from 1981 to 1999. The model used pattern recognition that efficiently and accurately identified spikes in the trading volumes and forecasted future changes in price based on the historical price, traded volume, and the prime interest rate. Three years later, an ensemble model comprising SVM and artificial neural networks (ANNs) for stock price prediction was proposed by [4]. The ensemble model outperformed the single model according to their empirical findings. In 2010, [5] compared SVM with a multi-layer back-propagation (BP) neural network in time series forecasting. Their results suggested that SVM outperformed the BP neural network. A year later, Reference [6] used ARIMA-Intervention time series analysis as both an analytical and forecast tool for the values of the stock price index of the chosen company in their study. Reference [7] later compared the traditional time series decomposition (TSD) model, the Hot Winters (H/W) exponential smoothing with trend and seasonality model, the Box-Jenkins (B/J) model using auto-correlation, and the machine learning model—neural network-based models. The result of their study showed that the machine-learning model outperformed the traditional models. However, [8] revealed that the ARIMA model has a high potential for short-term prediction and can compete favorably with

existing stock price prediction techniques. Feed-forward neural networks using a back propagation technique were compared with panel data regression analysis for the prediction of stock price [9]. The findings showed that the neural network method was superior and more effective than regression in predicting stock values. [10] using daily closing prices of SENSEX from 1997-2014 examined the efficiency of the Indian stock market if stock returns follow a random walk. Results show that the market is inefficient in weak form, suggesting that past stock prices may not reflect all information, and abnormal returns can be achieved by exploiting market inefficiency. A year later [11] studied the Random Walk Hypothesis (RWH) and market efficiency of stock market indices like the London Stock Exchange, EuroStoxx 50, NIKKI, Shanghai Composite Stock Exchange, and Bombay Stock Exchange. Their results showed that the null hypotheses were rejected, with few acceptances based on test statistics. In the same 2018, the Recurrent Neural Network (RNN) algorithm was applied to the prediction of the closing price of a specific stock [12]. The predicted closing prices were cross-checked with the true closing price, and it was suggested that the model could be used to make predictions for other volatile financial instruments. Their claim was supported by [13], who compared three models: ARIMA, ANN, and LSTM (Long-short-term memory). Their findings suggested that LSTM had the strongest prediction performance but was strongly affected by data processing. However, the ANN model outperformed the ARIMA model., The results of [14,15] concluded that their deep learning models outperformed other models in forecasting stock price movement and stock price due to the extensive feature engineering they developed. Other machine learning models have been studied also: Random Forest Regression outperformed Extra Tree Regression as in [16] while the geometric Brownian motion model outperformed the multilayer perceptron in [17].

Recently, Reference [18] used different machine learning algorithms using historical stock price data. In their research, they used five regression models: linear regression, random forest, support vector regression (SVM), vector auto-regression (VAR), and long-term short-term memory (LSTM). The results showed that the LSTM model outperformed all of the others. Reference [19] compared time series models (autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA)) with integrated artificial neural networks and meta-heuristic algorithms on stock price forecasting. The artificial neural network (ANN) was trained with meta-heuristic algorithms, including the social spider optimization (SSO) and bat algorithm (BA). Their findings showed that a hybrid meta-heuristic-based ANN outperformed others. Reference [20] proposed a recurrent neural network (RNN) and long short-term memory (LSTM) model by using machine and deep learning models to predict the trend in stock prices that would be more accurate. In his experiment, by increasing the epochs and batch size, the accuracy of prediction increased. The proposed method is capable of tracing and predicting stock market movements with highly accurate results.

This study aims to compare the performance of the Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Variance (GARCH), Long Short-Term Memory-Recurrent Neural Network (LSTM-RNN), and Support Vector Regression (SVR) in the prediction of Microsoft Stock closing prices. The specific objectives are to: (i) model the stock price data of Microsoft using the ARIMA model (ii) model the stock price data of Microsoft using the GARCH model (iii) model the stock price data of Microsoft using Long Short Time Memory-Recurrent model (iv) model the stock prices data of Microsoft using Support Vector Regression model (v) compare (a) the predictive power of (i) and (ii) which are the traditional statistical time series models (b) the predictive power of (iii) and (iv) which are the machine learning models and finally (vi) use the best model out of the four models to forecast future values of the stock prices of Microsoft.

The rationale for this study is that precise stock price forecasting can lead to a higher profit yield for stock investors especially those who invest in Microsoft stocks. This is because although Apple's stock is trading at a better value and is the better growth stock to buy right now keeping Microsoft stock on your radar for future investment is still a good idea [21]. Therefore, the motivation for studying Microsoft stock is to assist investors who keep Microsoft stock to choose securities that will deliver a higher return, according to predictions. However, this study is limited to Microsoft stock close prices for a period of 27 months, from September 2020 to December 2022 because we intend to model the behavior of Microsoft stock prices as influenced by the COVID-19 pandemic. This paper is divided into sections; Section 2: Materials and Methods is the next section to be discussed, followed by the other remaining sections and the references.

2. Materials and Methods

This section contains the data collection method, the descriptions of the [2] model, the GARCH model, the LSTM model, and finally SVM model.

2.1 Method of Data Collection

The data used in this study was historical daily stock prices for the Microsoft Corporation, and secondary data obtained from the YahooFinance website (www.yahoofinance.com). The date, open price, low price, high price, closing price, and volume traded are all part of the stock data. However, in this study, the closing price is used for modeling and prediction of the stock prices. This is because the closing price reflects the entire day's activities.

2.2 ARIMA (p, d, q) model for Stock Price of Microsoft Incorporation

ARIMA is a class of models that estimate future values by explaining a given time series based on its past values, that is, its lags and lagged prediction errors. ARIMA models can be used to simulate any non-seasonal time series with a pattern that is not random white noise [22]. According to [23], ARIMA is an acronym for autoregressive integrated moving average. It combines the autoregressive (AR) and moving average (MA) models, as well as the integrating process, which converts a non-stationary time series variable to a stationary one. Reference [2] established a three-step technique for deciding the best ARIMA model, which is critical in the selection process. The three steps are identification, estimation, and diagnostic testing. These three steps can be done several times to find the optimal model [2]. While selecting the optimum ARIMA model, the notion of parsimony is critical to avoid overfitting the model. As explained by [23], "d" is the number of times the data is differenced to make it stationary, "p" is the number of lags that crosses the significant limit in the Partial Autocorrelation Function (PACF) plot, and "q" is the number of lags that crosses the significant limit in the Autocorrelation Function (ACF) plot. The method in [2] has been widely applied by researchers in modeling financial data like [24,25], etc

According to [23], given the Microsoft closing stock price x_t , the ARIMA (p, d, q) model is given as in equation (1):

$$\phi(B) (1-B)^d X_t = \theta(B) Z_t \tag{1}$$

where

 $\phi(B)$ is the characteristic polynomial of order "p" for the autoregressive component of the model.

