

Time series forecasting of stock market using ARIMA, LSTM and FB prophet

Asha Sunki¹, C SatyaKumar^{2*}, G. Surya Narayana², Vinith Koppera¹ and Manish Hakeem¹

¹Department of Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad

²Vardhaman College of Engineering, Hyderabad

Abstract. Considering the stock market's dynamic and turbulent character, predicting the future stocks of a company is a difficult endeavor. The goal of this study is to analyze the performance of three widely used forecasting methods: ARIMA, LSTM, and FBProphet. ARIMA is a time series data statistical model that captures linear relationships and stationarity. Recurrent neural networks, such as LSTM, are able to recognize nonlinear patterns and long-term dependencies. FB Prophet is a Facebook-developed time series forecasting library that use an additive regression model to account for trend, seasonality, and holiday impacts. The results show that each strategy has advantages and disadvantages in projecting stock market values. When the underlying data is steady and linear, ARIMA works well. In contrast, LSTM excels in capturing nonlinear and complicated relationships. FB Prophet performs admirably when dealing with trend and seasonality patterns. This study examines the performance of ARIMA, LSTM, and FB Prophet in stock market forecasting, allowing practitioners to choose the best approach depending on the peculiarities of their data and forecasting objectives. Further study might look at ensemble methods or hybrid approaches that combine the capabilities of these techniques to increase stock market forecast accuracy.

Keywords: ARIMA, LSTM, FB Prophet, Stock market, forecasting, MSE, RMSE, model, prediction.

1 Introduction

Stock market forecasting is the practice of predicting the future direction and behavior of stock prices, indexes, or specific securities traded in financial markets using various analytical approaches and models. It entails analyzing past price and volume data, as well as other important aspects including economic indicators, corporate financials, news events,

* Corresponding author: csatyakumar@gmail.com

and market sentiment, in order to create educated forecasts regarding future market movements. Because financial markets are dynamic and unpredictable, stock market forecasting is a complicated and difficult endeavor. However, it is a necessary activity for investors, traders, financial institutions, and analysts who want to make educated judgments and successfully manage risks. In earlier work, several regression models and RNN had been used. In present work, we have implemented ARIMA, LSTM and FB Prophet algorithms in order to forecast the future trends.

In the stock market, time series forecasting is critical for investment decision-making, portfolio management, and risk assessment. Accurate stock price and market trend forecasting is required for traders, investors, and financial institutions to make educated decisions and maximize their gains. Forecasting stock market values, on the other hand, is a difficult undertaking due to the complex and dynamic structure of financial data. A variety of variables impact the stock market, including economic statistics, corporate performance, geopolitical events, investor mood, and market psychology. These variables contribute to stock price volatility and nonlinearity, making it challenging to effectively capture and predict the underlying trends. As a result, new forecasting approaches are necessary to deal with the distinct properties of stock market data. Several forecasting approaches for time series analysis in the stock market have gained prominence in recent years. The FB Prophet forecasting library, ARIMA and LSTM neural networks are the subject of this study.

The ARIMA model, as a traditional statistical method, accounts for both autocorrelation and moving average elements within time series data. They've been widely utilized in finance and economics to forecast stock market prices using past price and volume data.[7]. In capturing long-term relationships and nonlinearity in time series data, LSTM, a form of RNN, has demonstrated promising results. Because of its capacity to analyze sequential information, it is well-suited for modeling stock market prices, which display complicated patterns and changing trends [6].

FB Prophet, created by Facebook's Data Science team, provides a versatile and user-friendly method to forecast time series. It uses an additive regression model that integrates trend, seasonality, and holiday effects, making it suited for modeling stock market data with various temporal patterns. This study aims to assess and compare the predictive accuracy of ARIMA, LSTM, and FB Prophet Models specifically for predicting stock market values. Historical stock market data will be used to train and evaluate the models, judging their accuracy and capacity to reflect the stock market's complex dynamics. Investors, financial analysts, and researchers can get insights into the best approaches for anticipating stock market movements by studying the strengths and limits of different forecasting methodologies. This knowledge may aid in the improvement of decision-making processes, the optimization of investment strategies, and the reduction of risks connected with stock market investments.

To summarize, accurate and consistent time series forecasting of stock market values is a difficult undertaking owing to the complex and volatile nature of financial data. This work intends to contribute to a better understanding of the strengths and shortcomings of ARIMA, LSTM, and FBProphet in predicting stock market movements by examining their performance.

2 Literature Survey

[1] Tiwary and Mishra studied Tesla (TSLA) and NIO in their case study of using ARIMA models to forecast stock prices over time. Using past data, the study attempted to forecast future stock prices. Traditional statistical methods such as ARIMA models were used because of their capacity to extract the moving average and autocorrelation components from time series data. The methodology, data gathering from Tesla and NIO, model training, and evaluation using metrics like MAE or RMSE are probably covered in the paper. The study's conclusions and ramifications are probably talked about in relation to how well ARIMA models predict Tesla and NIO stock values.

[2] This study looks at how well time-series models anticipate earnings for six different companies using historical data from January 2010, until December 2020. The study demonstrates that the ARIMA family of models outperforms simple time-series models in terms of mean percentage errors, AIC, and average ranks. Investors should use the selected ARIMA model to help them set expectations. Financial Time Series Data (TSD) mining provides investors, banks, and insurance providers with important information to help them manage their money more profitably.

[3] This study introduces a novel approach called the TSRM, which leverages both relationship and time data to enhance stock market analysis. Using a K-means model, it automatically classifies stocks based on transaction data and determines the linkages between stocks. Using a K-means model, the TSRM automatically classifies stocks based on transaction data and establishes stock linkages. The integration of LSTM and relationship data retrieved using a GCN to forecast a stock price is one of the text's most crucial details. In comparison to the baseline, the results demonstrated that the cumulative returns increased by 44% and 41%, respectively, while maximum drawdown decreased by 4.9% and 6.6%. Although a lot of study has been done on stock time series, not much has been done on how stock correlation affects stock price forecasting.

[4] The paper delves into the application of machine learning techniques for forecasting the stock market. It acknowledges stocks as a popular financial instrument and emphasizes the need for accurate predictions in the dynamic stock market landscape. Leveraging advances in trading technology and the growing interest of the general public in stock markets, the study focuses on utilizing machine learning techniques. Specifically, 3MMA, ES, and Time Series Forecasting are highlighted. Yahoo Finance data for Amazon (AMZN), Apple (AAPL), and Google (GOOGLE) stocks are used. The paper successfully forecasts stock market trends for the following month and evaluates the accuracy of predictions.

