Stock Market Prediction using LSTM model

line 1: 1st Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 3rd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 4th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 5th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 6th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

Abstract*—The rapid advancement in artificial intelligence and machine learning techniques, availability of large-scale data, and increased computational capabilities of the machine opens the door to develop sophisticated methods in predicting stock price. In the meantime, easy access to investment opportunities has made the stock market more complex and volatile than ever. The world is looking for an accurate and reliable predictive model which can capture the market's highly volatile and nonlinear behavior in a holistic framework. This study uses a long short-term memory (LSTM), a particular neural network architecture, to predict the next-day closing price of the TATAMOTORS. A well-balanced combination of nine predictors is carefully constructed under the umbrella of the fundamental market data, macroeconomic data, and technical indicators to capture the behavior of the stock market in a broader sense. Single layer LSTM models are developed using the chosen input variables, and their performances are compared using standard assessment metrics–Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Correlation Coefficient (R). The experimental results show that the single layer LSTM model provides a superior fit and high prediction accuracy compared to multilayer LSTM models.

***Keywords*—Stock market, LSTM, Prediction, Machine learning**

I. Introduction

The stock price fluctuations are uncertain, and there are many interconnected reasons behind the scene for such behavior. The possible cause could be the global economic data, changes in the unemployment rate, monetary policies of influencing countries, immigration policies, natural disasters, public health conditions, and several others. All the stock market stakeholders aim to make higher profits and reduce the risks from the thorough market evaluation. The major challenge is gathering the multifaceted information, putting them together into one basket, and constructing a reliable model for accurate predictions.

Stock price prediction is a complex and challenging task for companies, investors, and equity traders to predict future returns. Stock markets are naturally noisy, non-parametric, non-linear, and deterministic chaotic systems (Ahangar, Yahyazadehfar, & Pournaghshband, 2010). It creates a challenge to effectively and efficiently predict the future price. Feature selection from the financial data is another difficult task in the stock prediction for which many approaches have been suggested (Hoseinzade & Haratizadeh, 2019). There has been a trend in which some researchers use only technical indicators, whereas others use historical data (Di Persio and Honchar, 2016, Kara et al., 2011, Nelson et al., 2017, Patel et al., 2015, Qiu and Song, 2016, Wang and Kim, 2018). The performance of the predictive model may not be top-notch due to the use of limited features. On the flip side, if all the available features from the financial market are included, the model could be complex and difficult to interpret. In addition, the model performance may be worse due to collinearity among multiple variables.

A proper model developed with an optimal set of attributes can predict stock price reasonably well and better inform the market situation. A plethora of research has been published to study how certain variables correlate with stock price behavior. A varying degree of success is seen concerning the accuracy and robustness of the models. One possible reason for not achieving the expected outcome could be in the variable selection process. There is a greater chance that the developed model performs reasonably better if a good combination of features is considered. One of the contributions of this study is selecting the variables by looking meticulously at multiple aspects of the economy and their potential impact in the broader markets. Moreover, a detailed justification is supplied why the specific explanatory variables are chosen in the present context in Section 4.

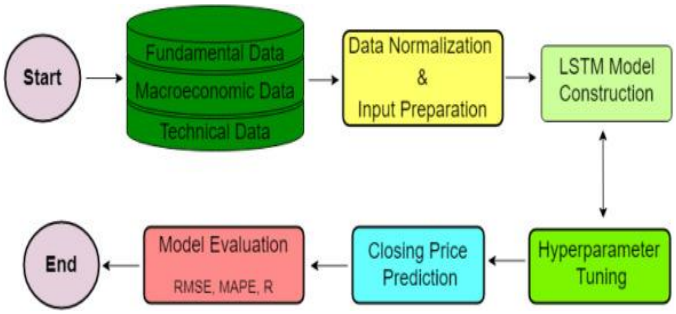
The field of quantitative analysis in finance has a long history. Several models ranging from naive to complex have been developed so far to find the solution to financial problems. However, not all quantitative analyses or models are fully accepted or widely used. One of the first attempts was made in the seventies by two British statisticians, Box and Jenkins, using mainframe computers (Hansen, McDonald, & Nelson, 1999). They developed the Auto-Regressive Integrated Moving Average (ARIMA) model utilizing only the historical data of price and volume. The ARIMA is used to handle only stationary time series data by default. Performance can be abysmal if it is used for non-stationary data. Therefore, it is essential to convert non-stationary time series data to stationary before implementation, which may lose the original structure and interpretability of the feature. With very few exceptions, almost all classical models assume that data has a linear relationship. This assumption vividly raises the questions about the robustness of the classical time series models as the real-world time series data are often nonlinear.

