UNIVERSITY OF INFORMATION TECHNOLOGY
FACULTY OF INFORMATION SYSTEMS

Stock Price Prediction Using Statistical Methods

• LÊ TRÍ KHOA-20521466
• LÂM LÊ PHÚC HUY-20521388
• NGUYỄN HỮU THIÊN-20521951

Stock Price Prediction Using Statistical Methods

1st Khoa Le Tri
Faculty of Information Systems
University of Information Technology
Ho Chi Minh City, Vietnam
20521466@gm.uit.edu.vn

2nd Huy Lam Le Phuc
Faculty of Information Systems
University of Information Technology
Ho Chi Minh City, Vietnam
20521388@gm.uit.edu.vn

3rd Thien Nguyen Huu
Faculty of Information Systems
University of Information Technology
Ho Chi Minh City, Vietnam
20521951@gm.uit.edu.vn

***Abstract*—In our recent time, there has been an increasing demand for creating a future strategy which minimizes the risks taken and maximizes the benefits gained. And the time series analysis has become an important tool of choice whether it's economics, social science, business or finance. Therefore, investors and researchers are improving different kinds of models in order to improve their accuracy of the forecasting result. Originally, there are Linear Regression (LNR), Nonlinear Regression (NLR), Autoregression (AR) and Moving Average (MA) models, which were developed to forecast the next period data. And then, ARIMA was also developed to solve the non-stationarity of data. Particularly, ARIMA model has demonstrated its outperformance in precision and accuracy when predicting the lags of time series. As the time goes on, many advancements in computer science have been archived, which also leads to the introduction of more advanced machine learning algorithms developed to analyze and forecast time series data. The research's question, which is investigated in this article, is that whether and how the newly developed deep-learning-based algorithms such as Long Short-Term Memory (LSTM) or machine learning algorithms-based such as Prophet, are superior to the original algorithms. The experienced studies conducted on several stock data set and reported in this article show that deep learning-based algorithms and machine learning-based algorithms in fact do outperform traditional-based algorithms.**

***Index Terms*—Stock forcasting, Time Series, ARIMA, LSTM, PROPHET, Linear Regression, Nonlinear Regression**

I. Introduction

Prediction of time series data such as the price movement of stock market is always a challenging and crucial task for investors as the market is a volatile environment. In recent years, market volatility has introduced some serious difficulties to economic and financial time series forecasting. Therefore, assessing the accuracy of forecasts is important for investors who want to maximize their profit earned. As the financial market grows, there is also an increasing availability of historical data that gives usability to Time Series Forecasting (TSF), which is an important area of machine learning with a sequence of time components. Fortunately, in the past several decades, there are lots of models and techniques with time series have been developed for stock prices forecasting.

The main objective of this article is to investigate which forecasting methods offer best predictions with respect to lower forecast errors and higher accuracy of forecasts. From a traditional perspective, the Box-Jenkins Method is a linear model extensively used which includes the auto-regression model (AR), the moving average model (MA), the auto-regressive moving average model (ARMA), and auto-regressive integrated moving average model (ARIMA). Particularly, the ARIMA model which is used for analysis and prediction has been considered as a very effective prediction technique, especially in social sciences because ARIMA's forecast results are derived from the values of the input variables and error terms. However, it is limited as a linear regression model, which means ARIMA model may have some deviations when facing complex nonlinear practical problems, but in terms of short-term forecasting, the linear models usually outperform the complex structural models.

From an innovative perspective, the Artificial Neural Network (ANN) model gain its favor for its ability to learn patterns from the nonlinear, nonstationary, and high-noise time series data of stock prices. The development of the ANN model is inspired by the animal's brain which can process complex information through pattern learning behavior [1]. Thus, ANN model has great ability in model stationarity, fault tolerance, and data processing [2]. Recurrent Neural Network (RNN), a model that is more complicated model than ANN, becomes popular since it makes information flow in different directions in its layers. Moreover, Long Short-Term Memory (LSTM), an improvement for Recurrent Neural Network (RNN), has performed well in time series analysis in recent years because its feedback connection makes it easier to find development trends through the back propagation of current historical prices and current prices.

In 2017 a new tool for predicting time series was open sourced by Facebook: Facebook Prophet tool [3]. This model has already been used in very interesting applications like Land-Use/Land-Cover classification [4] and others, but has not been as extensively studied as a market forecaster as neural networks.

