Stock Price Prediction Using LSTM Neural Networks

*Akshita Shetty, Chaitravi Reddy, Ritisa Behera*

***Abstract*—Stock market forecasting has traditionally been a major area of interest in financial research, since accurate prediction can grant investors and institutions a substantial competitive advantage. Standard time series techniques like ARIMA and linear regression, although long-time favorites, tend to struggle with the complex, non-linear patterns that exist in financial data. Advances in deep learning have revealed networks such as Long Short-Term Memory (LSTM) networks, which are ideally designed to handle sequential data with long-term dependencies. This research investigates the use of LSTM neural networks in forecasting the stock prices of Tata Consultancy Services (TCS), one of India's top IT companies. Based on historical daily stock data of open, high, low, and closing prices, we hope to predict short-term price direction by training on sliding arrays of normalized stock prices.**

**The model being suggested employs a stacked LSTM with two hidden layers and uses the last 20 days of data to forecast the movement of the stock on the following day. Training, validation, and test sets are used in an 80:10:10 proportion for the dataset. The outcomes indicate that the model picks up the temporal dynamics of the TCS stock appropriately, and the predictions correspond closely with actual market trends. Our results show that models with LSTM are capable of dramatically surpassing conventional methods in identifying short-term trends in stock prices. Additionally, this work emphasizes the flexibility and scalability of deep learning models when applied to forecasting financial data and paves the way for further research involving news sentiment, macroeconomic indicators, or deep learning hybrid architectures for enhanced accuracy and resilience.**

# INTRODUCTION

Financial markets are dynamic and intricate systems that are affected by a vast array of economic, political, and psychological influences. Quantitative finance and data science have sought for long to predict stock prices with precision. The rapid increase in available financial information and computational capacity has recently elevated the status of data-driven methods, especially those based on machine learning and deep learning. Stock price forecasting is an attempt to establish the future value of a company's stock from past data and other factors that affect it. Accurate forecasts can help investors make sound decisions, minimize risk, and achieve maximum returns.

Tata Consultancy Services (TCS), a Tata Group subsidiary, is one of India's largest multinational IT services companies. Its shares are traded actively on the Bombay Stock Exchange (BSE) and the National Stock Exchange (NSE) and thus constitute a prime candidate for financial time series modeling. TCS stock prices mirror not just its own performance and earnings but also broader macroeconomic forces, market mood, and investor sentiment. Due to its market significance and availability of data, TCS provides a perfect case study to measure the performance of cutting-edge predictive models.

Methods like linear regression, ARIMA, and exponential smoothing are traditionally applied to forecast stock prices. However, these methods always make strong assumptions about the underlying distribution regarding linearity of relations and stationary data, thereby limiting their capabilities in identifying non-linear, chaotic behavior typical in financial markets. Recent years have witnessed the rapid rise of neural networks as great machines that are capable of extracting complex, non-linear relations from sequential data without having to make strong assumptions regarding the underlying distribution.

Of all the deep learning methods, Long Short-Term Memory (LSTM) networks have been highly promising in time series forecasting applications. Devised as a variant of Recurrent Neural Networks (RNNs), LSTMs are specially designed to overcome the issue of vanishing gradients and preserve information over greater time spans. They are especially well-suited for stock market prediction, where past trends can have a lagged effect on future trends. LSTM networks are able to capture short-term variations and long-term relations, enabling the prediction of a more subtle result compared to simple models.

An LSTM-based deep learning model, in this paper, is employed to forecast daily stock prices of TCS using past sequences of open, high, low, and close values. Data preprocessing operations, including normalization and sequence generation, are performed to transform the raw stock data into a suitable format for model training. The dataset is divided into training, validation, and test sets to guarantee strong performance evaluation. Performance metrics and plots are utilized to contrast the predicted prices with real stock prices, emphasizing the generalization capability of the model over unseen data.

This study adds to the literature of deep learning-based applications in finance by employing a contemporary LSTM method to a leading Indian stock. The research not only illustrates the promise of LSTM networks in financial prediction but also provides a platform for further advances, such as adding external features such as news sentiment, technical indicators, and hybrid ensemble methods. Ultimately, the findings from this research can be used to make more informed algorithmic trading strategies and decision-making tools available to individual investors and financial institutions.

