# Stock Price Prediction: A Time Series Analysis

*Abstract*—**Predicting future stock volatility has always been a demanding chore for research studies. Individuals around the world have long regarded the stock market as a substantial profit. This research study seeks to comprehend historical stock market information and extract analysis from it in order to remedy the disparity between price movements and risk attitudes. A stock data set contains numerous precise terms that are difficult for an individual to comprehend when considering stock market expenditures. This thesis tries to fill that knowledge gap. This study aims to provide a future market scenario supported by statistical data. To foresee stock market volatility, we used time series analysis with the Long Short-Term Memory (LSTM), Bidirectional Long-short Term Memory (Bi-LSTM), Autoregressive Integrated Moving Average (ARIMA), Hidden Markov Model (HMM), and Multi-Head Attention.**

*Keywords*—**Stock market prediction, LSTM, Bi-LSTM, ARIMA, HMM, Multi-Head Attention, and Time series analysis.**

## I. Introduction

The stock exchange refers to the network of trades where shares of major corporations can be traded. A stock price prediction is an attempt to forecast the future value of an individual stock. The Stock Market Prediction is an indication of time-series forecasting in which prior data is quickly examined and subsequent data values are estimated. A time series is an accumulation of analyses of data items over a specific time period.

Because of the complexity of the stock market, stock value forecasting is regarded as being among the most difficult aspects of sales forecasting [1]. Accurate stock market prediction is essential for any company as it allows it to build and optimize strategies, explore new scopes, find insights from the business data, and help achieve tangible business results. The principal target of every business is making a profit; for this purpose, time series analysis for their stock data is fundamental. Many stockholders want to get their hands on forecasting techniques that guarantee enormous profit and reduce potential losses in the share market. This motivates investigators to grow, develop, and build innovative predictive models [2]. A few other research on forecasting stock prices has also been undertaken. Similar techniques, like Linear Regression and Support-Vector Regression, have been employed but the results were inadequate. To forecast future stock values, we used time series analysis with the Long Short-Term Memory (LSTM), Bidirectional Long-Short Term Memory (Bi-LSTM), Autoregressive Integrated Moving Average (ARIMA), Hidden Markov Model (HMM), and Multi-Head Attention.

Researchers could see the development of appropriate models of deep learning in [3] analyses of time series with numerous factors. The very first effort to model a time series of financial data applying a neural network was done in [4]. The study for IBM aimed to create a neural network model for detecting nonlinear symmetries in commodity market volatility. The creation of LSTM [5] improved the efficiency of the time-dependent analysis of the data. These networks have the ability to store historical data. [6] have used them to forecast stock prices. The Transformer model and Hidden Markov models are rarely employed in time series analysis, some scholars have already presented novel methods for forecasting stock prices utilizing several classical models. In order to accurately anticipate stock price, we used HMM and Transform models, which gave us good accuracy.

We're working with the Microsoft Corporation Stock dataset from 2022 here. We are primarily concerned with the date and the closing price of the stock. In stock price forecasting, the closing price of the stock market has a different meaning. The Closing Price assists the shareholder in understanding the stock market viewpoint over time. It is the most precise matrix for determining stock valuation until the market resumes trading the following day.

## II. Related Works

Professional investors utilize the share price analysis technique to decide whether to buy or sell a stock. By critically thinking, examining, and evaluating both historical and recent data, traders and investors try to obtain an advantage in the markets. There are extensive research papers on this topic, which makes it more convenient.

The paper [7] emphasizes the importance of time series by analyzing historical stock market data and thoroughly examining its fundamental principles to forecast stock price movements utilizing the LSTM neural network model's improved deep learning function for selective memory. The input layer, hidden layer, attention layer, and output layer are the four layers that make up this model. Model training is accomplished using the gradient descent approach. For the dataset, which comprised six criteria including the trading date, starting price, closing price, lowest price, highest price, and daily volume of the stock, they chose 300 indices of relevant importance for the Chinese stock market. They translate each input data element into the range [0,1] after standardizing the data, then split the input data into training and test sets in a 7:3 ratio. The hidden layer and attention layer of the stock deep learning model then predict the closing price of the following trading day. The initial step in preprocessing these 300 Chinese stock market data is the removal of invalid and null data as well as data normalization. They employed mini-batch gradient descent to train the model, using a learning rate of 0.001. LSTM gives 95.70% accuracy on the training dataset and 80% on the test dataset.