 $\theta(B)$ is the characteristic polynomial of order "q" for the moving average component of the model.

 $(1-B)^d$ is the differencing of order "d" of the data.

 X_t is the observed value at the time t

 Z_t is the random error associated with observation at time t

3.2.1 Model Identification

The series (Microsoft stock closing prices) are tested for stationarity assumption using different approaches. The approaches include:

i. Observing time series, ACF, and PACF plots.

The autocorrelation between X_t and its value X_{t-k} , separated by k intervals of time is called the autocorrelation (ρ_k) at lag k and is defined by equation (2):

$$\rho_{k} = \frac{E((x_{t} - \mu)(x_{t+k} - \mu))}{\left\{E((x_{t} - \mu)^{2} E(x_{t+k} - \mu)^{2})\right\}^{\frac{1}{2}}}$$
(2)

where

 $E((x_t - \mu)(x_{t+k} - \mu))$ is the autocorrelation at lag $k, k = 0, \pm 1, \pm 2, ...$ and the denominator is variance at lag k which is the same at lag k + 1 for stationary series.

Similarly, for any given process X_t , $t \in z$, a partial autocorrelation at lag k is given by equation (3):

$$\varphi_{k+1,k+1} = \frac{\rho_{k+1} - \sum_{j=1}^{k} \varphi_{kj} \rho_{k+1,j}}{1 - \sum_{j=1}^{k} \varphi_{kj} \rho_{j}}$$
(3)

where,

$$\varphi_{_{k+1,j}} = \varphi_{_{kj}} - \varphi_{_{k+1,k+1}} \varphi_{_{k,k-j+i}}; j=1,2,...,k$$
 and

$$\varphi_{11} = \rho_1$$
, for $k = 1$

In the same way, the matrix version is given by equation (4):

$$\varphi_{kk} = \frac{\left| \rho_k^* \right|}{\left| \rho_k \right|} \text{ for } k > 1 \tag{4}$$

where

 ρ_k is an k imes k autocorrelation matrix

with the last column replaced by $\rho^T = (\rho_1, \rho_2, ..., \rho_k)$ The autocorrelation matrix for any series of length n is given by:

[1	$ ho_{_1}$	$ ho_{_2}$		$ ho_{\scriptscriptstyle n-1}$
$ ho_1$	1	$ ho_{_1}$		$ ho_{\scriptscriptstyle n-2}$
$ ho_2$	$ ho_{_1}$	1		$ ho_{\scriptscriptstyle n-3}$
	•	•		
-	•	•		
	•	•		
$ ho_{{}_{n-1}}$	$ ho_{\scriptscriptstyle n-2}$	$ ho_{\scriptscriptstyle n-3}$	•••	1

The Partial Autocorrelation Plot (PACF) and Autocorrelation Plot (ACF) are used to determine the values of p and q respectively as described in [23]. The determination of the values q depends on the number of significant spikes that cross the significant region (the blue-shaded region, this blue area depicts the 95% confidence interval and is an indicator of the significance threshold) in Figure 11. According to [26], the basic guidelines for interpreting the ACF and PACF plots are as follows: Look for tail-off patterns in either ACF or PACF. If tail off at ACF \rightarrow AR model \rightarrow Cut off at PACF will provide order p for AR(p). If tail off at PACF \rightarrow MA model \rightarrow Cut off at ACF will provide order q for MA(q).

(ii) The second approach for identifying the appropriate ARIMA model is by conducting an Augmented Dickey-Fuller test on the series. This test considers different assumptions such as under constancy, alongside no drift, or along a trend and a drift term. If the series is not stationary, then the first or second difference is likely to be stationary.

The hypothesis as given in [27] is:

 $H_0: |\phi_1| = 1$, that is: the process contains a unit root and therefore it is non-stationary.

 $H_1: |\phi_1| < 1$, that is: the process does not contain a unit root and is stationary.

Decision: If the p-value < 0.05, we reject the null hypothesis. This means that there is stationarity in the stock data.

2.2.2 Model Estimation

Once stationarity is attained, the next thing is to fit different values of p and q, and then estimate the parameters of the ARIMA model. We use iterative methods to select the best model based on the following measurement criteria: AIC (Akaike information criteria) BIC (Bayesian information criteria) and log-likelihood.

i. Akaike's Information Criteria (AIC) =
$$-2*Log(L)+2(p+q+k+1)$$
 (5)

where

L is the likelihood of the series

and k = 0, if c = 0

ii. Bayesian Information Criteria (BIC) =
$$AIC + [Log(T) - 2]*(p + q + k + 1)$$
 (6)

where

T is the maximum likelihood estimation limit

iii. **log-likelihood of the data**: This is the logarithm of the probability of the observed data coming from the estimated model. The larger the log-likelihood, the better the model.

Note: smaller values of AIC, and BIC with maximum log-likelihood indicate a better model.

2.2.3 Model Diagnosis

The conformity of the white noise residual of the model fit will be judged by plotting the ACF and the PACF of the residual to see whether it does not have any pattern, when steps 1-3 are achieved, we go ahead and fit the model.

2.3 Symmetric Models

2.3.1 Autoregressive Conditional Heteroscedastic (ARCH) Model

The Autoregressive Conditional Heteroscedastic (ARCH) model is used to model the error of the conditional variance of a series. Suppose we are modeling the error variance of a series X_t , the ARCH (1) model for the error variance is the condition X_{t-1} at time t (it implies q = 1).

Mathematically, the ARCH (1) model is represented in equation (7):

$$\sigma_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + \dots + \alpha_q u_{t-q}^2$$
(7)

We impose the constraints that $\alpha_0 \ge 0$, $\alpha_1 \ge 0$ and $\alpha_q \ge 0$ to avoid negative variance.

An ARCH(q) model can be estimated using ordinary least squares. The mean equation for prediction purposes which was adopted in this study for predicting the stock price is available in [28].