This paper compares Linear Regression, Nonlinear Regression, ARIMA, LSTM and Prophet models with respect to their performance in reducing error rates. As a representative of traditional forecast model, Linear Regression, Nonlinear Regression and ARIMA are chosen. In an analogous way, as a representative of deep learning-based and machine learning-based algorithms, LSTM and Prophet methods are used due to theirs use in preserving and training features of given data for a longer period of time. The paper provides an in-depth guidance on data processing and training models for a set of economic and financial time series data. The key contributions

of this paper are:

– Conduct an empirical study and analysis with the goal of investigating the performance of traditional forecasting techniques, deep learning-based and machine learning-based algorithms.

– Compare the performance of Linear Regression, Nonlinear Regression, ARIME, LSTM and Prophet with respect to minimization achieved in the error rates in prediction.

The motivation for this project is to explore a field where algorithms are rapidly replaced by more efficient ones and where solutions hardly ever work well for all problems.

II. Related Works

This chapter will discuss the previous work done on stock price forecasting using regression machine learning and deep learning domain.

Dinesh Bhuriya et al [5] used linear regression, polynomial and RBF regression to predict the stock prices using 5 variables and compared the above models and concluded that linear regression is best among all other used.

ARIMA model was first developed by George Box and Gwilym in their textbook Time Series Analysis: Forecasting and Control to engage in Time Series Analysis (TSA) in 1970. They proposed to use ARMA when the sequence is stationary and use ARIMA when the sequence is non-stationary [6].

Many investors then apply the ARIMA model when analyzing their trading strategy and some researchers have found that ARIMA is effective in forecasting stock prices. Nau believes that the ARIMA model is a relatively sophisticated and accurate algorithm for time series forecasting [7].

Zumbo et al. researched that ARIMA is a good method for nonstationary time series prediction that is composed of an autoregressive and a moving average model and was successfully utilized for time series prediction in different areas which includes financial markets [8].

Bollerslev developed the GARCH model which is based on the ARCH model constructed by Robert Engle in 1982 [9]. Shengtantu applied the GARCH model in financial markets and found that it not only accurately depicts and describes whether the impact of positive and negative financial market shocks on stock prices is asymmetric, but also has obvious significance and role in the study of the symmetrical relationship between expected operating returns and expected economic risks of the financial market [10].

Assous et al. researched that the GARCH model can effectively solve the volatility problem of time series since it can accurately describe the basic characteristics of the ”thick tail” of stock prices in financial markets [11].

However, some researchers argued that the accuracy of classic TSF models is not satisfactory. Yang and Wang questioned the accuracy of the ARIMA and GARCH models because financial data have high noise and dynamic characteristics. The flexible relationship between the dependent variable and independent variable limits the further application and expansion of traditional TSF. They proposed that LSTM is a better forecasting method for stock prices [12].

By comparing prediction results of stock prices under the ANN model and the ARIMA model, Milad and Seyed researched that the ANN model has higher accuracy than ARIMA Model [13].

Sima et al. compare the ARIMA model with the LSTM model, arguing that the LSTM model can achieve higher accuracy than the ARIMA model [14].

Lu et al. applied different models to forecast stock prices and the result shows that the CNN-LSTM model has the highest accuracy for the next day stock price [15].

Kumar Jha and S. PandeAs [16] has examined few forecasting models such as- The additive model, the Autoregressive integrated moving average (ARIMA) model, FB Prophet model. From the propsoed research work, it is concluded that, FB Prophet is a better prediction model in terms of low error, better prediction, and better fitting.

W. -X. Fang, P. -C. Lan [17], in their research LSTM and Prophet are used to predict the trend of time series data, and the prediction trend is combined with the inverse neural network model (BPNN) for prediction. The empirical results show that this method can indeed achieve accurate forecasting trends and reduce errors.

As the reviewed papers above, it can be inferred that studies on forecasting stock prices are still being raised among researchers and it seems that newly proposed models are more advanced than the classic one. However, there still exists debate between different models.

III. Background

A. Regression

1) Linear Regression: Linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables. In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data which are called linear models. Linear regression model follows a very particular form, a regression model is linear when all terms in the model have a constant or a parameter multiplied by an independent variable. And buy adding the terms together, the equation is formed:

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k + \epsilon$

Where Y is a dependent variable, X are independent variables, β_i is the parameter and ϵ are errors.

2) Non Linear Regression: Nonlinear regression is a form of regression analysis in which observational data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables . In statistic, nonlinear regression can be demonstrated by the equation:

$Y = F(X, \beta) + \epsilon$

Where X is a vector of P predictors, β is a vector of k parameters, F is the known regression function. Systematic error may be present in the independent variables, but its treatment is outside the scope of regression analysis.

