Predicting Stock Prices: A Deep Dive into Machine Learning and Deep Learning Techniques

Aditya Shah[1†], Hrishita Shah[1†], Shrutam Shah[1†], Vishal Parikh[1*]

[1*]CSE Department, Institute of Technology, Nirma University, S G Hoghway, Ahmedabad, 382481, Gujarat, India.

*Corresponding author(s). E-mail(s): vishalparikh@nirmauni.ac.in;
Contributing authors: 21bce008@nirmauni.ac.in;
21bce089@nirmauni.ac.in; 21bce273@nirmauni.ac.in;
†These authors contributed equally to this work.

Abstract

The behaviour of stock prices is random and uncertain, making it difficult to predict, and hence, one of the most complicated tasks of Machine Learning and Deep Learning is Stock Price Prediction. Traditional Models like SVM, Random Forest, etc. fall short in trying to capture the long-term time dependencies of the stock market. This paper explores the usage of Long Short Term Memory (LSTM) networks, a type of Recurrent Neural Networks, which has proven to be efficient in capturing temporal dependencies in the time-series data by comparing other deep learning and machine learning models. The study utilizes different LSTM architectures and compares them at different timesteps to understand and contrast between the results achieved by simpler and complex architectures. The key factors of developing the LSTM models, including the process of feature engineering including technical indicators, normalization of data, creating the input-output sequences, and training of the models, are discussed in depth. Evaluation metrics show that a one-layer LSTM model, optimized for different time-step intervals, outperforms the complex multilayer architectures and traditional models, by achieving the highest accuracy in predicting the price of a stock for the next day. This research highlights the potential of LSTM models for financial forecasting and provides valuable insights for improving stock market applications.

Keywords: Machine Learning, Deep Learning, RNN, LSTM, Data Preprocessing, Stock Price Prediction

1 INTRODUCTION

The stock market is known for its volatility, randomness, and unpredictability. It is a chaotic place with an unbelievably huge, continuously changing stream of data, making predicting the stock price and acting on those predictions difficult. It is one of the most challenging tasks in times series forecasting[1]. The buying and selling price of the stock changes continuously because of the law of demand and supply. As the demand for a stock increases, we see a substantial increase in the buying price of the stock, and also, when the supply increases, there is a decrease in the buying price of the stock. Buyers will try to buy the stock at the lowest price possible, and sellers will try to sell the stock at the highest price possible. The condition of the financial market when the prices of the stocks are expected to rise is called the Bullish Market. The condition when a market experiences price decline is called a bearish market. There have been many researchers who have tried to predict the price of the stock market using many different Machine Learning models like ARIMA (Auto Regressive Integrated Moving Average) [2], SVM (Support Vector Machine) [3] and Linear Regression but these traditional techniques have limitations that the accuracy is not adequate.[4] Hence, we move forward to another domain called Deep Learning which focuses on using models like Artificial Neural Networks (ANN)[5], Recurrent Neural Networks (RNN)[6], Long Short Term Memory (LSTM)[7], Convolutional Neural Network (CNN)[8]. Deep learning is a fast-growing domain in the present world with numerous applications, one of them being Stock Price prediction. The study aims to use different machine learning and deep learning models to predict stock prices with the best accuracy. Neural networks have the best results over time series data, and stock prices are time series data. The neural network used here is Long Short Term Memory(LSTM)[7]. There have been many technological advancements in this field, but still, a single method that can be considered the most accurate for prediction has not been found. This can be due to the global economy, company financial reports, natural disasters, public health conditions, political factors, COVID-19, etc. Hence, different models give different accuracy based on the factors that are taken into account. LSTM model offers a mechanism that combines weak sources of information and makes it a strange tool that can be used efficiently[9]. Some researchers use only technical data, while others use historical data. The developed model will give accurate results if a good combination of data is used. In this paper, we try to compute various ML and DL models to predict the stock price.