2.3.2 Generalized Autoregressive Conditional Heteroscedastic: GARCH (p, q) Model

One of the most challenging problems nowadays for econometrics academics, time series analysts, and policymakers is modeling the variance that happens in an econometric series, which has become a significant source of worry in the financial markets. This has been the subject of all research for a very long time. Once there is volatility clustering in the time series, the ARIMA is not appropriate. Volatility clustering is simply a continuous rise and decrease over time in a time series. Reference [28] argues that an adequate volatility model is one that sufficiently models heteroscedasticity in the disturbance term and also captures the stylized facts inherent in stock return series, such as volatility clustering, the autoregressive conditional heteroscedasticity and its extensions, such as generalized ARCH, threshold GARCH, exponential GARCH, and power GARCH. In most cases, first-order GARCH models have been extensively proven to be adequate for modeling and forecasting financial time series. Reference [29] employed empirical data and various GARCH models to evaluate the volatility of the Malaysian stock market (symmetric and asymmetric). Other researchers include [30,31,32,33].

The Generalized Auto-regressive Conditional Heteroskedasticity (GARCH) model is a statistical model used to analyze time series data that exhibit conditional heeroskedasticity, which is the phenomenon where the variance of the errors in a time series changes over time. In other words, the GARCH model allows for the estimation of time-varying volatility in a time series. The model was first introduced by [28] in the 1980s, and it has since become a widely used tool in financial econometrics for modeling asset returns, volatility, and risk. The GARCH model builds on the Autoregressive Conditional Heteroskedasticity (ARCH) model, which assumes that the variance of a time series is a function of its past values. The GARCH model extends this concept by allowing the variance to depend not only on the past values of the time series but also on the past values of its squared errors.

The past squared observation value and past variance are used by the GARCH model to model the variance at time t. The conditional variance is allowed to depend on prior lags in the model. The models gauge how much a volatility shock from today will affect volatility in the coming term. It gauges how quickly this effect has subsided over time. The GARCH (1, 1) variance process model is defined in equation (8).

$$\sigma_{t}^{2} = \alpha_{0} + \alpha_{1} u_{t-1}^{2} + \beta_{1} \sigma_{t-1}^{2}$$
(8)

A GARCH(p,q) model can be estimated using ordinary least squares. The mean equation for prediction purposes which was adopted in this study for predicting the stock price is available in [34].

2.4 Methods of Estimation of GARCH Models

2.4.1 Maximum Likelihood Function (MLF)

The maximum likelihood estimator is the technique used to estimate the GARCH model. The technique is used to determine the parameter value that is most likely given the actual series. The GARCH model is estimated in the following two phases.

(i) Specify the mean and variance equation, for example, AR(1) in equation (9) and GARCH(1,1) in equation (10)

$$y_{t} = \mu + \theta(y_{t-1}) + \mu_{t}; \quad \mu \sim (0, \sigma_{t}^{2})$$

$$\tag{9}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$
(10)

(ii) Estimate the likelihood function to maximize the normality assumption of disturbance terms given in equation (11)

$$\log L = -\frac{\tau}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^{\tau} \log(\sigma_{t}^{2}) - \frac{1}{2} \sum_{t=1}^{\tau} \frac{\mu_{t}^{2}}{\sigma_{t}^{2}}$$
(11)

2.5 Long Short-Time Memory Recurrent Neural Network (LSTM-RNN)

The LSTM-RNN (Recurrent Neural Network) is a soft computing method for modeling sequential data. It is made up of multiple self-connected LSTM cells that are utilized to store the networks' temporal state utilizing three gates: input, output, and forget gates. Stock price prediction is a tough task that can be represented using Machine Learning and Artificial Neural Networks techniques. RNNs (Recurrent Neural Networks) are a type of neural network that is particularly effective at processing time series and other sequential data [35]. The LSTM-RNN is a form of Recurrent Neural Network composed of several self-connected LSTM cells that are used to record the network's temporal state using three gates: input, output, and forget.

Neural networks are computer systems that are designed to mimic the organic neural networks seen in human brains. It is a network of connecting nodes inspired by the simplicity of neurons in the brain. Artificial neurons are a network of connected units, or nodes, that are loosely modeled after the neurons in the human brain. Each link, like synapses in the human brain, can transmit a signal, process it, and signal neurons attached to it [36]. In a recurrent neural network (RNN), the output from the previous step is supplied as input to the next step. Traditional neural network inputs and outputs are independent of one another [37].

An RNN recognizes that information evolves. Because it can recall earlier inputs, long-short-term memory (LSTM) is useful in time series prediction. RNNs, unlike standard neural networks, include loops to allow signals to travel both forward and backward through the network. Neural networks, an artificial intelligence-based concept, are fast gaining prominence in the building of stock models. The architecture of RNN-LSTM is shown in Figure 1.



Figure 1: The Architecture of Recurrent Neural Network (RNN-LSTM)

The gate mechanism in an LSTM cell regulates the amount of data that can be sent through the network. The forget gate is located in the cell's first section and is used to control how much of the preceding cell's hidden state can be forgotten. The input gate is then utilized to determine what fresh information in the current cell state will be sorted. Finally, the output gate is utilized to find the value that will be the current cell's output. There are a few equations that are connected to this mechanism, as shown below [38]:

$$f_{t} = \alpha \left(W_{f} h_{t-1} + U_{f} X_{t} + b_{f} \right)$$

$$\tag{12}$$

$$i_{t} = \alpha \left(W_{i} h_{t-1} + U_{i} X_{t} + b_{i} \right)$$

$$\tag{13}$$

$$\hat{C}_{t} = \tanh\left(W_{c}h_{t-1} + U_{c}X_{t} + b_{c}\right)$$
(14)

$$C_{t} = f_{t}\Theta C_{t-1} + i_{t}\Theta \hat{C}_{t}$$
⁽¹⁵⁾

$$O_t = \alpha \left(W_o h_{t-1} + U_o X_t + b_o \right) \tag{16}$$

$$H_t = O_t \Theta \tanh(C_t) \tag{17}$$

Where f_t is the forget gate value at the current cell, i_t is the input gate value, C_t is the current state, \hat{C}_t is the cell candidate value, and O_t is the output gate value. W_f , W_i , W_c , W_o , U_f , U_i , U_c , and U_o are weights of the networks, b_f , b_i , and b_c , are bias variable values, h_t is the current hidden state value, h_{t-1} is the prior hidden state value, and X_t is the new input value at the current cell. There are two activation functions (AFs) being used here, namely the sigmoid activation function (σ) and the tanh activation function. Both of them are the most frequently used nonlinear activation functions in artificial neural networks.