The hypothesis that merging RNNs with useful input variables might offer a more effective method for anticipating stock indicating the following day is examined by Gao, Tingwei et al. (2017) [8]. The goal of this study was to find out whether the stock prediction model could be implemented using long-short term memory (LSTM) and main trade data. For the study, Yahoo Finance stock trading data was gathered to assess and validate the efficacy of several models. A case study was presented based on Standard & Poor's (S&P500). Five different models were tested to demonstrate the utility of our system. The mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and average mean absolute percentage error (AMAPE) were computed to evaluate the model's performance. For prediction studies, the LSTM model produces better results.

The author [9] Gourav Bathla, attempted to elevate stock price prediction using ARIMA in this work. Because of the high variability of stock prices, techniques for deep learning are employed because of their verified efficiency in a variety of statistical domains. Recurrent neural networks were used in this case to boost prediction accuracy even further. For stock price prediction, they employed Linear Regression, Support Vector Machines, LSTM, CNN, RBM, ANN, ARIMA (Auto Regressive Integrated Moving Average), and standard machine learning approaches. Training took place from January 2015 to June 2019, and testing and validation took place from July 2019 to January 2020. They experimented from January 2015 to January 2020. In this research, The use of LSTM to different macroeconomic indicators is equated to ARIMA and regression models. The NYSE, NSE, NASDAQ, the Dow Jones Industrial Average and S&P500 are the market indices utilized. In this study, LSTM performs better than SVR and offers more accurate prediction. On several stock markets, convolutional neural networks achieve an accuracy of 92%.

Adil Moghar et al. (2020) [10] studied the ML algorithm based on LSTM RNN to predict the adjusted daily closing price for a portfolio of assets in order to discover the most accurate trained algorithm for forecasting future values. The model developed for this research has shown some encouraging results, showing that it can follow the development of opening prices for both assets. While only 20% of the data is used for testing, the remaining 80% is used to train the model. Utilizing mean square error, the model is enhanced throughout training. Additionally, precision is improved by using more training data epochs (12, 25, 50, and 100).

## III. Background Study

### A. Autoregressive Integrated Moving Average (ARIMA)

ARIMA is an acronym that stands for AutoRegressive Integrated moving average, and it's miles a category of the version that captures a collection of various trendy temporal systems in time collection information. Model identification, parameter estimation, and diagnostic checking are the processes in developing an ARIMA prediction model [11]. ARIMA version is used to are expecting destiny time collection motion via way of means of analyzing the variations among values withinside the collection in place of thru real values and additionally It may be used for any nonseasonal collection of numbers that famous styles and isn't always a chain of random events. In desk-bound time-collection information, the houses or cost of a pattern statement does now no longer rely on the timestamp at which its miles are observed. To cast off the correlation, ARIMA makes use of differencing to make the information desk-bound. ARIMA is super healthy to seize ancient trends, seasonality, randomness, and other non-static conduct that people miss. ARIMA models have demonstrated their effectiveness at generating short-term projections. In regards to short-term forecasting, it consistently beats complicated structural components [12].