2 RELATED WORKS

In the past, artificial neural networks were used to deal with non-linear data, but they could not make predictions with adequate accuracy. ANN includes a set of threshold functions. These functions are trained on historical data after connecting each other with adaptive weights, and they are used to make future predictions. [10]. Usmani et al. [11] used models like ARIMA and SMA to predict the Karachi Stock Exchange (KSE) stock prices. Many other models like Gated Recurrent Unit(GRU), Support Vector Machine(SVM), Support Vector Regressor(SVR), and Convolutional Neural Network(CNN) were also brought into effect, and later, researchers started combining

these models into one to create models that could give better accuracy and precision. Many researchers have used the Bi-LSTM, LSTM-LSTM, CNN-LSTM, and ARIMA-LSTM models to make predictions. The model was chosen according to the application it was being used for. Later, other researchers started diving deeper and exploring new models to make more accurate predictions.

In their paper, L. Di Persio et al. [12] compared different models to forecast different stock price movements. They took into account the daily data of the SnP 500 index to train the dataset. It was discovered to show the best outcome when working on large-scale classification tasks.

Dr. Karunakar Pothuganti et al. [13] used transfer learning in their research to take the advantage of the pre-built neural networks. He integrated various datasets using machine learning techniques because relying on a single dataset is not advisable for these predictions.

Sakshi Kulshreshtha et al. [14] developed a novel LSTM-ARIMA hybrid model. They also used the Facebook library called Prophet. They used MSE, RMSE and MAPE as their evaluation metrics and later concluded that the ARIMA-LSTM model provided the best accuracy.

Zhenbin Gao et al. [15] researched by finding a correlation between investor sentiments and the hybrid model of VMD-LSTM. It included sentiment analysis and deep learning techniques for the best prediction accuracy.

Kotrappa et al. [16] also performed a bibliometric survey to predict stock prices using sentiment analysis and LSTM.

Tengku Nurul et al. [17] used the LSTM model on the FTSE Bursa Malaysia KLCI (FBM KLCI) stock market. The closing prices were used to build the model, and RMSE and MAE were the evaluation metrics considered to choose the best model.

3 MACHINE LEARNING AND DEEP LEARNING IN STOCK PRICE PREDICTION

Predicting stock prices is a complex task due to the influence of multiple factors like open price, volume, market sentiment, and government policies. Machine learning algorithms like Random Forest and Artificial Neural Networks (ANN) can help, but time series data like stock prices are better handled by models such as LSTM, CNN, and RNN. CNN, traditionally used in image processing, is also explored for stock prediction, while RNN models are favored for their ability to capture time dependencies. LSTM, an extension of RNN, excels in stock price prediction by using a "forget gate" to discard irrelevant information, making it highly effective with data from sources like NIFTY. The LSTM variant with "peephole connections" further enhances its predictive capabilities by allowing gate layers to access the cell state directly. regressions ones.

3.1 TOTAL ANALYSIS OF ML AND DL MODELS:

From the results, both ML and DL models were thoroughly evaluated for predicting Reliance Stock prices. Among the machine learning models (SVM, ANN, Logistic

Prediction Models	Stock	Accuracy	Dataset
ANN	Reliance	55.12	Yahoo Finance
CNN	Reliance	68.91	Yahoo Finance
LSTM	Reliance	87.23	Yahoo Finance
AutoEncoderDecoder	Reliance	61.28	Yahoo Finance
SVM	Reliance	51.25	Yahoo Finance
RandomForest	Reliance	71.34	Yahoo Finance
LogisticRegression	Reliance	42.26	Yahoo Finance

Table 1: Assessing Model Performances

Regression, and Random Forest), Random Forest showed the highest predictive accuracy due to its ability to handle complex relationships and capture nuanced stock patterns. In deep learning models, LSTM excelled in capturing temporal dependencies, retaining long-range patterns, and understanding trends in stock movements, outperforming CNN and Auto Encoder-Decoder architectures. This emphasizes LSTM's significance in stock price prediction, particularly due to its tailored design for time series analysis and memory retention capabilities. Overall, LSTM proved to be the most accurate model for stock prediction, making it a powerful tool for financial forecasting. The following sections focus on a detailed study of the LSTM model.