[5] Kim and Han conducted a study on ensemble learning, which is used to predict stock values by combining ANN and Decision Trees. Using data from the Taiwanese stock market, they created a dataset that included macroeconomic, technical, and fundamental

indexes. A 77% F-score was found in the performance evaluation of their Decision Tree + ANN ensemble model, which was trained using data from the Taiwan Stock Exchange. In comparison, individual algorithms achieved F-scores of up to 67%. This research highlights the effectiveness of ensemble learning approaches in enhancing predictive accuracy for stock price forecasting.

[6] Min and Lee conducted research on bankruptcy prediction using machine learning techniques, including Multiple discriminant analysis, logistic regression analysis, SVM, and three-layer fully connected back-propagation neural networks. Their study revealed that SVMoutperformed other methods. Lee focused on predicting company credit ratings employing SVM and a variety of financial metrics and ratios, including net income to shareholders' equity and the interest coverage ratio. They achieved an accuracy rate of approximately 60%. Additionally, credit rating prediction was explored that Tsai and Wang used neural networks to achieve accuracy rates for the US and Taiwan markets that ranged from 75% to 80%.

[7] This study introduces a Posteriori HMM method for prediction of next-day stock prices based on historical data, considering fractional changes and intra-day high/low values. The continuous HMM is trained using these features and used to make predictions. Performance assessment in comparison to current techniques, including HMMs and ANN, is conducted using MAPE. The approach is tested across various stocks, showcasing its potential for improved stock market prediction accuracy.

3 Methodologies Used for Analysis

Machine learning and deep learning have made major advances in time series analysis, and numerous outstanding models have emerged as excellent forecasting tools. We recommend using ARIMA, LSTM, and FB Prophet models in this project because of their expertise and efficacy in time series forecasting.

3.1 A moving average that is both autoregressive and integrated: ARIMA

An established time series analysis model for predicting and examining time-dependent data is ARIMA. It is composed of 3 main parts: auto regression (AR), moving average (MA), and differencing (I for integrated). ARIMA models may capture and model many patterns, trends, and relationships seen in time series data.[8].

The mathematical formulas for each component are as follows:

1. The autoregressive component, also known as AR, describes the linear relationship between the present observation and a set number of delayed data points. Order p is used to represent it.

The autoregressive component's formula is as follows: $AR(p) \Rightarrow y(t) = c + \sum(\varphi_i * y(t-i)) + \varepsilon(t)$

Where:

- $y(t) \rightarrow$ time series' current value at time. t. c is a fixed term.
- ϕ_i is a representation of the autoregressive coefficients for each lag observation.
- white noise error term $\rightarrow \epsilon(t)$.

2. The time series is made stationary by removing trends and seasonality using the differencing (I) component. The order d is used to symbolize it.

The differencing formula is as follows: $I(d) \Rightarrow y'(t) = y(t) - y(t-d)$

Where:

- $y'(t) \rightarrow$ differenced series at time t.
- $y(t) \rightarrow$ original time series at time t.
- $y(t-d) \rightarrow$ lagged value of the time series.

3. Component for Moving Average (MA): The MA component represents the dependence of the present observation on a residual error term that is derived from earlier errors. The order q is used to symbolize it.

The moving average's component formula is as follows:

$$MA(q): y(t) = \mu + \sum (\theta_i * \epsilon(t-i))$$

Where:

- $y(t)$ indicates the time series' current value at time t.
- θ_i is a representation of the moving average coefficients for each lag in the error term.
- The expression $(t-i)$ represents the lag terms in the mistake.

This is the generic formula for the ARIMA (p, d, q) model, which combines the autoregressive, differencing, and moving average components:

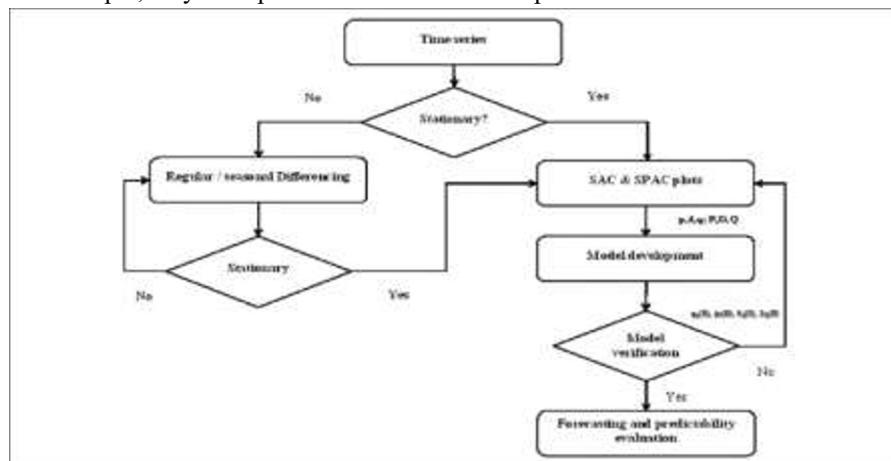
$$ARIMA(p, d, q): y'(t) = c + \sum (\phi_i * y'(t-i)) + \sum (\theta_i * \epsilon(t-i)) + \epsilon(t)$$

Where:

- $y'(t)$ represents the differenced series at time t.
- Autoregressive coefficient $\rightarrow \phi_i$.
- Moving average coefficient $\rightarrow \theta_i$.
- Error term for white noise $\rightarrow \epsilon(t)$.

The parameters p, d, and q must be calculated using an examination of the differenced series' autocorrelation and partial autocorrelation functions.

The exact formulas and implementation may differ somewhat based on the software or programming language used for ARIMA modeling. Seasonal ARIMA (SARIMA) models, for example, may incorporate extra seasonal components and formulae.



3.2 LSTM (Extended Short-term Memory):

RNN with LSTM are especially adept at seeing long-term trends and linkages in time series data. It is commonly used for sequence modeling, natural language processing, and time series forecasting. LSTM, as opposed to typical RNNs, avoids the vanishing gradient problem by incorporating specialized memory cells that may selectively store or forget information over lengthy periods.