Things were getting more interesting from the eighties because of the development in data analysis tools and techniques. For instance, the spreadsheet was invented to model financial performance, automated data collection became a reality, and improvements in computing power helped predictive models to analyze the data quickly and efficiently. Because of the availability of large-scale data, advancement in technology, and inherent problem associated with the classical time series models, researchers started to build models by unlocking the power of artificial neural networks and deep learning techniques in the area of sequential data modeling and forecasting. These methods are capable of learning complex and non-linear relationships compared to traditional methods. They are more efficient in extracting the most important information from the given input variables.

Several deep learning architectures have been developed to deal with various problems and the intrinsic structure of datasets. Information flows only in the forward direction in a basic feedforward neural network architecture. Since each input is processed independently, it does not retain information from the previous step. Thus, these models are ineffective in dealing with sequential data where series of prior events are essential in predicting future events. Recurrent neural networks (RNN) are designed to perform such tasks. The RNN architecture consists of loops, allowing relevant information to persist over time. Information is being passed from one timestep to the next internally within the network. Therefore, the RNN is more suitable for sequential data modeling and time series applications such as stock market predictions, language translations, auto-completion in messages/emails, and signal processing. During the training process of the RNN, the cost or error is calculated between the predicted values and the actual values from a labeled training dataset. The error is minimized by repeatedly updating the networks' parameters (weights and biases) until the lowest possible value is obtained. The training process utilizes a gradient, the rate at which cost changes with respect to each parameter. The gradient provides a direction to move in the error surface by adjusting the parameters iteratively. This strategy is called backpropagation, where the error is propagated backward from the output layer all the way up to the input layer. One of the challenges of this technique is that parameters can be anywhere in the networks, and finding a gradient involves calculations of partial derivatives with respect to all the parameters. This process sometimes needs a long chain rule, especially for the parameters in earlier layers of the networks. As a result, gradients could ultimately vanish or decay back through the networks, known as the vanishing gradient problem, a common issue in neural networks training. Unfortunately, this problem also persists in the RNN architecture. LSTM, a typical recurrent neural network architecture, is designed to overcome the vanishing gradient problem (Hochreiter, 1998). Memorizing information for a longer period of time is the default behavior of the LSTM model.

This study considers the computational framework to predict the stock index price using the LSTM model, the improved version of neural networks architecture for time series data. The bird's-eye view of the proposed research framework via the schematic diagram is expressed in Fig. 1. As outlined in the diagram, the proposed study utilizes the carefully selected features from fundamental, macroeconomic, and technical data to build the model. After that, the collected data has been normalized using the min–max normalization technique. Then input sequence for the LSTM model is created using a specific time step. The hyperparameters such as number of neurons, epochs, learning rate, batch size, and time step have been incorporated in the model. The regularization techniques have been utilized to overcome the over-fitting problems. Once the hyperparameters are tuned, the input data is fed into the LSTM model to predict the closing price of the stock market index. The quality of the proposed model is assessed through RMSE, MAPE, and R.

In a nutshell, plenty of research has been done in predicting the stock market. Some research focuses on complex statistical or machine learning techniques without focusing on the type of attributable variables. Others use only the fundamental data without exploring additional factors that could influence the stock market prediction. There is a need to develop a model with a good combination of features of the stock market variables and simplicity in model architecture. Thus, our contribution is to create a model without adding any complexity in model architecture and maintaining well-balanced set of variables to capture the behavior of the stock market from multiple



The rest of the paper is organized as follows. Section 2 explains the related work in this field. Section 3 explores the implementation of the LSTM model in the TATAMOTORS data. The data collection and feature selection procedure are explained in Section 4. Model outcomes are discussed in Section 5. It also explains the predictive capability of the model after tuning the hyperparameters. Finally, Section 6 presents the conclusion and future work, followed by acknowledgments, list of references, and appendix.