In general, there is no closed-form expression for the best-fitting parameters, as there is in linear regression. In contrast,

there may be many local minima of the function to be optimized and even the global minimum may produce a biased estimate.

In practice, estimated values of the parameters are used, in conjunction with the optimization algorithm, to attempt to find the global minimum of a sum of squares.

B. ARIMA

Autoregressive Integrated Moving Average Model (ARIMA) is a generalized model of Autoregressive Moving Average (ARMA) that combines Autoregressive (AR) process and Moving Average (MA) processes and builds a composite model of the time series. The reason for stationarity is that ARIMA can only be applied to stock price prediction if the time series is not white noise and not seasonal . As acronym indicates, ARIMA (p, d, q) captures the key elements of the model:

- AR: Autoregression - A regression model that uses the dependencies between an observation and the lagged observations (p).

- I: Integrated - To make the time series stationary by measuring the differences of observations at different time (d).

- MA: Moving Average. An approach that takes into accounts the dependency between observations and the residual error terms when a moving average model is used to the lagged observations (q).

A simple form of an AR model of order p, i.e., AR(p), can be written as a linear process given by:

$x_t = c + \sum_{i=1}^{p} \Phi x_{t-1} + \varepsilon_t$

Where x_t is the stationary variable, c is constant, the terms in θ_i are autocorrelation coefficients at lags 1, 2, p and t, the residuals, are the Gaussian white noise series with mean zero and variance σ^2. An MA model of order q, i.e., MA(q), can be written in the form:

$x_t = \mu + \sum_{i=0}^{q} \theta_i \epsilon_{t-i}$

Where μ is the expectation of x_t (usually assumed equal to zero), the θ_i terms are the weights applied to the current and prior values of a stochastic term in the time series, and θ_0 = 1. We assume that t is a Gaussian white noise series with mean zero and variance σ^2. We can combine these two models by adding them together and form an ARIMA model of order (p, q):

$x_t = c + \sum_{i=1}^{p} \Phi x_{t-1} + \varepsilon_t + \sum_{i=0}^{q} \theta_i \epsilon_{t-i}$

Where $\varphi_i \neq 0$, $\theta_i \neq 0$, and $\sigma^2 > 0$. The parameters p and q are called the AR and MA orders. ARIMA forecasting, also known as Box and Jenkins forecasting, can deal with non-stationary time series data because of its "integrate" step. In fact, the "integrate" component involves differencing the time series to convert a non-stationary time series into a stationary. The general form of a ARIMA model is denoted as ARIMA (p, d, q).

With seasonal time series data, it is likely that short runnon-seasonal components contribute to the model. Therefore, we need to estimate seasonal ARIMA model, which incorporates both non-seasonal and seasonal factors in a multiplicative model. The general form of a seasonal ARIMA model is denoted as ARIMA (p, d, q) × (P, D, Q)S, where p is the non-seasonal AR order, d is the non-seasonal differencing, q is the non-seasonal MA order, P is the seasonal AR order, D is the seasonal differencing, Q is the seasonal MA order, and S is the time span of repeating seasonal pattern.

The most important step in estimating seasonal ARIMA model is to identify the values of (p, d, q) and (P, D, Q). Based on the time plot of the data, if for instance, the variance grows with time, we should use variance-stabilizing transformations and differencing. Then, using autocorrelation function (ACF) to measure the amount of linear dependence between observations in a time series that are separated by a lag p, and the partial autocorrelation function (PACF) to determine how many autoregressive terms q are necessary and inverse autocorrelation function (IACF) for detecting over differencing, we can identify the preliminary values of autoregressive order p, the order of differencing d, the moving average order q and their corresponding seasonal parameters P, D and Q. The parameter d is the order of difference frequency changing from non-stationary time series to stationary time series.

C. LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) with the capability of remembering the values from earlier stages for the future usage. Before getting into LSTM, it is essential to have a look of what a neural network looks like.

1) Artificial Neural Network (ANN): A neural network consists of at least three layers: an input layer (1), hidden layers (2), and an output layer (3). The number of features of the data set determines the dimensionality or the number of nodes in the input layer. These nodes are connected through links called "synapses" to the nodes created in the hidden layer(s). The synapses links carry some weights for every node in the input layer. The weights play the role of a decision maker to decide which signal, or input, can pass through and which cannot. The weights also show the strength or extent to the hidden layer. A neural network learns by adjusting the weight for each synopsis.