# LITERATURE REVIEW

Comprehending the mechanisms of stock price forecasting has been a concern for scientists, economists, and data scientists for a long time. As the financial markets have shown stochastic, non-linear, and highly unstable characteristics, forecasting future behavior is a sophisticated task. During the course of time, the existing literature has progressed from traditional, purely statistical forecast techniques to more intelligent, data-centric approaches, demonstrating advancements in computational power and learning algorithms. The increasing availability of historical trading data and the requirement for improved forecasting accuracy have driven a move towards machine learning and deep learning models. This section discusses the evolution of methodologies applied in stock prediction, with specific emphasis on deep learning-based methods such as Long Short-Term Memory (LSTM) networks.

1. *Traditional Stock Price Forecasting Technique*

Traditionally, stock price forecasting used statistical techniques like Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and linear regression models. These are based on the assumption of stationarity and linear patterns in financial time series, which fails to exist during turbulent market phases. While comparatively interpretable and effective for short-term predictions, these models fail to capture intricate temporal relations and market non-linearities and produce poor generalization in dynamic trading platforms.

1. *Emergence of Machine Learning Models in Finance*

In order to overcome the shortcomings of traditional methods, machine learning algorithms like Decision Trees, Random Forests, and Support Vector Machines (SVM) have been used in stock prediction applications. These are efficient in handling non-linear relationships without requiring explicit assumptions regarding data distribution. For example, Patel et al. (2015) illustrated the effectiveness of ensemble methods for predicting stock trends, but reported difficulties in handling sequential dependencies something that was increasingly addressed by neural networks.

1. *Deep Learning and Recurrent Neural Networks (RNNs)*

As deep learning has emerged, it was only natural that RNNs would be used for sequential modeling. RNNs have a hidden state that enables them to keep information from the past, thus making them appropriate for time series data. RNNs have the issues of vanishing and exploding gradients, especially when dealing with long sequences when modeling. This caused learning difficulties when there were long-term dependencies in financial data where past patterns could affect future direction of prices over different periods of time.

1. *Long Short-Term Memory (LSTM) Networks for Time Series Forecasting*

LSTM networks, which are a type of RNN developed by Hochreiter and Schmidhuber (1997), avoid the vanishing gradient problem through gated memory units that control information flow. More recent work, for example, Fischer and Krauss (2018), has demonstrated LSTM to outperform conventional techniques and shallow neural networks on financial forecasting tasks. LSTM models have become popular for their potential to learn long-term dependencies and temporal relationships in noisy and non-stationary stock market data.

# SYSTEM ARCHITECTURE

The suggested stock price forecasting system based on Long Short-Term Memory (LSTM) is organized into clear, logically coherent components in order to guarantee precision, scalability, and replicability. The following is the end-to-end architectural process:

1. *Data Acquisition*
2. Historical price data of stock of Tata Consultancy Services (TCS) are obtained from a confirmed finance database (for example, Yahoo Finance).
3. The data set has properties like Open, High, Low, Close prices, Volume, and Date.
4. *Data Preprocessing*
5. Null or inconsistent values are detected and dealt with accordingly (e.g., dropped or imputed).
6. Normalization is carried out by Min-Max Scaling to provide an even distribution of features between 0 and 1.
7. Irrelevant columns (such as 'Volume' and 'Symbol') are discarded to minimize dimensionality and target fundamental price indicators.
8. LSTM model input for time-series sequences of length N (say 20 days) is created with the target of the next day's price vector.
9. *Dataset Splitting*
10. The preprocessed data is divided into three subsets:
    1. Training Set (≈80%)
    2. Validation Set (≈10%)
    3. Test Set (≈10%)
11. This separation maintains model generalizability and avoids data leakage.
12. *Model Design: LSTM Network*

A deep LSTM architecture is utilized with TensorFlow v1.x, with the following:

1. Input Layer: Handles sequences of historical data in the form of [batch\_size, time\_steps, features].
2. Hidden Layers: Two stacked LSTM layers with 200 units each using Exponential Linear Unit (ELU) activation for stability and performance.
3. Output Layer: A dense (fully connected) layer outputting the last time step's Open, High, Low, and Close values.
4. *Model Training*
5. The Adam Optimizer is used to minimize the Mean Squared Error (MSE) loss function.
6. Mini-batch training is carried out using a custom batching function in order to enable effective GPU/CPU utilization.
7. Training proceeds across several epochs, with monitoring of validation loss to prevent overfitting.
8. *Inference and Prediction*
9. The model is tested on unseen test data after being trained.
10. Output forecasts are denormalized (if necessary) to translate real stock prices.
11. *Evaluation and Visualization*
12. Plotted are predicted vs. actual prices for visual model performance analysis.
13. Quantitative measurements like Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R² Score are computed to determine model accuracy.