2.5.1 Data preprocessing

The data is divided into a training and test set with a 70:30 ratio for the training and test set, respectively. The training set was used to train the model, while the test set was used to test and evaluate the model. The data were normalized by using the feature normalization method called Min-Max normalization to enable the recurrent

neural network to converge very fast. The normalization was done to keep the data within the range of 0 to 1. The training set was further converted into an X-train and a Y-train, where the X-train is the input variable and the Y-train is the output variable. Next, we reshaped the data into a 3D array shape (sample, time steps, feature) accepted by the LSTM model. The 60-time steps were used in the LSTM-RNN. The model was then fitted to the training set, and evaluation was done on the test set.

2.6 Support Vector Regression (SVM)

Support Vector Regression is a supervised machine learning technique that finds a function that approximates mapping from an input domain to real values based on a training sample. It detects non-linearity in the data and provides a valid prediction model. Because of its ability to address nonlinear regression estimation problems, support vector regression is successful in time series forecasting. It has been a prominent topic of research due to its performance in regression tasks [5]. With the introduction of new algorithms and ideas, an increasing number of studies on the prediction of stock prices using machine learning algorithms have been conducted. As a result, stock prices can be forecasted using a support vector regression.

The main principle behind SVM for function approximation is to execute a nonlinear mapping of the data X into a high-dimensional feature space, followed by a linear regression in the feature space. Given a training set of n data points $\{X_i, Y_i\}^n I_n = 1$ with input data $X_i \in \mathbb{R}^p$, where p is the total number of data patterns and the output $Y_i \in \mathbb{R}^p$. The SVM function approximation is described in equation (18):

$$y(X) = w^{T} \varphi(X) + b \tag{18}$$

where,

 $\boldsymbol{\varphi}(\mathbf{X})$ is a high-dimensional feature space.

w and b are the coefficients which are estimated by minimizing the regularized function:

$$R(C) = \frac{1}{2} \|w\|^2 + \frac{c}{n} \sum_{i=1}^n L_i(d_i, y_i)$$
⁽¹⁹⁾

$$L_{i}(d_{i}, y_{i}) = |d_{i} - y_{i}| - \varepsilon; \quad |d_{i} - y_{i}| \ge \varepsilon$$

$$0 \quad otherwise \qquad (20)$$

To obtain the estimation of w and b, Equation (19) is transformed to the primal function given by Equation (21) by the introduction of positive slack variables ξ and ξ^* as follows:

minimize
$$R(w, \xi^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*)$$
 (21)

Subject to
$$\begin{cases} d_i - w\Phi(x_i) - b_i \le \varepsilon + \xi_i \\ w\Phi(x_i) + b_i - y_i \le \varepsilon + \xi^* \\ \xi, \xi_i^* \ge 0 \end{cases}$$

where

$$\frac{1}{2} \|w\|^2$$
 is the weights vector norm.

 d_i is the desired value.

C is referred to as regularized constant determining the tradeoff between empirical error and regularized term. ε is called a tube size of SVM.

 ξ and ξ^* are slack variables that allow the model not to overfit the data.

introducing Lagrange multipliers (a and a_i^*) and exploiting the optimality constraints, the decision function given in equation (22) takes the following explicit form:

$$y(X) = \sum_{i=1}^{n} (a_i - a_i^*) K(X_i, X_j) + b$$
(22)

where,

 $K(X_i, X_j) \text{ is the kernel function. Some examples of the kernel function are as follows:}$ Linear: $K(X_i, X_j) = X_i^T X_j$ Sigmoid: $K(X_i, X_j) = \tanh(\sqrt{X_i}^T X_j + r)$ (24)

Polynomial: $K(X_i, X_i) = (\sqrt{X_i}^T X_i + r)^4, \ \gamma > 0$ (25)

Radial basis function (RBF): $K(X_i, X_j) = \exp(-y \parallel X_i - X_j \parallel^2), y > 0$ (26)

The Lagrange multipliers (a and a_i^*) that satisfy a x $a_i^* = 0$, $a_i \ge 0$ and $a_i^* \ge 0$; I =1, minimizing the dual function of equation 17 which has the following form: R(a_i, a_i^*) = $\sum_{i=1}^n d_i(a_i - a_i^*) - \varepsilon \sum_{i=1}^n a_i + a_i^*$) $-\frac{1}{2} \sum_{i=1}^n \sum_{i=1}^n (a_i - a_i)^*$

(27)

$$a_i^*$$
) $(a_j - a_j^*)K(X_i, X_j)$

With the constraints

$$\sum_{i=1}^{n} = \sum_{i=1}^{n} a_{i}^{*}$$

0\le a_{i} \le C, i = 1,2,...,n
0\le a_{i}^{*} \le C, i = 1,2,...,n

3. Results and Discussions

This Section consists of the description of the actual series, and results of ARIMA, GARCH. LSTM-RNN and SVM.



3.1 Data Description

Figure 2: Time Series plot of Microsoft Stock prices

The Microsoft stock closing price from October 2010 to January 2023 is plotted in Figure 2. It is important to also observe that the price reached an all-time high between October 2021 and January 2022 before it started dropping. The pattern of the series suggests the presence of a trend component in the series, which means that the series is not stationary. To further verify the stationarity assumption, the Augmented Dickey-Fuller test, which tests the null hypothesis that the series is not stationary, was conducted, with the following result: Dickey-Fuller: Lag order = 0, p-value = 0.27. Since the p-value is > 0.05, we reject the null hypothesis and conclude that the Microsoft Stock closing price within the interval investigated is not stationary.



Figure 3: ACF and PACF plots of the time series data

Since significant spikes in Figure 3 are decreasing extremely slowly (ACF plot) and there is only one significant spike at lag 1 (PACF plot), we conclude that the series is not stationary. Given the outcomes of the time series plot, ADF test, and ACF/PACF plots, we consider transforming the series to make the series stationary. The differencing transformation was applied to the series and reinvestigated in Figures 4 and 5, including another ADF test.



Figure 4: First differencing of the time series plot

The first differenced series re-plotted in Figure 4 shows no patterns, and the ADF test (Dickey-Fuller: lag order = 0, p-value =0.000) is now significant at the 5% alpha level. We therefore conclude that the series is now stationary. Therefore, we begin with the first step of [2].