In the hidden layers, the nodes apply an activation function on the weighted sum of inputs to transform the inputs to the outputs or predicted values. The output layer generates a vector of probabilities for the various outputs and selects the one with minimum error rate or cost, which minimizes the differences between expected and predicted values, which also known as the cost, using a function called SoftMax.

The assignments to the weights vector and thus the errors obtained through the network training for the first time might not be the best. In order to find the most optimal values for errors, the errors are "back propagated" into the network from the output layer towards the hidden layers and as a result the weights are adjusted. The procedure is repeated several times with the same observations and the weights are re-adjusted until there is an improvement in the predicted values and

subsequently in the cost. When the cost function is minimized, the model is trained.

2) Recurrent Neural Network (RNN): A Recurrent Neural Network (RNN) is a special case of neural network where the objective is to predict the next step in the sequence of observations with respect to the previous steps observed in the sequence. In fact, the idea behind RNNs is to make use of sequential observations and learn from the earlier stages to forecast future trends. As a result, the earlier stages data need to be remembered when guessing the next steps. In RNNs, the hidden layers act as internal storage for storing the information captured in earlier stages of reading sequential data. RNNs are called recurrent because they perform the same task for every element of the sequence, with the characteristic of utilizing information captured earlier to predict future unseen sequential data. The major challenge with a typical generic RNN is that these networks remember only a few earlier steps in the sequence and thus are not suitable to remembering longer sequences of data. This challenging problem is solved using the memory line introduced in the Long Short-Term Memory (LSTM) recurrent network.

3) Long Short-Term Memory (LSTM):: LSTM is a special type of RNNs with additional features to memorize the sequence of data. The memorization of the earlier trend of the data is possible through some gates along with a memory line incorporated in a typical LSTM.

LSTM is a special type of RNNs with additional features to memorize the sequence of data. Each LSTM is a set of cells, or system modules, where the data streams are captured and stored. The cells resemble a transport line (the upper line in each cell) that connects out of one module to another one conveying data from past and gathering them for the present one. Due to the use of some gates in each cell, data in each cell can be disposed, filtered or added for the next cells. Thus, the gates which are based on sigmoidal neural network layer enable the cells to optionally let data pass through or disposed.

Each sigmoid layer yields numbers in the range of zero and one, depicting the amount of every segment of data ought to be let through in each cell. More precisely, an estimation of zero value means that let nothing pass through, whereas an estimation of one indicates that let everything pass through. There are three types of gates involved in each LSTM with the goal of controlling the state of each cell:

- Forget Gate: outputs a number between 0 and 1, where 1 indicates completely keep this, whereas 0 indicates completely ignore this.

- Memory Gate: chooses which new data need to be stored in the cell. Firstly, a sigmoid layer, called the input door layer chooses which values will be modified. Next, a tan layer makes a vector of new candidate values that could be added to the state.

- Output Gate: decides what will be yield out of each cell. The yielded value will be based on the cell state along with the filtered and newly added data.

D. Prophet

Facebook Prophet is a model and a library that provides features both from generalized linear models (GLM) and additive models (AM), mainly extending GLM by using non-linear smoothing functions. It was specified by Taylor and Letham [18] in 2017.

The main difference between Prophet and other statistical methods is the analyst-in-the-loop approach. This approach allows the the analyst to apply their domain knowledge about the data to the forecasting algorithm, without having any knowledge of the statistical methods working from within. This approach, therefore, tries to take advantage from both the statistical forecasting and the judgmental forecasting, the latter being the forecasting methods based on human experts decisions.

The general function to define the time series is the following:

$y(t) = g(t) + s(t) + h(t) + \epsilon_t$

where g(t) represents the non-periodic changes in the value of the time series, s(t) model seasonality (which can be daily, weekly, monthly, yearly or any other), h(t) represents the effects of holidays and ϵ_t is the error term.

IV. Data preparation and Tools

A. Data

The authors extracted daily VinaCapital Vietnam Opportunity Fund Limited (VOF.L) index for the period between 12/28/2017 - 12/23/2022 from the Yahoo finance Website [19].