The LSTM architecture is made up of LSTM cells, which serve as the network's building blocks. Memorization cells, forget gates, output gates, and input gates are all included in each LSTM cell.

Input Gate:

- The amount of new data that is kept in the memory cell is managed by the input gate.
- It processes the inputs using a sigmoid activation function that returns values ranging from 0 to 1 using the current input and the previously hidden state as inputs.

Forget Gate:

- The information in the memory cell that should be erased is decided by the forget gate.
- A sigmoid activation function is used to the inputs, which consist of the current input and the previously concealed state.

Memory Cell:

- The memory cell saves data over time by mixing the input from the forget gate and the input from the input gate.
- To generate a new candidate cell state, it employs a hyperbolic tangent (tanh) activation function.

Output Gate:

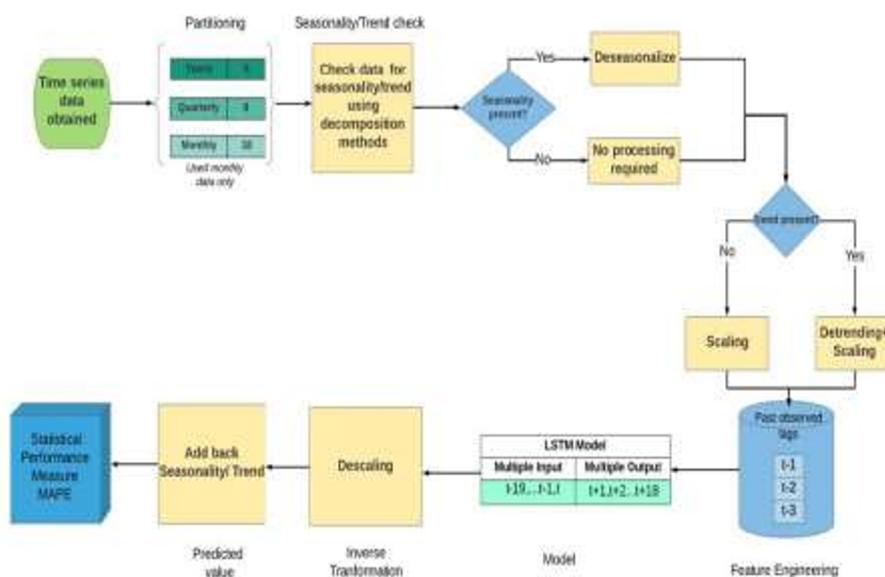
- The data that the LSTM cell will output is determined by the output gate.
- It applies a sigmoid activation function to the inputs, Which are the current input and the prior hidden state.

The formulas involved in LSTM modeling are related to the internal operations of the LSTM cells. The key equations are as follows:

1. LSTM Cell Update:
 - o $\sigma(W_f \cdot [h_{(t-1)}, x_t] + b_f) = f_t$
 - o $\sigma(W_i \cdot [h_{(t-1)}, x_t] + b_i) = i_t$
 - o $\sigma(W_o \cdot [h_{(t-1)}, x_t] + b_o) = o_t$
 - o $f_t * c_{(t-1)} + i_t * \tanh(W_c \cdot [h_{(t-1)}, x_t] + b_c) = c_t$
 - o $o_t * \tanh(c_t) = h_t$

2. Where:

- o The input, output and forget gates are denoted by the letters i_t , o_t and f_t .
- o c_t → cell state or memory.
- o h_t → hidden state or output.
- o At step t, the input is x_t .
- o The preceding hidden state → $h_{(t-1)}$ and cell state → $c_{(t-1)}$.
- o The sigmoid activation function has the value σ .
- o W → weight matrices and b → bias terms.



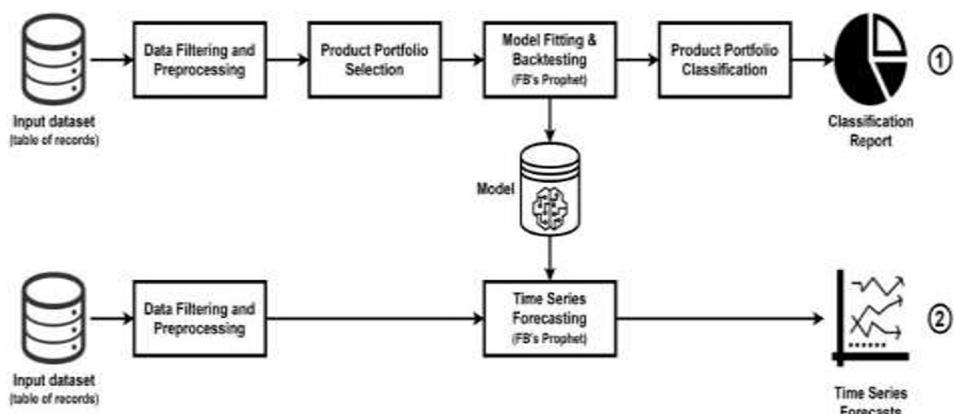
3.3 FB Prophet (Facebook Prophet):

FB Prophet is a prominent open-source library for time series forecasting created by Facebook. It is intended to give a simple yet effective method for modeling and predicting time series data, with a focus on commercial applications. To capture trend and seasonality in data, FB Prophet employs a combination of statistical models and machine learning approaches.

FB Prophet's particular mathematical facts and formulae may not be publicly disclosed or accessible.

Nonetheless, it's worth mentioning that FB Prophet's mathematical foundations are based on well-established statistical approaches. In order to predict and model time series data, the library uses a combination of linear and nonlinear regressions, Fourier series expansions, and Bayesian inference. While the specific formulae used by FB Prophet are not released, the library offers a high-level API that allows users to specify the Time series data components including holidays, seasonality, and trend. FB Prophet then uses a collection of mathematical methods and optimization approaches to estimate the model's parameters and create forecasts.

FB Prophet's major goal is to provide a user-friendly interface for time series forecasting, abstracting away difficult mathematical intricacies while producing accurate and interpretable forecasts. As a consequence, consumers may utilize FB Prophet without having to comprehend the underlying mathematics and equations.



4 Implementation and Tools:

4.1 Dataset:

Netflix dataset which comprises 4882 rows and 7 columns. Our Goal is to estimate future trends of Netflix data based on the historical analysis of past data.

4.2 Frameworks used:

TensorFlow and Keras are two well-known deep learning frameworks for building and training neural network models. Keras is a high-level API that runs on top of TensorFlow (and other deep learning frameworks), while TensorFlow is an open-source library developed by Google.