II. RELATED WORK

Chen et al. used the LSTM model to predict China stock returns (Chen, Zhou, & Dai, 2015). The historical data was transformed into 30-days long sequences with ten learning features and 3-day learning rate labeling. The authors claimed that the model improved the accuracy from 14.3% to 27.2% compared to the random prediction method. Bao et al. applied the Haar wavelet transformation to denoise the financial time series data and implemented the stacked autoencoders to learn the deep features of the data and then used LSTM to predict the closing price of stock indices (Bao, Yue, & Rao, 2017). Their average R score was below 88% on the LSTM model for S&P 500. Roondiwala et al. used the LSTM model for the NIFTY 50 data ranges from 2011 to 2016 (Roondiwala, Patel, & Varma, 2017). The authors used fundamental data (open, close, low, and high) without incorporating macroeconomic and technical indicators to predict the closing price.

Fischer and Krauss used LSTM networks for the classification problem of predicting directional movements for the constituent stocks of S&P 500 from 1992 until 2015 (Fischer & Krauss, 2018). The authors concluded that the LSTM network could effectively extract meaningful information from the financial time series data. Based on prediction accuracy and daily returns after transaction costs, LSTM outperforms random forests, standard deep networks, and logistic regression. Qiu et al. implemented LSTM based model on historical data of S&P 500, Dow Jones Industrial Average (DJIA), and Hang Seng Index dataset (Qiu, Wang, & Zhou, 2020). They applied an attention mechanism extracting the information in the news to evaluate the price fluctuation. The authors denoised the data using wavelet

transformation and then implemented their attention-based LSTM framework to predict the opening index price using only fundamental market data. Lanbouri and Achchab used the LSTM model for the high-frequency trading perspective in which their goal was to use the S&P 500 stock trading data to predict the stock price in the next 1, 5, and 10 minutes (Lanbouri & Achchab, 2020). Yadav et al. implemented LSTM model with various hidden layers to Indian stock market data removing the trend and seasonality components to predict the closing price (Yadav, Jha, & Sharan, 2020). They concluded that the single layer LSTM model had better prediction accuracy.

Kara et al. used support vector machine (SVM) and artificial neural network to predict movement in the daily Istanbul Stock Exchange National 100 Index from 1997 to 2007 (Kara et al., 2011). The authors selected ten technical indicators as input for their model. Experimental results showed that the average performance of the artificial neural network model was significantly better than that of the SVM model. Karmiani et al. compared LSTM, Backpropagation, SVM, and Kalman filter to predict stock price (Karmiani, Kazi, Nambisan, Shah, & Kamble, 2019). The stock price of selected nine companies were considered for the prediction. LSTM was the best choice in terms of prediction accuracy with low variance. Yu and Yan combined phase-space reconstruction method for time series analysis and LSTM model to predict the stock price (Yu & Yan, 2019). Various market environments such as the S&P 500, DJIA, Nikkei 225, Hang Seng Index, China Securities Index 300, and ChiNext index were considered for the analysis. This prediction model was built by taking the historical price only. The outcomes of LSTM with Multilayer Perception, Support Vector Regressor, and ARIMA were compared. Authors claimed that the LSTM model outperformed other models for S&P 500 data. Gao et al. conducted a comparative study of four machine learning algorithms —Multilayer Perceptron, LSTM, Convolutional Neural Network, and Uncertainty-Aware Attention —to predict the next day's stock price (Gao, Zhang, & Yang, 2020). The S&P 500 index, CSI 300 index, and Nikkei 225 index were taken to represent the most developed market, the less developed market, and the developing market. Open price, close price, trading volume, Moving Average Convergence Divergence, Average True Range, exchange rate, and interest rate were considered predictors. The outcome of the study suggested that Uncertainty-Aware Attention's performance was slightly better than other models. Moreover, additional predictors such as the volatility index and the unemployment rate could improve the model's performance.