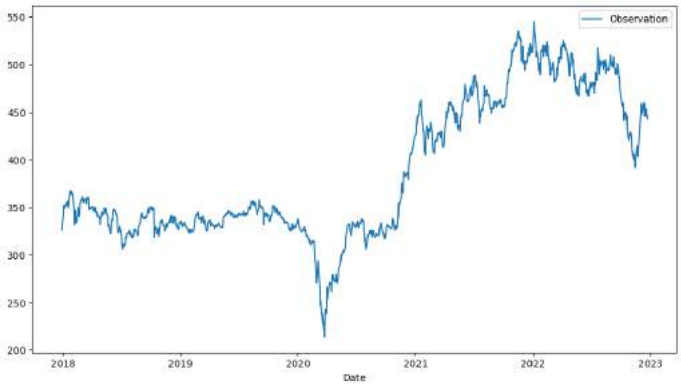

Fig. 1. Visualization for Close stock price of VOF.L

The data set consists of 1262 observations. Fig 1. shows the visualization of the data for the period while Fig 2. shows the description of the data.

B. Data preparation

Each financial time series data set features a number of variables: Open, High, Low, Close, Adjusted Close and Volume. The authors chose the "Close" variable as the only feature of financial time series to be fed into each models. The vinacapital data set was split into two subsets: training and test datasets where we divide 4 different ratio: 90/10, 80/20, 70/30, and 60/40 . Table I lists the number of time series observations for each ratio.

	Close
count	1262.000000
mean	387.623811
std	74.004481
min	214.000000
25%	331.000000
50%	348.250000
75%	459.375000
max	545.000000

Fig. 2. Data Description

TABLE I
NUMBER OF OBSERVATIONS FOR DIFFERENT TRAIN/TEST SPLIT

	Train	Test
90/10	1135	127
80/20	1009	253
70/30	883	379
60/40	757	505

C. Tools selection

The code for the project was written in Python [20]

For the implementation of ARIMA, the library pmdarima.arima [21] were used to automatically choose the best hyperparameter for the model without bias. In addition, statsmodels.tsa.arima [22] were used.

For LSTM, we used Keras [23].

For the Facebook Prophet model, the implementation in Python was used [24].

Some other libraries were used, the most important of them being: sklearn (used for preprocessing tasks), matplotlib (for plotting), numpy (for array handling) and pandas (for reading and writing the datasets).

D. Assessment Metric

This research used 3 differnt performace metric for the models: Root-Mean-Square Error (RMSE), Mean absolute error (MAE), Mean absolute percentage error (MAPE).

The Root-Mean-Square Error (RMSE) is a measure frequently used for assessing the accuracy of prediction obtained by a model. It measures the differences or residuals between actual and predicated values. The metric compares prediction errors of different models for a particular data and not between datasets. The formula for computing RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}$$

Where N is the total number of observations, x_i is the actual value; whereas, $\hat{x}_i$ is the predicated value. The main benefit of using RMSE is that it penalizes large errors. It also scales the scores in the same units as the forecast values (i.e., per month for this study).

Mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as the sum of absolute errors divided by the sample size:

$$\mathbf{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics. It usually expresses the accuracy as a ratio defined by the formula:

$$\mathbf{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Where A_t is the actual value and F_t is the forecast value. Their difference is divided by the actual value A_t. The absolute value of this ratio is summed for every forecasted point in time and divided by the number of fitted points n

V. IMPLEMENTATION

A. Regression-based models

For the implementation of Linear regression on the data, the function LinearRegression() from sklearn.linear_model package was used. Whereas, the fuction SVR() from sklearnex.svm was used to perform nonlinear regression forecasting.

B. ARIMA model

We begin our forecasting with the most common one that is ARIMA model.

Check Stationarity Of Data: we used the fuction adfuller from statsmodels.tsa.stattools to perform Dickey-Fuller test which result in p-value of 0.667. From that we can interpret the data is non stationary.

Finding P, Q, D value: we used auto_arima fuction from pmdarima library to find the best hyperparameter in Fig.3 . From the result of the test we compiple the model ARIMA(0,1,0) and fit the training set to the model

C. LSTM model

The LSTM neural network was modeled using Python 3.9. To improve the training efficiency of the model, we first normalized the data before feeding it into the LSTM model. The main parameters in the LSTM model are the activation function, dropout, batch size, epoch, neurones in the hidden layer and the optimizer. The parameters of the LSTM model are shown in table II