4.3 Making the data stationary:

Making the data stationary is an important preprocessing phase of the examination of time series since it is beneficial to stabilize the statistical features of the data and enhance the forecasting model performance. Here are some popular methods for making data stationary:

4.3.1 Differencing:

- a. The difference between successive observations is one method.
- b. By subtracting the past value from the present value, the trend component is removed, and the data becomes more stationary.
- c. For seasonal data, the difference between observations with the same seasonality lag may be calculated.

Different approaches may be better suited to different datasets. Furthermore, the stationarity of the modified data must be validated by applying statistical tests like the Kwiatkowski Phillips Schmidt Shin (KPSS) test or Augmented Dickey-Fuller (ADF) test. By removing or reducing the impacts of trends and seasonality, you make it easier for time series models in order to identify the underlying relationships and patterns. As a result, predictions and analysis can become more precise.

4.4 Lagging the data:

The process of lagging time series data entails producing lagged variables by moving the observations backward in time. In time series analysis, this approach is widely used to include past values of the target variable as features in a forecast model. Lagged variables can capture data dependencies and trends, allowing the model to include data from past time points.

ACF and PACF plots are commonly used in time series analysis to examine the autocorrelation pattern of time series data. These plots help identify the appropriate sequence for the AR and MA components by displaying the relationship between the data at various lags.

4.4.1 ACF (Autocorrelation Function) Plot:

- The association between a time series and its lag values at different delays is evaluated using the ACF plot.

4.4.2 PACF (Partial Autocorrelation Function) Plot:

- After eliminating the impact of intermediate delays, the relationship between a time series' lag values and correlation is computed using the PACF graphic.
- If the autocorrelation values in an ACF plot decay slowly and stay significant for numerous lags, it indicates a non-stationary time series that may require differencing.
- Significant partial autocorrelation at various lags in a PACF graphic suggests a probable AR factor in the model of time series.

- If the PACF and ACF plots show a smooth decline with no notable spikes after a particular lag, it indicates a stationary time series with no strong or partial autocorrelation patterns.

Metrics used:

While assessing a time series model, the most popular statistic is mean squared error. The model with the least RMSE is regarded as the model with the finest fit.

The following formula can be used to get the RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (y_{\text{pred}} - y_{\text{true}})^2}$$

where:

- RMSE → Root mean squared error
- N → no. of points.
- Σ → summation.
- The anticipated value is y_{pred} .
- y_{true} → true value(observed).

5 Experimental Results and Analysis

This result helps us to analyze the future trends shown by various models by analyzing the historical data. It also helps us to identify the most suitable model for the given data.