3.2 ARIMA Modeling of the Microsoft Stock Market Closing Prices (Oct. 2020 – Jan. 2023)

3.2.1 Model Identification:

The ACF and PACF of the first differenced series are plotted in Figure 5.

The partial autocorrelation function (PACF) (p) indicates that there are two spikes crossing the significant limit at lag 3 and lag 32. This indicates that the p = 2. The autocorrelation function (ACF), q, indicates 1 spike crossing the significant limit at lag 32. Indicating that q = 1, and since the series was differenced once, d = 1. Hence, the suggested identification of the ARIMA model is given as (2, 1, 1). Model comparison was performed with results (1,1,1), (1,1,2), and (3,1,2) to determine the model using the criteria discussed in Section 3.2.2. ARIMA (2,1,1) outperformed the rest and was chosen as the best model for the series.



Figure 5: ACF and PACF after the first differencing

3.2.2 Model Estimation

The parameters of the ARIMA (2,1,1) model were estimated by implementing equation (1) in Python programming (version 3.8) language

Table 1:	The different	ARIMA	Models
----------	---------------	-------	--------

AIC	Log-likelihood	BIC
3390.243	-1691.121	3407.604
3389.766	-1689.883	3411.468
3389.771	-1689.885	3411.473
3390.751	-1688.375	3421.133
	AIC 3390.243 3389.766 3389.771 3390.751	AICLog-likelihood3390.243-1691.1213389.766-1689.8833389.771-1689.8853390.751-1688.375

Table 2: The Estimate of the coefficients of ARIMA (2, 1, 1) Model

Terms	constant	AR_1	AR ₂	MA_1
Coefficient	0.0703	0.7580	-0.0173	-0.7916
<i>p</i> -value	0.043	0.015	0.002	0.010
Standard Error	0.161	0.311	0.055	0.308

3.2.3 Model Diagnosis

The errors of the model are tested to check if they are white noise by plotting as shown in Figure 6.



Figure 6: The residual of white noise

The residuals of ARIMA (2,1,1) were tested for white noise using different plots as shown in Figure 6. The residuals do not follow any known pattern, and the density looks normally distributed and centered at zero. These properties suggest that the errors are white noise and can be used for prediction. The predictive power of this ARIMA (2,1,1) will be compared with the ones of the GARCH model, and the best will be used to compare with the best machine learning model in this study. The GARCH model is implemented in Section 3.3.

3.3 GARCH Modeling of the Microsoft Stock Market Closing Prices (Oct. 2020 to Jan. 2023)

The GARCH model with normally distributed errors and the GARCH model with t-studentized distributed errors will be fitted to the series and compared in this section. The prediction power of the best GARCH model will be used to compare with the ARIMA (2,1,1) prediction power.

3.3.1. Garch Model with Error Normal Distribution

First, we need to identify volatility in the series by investigating the volatility plot in Figure 7.





Consider Figure 7. Some months have very high volatility, and some months have very low volatility, suggesting the stochastic model for conditional volatility. This is evidence of the presence of volatility in the series, but we further subjected the series to the chow test at a 10% level of significance to investigate the presence of structural break, which is an inferential test for volatility. The Chow test (F = 2.7148, p = 0.06819) shows enough evidence at the 10% level of significance that the series contains structure breaks. The Lagrange multiplier (LM) test for ARCH in the residuals for the Microsoft stock closing prices, which tests the null hypothesis that there is no ARCH up to order q, is conducted and presented in Table 3.

Table 3: Results of the ARCH LM test for Microsoft stock prices

Lag	LM statistic	<i>p</i> -value
ARCH Lag[3]	0.001604	0.9680
ARCH Lag[5]	0.011442	0.9994
ARCH Lag[7]	0.507811	0.9777

Given the smaller values of the LM statistic and their corresponding large p-values up to lag 7, there is no evidence to conclude that there is a presence of the ARCH effect in the return series, even at the 5% significant level. This simply means that there may not be a need to estimate the GARCH model with normally distributed errors for this series. Nonetheless, the estimation is done and its results are compared with the results of estimation with the GARCH model of t-distributed errors. For the standard GARCH model, we specify a constant to mean as ARMA (0, 0) and the distribution of the conditional error term as the normal distribution. The weighted Ljung-Box Test for the ARMA (0, 0) series is presented in Table 4.

Table 4: Results of the weighted Ljung-Box Test on Standardized residuals

Lag	Statistic	<i>p</i> -value
Lag[1]	0.5810	0.4459
Lag[2*(p+q) + (p+q)-1][2]	0.5858	0.9437
Lag[4*(p+q)+(p+q)-1][5]	1.0256	0.9852

The statistical values of each lag and associated p-value suggest that the errors are normally distributed at the various lags tested in Table 4. Therefore, the parameters of the sGARCH model are estimated and presented in Table 5.

From Table 5, the appropriate sGARCH model is sGARCH (1,1) with normally distributed errors and model information as follows: Log-Likelihood: 736.0304, AIC: -2.6716, and BIC: -2.6402. Further, we fitted the t-distributed GARCH model to the series as given in Section 3.3.2.

	Estimate	Std. Error	t value	Pr(> t)
mu	0.249396	0.003412	73.08589	0.000000
omega	0.000190	0.000054	3.48942	0.000484
Alpha1	0.963713	0.081338	11.84818	0.000000
Beta1	0.035287	0.045611	0.77365	0.439141

Table 5: Estimation Results of the sGARCH model

3.3.2 GARCH Model with Error t-Distribution

Again, the ARCH LM test for the Microsoft stock closing prices is conducted and shown in Table 6.

Table 6: Results of the ARCH LM test for Microsoft stock prices (t-distribution)

Lag	LM statistic	<i>p</i> -value
ARCH Lag[3]	10.22	0.001390
ARCH Lag[5]	27.08	0.000000
ARCH Lag[7]	39.02	0.000000

Given the high values of the LM statistic and their corresponding small *p*-values up to lag 7, there is evidence to conclude that there is the presence of an ARCH effect in the return series, even at a 1% significant level. The distribution of the errors is tested using the weighted Ljung-Box test in Table 7.

 Table 7: Weighted Ljung-Box Test on Standardized Residuals (t-distribution)

Lag	Statistic	p-value
Lag[1]	23.84	0.0000
Lag[2*(p+q) + (p+q)-1][2]	58.90	0.0000
Lag[4*(p+q)+(p+q)-1][5]	79.93	0.0000

The statistic values of each lag and associated p-value suggest that the errors are not normally distributed at the various lags tested in Table 7. Having determined the presence of the ARCH effect with the t-distributed errors, the sGARCH model is now estimated in Table 8.