4 INTRODUCTION TO LSTM

4.1 What is Long Short Term Memory

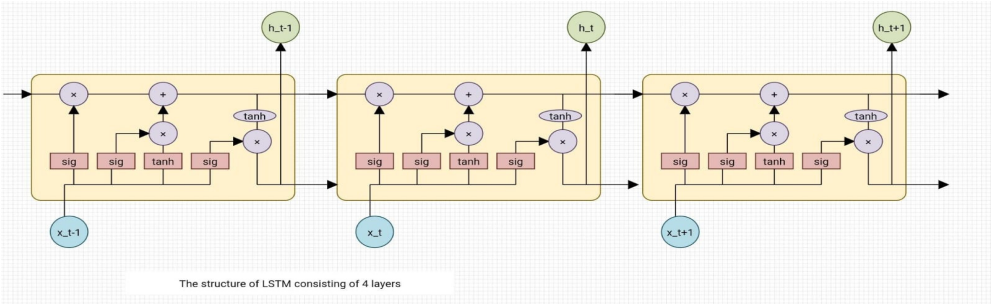


Fig. 1: Basic Structure of LSTM Cell.

LSTM, introduced by Hochreiter and Schmidhuber in 1997, addresses RNNs' difficulty in retaining long-term information, making it ideal for time-series tasks like stock price prediction. LSTM can manage sequential data through its ability to store information over time.

Each LSTM node has cells that pass information through various stages, controlled by a gating mechanism to determine what data to keep, update, or discard [3]. This is achieved with components like the input layer, hidden layer, cell state, and output

layer, using a combination of a sigmoid layer, hyperbolic tangent layer, and point-wise multiplication.

LSTM manages input, short-term, and long-term memory through three gates: the forget gate discards irrelevant data, the input gate updates relevant information, and the output gate produces the final result, enabling efficient processing of sequential data.

4.2 BASIC ANATOMY OF A CELL

- Forget Gate: Decides what information to discard from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

  Let f_t be the forget gate activation at time t, where σ is the sigmoid activation, W_f is the forget gate weight matrix, h_{t-1} is the previous hidden state, x_t is the input at time t, and b_f is the forget gate bias.
- Input Gate: Decides which new information to store in the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

  Let i_t be the input gate activation at time t, where W_i is the input gate weight matrix, and b_i is the input gate bias.
- Cell State Update: Updates the cell state with new candidate information.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

  Let $\tilde{C}_t$ be the candidate cell state at time t, where tanh is the hyperbolic tangent activation, W_C is the weight matrix for the candidate cell state, and b_C is the bias for the candidate cell state.
  The cell state update is:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

  Let C_t be the cell state at time t. It is updated by combining the previous cell state C_{t-1} (multiplied by f_t) and the new candidate cell state $\tilde{C}_t$ (multiplied by i_t).
- Output Gate: Decides what the next hidden state will be.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

  Let o_t be the output gate activation at time t, where W_o is the output gate weight matrix and b_o is the output gate bias.
- Hidden State Update: The hidden state is updated with the filtered information from the cell state.

$$h_t = o_t * \tanh(C_t) \tag{6}$$

  Let h_t be the hidden state at time t. It is updated by filtering the current cell state C_t through the output gate o_t and the tanh activation function.

Algorithm 1 Feature Engineering

Input: Dataset
Output: Modified Dataset with Additional Features
Initial Features Extraction:
Extract initial features: close_price, open_price, high_price, low_price, volume
Technical Indicators Extraction:
Extract different technical indicators: EMA(50),EMA(200),RSI etc.
return Modified Dataset with Additional Features

5 LSTM MODEL DEVELOPMENT

For this research, historical data from stocks like Reliance and ITC, sourced from the Nifty 50 website, was used, covering the period from March 1st, 2003, to March 1st, 2023. Feature engineering is crucial for improving model accuracy. Key initial features include Open Price, Close Price, High Price, Low Price, and Volume. To help the LSTM model recognize stock patterns, technical indicators such as EMA(50), EMA(200), SMA(50), SMA(200), RSI, MACD, Bollinger Bands (upper, middle, lower), %K, %D, OBV, and ATR were used, utilizing a Python library for implementation.