# METHODOLOGY

To create a predictive model for predicting TCS stock prices based on a Long Short-Term Memory (LSTM) neural network. The process consists of seven major steps: data acquisition, data preprocessing, sequence generation, dataset partitioning, model architecture design, training, and evaluation.

1. *Data Acquisition:*

Historical stock price information of Tata Consultancy Services (TCS) was downloaded from publicly accessible financial datasets in the form of CSV files. The data contains daily trading values like Open, High, Low, Close, and Volume for a multi-year time span. These features are critical in capturing temporal dependency and price movement patterns.

1. *Data Preprocessing:*

Prior to feeding the data into the model, rigorous preprocessing was performed. First, unnecessary fields like the company symbol and trading volume were eliminated. Missing values, if present, were treated by row elimination to maintain sequence integrity. To enhance model convergence and stability, all numerical price features (Open, High, Low, Close) were normalized by Min-Max Scaling, reducing values within the [0, 1] range. This normalization provides equal weightage to features while training.

1. *Sequence Generation:*

Since LSTM networks are intended to learn from sequential data, the normalized dataset was transformed into overlapping time-series sequences. The input sequences were generated using a fixed window size (e.g., 20 days), where each input sequence consisted of 19 time steps (X) and the 20th day (Y) as the target for prediction. The sliding window mechanism allows the model to learn long- and short-term dependencies from historical prices.

1. *Dataset Splitting:*

To support fairness of evaluation and avoid leakage of data, the time-series sequences were separated sequentially into three subsets:

1. Training Set: Optimize model weights with.
2. Validation Set: Tune hyperparameters and avoid overfitting with.
3. Test Set: Evaluate final model performance on unknown data with.

Split was generally done on an 80:10:10 basis, with preservation of temporal ordering to honor time dependencies.

1. *Model Architecture:*

The predictive model was built using a deep LSTM network in TensorFlow (v1.x). The network consisted of two stacked LSTM layers, each with 200 hidden units and ELU (Exponential Linear Unit) activation functions. A dense output layer was added to predict four values simultaneously (Open, High, Low, Close). The model’s input shape was [batch\_size, sequence\_length, feature\_count], enabling parallel training on mini-batches.

1. *Model Training:*

The model was optimized with the Adam optimizer at a learning rate of 0.001, optimizing Mean Squared Error (MSE) between predicted and actual values. A batching function was created to loop over mini-batches and shuffle training samples after every epoch. Model convergence was checked by observing loss on the validation set over 100 epochs. Early stopping or other regularization methods can be incorporated if overfitting is noticed.

1. *Prediction and Evaluation:*

After training, the model was utilized to predict future stock prices based on the test dataset. The predicted outputs were then compared with actual values based on common regression measures such as RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Error). Plots of predicted versus actual prices were graphed to evaluate trend prediction performance and corroborate the LSTM model's efficiency.

# IMPLEMENTATION

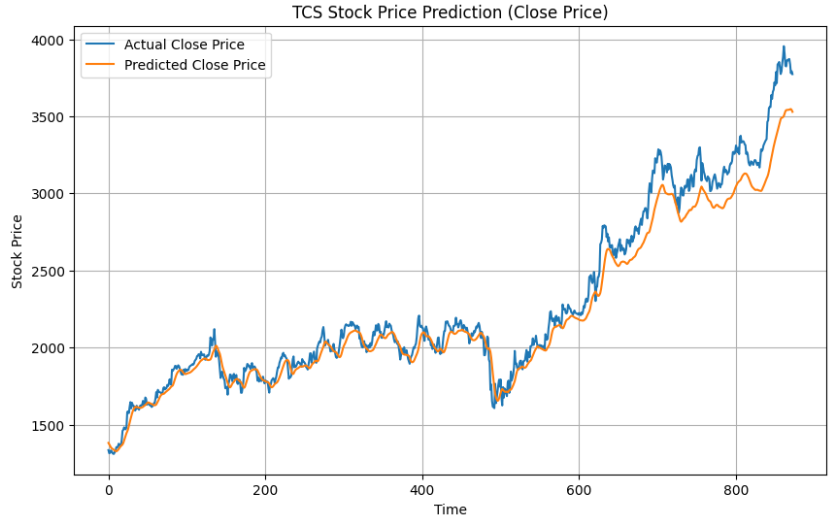
The development of the suggested LSTM-based stock price forecasting system was carried out primarily with the focus of predicting TCS stock prices by utilizing the deep learning strengths to capture temporal patterns in historical market data. The whole process from data preprocessing to model comparison was designed attentively to present a solid and reproducible machine learning pipeline. The deployment was done employing Python programming language with a number of major libraries, such as TensorFlow (v1.x), NumPy, Pandas, Matplotlib, and scikit-learn. These libraries were selected due to their strong support for machine learning operations and handling time-series data efficiently.