Table 8: Estimation Results of the sGARCH model (t-distribution)

	Estimate	Std. error	t value	Pr(> t)
mu	0.208307	0.008563	24.3258	0.000000
omega	0.000689	0.000233	2.9551	0.003126
Alpha1	0.961241	0.153320	6.2695	0.000000
Beta1	0.037759	0.024180	0.1714	0.863907

From Table 8, the appropriate sGARCH model is sGARCH (1,1) with t-distributed errors and model information as follows: Log-Liklihood: 818.6406, AIC: -2.9658, and BIC: -2.9187. Now, based on the AIC, BIC, and log-likelihood, we compared the sGARCH with normally distributed errors and the sGARCH with t-distributed errors in Table 9.

Table 9:	GARCH Model	Comparison
----------	-------------	------------

	Normal errors	t-distributed errors
AIC	-2.6716	-2.9658
BIC	-2.6402	-2.9187
Log Likelihood	736.0304	818.6406

Given that the sGARCH model with t-distributed errors gave the smallest AIC and BIC but the largest loglikelihood values, we conclude that sGARCH (1,1) with t-distributed errors is the best GARCH family model for the Microsoft stock closing prices. This sGARCH (1,1) with t-distributed errors is now compared with the best ARIMA model (2,1,1) in Table 10.

Table 10: Comparison of ARIMA (2,1,1) and sGARCH(1,1)

	ARIMA(2,1,1)	sGARCH(1,1)
AIC	3389.766	-2.9658
BIC	3411.468	-2.9187
Log Likelihood	-1689.883	818.6406

Table 10 shows that sGARCH(1,1) of t-distribution performed better than ARIMA(2,1,1) given that it produced the smallest AIC and BIC values with higher Log likelihood and therefore GARCH model is chosen as the best traditional statistical model.

3.4. Machine Learning Models

This Section will show the analysis results of the two machine learning models starting with the LSTM-RNN and then SVM.

3.4.1 Long Short Time Memory-Recurrent Neural Network (LSTM-RNN)

In this paper, a four-layer LSTM-RNN network containing an input layer, two LSTM layers, and a dense layer was fitted to the Microsoft stock closing price data. The model summary is shown in Figure 8. 50 neurons are being used in the LSTM layers. For the loss function in the networks, we used the simple mean square error (MSE) with Adam's optimizer. The hyper-parameters, such as batch size and epochs, were tuned to get the best model as shown in Table 11. The built model will then be used to predict the closing price on the test set. However, data inversion was done to convert the predicted results into the original scaling of the data.



Figure 8: Model summary of LSTM-RNN

Figure 8 shows the model summary of the LSTM-RNN model with the input component of training data having dimensions or shapes (samples, time steps, features). The first shell indicates the input layer, which has 60 input time steps; the second and third layers are the LSTM layers, each having 50 neurons, and the last layer is the output layer.

 Table 11: Hyper-parameter Tuning

Epochs	Batch size	RMSE
50	5	6.70
100	32	8.12
1	1	7.892
200	32	6.47

From Table 11 above, different hyper-parameters were tuned to obtain the model with the best accuracy using RMSE as the evaluation metric. A model with an epoch and batch size of 200 and 32, respectively, performed best with an RMSE of 6.47 and was fitted to the series. The SVM model is now implemented in Section 3.4.2.

3.4.2 Support Vector Machine (SVM)

The data were trained using different kernels of support vector regression, with values for epsilon and the regularization parameter (c) of 0.4 and 1,000, respectively, [39]. The performances of the models were tested using the test dataset. The different kernels and their error measures are shown in Table 12. Root Mean Squared Error (RMSE) was used as an evaluation metric.

Kernel	Е	с	RMSE
Radia Basic Function (RBF)	0.4	1000	35.15
Polynomial	0.4	1000	35.85
Linear	0.4	1000	32.05

Table 12: SVM Kernels and their error measures

The best SVM model from Table 12 is the one with a linear kernel, error of 0.4, and cost of 1,000 because it produced the smallest RMSE. The best model was fitted to the series and the results compared with that of the LSTM-RNN.

Table 13: Comparison of LSTM-RNN and SVM models

	RMSE
LSTM-RNN	6.42
SVM	32.05

The RMSE of LSTM-RNN in Table 13 is substantially smaller than the one produced by the best SVM model after tuning. Therefore, the LSTM-RNN outperformed the best SVM model and will now be used to compare the sGARCH (1,1) model in Table 14.

Table 14: Comparison of LSTM-RNN and GARCH models

RMSE
12.83
6.47

Table 14 shows the model comparison of the sGARCH (1,1) with t-distributed errors and LSTM-RNN models. LSTM-RNN outperformed the GARCH model because it has the least RMSE, but GARCH (1,1) with an RMSE of 12.83 outperformed the best SVM model with an RMSE of 32.05. Further, the model confidence set test described in [40] was used for comparing the competing models and presented in Table 15.

Table 15: Confidence Set Test (95%)

	LSTM-RNN	ARIMA	SVR	GARCH
p-value	0.7234	0.6326	0.6812	0.6581

All the models returned their associated *p*-values which are all greater than 0.05 (5% alpha level) but the best model is the model with the highest *p*-value that leads to the non-rejection of the null hypothesis. In this case, the LSTM-RNN model is the best. Therefore, the Microsoft stock closing prices from 2020 to 2023 were predicted using the LSTM-RNN model in Figure 9.



Figure 9: Graph of Microsoft stock prediction using LSTM-RNN

The predicted values of the Microsoft stock closing prices from 2020–2023 using the LSTM-RNN model are plotted in Figure 9. The graph shows that the LSTM-RNN correctly predicted the closing prices within the scope of this study. Further, the forecasted Microsoft stock closing prices for the years 2023–2024 are given in Figure 10.





4. Summary

In this paper, the results are summarized as follows:

- i. The Microsoft stock closing prices were fitted with the ARIMA model and the best model obtained was ARIMA (2,1,1) because it produced the smallest AIC, BIC, and highest Log-likelihood values when compared with the rest ARIMA models.
- ii. The Microsoft stock closing prices were fitted with standard GARCH models (GARCH with normally distributed and with t-distributed errors). The two families of GARCH models were compared. The sGARCH (1,1) model with t-distributed errors was chosen as the best GARCH model for the series.