Algorithm 2 Normalization

Input: Features
Output: Preprocessed Features, Output Variable
Remove Null Values:
Remove null values from Features
Assign Output Variable:
Set Output_Variable as close_price
Features Transformation:
Apply Min-Max Scaling to Features
Features_Transform ← MinMaxScaler(Features)
return Preprocessed Features (Features_Transform), Output Variable (close_price)

To preprocess the data, first check for any null values, mainly from the technical indicators (EMA and SMA), which require 200 days of data. Therefore, remove the first 200 entries as their EMA and SMA values will be null. Only numerical data is considered in the dataset. After filtering, apply Min-Max normalization to scale the attributes between 0 and 1.

The next step involves preparing the training and testing data. In this study, four timesteps (5 days, 14 days, 30 days, and 50 days) are used to predict the next day's closing price. Input data will be taken from the first n days, and the goal is to predict the closing price for the (n+1)th day. The training and testing inputs should be sliced accordingly, with the outputs being the closing price of the (n+1)th day based on the chosen n value.

Algorithm 3 Create Input and Output Sequences

Input: Data, Sequence Length
Output: Input Sequences (X), Output Sequences (Y)
Initialize empty lists to store input sequences (X) and output sequences (Y): X ← [], Y ← []
for $i = 0$ to $len(Data) -$ Sequence Length **do**
    Append a sequence of length Sequence Length starting from index i to X:
    X.append(Data[$i : i +$ Sequence Length])
    Append the data point occurring Sequence Length steps ahead of index i to Y:
    Y.append(Data[$i +$ Sequence Length])
end for
return Input Sequences (X), Output Sequences (Y)

Algorithm 4 Model Training

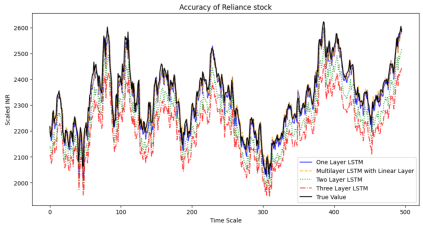
Input: Features, close_price
Output: Trained Model, Predicted Test Output
Split Dataset:
Split Features and close_price into train and test sets:
train_x, test_x, train_y, test_y ← train_test_split(Features, close_price)
Define Loss Function:
Set loss function as Mean Squared Error: loss ← mean_squared_error
Select Optimizer:
Choose optimizer: optimizer ← Adam
Train Model:
Train the model using train_x and train_y:
Model ← train_model(train_x, train_y)
Make Predictions:
Generate predictions for test data using the trained model:
pred_test_y ← Model(test_x)
return Trained Model (Model), Predicted Test Output (pred_test_y)

In this study, four different LSTM architectures were considered. The first architecture has a single LSTM layer with 450 hidden neurons connected to a linear layer and a Dense layer with one neuron, using 841,051 parameters. The second architecture has a single LSTM layer with 250 hidden neurons, followed by a Dense layer with 200 neurons, another LSTM layer with 250 neurons, and a Dense layer with one neuron, utilizing 768,451 parameters. The third architecture includes two LSTM layers, each with 230 hidden neurons, and a Dense layer with one neuron, with 651,591 parameters. The fourth architecture has three LSTM layers, with 220, 180, and 180 hidden neurons, followed by a Dense layer with one neuron, using 757,381 parameters. All models were trained for 50 epochs with a batch size of 64, using Mean Squared Error as the loss function, Adam as the optimizer, and the tanh activation function.

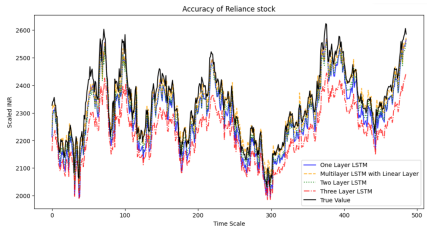
6 EVALUATION METRICS

Evaluation metrics like Mean Absolute Error(MAE) and the Mean Squared Error(MSE) are important because they help us compare the different models by providing a quantitative basis and, hence, help us find the most suitable model. This has been performed on Reliance Stock data of the past 20 years against different LSTM Models, and MAE and MSE values have been noted down.