Figure 4 shows the performance of the loss functions for the training and test sets. The results show that the LSTM model is well-trained and does not show any over-fitting.

```
Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 1.137 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1262
Model:               SARIMAX(0, 1, 0)   Log Likelihood               -3824.648
Date:                Thu, 29 Dec 2022   AIC                           7651.296
Time:                        11:27:41   BIC                           7656.435
Sample:                             0   HQIC                          7653.227
                                 - 1262
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2        25.2348      0.597     42.300      0.000      24.066      26.404
===================================================================================
Ljung-Box (L1) (Q):                   0.90   Jarque-Bera (JB):               710.27
Prob(Q):                              0.34   Prob(JB):                         0.00
Heteroskedasticity (H):               3.12   Skew:                             0.04
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.48
===================================================================================
```

Fig. 3. Result of auto arima

TABLE II
LSTM AND PROPHET PARAMETERS AND THEIR VALUES

Method	Parameters	Values
LSTM	Layers	2
	No. of neurons	{128,64}
	Dropout	0.5
	Optimizer	Adam
	Batch size	30
	Epcohs	100
	Activation Fuction	Linear
Prophet	changepoint_prior_scale	0.5
	seasonality_prior_scale	0.01
	seasonality_mode	additive
	yearly_seasonality	TRUE
	daily_seasonality	TRUE
	daily_seasonality	FALSE

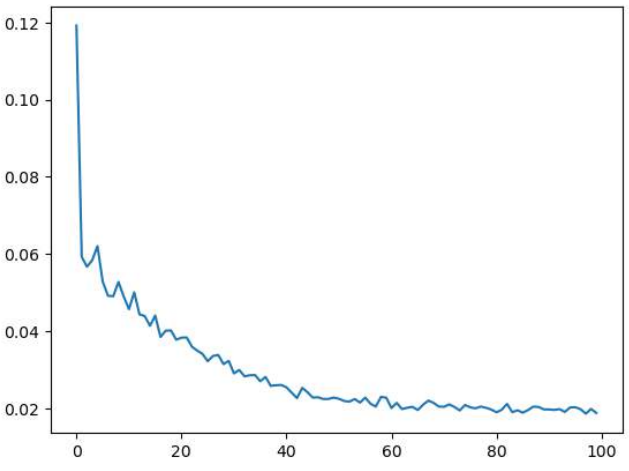

Fig. 4. Loss function for LSTM models

D. Prophet model

This research uses the training data to build the Prophet model in Python 3.9. We implemented the cross-validation fucntion by facebook Prophet package to find the best parameter for the training data set. The parameters of the Prophet model are shown in Table II.

The Fig 5. show the component of Prophet model.

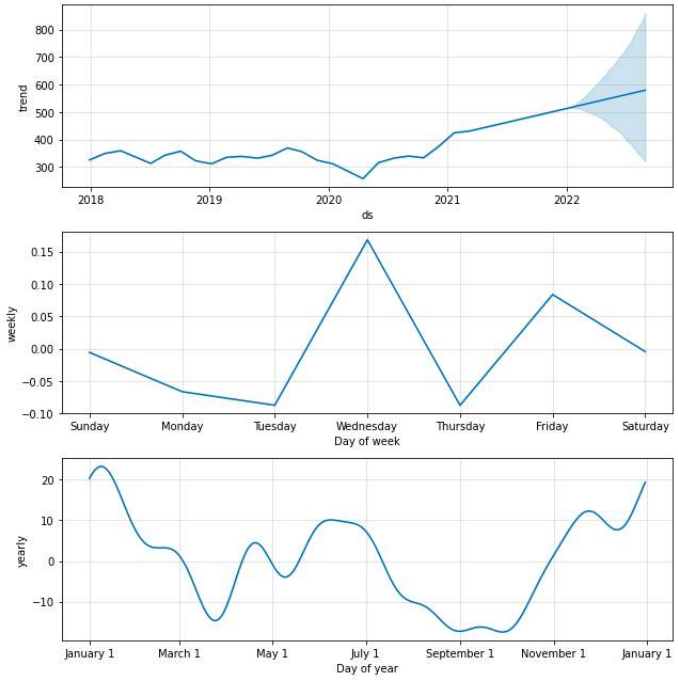

Fig. 5. Component of Prophet model

VI. RESULT

A. Performance Review

The Accuracy of 5 models and their respective train/test ratio is presented in Table III. The best performance for each model is highlighted with red color.

After analyzing, it was found that most models (Linear Regression, Nonlinear Regression and Prophet) have given the best performance with the more data fitted in. While the ARIMA model work best with 70% trainset and 20% testset. Most notably, the result clearly indicate that the LSTM model outperform other models with a high margin, especially with the train/test ratio of 80/20 with RME value of 7.06

B. 20% Test Forecasting

From the result of performance review, it was found that that most of the models have given the best performance for 20% and 10% test dataset. Hence for the next step which is forecasting, therefore, performed forecasting analysis on 20% and 10% test dataset.