5.1 Overall stocks of the data:

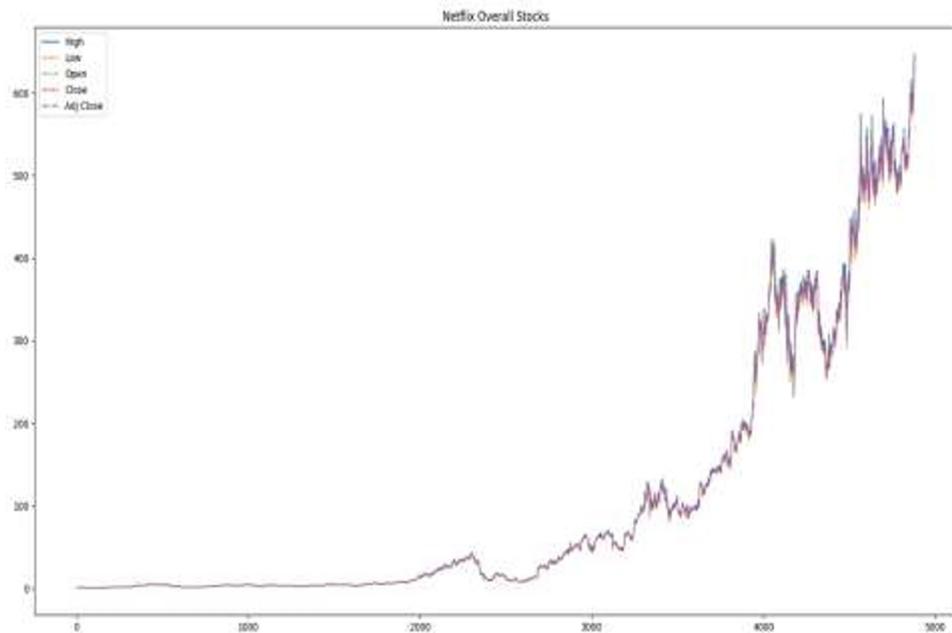


Fig. 1. Overall stocks

5.2 Closing Price:

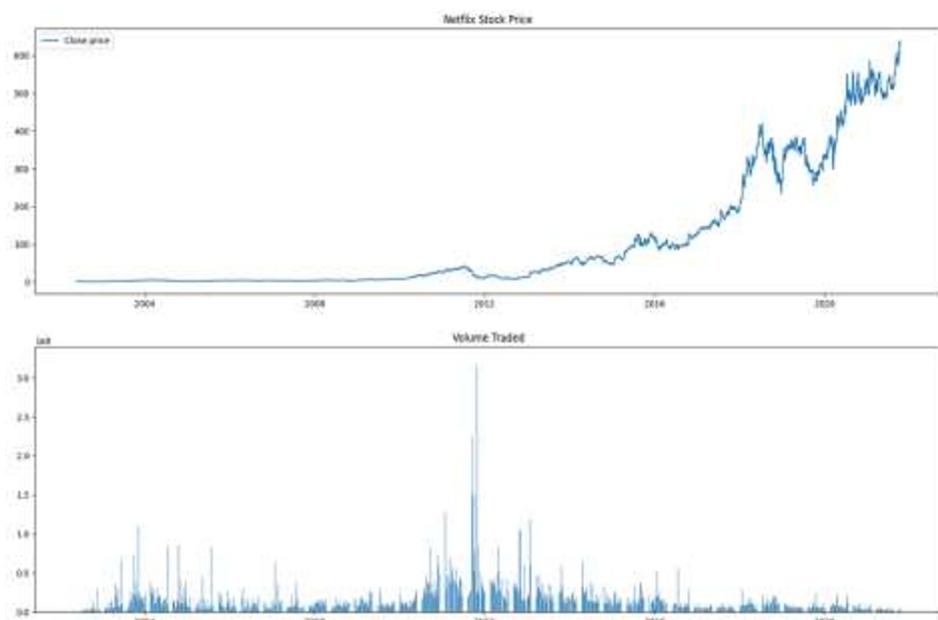
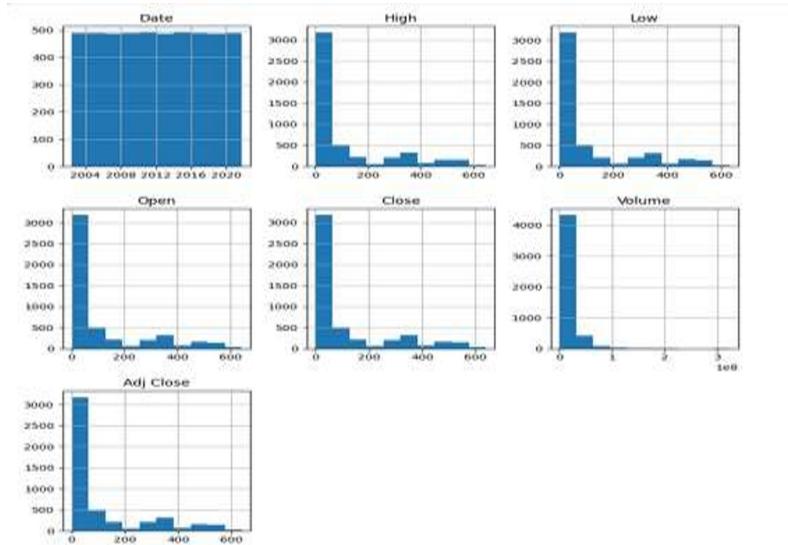


Fig. 2. Closing Price from 2004 to 2020

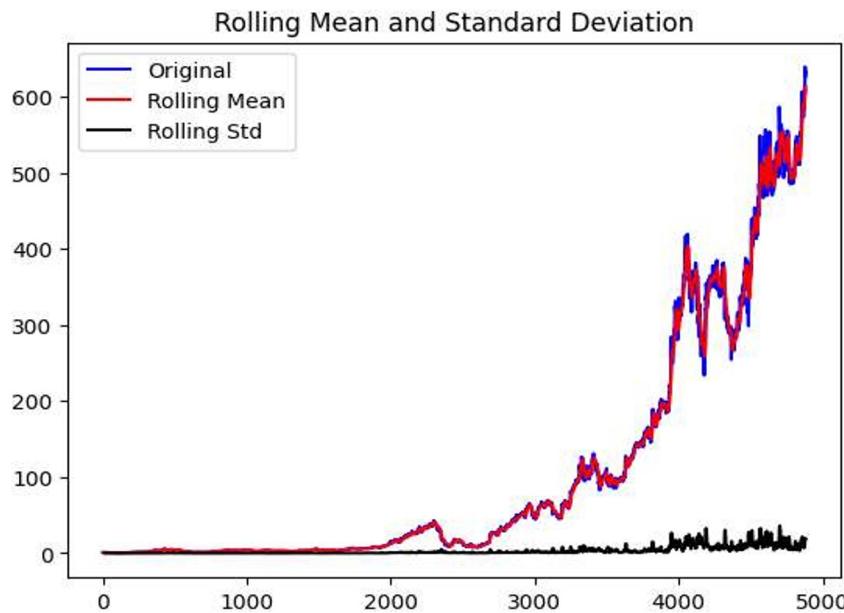
5.3 Detailed Columns:

It represents the histogram of each detailed column in the dataset.



5.4 Dicky Fuller Test Results:

Generally, this test is performed to identify whether the data is stationary or not. Because, ARIMA model performs well on stationary data. It transforms the non-stationary data into stationary data inorder to perform analysis.