### B. Long Short-term Memory (LSTM)

Time-series data may be accommodated by recurrent neural networks created specifically for this purpose, termed LSTMs.The LSTM is a more sophisticated version of the Recurrent Neural Network. This model was created to address the hard issues of gradient expansion and gradient extinction in RNN [13]. Accuracy diminishes because the states of many hidden layers cannot be preserved in RNN. In the long run, LSTM can save cell state. Because previous days' stock prices can be held, it can effectively forecast the stock price. In the LSTM model, the input gate, forget gate, and output gate are used to verify previous states. It has a feedback line that solves the problem of vanishing gradients. The feedback line is capable of tracking the entire data sequence. This demonstrates outstanding performance on a wide range of successive data. They are able to selectively learn, unlearn, or retain knowledge from each of the units using a cell state, three gates, and an LSTM module. It has an output, a forget gate, and three inputs. Here, a sigmoid is used to create a number between 0 and 1. A forget gate erases the past value at a specific time step, and the result is passed through an activation function that returns 0 when the information should be erased or 1 when it must be retained for use in the future. An input gate evaluates the relevance of new information carried mostly by input, while an output gate chooses which documentation should be conveyed onto the subsequent hidden layer, and also forget gate deletes the previous value. Long Short-Term Memory can be used to address the vanishing gradient problem. The network's LSTM structure is highly fitted for analyzing time-series data and therefore is commonly used in the analysis of time series tasks [14].

### C. Bidirectional Long-short Term Memory (Bidirectional LSTM)

The RNN version is Long-Term Short-Term Memory (LSTM). To remember a particular part of the input sequence, the LSTM uses a combination of different RNNs. Sometimes it's important to remember a certain block of data, but that's not always the case, and there are ways to deal with that. When trying to recall specific calculations, LSTM is used. Estimates can be used to control how much he should keep. The RNN can also reasonably recall part of the previous input, but it has two disadvantages. Gradients fade and explode. Then LSTM - embedding a memory element called a cell in the array - was used to solve this problem. Any neural network may keep the sequence in both directions: backward (future to past) or forward (past to future), utilizing the bi-LSTM approach. BiLSTM is made up of two LSTM, one forward and another backward. In comparison to the regular LSTM's yet another transfer, BiLSTM examines the shifting principles of the data during and after transmitting data and that can make higher complete and specific conclusions using prior and upcoming knowledge [15]. Our input works bidirectionally with bidirectional LSTM, unlike conventional LSTM.

### D. Hidden Markov Model (HMM)

HMM includes statistical models to capture confidential information from observable sequential emblems [16]. This statistical technique has grown in popularity in recent years due to its robust numerical method and development of hypotheses for usage in a broad range of situations [17]. In an HMM, the task is to extract the hidden parameters from the visible parameters in a system considered a Markov process with unknown parameters. According to this statement, an observed event will relate to a set of probability distributions rather than its step-by-step state. Making probabilistic models of problems involving the "labeling" of linear sequences is formally supported by hidden Markov models (HMMs). Additionally, they offer a conceptual toolkit for creating intricate models by creating an understandable image. As-transition probability, Emission probability, and starting probability are parameters of the HMM model. Learning in HMMs includes estimating the state transition probabilities A and output combustion probability B, which enhance the likelihood of an observed sequence.

### E. Multi-Head Attention (Transformer Model)

The words in this statement serve as an example of how a transformer model learns context and meaning by examining relationships in sequential data. Transformer models apply an evolving set of mathematical techniques, known as attention or self-attention, to uncover the subtle ways in which even data items are distant from the chain of influence and interdependence. Transfiguration translates text and speech near real-time, opening meetings and classes to diverse and hearing-impaired participants. To efficaciously extract the embedded input and solidify the model ultimately, numerous thick layers and a global average pooling layer are included in the model. The envisaged model involves time embedding, which should enhance its performance in time series prediction problems like stock price prediction. The test data is used to evaluate the trained models. Even though the test data are not utilized during the preprocessing step, the results emphasize how accurately the model fits the data on extracted features, which is essential as it demonstrates how effectively the model can predict price movements [18].

### F. Performance Matrices

Machine learning models' quality is assessed using performance indicators. Here, the study was evaluated using three distinct error matrices, including Mean absolute error (MAE), Mean squared error (MSE), and Root Mean square error (RMSE).

Below $x$ and $y$ are $D$ dimensional vectors, and $x_i$ denotes the value on the $i$th dimension of $x$.