- iii. The Microsoft stock closing prices were fitted with the LSTM-RNN and the SVM models. The results of the two machine learning models were compared. The LSTM-RNN outperformed the SVM model and was chosen as the best model for the Microsoft stock closing price.
- iv. The sGARCH(1,1) with t-distributed errors model was compared with the ARIMA(2,1,1) model at first and then with the LSTM-RNN model that outperformed the SVM. The LSTM-RNN proved to be the best model, followed by the sGARCH(1,1), then the SVM, and finally the ARIMA(2,1,1) model.
- v. The future closing prices of the Microsoft stock were forecasted using the LSTM-RNN model as shown in Figure 10.

5. Conclusion

Based on the forecasted values of the Microsoft stock closing prices from 2023–2024 in Figure 10, we strongly recommend that stock buyers apply proper risk management before investing their money. The forecasted Microsoft stock closing prices ranging between the periods of March 2023, and September 2023, were characterized by strong support and resistance zones in the graph, price breaks in the resistance zone around September 2023, to make uptrend movement within a short period, then later comes back to previous support zones. When the price breaks up to the upside, what it breaks is the resistance level, not the support level. It is a support level break only when the price breaks into a downtrend. The bottom where the price touches several times and reverses back is the support while the 'roof' where the price touches several times and reverses is the Resistance. A ranging market is characterized by Support and Resistance while a trending (uptrend) market is characterized by Higher highs and Higher lows. From November 2023 to January 2024, we noticed an uptrend movement up to a price of around \$275. This clearly shows the volatile nature of the financial market. Investors are therefore advised to apply proper risk and money management before investing their money. In this study, we used historical data of Microsoft stock closing prices. Other factors such as news, political factors, natural calamities, etc. affect the financial market; such factors should be taken into consideration before investing in the stock market. The findings of this study have revalidated the claims of [12] and [13] that the LSTM-RNN model can be used to predict volatile financial series. According to [18], LSTM-RNN outperforms SVM in predicting volatile series yet again. Given that LSTM-RNN outperformed other models in this study and agrees with [20] findings, we conclude that LSTM-RNN is capable of tracking and forecasting stock market movements with high accuracy. However, this research is limited to Microsoft stock close prices for a period of 27 months, from September 2020 to December 2022, thus this conclusion is only applicable within this scope.

6. Conflict of Interest

The authors declare that there is no conflict of interest.

7. Funding

This study did not receive any form of funding from any organization.

8. Data Availability

Yes (www.yahoofinance.com)

9. Authors' contribution

All authors contributed to the study's conception and design. Data collection and analysis were performed by Desmond Chekwube Bartholomew and Emmanuel Chigozie Umeh. The methodology was written by Chrysogonus Chinagorom Nwaigwe, Desmond Chekwube Bartholomew, and Emmanuel Chigozie Umeh. Godwin Onyeka Nwafor managed the literature searches and the background of the study. The first draft of the manuscript was written by Chrysogonus Chinagorom Nwaigwe, Godwin Onyeka Nwafor, and Desmond Chekwube Bartholomew. All authors commented on previous versions of the manuscript and approved the final manuscript.

References

- W. Bao, J. Yue, & Y. Rao, Y. "A deep learning framework for financial time series using stacked autoencoders and long-short-term memory". *PLOS ONE*, 12(7), (2017). https://doi.org/10.1371/journal.pone.0180944
- [2]. G.E. Box, & G. Jenkins, "Time Series Analysis, Forecasting, and Control", (June, 1976). https://doi.org/10.1604/9780816211043.
- [3]. W. Leigh, R. Hightower, & N. Modani, "Forecasting the New York Stock Exchange composite index with past price and interest rate on condition of volume spike", *Expert Systems with Applications*, 28(1), 1–8, (2005). <u>https://doi.org/10.1016/j.eswa.2004.08.001</u>.
- [4]. Q. Wu, Y. Chen & Z. Liu, "Ensemble model of intelligent paradigms for stock market forecasting," Proceedings of the IEEE 1st International Workshop on Knowledge Discovery and Data Mining, (2008), 205-208.
- [5]. R. Samsudin, A. Shabri, & P. Saad, "A Comparison of Time Series Forecasting using Support Vector Machine and Artificial Neural Network Model." *Journal of Applied Sciences*, 10(11), 950–958, (2010). <u>https://doi.org/10.3923/jas.2010.950.958.</u>
- [6]. J. E. Jarrett, & E. Kyper, "ARIMA Modeling with Intervention to Forecast and Analyze Chinese Stock Prices." International Journal of Engineering Business Management, 3(17), (2011). <u>https://doi.org/10.5772/50938</u>
- [7]. K.C. Tseng, O. Kwon, & L.C. Tjung, "Time series and neural network forecast of daily stock prices." *Investment Management and Financial Innovations*, 9(1), 32 – 54, (2012).
- [8]. A.A. Adebiyi, A.O. Adewumi, & C.K. Ayo, "Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction." *Journal of Applied Mathematics*, 4(1), 34-43, (2014). https://doi.org/10.1155/2014/614342
- [9]. A.E. Soray, A. Zahra, & M. Ghiami, "Prediction of stock price using artificial neural network." *Bulletin de la Societe Royale des Sciences*, 85, 991-998, (2010).
- [10]. S. Kumar, & L. Kumar, "Testing of Market Efficiency in India a Study of Random Walk Hypothesis of

Indian Stock Market (BSE)." *SSRN Electronic Journal*, 35, 23 – 35, (2010). https://doi.org/10.2139/ssrn.3095177.