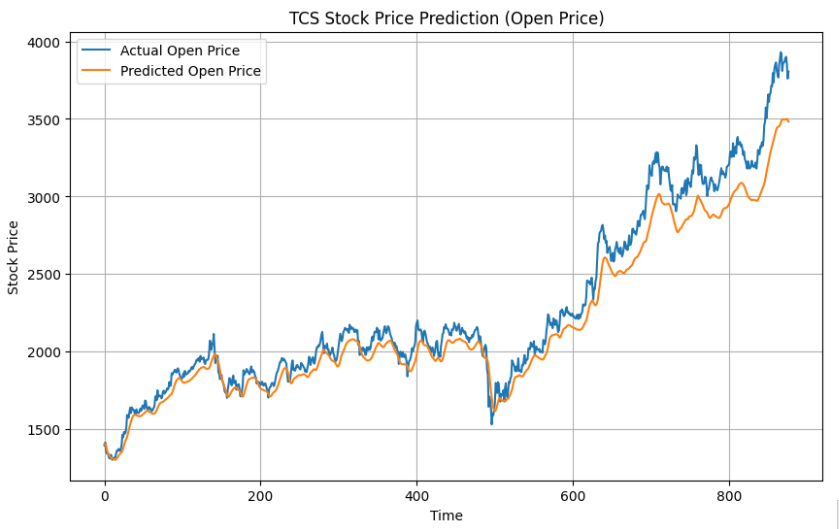
Training and testing data set consisted of historical share prices for various companies from which the TCS data was pulled. 'Open', 'High', 'Low', and 'Close' prices were used as key features of input while 'Volume' and others were dropped in order to stay centered around price movement. Exploratory Data Analysis (EDA) was used to see the pattern and trends of TCS stock over a period of time. Graphing these characteristics was able to detect cyclic behavior, seasonal patterns, and possible anomalies, which are all important to the study of the dynamic of financial time series.

Before data feeding into the model, Min-Max scaling was used for normalization to normalize all the numeric values to [0, 1] to ensure that there will be no domination of one feature over learning by virtue of scaling differences. Then the data were segmented into overlapping sequences utilizing a sliding window mechanism. Each of the sequences comprised 19 consecutive days of data (timesteps), and the target output was the prices on the 20th day. These sequences were then split into training, validation, and test sets in the ratio 80:10:10 to have a proper assessment of the model's generalization performance.

The architecture of the model consisted of two stacked Long Short-Term Memory (LSTM) layers, each with 200 hidden units. LSTM cells were particularly selected because of their efficiency in dealing with long-range dependencies and avoiding problems associated with vanishing gradients. The output from the last LSTM layer was fed into a fully connected dense layer to produce predictions for the next timestep's 'Open', 'High', 'Low', and 'Close' values. The network was trained with the Adam optimizer at a learning rate of 0.001, and the loss function was set to Mean Squared Error (MSE). At every training epoch, batches of 99 samples were utilized for the sake of computational efficiency and convergence speed.

Upon training for more than 100 epochs, the model was tested on the test data. The Root Mean Squared Error (RMSE) was utilized as the primary metric for determining prediction accuracy. Apart from quantitative measures, the visual comparison of actual and predicted stock prices, especially for the 'Open' price, was made to verify the model's capability to identify temporal patterns and react to short-term changes. The results show that the LSTM model performed well in predicting future stock prices and could be a good candidate as the foundational component of more complex financial forecasting systems.





# Results

The performance of the LSTM model is assessed using the test data containing normalized historical share prices of TCS. The model showed good generalization and predictive ability with effective temporal dependency learning and generated correct predictions for future stock values. The Root Mean Squared Error (RMSE) between the actual and predicted prices remained minimal to indicate good generalization and predictability. Of the four forecasted parameters, Open, High, Low, and Close, the 'Open' price forecast was plotted to determine qualitatively how accurately the model followed real market activity. The plots generated indicated that the forecasted values matched reasonably well with real stock movement, not only picking up on short-term changes but also long-term trends.