Mean Absolute Error(MAE) = $\sum_{i=1}^{D} |x_i - y_i|$
Mean Squared Error(MSE) = $\sum_{i=1}^{D} (x_i - y_i)^2$
Root Mean Square Error (RMSE) = $\sqrt{MSE}$

## IV. Dataset

The dataset is collected from yahoo finance, which is a media property of Yahoo corporation. We used the 2022 Microsoft Corporation Stock dataset which is consisting of 9166 data. The dataset contains 7 features: date, high, low, opening price, closing price, Adj close, and volume. Our selected features will be 'date', and 'closing price'.

## V. Methodology

ARIMA, LSTM, HMM, Transformer, and Bidirectional layers are used in this research. Adam optimizer and relu activation function are also used. Furthermore, epochs are set to 10 for LSTM and bidirectional LSTM. For experiment setup, the python libraries such as Keras, Scikit-learn, Pandas, Numpy, and Matplotlib are utilized. These analyses are executed on the Google Colab GPU. Microsoft stock indexes are used.

### A. Dataset Prepossessing

- Eliminating Unnecessary Columns:
  There were several columns that were irrelevant to our work, such as high, low, opening price, Adj value, and volume. We have eliminated those columns.

- Null Value Handling:
  We have deleted the rows having null values. We have discarded all of that. The number of null values was so less. So there was any impact after deleting those columns in our dataset.

*B. Model building*

After pre-processing and using deep learning models (RNN, LSTM, Bi-LSTM, ARIMA, HMM, and Transformer (Multi-Head Attention)), we divided our data into training and testing data. Then, we calculated the MSE, RMSE, and MAE for each model and identified the one that performed the best. The model diagram of this project is given below.

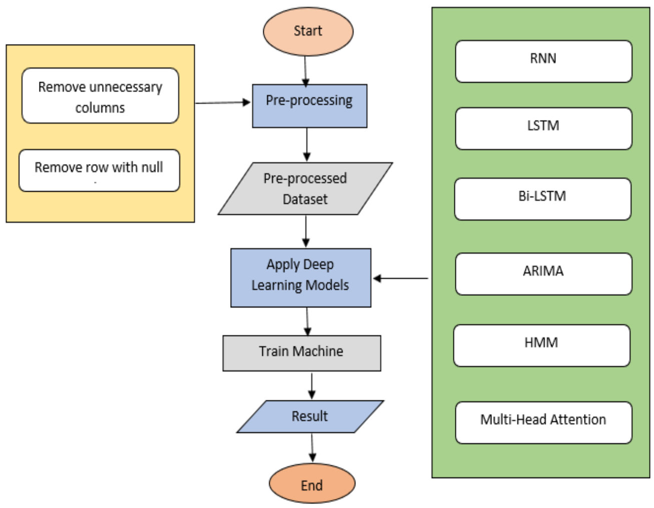


Fig. 1. Model Diagram

## VI. Result and Analysis

We used ARIMA, LSTM, BiLSTM, HMM, and Transformer models to predict the stock market price and plot it to understand the difference between actual and predicted prices. We also figured out the data by performance matrix.

- ARIMA:

  After applying the ARIMA model to our dataset, we get 5.467 MAE,and 44.539 MSE. In our research ARIMA model gives rmse 6.674, which is pretty good. 'AutoRegressive Integrated Moving Average' is a forecasting algorithm for stock market prediction. It has the capability to capture sudden ups and downs in stock price. ARIMA uses a moving average where random jumps to the time series plot that has an impact over two or more successive periods. These leaps show our ARIMA model's erroneous calculations and what the MA component would lag for. Similar to the exponential smoothing strategy, a model that uses MA would smooth out these abrupt jumps.