III. MODELING APPROACH

A. *A brief overview of LSTM*

LSTM is a popular deep learning technique in RNN for time series prediction. For example, LSTM is used for both classification and regression problems not only for the stock market prediction but the rainfall runoff modeling (Kratzert, Klotz, Brenner, Schulz, & Herrnegger, 2018), fMRI data analysis (Rahman et al., 2020), anomaly detection (Lindemann, Maschler, Sahlab, & Weyrich, 2021), mobile traffic prediction (Trinh, Giupponi, & Dini, 2018) to name a few. Although standard RNN is superior to traditional networks in preserving the information, it is not effective in learning long-term dependencies due to the vanishing gradient problem (Hochreiter, 1998). LSTM uses memory cells to overcome the issue of vanishing gradients. It consists of an input layer, a hidden layer, a cell state, and an output layer (Gers et al., 2000, Gers et al., 2003, Hochreiter and Schmidhuber, 1997). The key component of LSTM architecture is the cell state which runs through the chain, with only linear interaction, keeping information flow unchanged. The gate mechanism of LSTM deletes or modifies the information of the cell state. It is a way to pass the information selectively that consists of the sigmoid layer, hyperbolic tangent layer, and the point-wise multiplication operation.

Fig. 2 illustrates the architecture of LSTM at time t which is designed to model sequential input. In particular, four gates —output, change, input, and forget —are shown with their operations at time t.

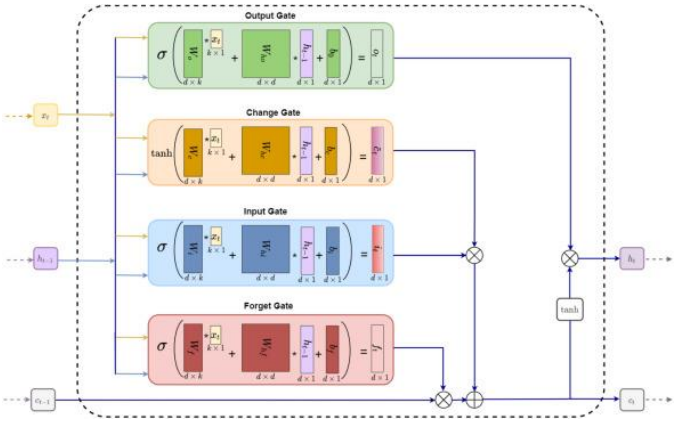


Fig. 2. Long short-term memory (LSTM) architecture.

LSTM cell takes 3 different pieces of information: the current input sequence xt, the short-term memory from the previous cell $ht-1$, and the long-term memory from the previous cell state $ct-1$ at time t. The forget gate takes the information from xt and $ht-1$ and produces the output between 0 and 1 through the sigmoid layer and then it identifies which information to discard from the previous cell state $ct-1$. When the value is 1, it stores all the information into the cell while with a value of 0, it forgets all the informatioType equation here.n from the previous cell state. Similarly, the input gate identifies which information to be updated from the change gate. The output gate decides which information to be taken as an output from the present cell state.

B. *Algorithm*

1. Input: Training data (X_train, Y_train), hyperparameters (units, epochs, batch size, dropout rates).
2. Initialize Sequential model.
3. Add LSTM layers with specified units and activation (ReLU).
4. Apply Dropout after each LSTM layer.
5. Add a Dense layer with one unit for the final prediction.
6. Compile the model with Adam optimizer and Mean Squared Error loss function.
7. Train the model for 50 epochs with a batch size of 32.
8. Output: Trained model capable of predicting time series values.

C. *Dataset Preparation*

In this study, the stock price dataset is prepared and modeled to train a machine learning model for predictive analysis. Two moving averages, computed over 100 and 200 days, are included to capture long-term trends and reduce the impact of short-term fluctuations in the stock's closing price. These features can help in understanding the general

momentum and smoothing out volatility in the data. After calculating the moving averages, the dataset is split into two parts: 80% is reserved for training the model, and the remaining 20% is kept for testing the model's performance.