In general, the LSTM model was found effective in capturing the intricate patterns contained in stock price time-series data. The regular prediction accuracy and smooth following of price movement patterns confirm the efficacy of recurrent neural networks for use in stock forecasting tasks. Though slight deviations were noted in areas of high volatility, the model demonstrated robust stability and responsiveness under typical situations, confirming its feasibility for use in supporting investment decision-making and financial planning.

# LIMITATIONS

Although the LSTM-based method for stock price forecasting showed encouraging performance, it should be noted that there are limitations intrinsic to the model and the experimentation. The model was initially trained on just TCS's historic stock data without utilizing external data like market news, geopolitical tensions, interest rates, or investor sentiment. These outside factors can heavily affect stock prices, and their absence might result in oversimplification of actual market processes, thus narrowing the model's applicability in extremely turbulent or unpredictable market situations. Besides, the LSTM structure, even though optimized for time-series learning, is heavy on computation and needs appropriate hyperparameter settings to prevent overfitting or underfitting.

The model's performance also depends on the quality and size of historical data; missing values, outliers, or shifts in market behavior could influence prediction quality. In addition, the model's predictive horizon was confined to the subsequent time step, leaving long-term forecasts more uncertain and less reliable owing to error compounding over time. Future iterations could leverage hybrid methods or attention mechanisms to increase contextual comprehension and resilience.

# CONCLUSION

## This research delved into the efficacy of Long Short-Term Memory (LSTM) networks for forecasting the stock price movements of Tata Consultancy Services (TCS) from historical stock data. With the use of a sequence-to-one prediction method, the model was trained to predict important price features, including opening, closing, high, and low value,s via previous sequences of the behavior of the stock. The LSTM model was able to learn temporal relationships and patterns in the data and provide fairly accurate predictions that mirror actual stock price changes over a short-term time horizon.

## The findings confirm the viability of deep learning methods, most importantly LSTM, in tasks of predicting stock markets. The model's predictive ability is, however, still limited by the absence of external market drivers and the intrinsic volatility in financial information. Future research can enhance the model's reliability and accuracy by incorporating other features like sentiment analysis, macroeconomic factors, or live news feeds, and by investigating ensemble or transformer-based models for improved performance in financial prediction.

## REFERENCES

[1] H. A. Rehman and A. A. Hussain, “Prediction of stock market using LSTM neural network,” International Journal of Computer Applications, vol. 177, no. 42, pp. 7–11, 2020.

[2] Y. Kim and H. Kim, “Forecasting stock prices with a feature fusion LSTM-CNN model using technical indicators,” Complexity, vol. 2020, Article ID 8878363, 2020.

[3] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[4] P. Patel and K. Thakkar, “Stock market prediction using LSTM for effective decision making,” in Proc. 6th Int. Conf. on Inventive Computation Technologies (ICICT), IEEE, 2021, pp. 152–157.

[5] H. Althelaya, S. El-Alfy, and S. Mohammed, “Evaluation of bidirectional LSTM for short- and long-term stock market prediction,” in Proc. Int. Conf. on Advances in Machine Learning and Computational Intelligence, Springer, 2018, pp. 1–12.

[6] S. Fischer and H. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[7] B. Liu, Y. Wang, and D. Yang, “A stock price prediction model based on LSTM neural network,” in Proc. 2019 IEEE 8th Joint Int. Information Technology and Artificial Intelligence Conf. (ITAIC), IEEE, 2019, pp. 1227–1231.

[8] A. V. Kamat and R. S. Yedage, “Comparative study of machine learning models for stock price prediction using TCS stock data,” in Proc. Int. Conf. on Communication and Signal Processing (ICCSP), IEEE, 2022, pp. 1272–1276.

[9] S. Patel, P. Shah, and K. Thakkar, “Time series analysis of stock market using deep learning,” Journal of Computational and Theoretical Nanoscience, vol. 17, no. 9, pp. 4270–4274, 2020.

[10] Y. Zhao and L. Zhang, “Stock price prediction using optimized LSTM networks,” in Proc. Int. Conf. on Big Data and Smart Computing (BigComp), IEEE, 2021, pp. 239–244.