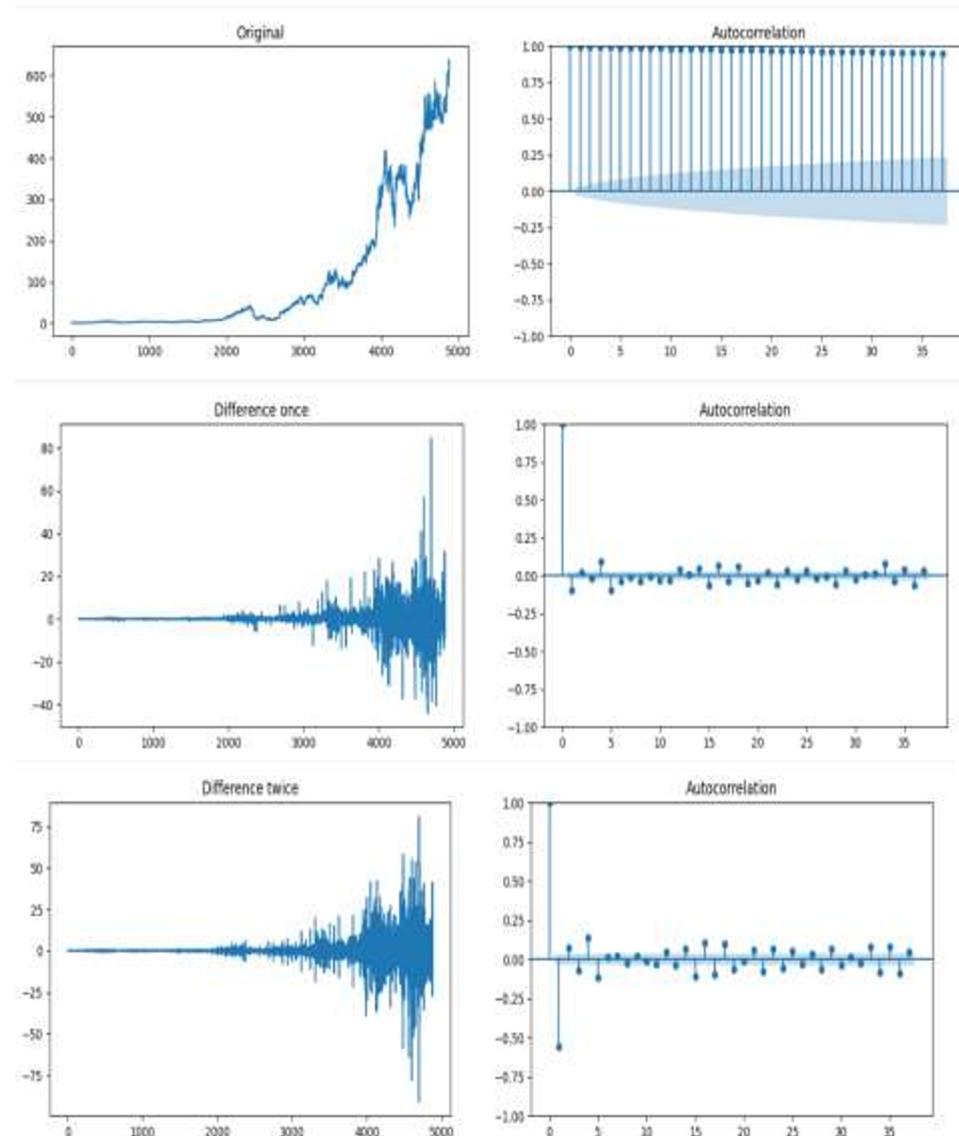


Results of dickey fuller test

```
Test Statistics      2.948081
p-value            1.000000
No. of lags used   30.000000
Number of observations used 4850.000000
critical value (1%) -3.431699
critical value (5%) -2.862136
critical value (10%) -2.567087
dtype: float64
```

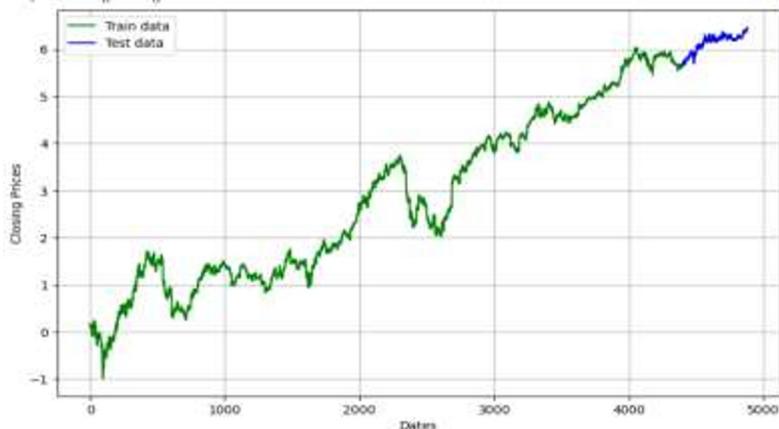
5.5 PACF and ACF Plots:

PACF (Partial Autocorrelation Function) and ACF (Autocorrelation Function) plots are useful for analyzing and determining the order of moving average (MA) and autoregressive (AR) components in an ARIMA model. The ACF and PACF charts can be used to establish the right values for the MA and AR variables in an ARIMA model. The ACF plot aids in the identification of MA terms by noticing significant spikes at certain lags, whereas the PACF plot aids in the identification of AR terms by observing significant spikes at specific lags.



5.6 Data splitting:

In this section, we partition the dataset into two parts: testing and training. The training dataset is used to train the model and define its parameters, while the testing dataset is used to assess the model's performance and prediction capacity.



ARIMA RESULTS

Dep. Variable:	Close	No. Observations:	4389			
Model:	ARIMA(1, 1, 2)	Log Likelihood	8258.444			
Date:	Thu, 16 Mar 2023	AIC	-16508.888			
Time:	17:01:48	BIC	-16483.341			
Sample:	0 - 4389	HQIC	-16499.875			
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-0.4153	0.463	-0.897	0.370	-1.323	0.492
ma.L1	0.4559	0.465	0.980	0.327	-0.456	1.368
ma.L2	-0.0010	0.027	-0.037	0.971	-0.054	0.052
sigma2	0.0014	8.25e-06	164.527	0.000	0.001	0.001
Ljung-Box (L1) (Q):	0.01	Jarque-Bera (JB):	101167.25			
Prob(Q):	0.93	Prob(JB):	0.00			
Heteroskedasticity (H):	0.38	Skew:	-0.89			
Prob(H) (two-sided):	0.00	Kurtosis:	26.46			

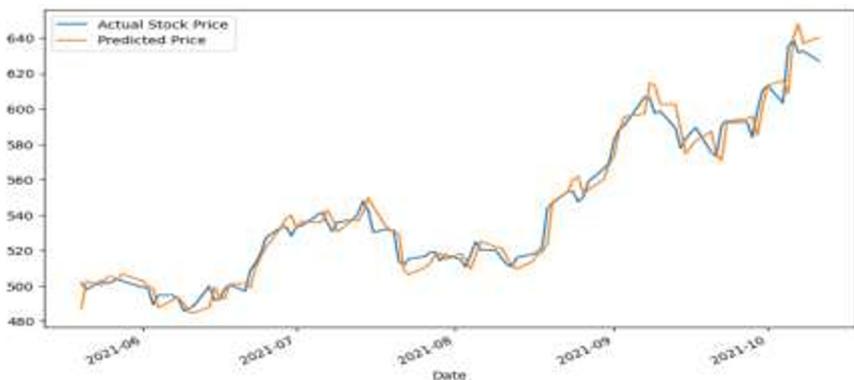
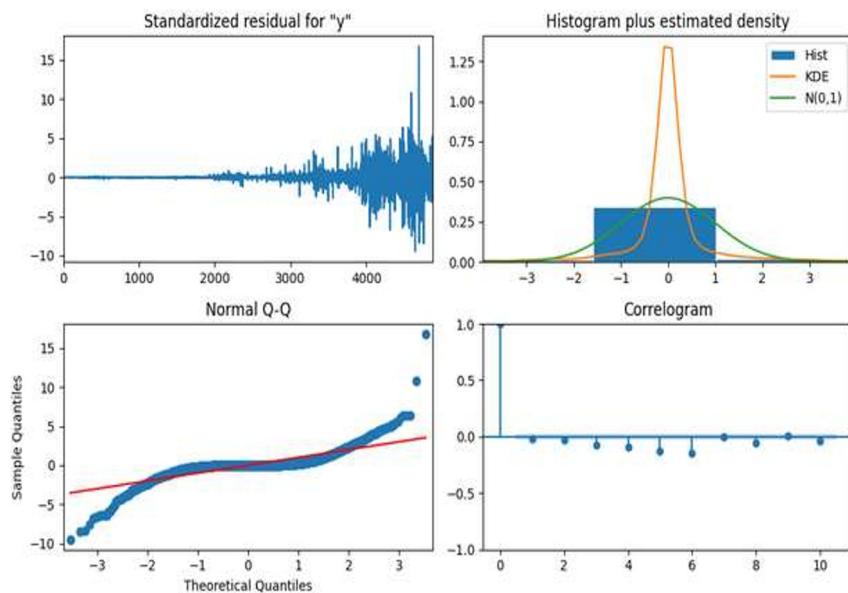
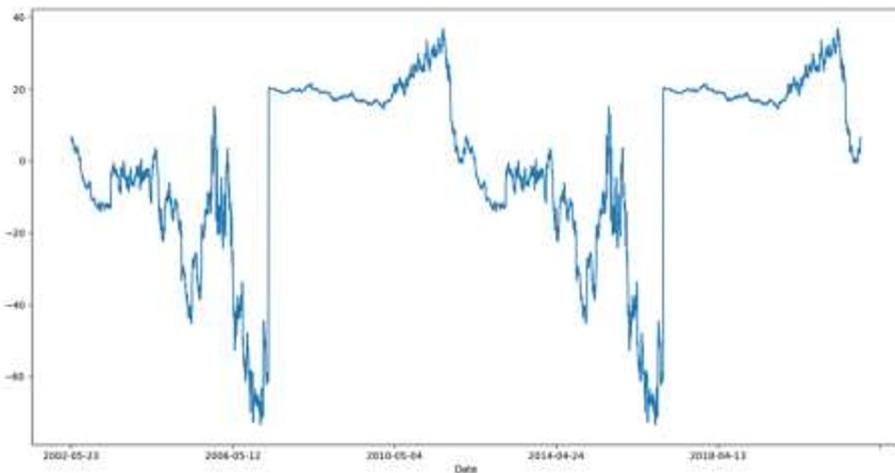