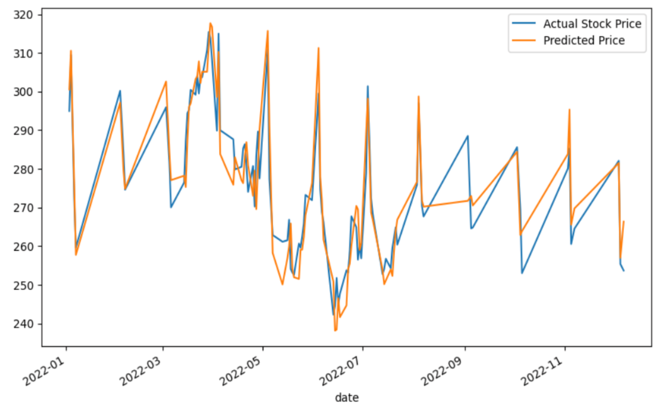


Fig. 2. Stock prediction by ARIMA

- LSTM:

  After applying the LSTM model to our dataset, we get 18.541 MAE, and 461.942 MSE. In our work, LSTM gives the rmse 21.493. So, LSTM could not perform well because our dataset is stationary, which means no trends or patterns concerning time. There are so many fluctuations in our dataset which couldn't be perfectly predicted by LSTM. LSTM works better when a considerable amount of sequential and pattern data is available. Time period from March 1986 to July 2022 is considered in which 13th march 1986 to 28th February 2022 is used for training and 3rd January 2022 to 22th July 2022 is used for testing. Here, we can see that the ARIMA model performs better than the LSTM model.

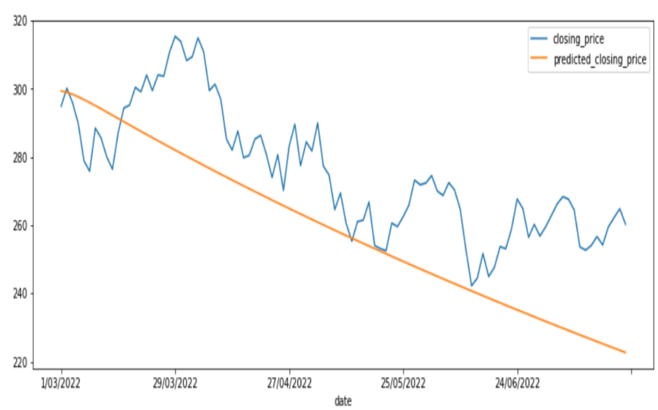


Fig. 3. Stock prediction by LSTM

- Bi-LSTM:

  After applying the BiLSTM model to our dataset, we get 11.256 MAE, and 217.860 MSE. We have gained rmse 14.760 using Bi-directional LSTM gives better optimization and prediction result. The method of enabling any neural network to retain sequence information in both directions, either backward or forwards, is widely used as bidirectional long-short term memory (BiLSTM). Our input is bidirectional and flows in both directions, setting a BiLSTM apart from a conventional LSTM. To store past and future information of stock at any time step, we can have input flow in both directions. Here, we also can see that the BiLSTM model performs better than the LSTM model. Moreover, the ARIMA model gives better performance than BiLSTM and LSTM.

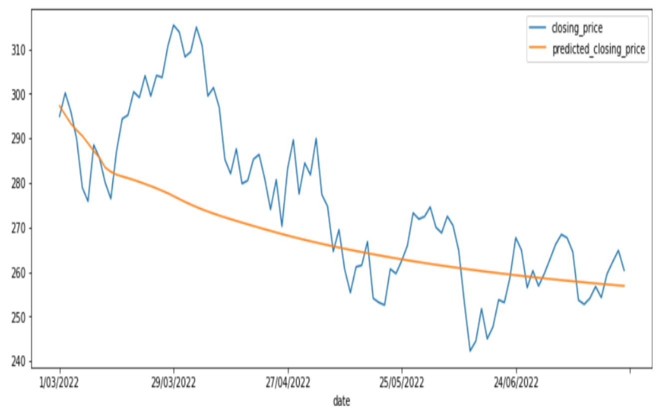


Fig. 4. Stock prediction by Bi-LSTM

- HMM:

HMM are Legos of computational sequence analysis. We have used three performance evaluation metrics on the HMM model. After applying HMM model to our dataset, we get 0.48553775 for MAE, 0.040804 for MSE, and 0.20203499 for RMSE. Due to inherent randomness, stock prediction is difficult. To identify hidden patterns, the Hidden Markov Model is employed for stock prediction. Here, the 10 hidden components (or states) of our data were quickly identified by the Hidden Markov model. The test data for the preceding 50 days will be used to test all potential outcomes for each of the days for which we will predict closing prices; we take the Cartesian product across a range of values for each feature to determine which sequence produces the highest score. The predictions for that day's closing price are then based on the result that receives the highest score. Hidden states and sequences have already been created to swiftly measure the level of both the sequence and if the value grows the next day. Additionally, it was determined whether the amount is moderately high, high, or very high, as well as whether it is moderately low, low, or deficient. For both long-term and short-term investors, this model will be constructive.