Fig 6. was shown forecasting for 20% test dataset. LSTM, Prophet and non linear regression had shown some trend and patterns for forecasting whereas ARIMA and Linear regression model had show straight line.

TABLE III
ACCURACY VALUES FOR ALL MODELS

Model	Train/Test ratio	RMSE	MAE	MAPE
Linear Regression	90/10	43.61	34.57162	0.07882
	80/20	50.88	45.95882	0.09439
	70/30	93.43	87.50885	0.17803
	60/40	159.3	155.54917	0.32645
Nonlinear Regression	90/10	27.56	22.40389	0.04958
	80/20	48.42	44.45939	0.09035
	70/30	82.62	76.62405	0.15650
	60/40	132.57	128.38354	0.26872
Arima	90/10	36.84	29.55512	0.06708
	80/20	47.47	35.93874	0.07951
	70/30	33.68	28.39182	0.05857
	60/40	69.73	61.78317	0.12683
LSTM	90/10	9.61	7.72610	0.01641
	80/20	7.06	5.28713	0.01100
	70/30	9.35	3.56217	0.01041
	60/40	8.22	6.61932	0.01393
Prophet	90/10	39.61	34.51245	0.07647
	80/20	77.38	64.35207	0.14023
	70/30	116.07	92.72411	0.19889
	60/40	93.66	63.75704	0.13872

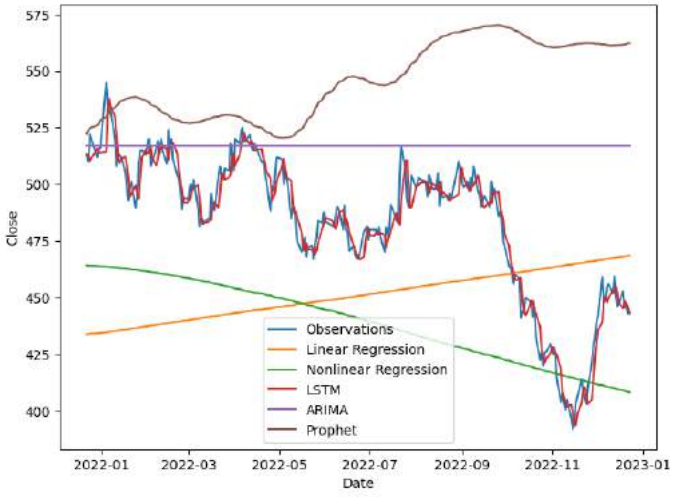

Fig. 6. VOL.F close stock price forecasting for 20% test dataset

C. 10% Test Forecasting

Fig 7. was shown forecasting for 10% test dataset. LSTM and Prophet had shown some trend and patterns for forecasting whereas ARIMA, Linear regression and Nonlinear regression model had show straight line.

VII. Conclusion

With the recent advancement on developing sophisticated machine learning-based techniques and in particular deep learning algorithms, these techniques are gaining popularity among researchers across divers disciplines. The major question is then how accurate and powerful these newly introduced approaches are when compared with traditional methods. This paper compares the accuracy of five different models, as representative techniques when forecasting time series data. The models were implemented and applied on a set of stock data on several train/test split ratio and the results indicated that LSTM with 80% train data set with was superior.

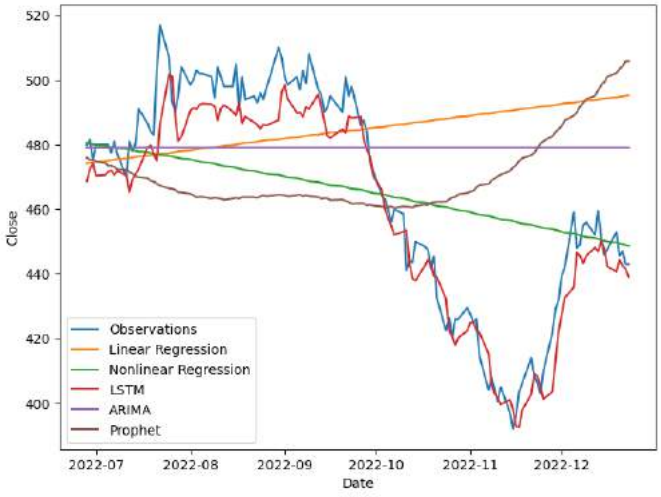

Fig. 7. VOL.F close stock price forecasting for 10% test dataset

References

[1] M. S. Mhatre, F. Siddiqui, M. Dongre, Paramjit Thakur, A Review Paper on Artificial Neural Network: A Prediction Technique, International Journal of Scientific & Engineering Research, vol. 16, no. 12, 2015, pp. 161-163.