Fig. 3. Actual vs Predicted for 2021

5.7 Error Calculation:

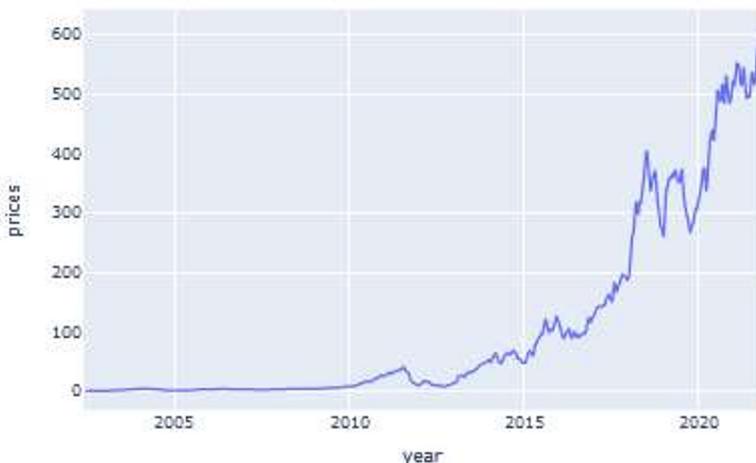


5.8 Seasonality variation:



5.9 LSTM RESULTS:

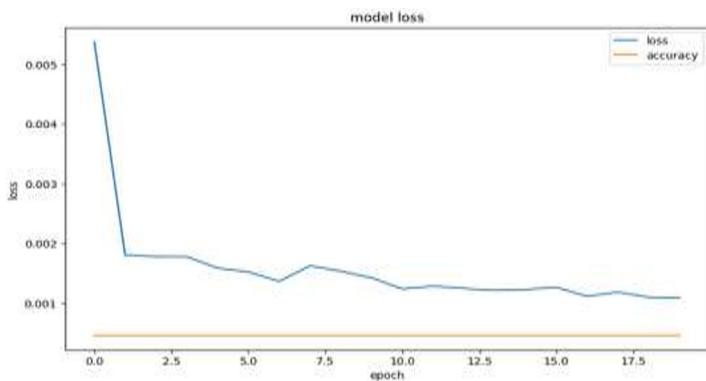
NETFLIX



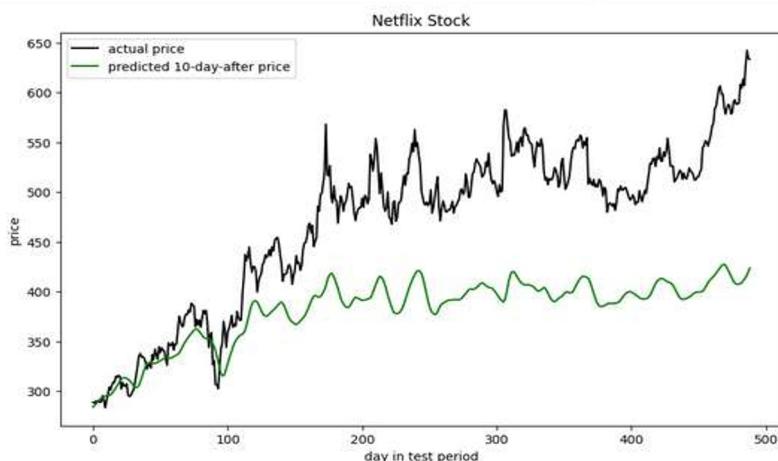
5.10 Summary of LSTM Model:

```
Model: "sequential"
-----
Layer (type)          Output Shape       Param #
-----
lstm (LSTM)           (None, 30, 50)      18400
dropout (Dropout)     (None, 30, 50)      0
lstm_1 (LSTM)         (None, 30, 50)      20200
dropout_1 (Dropout)   (None, 30, 50)      0
lstm_2 (LSTM)         (None, 50)          20200
dropout_2 (Dropout)   (None, 50)          0
dense (Dense)         (None, 1)           51
-----
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
```