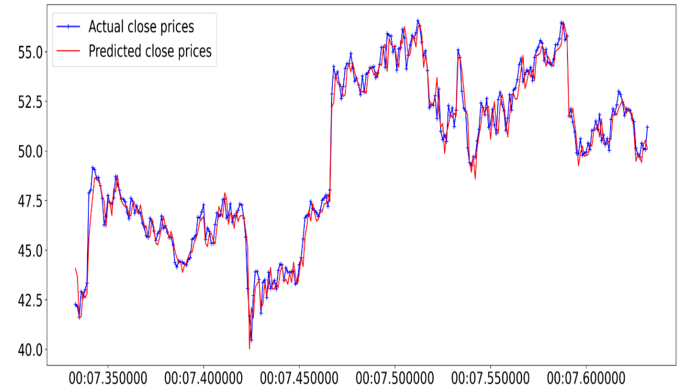


Fig. 5. Stock prediction by HMM

- Multi-Head Attention (Transformer Model):

After applying the transformer model, we get 0.09936 for MAE,0.0234 for MSE, and 0.15309 for RMSE. Transformers are an excellent method for determining the semantic relationships between the components of a short or lengthy sequence. The transformer model is so strong for time series analysis in the case of both stationary and non-stationary data. It can capture trends as well as sudden ups and downs in the stock movement. In our transformer model, we used 60 attention heads, and the Embedding size for attention was 46. The number of transformer blocks was 5, which is well fitted for our ten thousand data. Adam optimizer and 0.14% dropout were used. The transformer model is made by a multi-headed attention mechanism. The Multi-Head Attention mechanism was designed to allow the decoder to utilize the much more relevant elements of the input sequence in a responsive manner by weighting all of the encoded input variables, resulting in the most relevant vectors being assigned the highest weights. A module for attention mechanisms called 'Multi-head Attention' cycles repeatedly and simultaneously through an attention mechanism. The predicted dimension is then created by linearly combining the separate attention outputs.

The performance of different architectures are summarized in the table given below.

TABLE I
PERFORMANCE OF ALL THE MODELS.

| Model | MAE | MSE | RMSE |
|---|---|---|---|
| Transformer | 0.099 | 0.023 | 0.153 |
| HMM | 0.486 | 0.041 | 0.202 |
| ARIMA | 5.467 | 44.539 | 6.674 |
| BiLSTM | 11.256 | 217.860 | 14.760 |
| LSTM | 18.541 | 461.942 | 21.493 |

In our research transformer model performs best. The Attention module of the Transformer performs its calculations repeatedly and concurrently. These are referred to as Attention Heads each. Attention models, also renowned as attention mechanisms, are deep learning algorithms that emphasize a certain aspect. In deep learning, attention refers to dwelling on something specific and highlighting its explicit significance. So multiple attention modules can easily predict the fluctuation in the time series. The Hidden Markov model also gives a good result with rmse 0.202. HMM offers a strong stochastic foundation and high intellectual methods, allowing the acquisition to occur straight from the raw sequence analysis. It supports variable length inputs and enables systematic processing of insertion and deletion costs in the context of selectively surmountable algorithms. Here we used the Gaussian Hidden Markov model with 10 hidden states. ARIMA model ranks third in our research with a rmse 6.674. ARIMA model has been used extensively in finance, having the qualities to be robust and efficient, and offers a promising short-term share market forecast method. In comparing LSTM and Bidirectional LSTM, Bidirectional LSTM performs better than LSTM. The input flows in both directions, which makes it capable of utilizing information from both sides. So accuracy comes better.