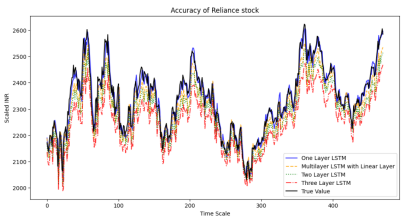
From the graphs, it is evident that increasing the number of timesteps increases the accuracy. Hence, the graph of 50 timesteps is most accurate for the One-Layer LSTM model compared to 5,15, and 30 timesteps. After implementing various LSTM models One-Layer, Two-Layer, Three-Layer, and Multilayer LSTM with a linear layer on 20 years of Reliance stock data, it was found that the One-Layer LSTM outperformed the others. Adding more layers increased complexity without improving performance. For the One-Layer LSTM, the accuracy for a 2% variation range was 82% for 5 timesteps, 83% for 15 timesteps, 84% for 30 timesteps, and 87% for 50 timesteps. This accuracy further rose to 97% for a 3% variation range. Therefore, One-Layer LSTM has an upper edge.



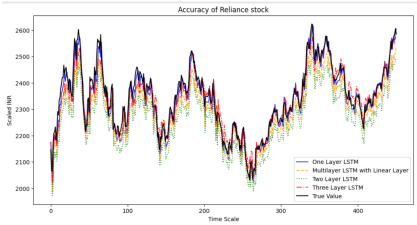
(a) Comparison Graph of LSTM Models for 5 timesteps



(b) Comparison Graph of LSTM Models for 15 timesteps



(c) Comparison Graph of LSTM Models for 30 timesteps



(d) Comparison Graph of LSTM Models for 50 timesteps

Fig. 2: Comparison Graphs of LSTM Models for Various Timesteps

From the MSE and MAE values for each timestep, it was observed that MultiLayer LSTM showed the least error values for MSE and MAE in the initial timesteps. As the timesteps increased, the lesser complex model, namely the One Layer LSTM proved

Table 2: MAE and MSE values for 5 Timesteps

Model	MSE	MAE
One Layer LSTM	1432.37	29.07
MultiLayer LSTM	1171.16	25.72
Two Layer LSTM	7129.14	77.86
Three Layer LSTM	16518.86	123.64

Table 3: MAE and MSE values for 15 Timesteps

Model	MSE	MAE
One Layer LSTM	3355.38	49.21
MultiLayer LSTM	1560.42	30.26
Two Layer LSTM	2640.93	41.71
Three Layer LSTM	13008.04	106.44

Table 4: MAE and MSE values for 30 Timesteps

Model	MSE	MAE
One Layer LSTM	1257.99	26.72
MultiLayer LSTM	2585.42	40.83
Two Layer LSTM	3956.09	53.04
Three Layer LSTM	8870.01	85.33

Table 5: MAE and MSE values for 50 Timesteps

Model	MSE	MAE
One Layer LSTM	1323.13	27.47
MultiLayer LSTM	3428.42	48.34
Two Layer LSTM	7768.41	80.69
Three Layer LSTM	1637.85	31.06

to be the most efficient. Two Layer LSTM and Three Layer LSTM did not provide good results in the chosen timesteps for our Reliance Stock in the chosen timeframe.

7 CHALLENGES IN STOCK PRICE PREDICTION

Stock prices are affected by market sentiment, making predictions difficult at times. Some of the reasons are:

- The abrupt and severe fluctuations in the input data features such as past price information, trade activity, editorial tone in the news, and economic indicators.
- The sudden profit booking made by foreign investors leads to an immediate reduction in the price of a stock.
- Statements made by famous financial influencers on various social media applications like Twitter, Instagram, etc, leading to sudden price variation.
- Changes in the price of a stock due to the winning of a certain political party.
- Introduction of new government policies regarding a certain sector, like banning the use of sugarcane for the production of ethanol, led to a sudden decline in all the stocks of the sugar industries.
- Same-sector companies are also interrelated, so a change in the price of one of the companies may affect the current price of other companies.

It is difficult to include this information in our LSTM model.