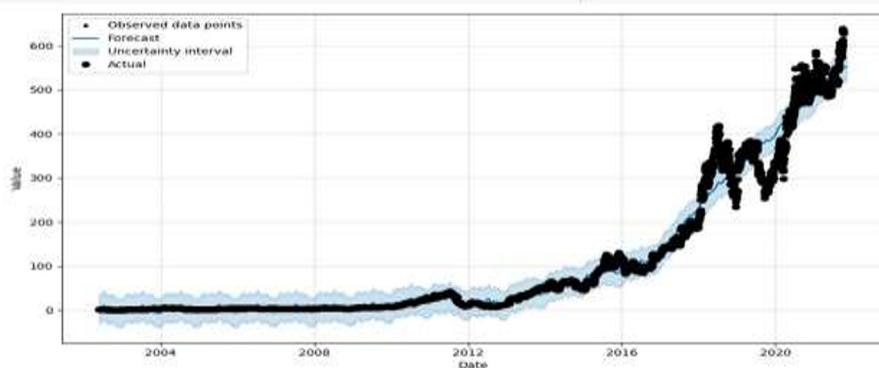
5.11 Model loss and accuracy:



5.12 Actual vs Predicted stock prices:



5.13 FBProphet Results:



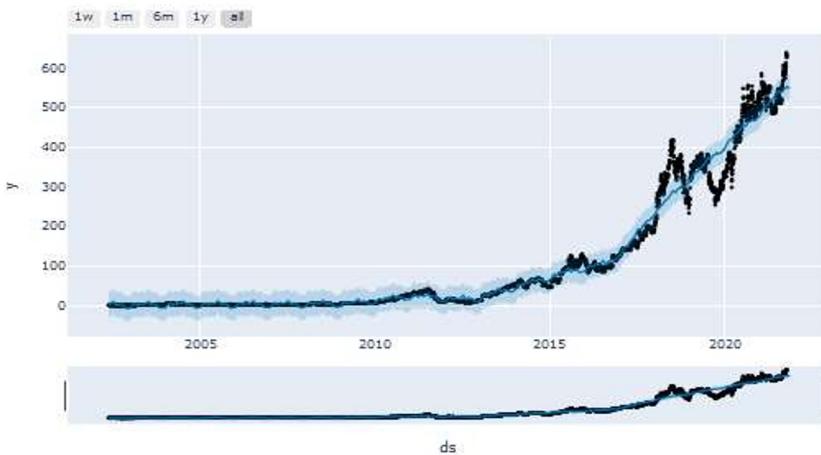


Fig. 4. FB Prophet Result

5.14 Overall Results:

Model	RMSE
ARIMA	7.891925316
LSTM	10.33765269
FBProphet	9.118639484

6 Conclusion and Future scope:

In conclusion, time series forecasting of the stock market is a difficult endeavor that demands complex methodologies and meticulous study. While no forecasting method can predict stock prices with absolute certainty, time series forecasting provides useful insights and enables investors to make informed decisions.

- ARIMA, LSTM, and Prophet time series forecasting models have been routinely utilized to forecast stock market values.
- To make forecasts, these models use numerous components such as trend, seasonality, and autocorrelation.

However, stock market forecasting is dynamic and volatile, predicting it is intrinsically difficult, and factors such as economic circumstances, geopolitical events, and market mood may all have a big influence on stock prices.

Finally, we may infer that the ARIMA model fits the presented data better than the LSTM and FBProphet models. This model was chosen as the final model to deal with the data since it had lower RMSE values than other models.

Incorporating sophisticated machine learning approaches, such as deep learning architectures (e.g., transformers, attention-based models), reinforcement learning, and ensemble models, can enhance forecasting accuracy.

It is critical to remember that the stock market is driven by a complex combination of forces that includes economic, psychological and political components. While time series forecasting techniques can aid in making educated judgements, when it comes to investing in stocks, it is critical to show care and analyze many sources of information. Ongoing research and breakthroughs in forecasting methodology, data availability, and processing capacity bode well for future improvements in the accuracy and efficacy of stock market forecasting.

References:

1. Tiwary, Sidharth, and Pramod K. Mishra. "Time-series Forecasting of Stock Prices using ARIMA: A Case Study of TESLA and NIO."
2. Liu, Peiqi. "Time Series Forecasting Based on ARIMA and LSTM." *2022 2nd International Conference on Enterprise Management and Economic Development (ICEMED 2022)*. Atlantis Press (2022).
3. Zhao, Cheng, et al. "Stock market analysis using time series relational models for stock price prediction." *Mathematics* **11.5** (2023): 1130
4. Umer, Muhammad, Muhammad Awais, and Muhammad Muzammul. "Stock market prediction using machine learning (ML) algorithms." *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* **8.4** (2019): 97-116.
5. Han, Kuk-Hyun, and Jong-Hwan Kim. "Genetic quantum algorithm and its application to combinatorial optimization problem." *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*. Vol. 2. IEEE (2000).
6. Min, Jae H., and Young-Chan Lee. "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters." *Expert systems with applications* **28.4** (2005): 603-614.
7. A. Gupta and B. Dhingra, "Stock market prediction using Hidden Markov Models," *2012 Students Conference on Engineering and Systems*, Allahabad, India (2012), pp. 1-4, doi: 10.1109/SCES.2012.6199099.
8. Mehtab, Sidra, and Jaydip Sen. "Stock price prediction using convolutional neural networks on a multivariate time series." *arXiv preprint arXiv:2001.09769* (2020).
9. Dwivedi, Subhash Arun, et al. "Analysis and forecasting of Time-Series data using S-ARIMA, CNN and LSTM." *2021 international conference on computing, communication, and intelligent systems (icccis)*. IEEE (2021).

10. Ariyo, Adebiyi A., Adewumi O. Adewumi, and Charles K. Ayo. "Stock price prediction using the ARIMA model." *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*. IEEE (2014).
11. I. Parmar *et al.*, "Stock Market Prediction Using Machine Learning," *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, (2018) , pp. **574-576**, doi: 10.1109/ICSCCC.2018.8703332.
12. Vijh, Mehar, et al. "Stock closing price prediction using machine learning techniques." *Procedia computer science***167** (2020): 599-606.
13. B. N. Varaprasad, C. Kundan Kanth, G. Jeevan and Y. K. Chakravarti, "Stock Price Prediction using Machine Learning," *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India, (2022), pp. 1309-1313, doi: 10.1109/ICEARS53579.2022.9752248.
14. Strader, Troy J., et al. "Machine learning stock market prediction studies: review and research directions." *Journal of International Technology and Information Management***28.4** (2020): 63-83.
15. Antad, Sonali, et al. "Stock Price Prediction Website Using Linear Regression-A Machine Learning Algorithm." *ITM Web of Conferences*. Vol. **56**. EDP Sciences, (2023).