Overall, we can say that the transformer model performs better for our data set. Also, we can write the sequence of performance as:

$$\text{Transformer} > \text{HMM} > \text{ARIMA} > \text{BiLSTM} > \text{LSTM}$$

## VII. Limitations and Future Work

When we first started using the models, we ran into a number of restrictions. These shortcomings also show where further developments can be made.

Firstly, the dataset we selected for our research was not so enormous. Our approach was univariate. We only worked on a single company's data. So, in the future, we will work with a multivariant approach. We will try to work with an extensive dataset. We will work with multiple companies to make it more convenient.

## VIII. Conclusion

The price of a stock indicates its present value to buyers and sellers. The fluctuations in the stock market may not always occur consecutively or continue that same trend. The prevalence of tendencies and the period of their persistence will vary depending on the enterprises and sectors. Understanding these types of patterns and cycles will result in greater profits for entrepreneurs. Nevertheless, volatility and stock values are much more than a collection of random numbers. As a result, they can be examined as a set of discrete-time data. In other terms, time-series data observations are undertaken at regular intervals. Time series forecasting applies well to stock forecasting. The main topic is stock price prediction, and we have used LSTM, Bi-LSTM, HMM, ARIMA, and Transformer models. By treating stock market data as a time series, one can use past stock prices to forecast stock prices for the coming month. The advantage of stock price forecasting is that it reveals the future behavior of the market, always helping investors understand when and what types of stocks can be purchased to grow their investment.

## References


[1] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[2] M. M. Mostafa, "Forecasting stock exchange movements using neural networks: empirical evidence from kuwait," *Expert systems with applications*, vol. 37, no. 9, pp. 6302–6309, 2010.

[3] B. Batres-Estrada, "Deep learning for multivariate financial time series," 2015.

[4] H. White, "Economic prediction using neural networks: The case of ibm daily stock returns," in *ICNN*, vol. 2, pp. 451–458, 1988.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] H. Jia, "Investigation into the effectiveness of long short term memory networks for stock price prediction," *arXiv preprint arXiv:1603.07893*, 2016.

[7] D. Wei, "Prediction of stock price based on lstm neural network," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, pp. 544–547, IEEE, 2019.

[8] T. Gao, Y. Chai, and Y. Liu, "Applying long short term momory neural networks for predicting stock closing price," in *2017 8th IEEE international conference on software engineering and service science (ICSESS)*, pp. 575–578, IEEE, 2017.

[9] G. Bathla, "Stock price prediction using lstm and svr," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 211–214, IEEE, 2020.

[10] A. Moghar and M. Hamiche, "Stock market prediction using lstm recurrent neural network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020.

[11] B. G. Tabachnick and L. S. Fidell, *SAS for windows workbook for Tabachnick and Fidell using multivariate statistics*. Allyn and Bacon, 2001.

[12] A. Meyler, G. Kenny, and T. Quinn, "Forecasting irish inflation using arima models," 1998.

[13] V.-D. Ta, C.-M. Liu, and D. A. Tadesse, "Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading," *Applied Sciences*, vol. 10, no. 2, p. 437, 2020.

[14] S. Mehtab and J. Sen, "Stock price prediction using cnn and lstm-based deep learning models," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 447–453, IEEE, 2020.

[15] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang, "A stock closing price prediction model based on cnn-bislstm," *Complexity*, vol. 2021, 2021.

[16] D. Venugopal, S. K. Kaliyaperumal, and S. M. Niraikulathan, "Stock market trend prediction using hidden markov model," in *Forecasting in Mathematics-Recent Advances, New Perspectives and Applications*, IntechOpen, 2020.

[17] M. R. Hassan and B. Nath, "Stock market forecasting using hidden markov model: a new approach," in *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pp. 192–196, IEEE, 2005.

[18] T. Muhammad, A. B. Aftab, M. Ahsan, M. M. Muhu, M. Ibrahim, S. I. Khan, M. S. Alam, *et al.*, "Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market," *arXiv preprint arXiv:2208.08300*, 2022.