8 CONCLUSION

Based on the research conducted, it can be concluded that LSTM proves to be one of the best options for the prediction of time series data available from the stock market. The model needs to be provided with as much past information to it as possible with various technical indicators as well which helps it to analyze the market patterns

efficiently. While it is easier to incorporate the technical analysis by adding features like close price, open price, high price, low price and volume, and stock market being chaotic, it is extremely difficult to include the various different factors mentioned above for the overall analysis to find the pattern of a stock. For long-term investment, LSTM models can still prove to be more efficient than its competing deep learning models. In future, with more features like government factors, technical factors, market sentiment indicators, etc. in the dataset and more complex Neural Architectures like Transformers, researchers would be able to produce better results with much higher accuracy.

References

[1] Drashti Talati PBP Dr Miral Patel. Stock Market Prediction Using LSTM Technique. 2022;.

[2] Ariyo AA, Adewumi AO, Ayo CK. Stock Price Prediction Using the ARIMA Model. 2014;p. 106–112. https://doi.org/10.1109/UKSim.2014.67.

[3] Chaajer P, Shah M, Kshirsagar A. The applications of artificial neural networks, support vector machines, and long-short term memory for stock market prediction. Decision Analytics Journal. 2021 11;2:100015. https://doi.org/10.1016/j.dajour.2021.100015.

[4] Alkhatib K, Khazaleh H, Alkhazaleh HA, Alsoud AR, Abualigah L. A New Stock Price Forecasting Method Using Active Deep Learning Approach. Journal of Open Innovation: Technology, Market, and Complexity. 2022;8(2):96. https://doi.org/https://doi.org/10.3390/joitmc8020096.

[5] Khan Z, Alin T, Hussain MA. Price Prediction of Share Market Using Artificial Neural Network 'ANN'. International Journal of Computer Applications. 2011 05;22:42–47. https://doi.org/10.5120/2552-3497.

[6] Zhu Y. Stock price prediction using the RNN model. Journal of Physics: Conference Series. 2020 10;1650:032103. https://doi.org/10.1088/1742-6596/1650/3/032103.

[7] Salimath S, Chatterjee T, Mathai T, Kamble P, Kolhekar M. In: Prediction of Stock Price for Indian Stock Market: A Comparative Study Using LSTM and GRU; 2021. p. 292–302.

[8] Sayavong L, Wu Z, Chalita S. Research on Stock Price Prediction Method Based on Convolutional Neural Network. 2019;p. 173–176. https://doi.org/10.1109/ICVRIS.2019.00050.

[9] Sarkar A, Sahoo AK, Sah S, Pradhan C. LSTMSA: A Novel Approach for Stock Market Prediction Using LSTM and Sentiment Analysis. 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA). 2020;p.

1–6.

[10] Selvamuthu D, Kumar V, Mishra A. Indian stock market prediction using artificial neural networks on tick data. Financial Innovation. 2019 12;5. https://doi.org/10.1186/s40854-019-0131-7.

[11] Usmani M, Adil S, Raza K, Ali SS. Stock market prediction using machine learning techniques; 2016. p. 322–327.

[12] Di Persio L, Honchar O. Artificial neural networks architectures for stock price prediction: Comparisons and applications. 2016 01;10:403–413.

[13] Pothuganti MMSM Karunakar M A Istiake Sunny, Alharbi AG. Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model;.

[14] Kulshreshtha S, A V. An ARIMA-LSTM hybrid model for stock market prediction using live data. Journal of Engineering Science and Technology Review. 2020;.

[15] Gao Z, Zhang J. The fluctuation correlation between investor sentiment and stock index using VMD-LSTM: Evidence from China stock market. The North American Journal of Economics and Finance. 2023;66(C). https://doi.org/10.1016/j.najef.2023.1019.

[16] Bagane P, Mr M, Mr K, Mr B, Mr S, Sirbi K. Bibliometric Survey for Stock Market Prediction using Sentimental Analysis and LSTM. 2021 03;.

[17] T N A B T M Busu NAA S A Kamarudin, Mamat NAMG. Prediction of FTSE Bursa Malaysia KLCI Stock Market using LSTM Recurrent Neural Network. 2022;.