[2] B. W. Wanjawa, L. Muchemi, ANN Model to Predict Stock Prices at Stock Exchange Markets, 2015, pp. 3-20.

[3] Facebook Prophet https://facebook.github.io/prophet/ (December 2022)

[4] Yan, Jining & Wang, Lizhe & Song, Weijing & Chen, Yunliang & Chen, Xiaodao & Deng, Ze. A time-series classification approach based on change detection for rapid land cover mapping. ISPRS Journal of Photogrammetry and Remote Sensing. 249-262. (2019). https://www.sciencedirect.com/science/article/abs/pii/S0924271619302400

[5] Dinesh Bhuriya, Ashish Sharma, Upendra Singh. "Stock Market Prediction using Linear Regression," International Conference on Electronics, Communication and Aerospace Technology ICECA 2017. DOI: 10.1109/ICECA.2017.8212716, https://ieeexplore.ieee.org/document/8212716/

[6] G. E. P. Box, G. M. Jenkins, Time series analysis: forecasting and control. Journal of Time, vol. 31, no. 3, 2010, pp. 190-200.

[7] R. Nau, Mathematical structure of ARIMA models, 2014, pp.1–8.

[8] B. D. Zumbo, E. Kroc, Heteroskedasticity in Multiple Regression Analysis: What it is, How to Detect it and How to Solve it with Applications in R and SPSS [Heteroscedasticidad en análisis de regresión múltiple: qué es, cómo detectarlo y cómo resolverlo con aplicaciones en R y], The Journal of Modern Applied Statistical Methods, vol. 17, 2019.

[9] T. Bollerslev, Generalized Autoregressive Conditional Heteroskedasticity, Journal of Econometrics, 1986, pp. 307–328.

[10] S. T. T. Wang, Research on the Volatility of BYD Stocks Price Based on GARCH Family Model, 1994, pp. 280–284.

[11] H. F. Assous, N. Al-Rousan, D. Al-Najjar, H. AlNajjar, Can international market indices estimate TASI's movements? The ARIMA model. Journal of Open Innovation: Technology, Market, and Complexity, 2020, pp. 1–17. DOI: https://doi.org/10.3390/joitmc6020027

[12] Q. Yang, C. Wang, A study on forecast of global stock indices based on deep LSTM neural network, Statistical Research, 2019, vol. 36, no. 6, pp. 65–77.

[13] S. F. Milad, S. H. R. Hajiagha, Forecasting Stock Price Using Integrated Artificial Neural Network and Metaheuristic Algorithms Compared to Time Series Models, Soft Computing, vol. 25, no. 13, 2021, pp. 483–513.

[14] S. Siami-Namini, N. Tavakoli, A.S. Namin, A Comparison of ARIMA and LSTM in Forecasting Time Series, 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 1394–1401.
[15] W.J. Lu, J. Z. Li, Y. F. Li, A. J. Sun, J. Y. Wang, A CNN-LSTM-Based Model to Forecast Stock Prices, Introduction, 2020, pp. 11.
[16] B. Kumar Jha and S. Pande, "Time Series Forecasting Model for Supermarket Sales using FB-Prophet," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 547-554, doi: 10.1109/ICCMC51019.2021.9418033.
[17] W. -X. Fang, P. -C. Lan, W. -R. Lin, H. -C. Chang, H. -Y. Chang and Y. -H. Wang, "Combine Facebook Prophet and LSTM with BPNN Forecasting financial markets: the Morgan Taiwan Index," 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2019, pp. 1-2, doi: 10.1109/ISPACS48206.2019.8986377.
[18] Taylor SJ, Letham B. Forecasting at scale. PeerJ Preprints. (2017). 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2
[19] Yahoo Finance website: https://finance.yahoo.com/ (December 2022)
[20] Python official site. https://www.python.org/ (December, 2022)
[21] pmdarima official site. http://alkaline-ml.com/pmdarima/ (December, 2022)
[22] statsmodel official site. https://www.statsmodels.org/dev/about.html (December, 2022)
[23] Keras official site. https://keras.io/ (December, 2022)
[24] https://facebook.github.io/prophet/ (December, 2022)