- [11]. S. Roy, "Testing Random Walk and Market Efficiency: A Cross-Stock Market Analysis." Foreign Trade Review, 53(4), 225–238, (2018). <u>https://doi.org/10.1177/0015732518797183.</u>
- [12]. J. Israt "Stock Price Prediction Using Recurrent Neural Networks." Graduate Faculty of the North Dakota State University of Agriculture and Applied Science, (2018)
- [13]. Q. Ma, "Comparison of ARIMA, ANN, and LSTM for Stock Price Prediction." E3S Web of Conferences, (2020), 21 – 28. 01026. https://doi.org/10.1051/e3sconf/202021801026.
- [14]. J. Shen, & M.O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system." *Journal of Big Data*, 7(1), 56 67, (2020). <u>https://doi.org/10.1186/s40537-020-00333-6.</u>
- [15]. S. Mehtab, & J. Sen, "A time series analysis-based stock price prediction using machine learning and deep learning models." *International Journal of Business Forecasting and Marketing Intelligence*, 6(4), 27 - 42, (2020). <u>https://doi.org/10.1504/ijbfmi.2020.115691.</u>
- [16]. R. Subba et al. "Stock Market Prices Prediction Using Random Forest and Extra Tree Regression." International Journal of Recent Technology and Engineering (IJRTE), 8(3), Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication - BEIESP, pp. 1224–28, (Sept. 2019). Crossref, doi:10.35940/ijrte.c4314.098319.
- [17]. M. Azizah, M.I. Irawan, & E.R.M. Putri, "Comparison of stock price prediction using geometric Brownian motion and multilayer perceptron", *Proceedings of the 5th International Symposium On Current Progress In Mathematics And Sciences* (ISCPMS2019), (2020). https://doi.org/10.1063/5.0008066.
- [18]. A. Majumder, M.M. Rahman, A.A. Biswas, M.S. Zulfiker, & S. Basak, "Stock Market Prediction: A Time Series Analysis." Smart Systems: Innovations in Computing, 389–401, 6, (2021). <u>https://doi.org/10.1007/978-981-16-2877-1_35.</u>
- [19]. S.M. Farahani, & S.H.R. Hajiagha, "Forecasting Stock Price Using Integrated Artificial Neural Network and Meta-heuristic Algorithms Compared to Time Series Models." *Soft Computing*, 25(13), 8483 8513, (2021). *Crossref*, doi:10.1007/s00500-021-05775-5.
- [20]. S. Shriram, K. Anuradha, K.P. & K.P. Uma, "Future Stock Price Prediction using Recurrent Neural Network, LSTM and Machine Learning." *International Journal of Engineering Research & Technology (Ijert) Icradl*, 9(5), 78 – 101, (2021).
- [21]. C. Dani, (2023). Better Growth Stock: Apple vs. Microsoft. https://www.fool.com/investing/2023/03/24/better-growth-stock-apple-vs-microsoft/
- [22]. P. Selva, "Complete Guide to Time Series Forecasting in Python." Retrieved from <u>https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/,</u> (2021).
- [23]. D.C. Bartholomew, U.C. Orumie, C.P. Obite, I.B. Duru, & F.C. Akanno, "Modeling the Nigerian Bonny Light Crude Oil Price: The Power of Fuzzy Time Series." *Open Journal of Modeling and Simulation*, 09(04), 370–390, (2021). https://doi.org/10.4236/ojmsi.2021.94024
- [24]. C.C. Nwaigwe, G.U. Atuzie, I.C. Obinwanne, & C.J. Ogbonna, "A time series analysis of daily

exchange rate of U.S dollar to naira from 2016-2017(recession period)." *International Journal of Engineering and Scientific Research*, Vol. 9(6), 36 – 43, (2018).

- [25]. C.P. Obite, U.I. Nwosu, & D.C. Bartholomew, "Modeling US Dollar and Nigerian Naira Exchange Rates During COVID-19 Pandemic Period: Identification of a High-performance Model for New Applications." *Journal of Mathematics and Statistics Studies*, 2(1), 40–52, (2021). https://doi.org/10.32996/jmss.2021.2.1.5
- [26]. D.C. Bartholomew, C.C. Nwaigwe, U.C. Orumie, & G.O. Nwafor, "Intervention Analysis of COVID-19 Vaccination in Nigeria: The Naive Solution Versus Interrupted Time Series." *Annals of Data Science*, 1 – 26, (2023). <u>https://doi.org/10.1007/s40745-023-00462-8</u>
- [27]. A. Marketa A. and F. Darina, "Selection of unit test based on length of the time series and value of AR(1) parameter." STATISTIKA, 96 (3), 45 – 65, (2016).
- [28]. R.F. Engle, "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation". Econometrica. 50 (4): 987–1007, (1982). doi:10.2307/1912773. JSTOR 1912773.
- [29]. L.C. Lim, & S.K. Sek, "Comparing the Performances of GARCH-type Models in Capturing the Stock Market Volatility in Malaysia." *Proceedia Economics and Finance*, 5, 478–487, (2013). https://doi.org/10.1016/s2212-5671(13)00056-7
- [30]. T. Maryam, & T.V. Ramanathan, "An overview of FIGARCH and related time series model." *Austrian Journal of Statistics*, 41(3), 175-196, (2012).
- [31]. J.A. Omorogbe, & C.E. Ucheoma, "An application of asymmetric GARCH models on the volatility of bank equity in Nigeria's stock market" *CBN Journal of Applied Statistics*, 8(1), 73-99, (2017).
- [32]. O.O. Cyprian, et al. "Using Conditional Extreme Value Theory to Estimate Value-at-Risk for Daily Currency Exchange Rates." *Journal of Mathematical Finance*, 7(4), 846 – 870, (2017). doi:10.4236/jmf.2017.74045.
- [33]. L. Wiri, & P. Sibeate, "A comparative study of Fourier series models and Seasonal ARIMA model of rainfall data in Portharcourt." *Asian Journal of Probability and Statistics*, 10(31), 36-46, (2020).
- [34]. T. Bollerslev, "Generalized Autoregressive Conditional Heteroskedasticity". Journal of Econometrics. 31 (3), 307–327, (1986). doi:10.1016/0304-4076(86)90063-1. S2CID 8797625.
- [35]. D.G. Robert & D. Hager, "Handbook of Medical Image Computing and Computer-Assisted Intervention." (2020).
- [36]. S. Haykin, "Neural Networks and Learning Machines." Bowker, (2008). doi:10.1604/9780131471399.
- [37]. N. Shajun, & M. Nagoor, "Handbook of Deep Learning in Biomedical Engineering." (2021).
- [38]. H. Seng, W. Arya, Q. Abdul, & M. Khaliq, "Comparative analysis of three recurrent neural networks." Journal of Big Data, 9(50), 56 – 74, (2022).
- [39]. C.P. Obite, D.C. Bartholomew, U.I. Nwosu, G.E. Esiaba, & L.C. Kiwu, "The Optimal Machine Learning Modeling of Brent Crude Oil Price." *Quarterly Journal of Econometrics Research*, 7(1), 31-43, (2021).
- [40]. P.R. Hansen, A. Lunde, & J.M. Nason, "The Model Confidence Set." *Econometrica*, 79(2), 453–497, (2011). <u>http://www.jstor.org/stable/41057463</u>.