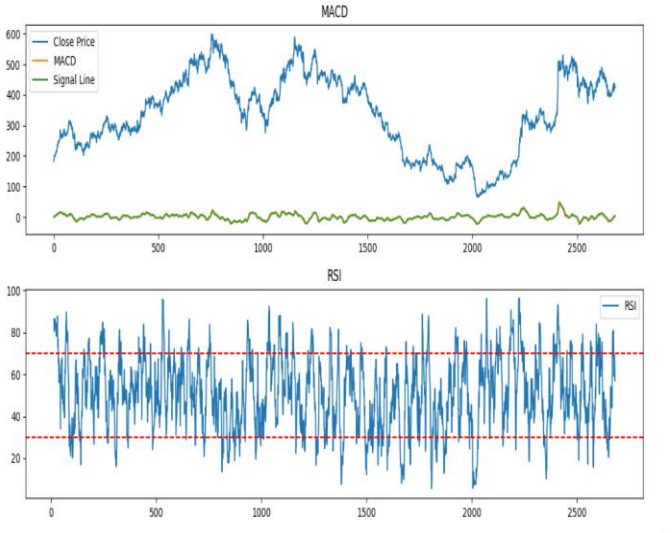
Price	Date	Adj Close	Close
Ticker		TATAMOTORS.NS	TATAMOTORS.NS
0	2012-01-02 00:00:00+00:00	175.647980	181.997894
1	2012-01-03 00:00:00+00:00	185.196655	191.891769
2	2012-01-04 00:00:00+00:00	192.119446	199.064835
3	2012-01-05 00:00:00+00:00	192.978821	199.955276
4	2012-01-06 00:00:00+00:00	194.411102	201.439362
...	...	...	...
2678	2022-11-14 00:00:00+00:00	431.022400	433.700012
2679	2022-11-15 00:00:00+00:00	434.451080	437.149994
2680	2022-11-16 00:00:00+00:00	428.885651	431.549988
2681	2022-11-17 00:00:00+00:00	420.537506	423.149994
2682	2022-11-18 00:00:00+00:00	421.183472	423.799988

2683 rows × 7 columns

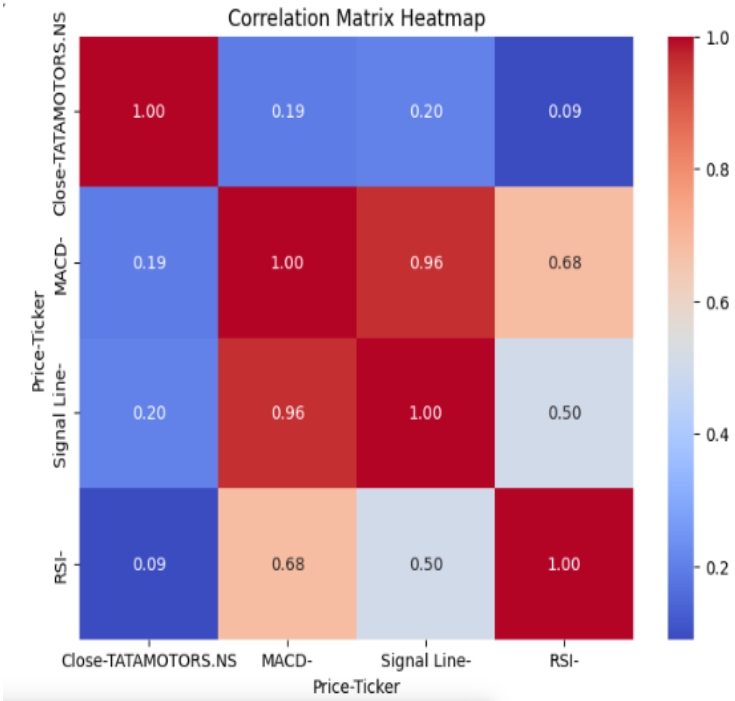
To normalize the data and facilitate efficient training, the training data is scaled to a range of [0,1] using the MinMaxScaler. This ensures that all input features are on a comparable scale, avoiding dominance by features with larger magnitudes and improving convergence during training. A sliding window technique is used for sequence modeling, where the scaled data is structured into sequences of 100 consecutive closing prices as input features (x) and the corresponding next day's closing price as the target output (y). This process creates a dataset suitable for time-series forecasting, allowing the model to learn temporal dependencies and patterns in the data. By dropping missing values resulting from the moving averages and preprocessing, the dataset becomes ready for input into machine learning models such as LSTM for sequential predictions. This approach effectively incorporates historical patterns to improve prediction accuracy.

Price	Adj Close	Close	High
Ticker	TATAMOTORS.NS	TATAMOTORS.NS	TATAMOTORS.NS
count	2683.000000	2683.000000	2683.000000
mean	334.009904	336.943550	342.306155
std	126.370914	126.865259	128.180599
min	64.896851	65.300003	66.900002
25%	244.009155	248.320290	252.495354
50%	339.055511	341.500000	347.250000
75%	434.873459	437.850006	445.335938
max	594.200989	598.134399	605.901123

Description of Dataset



For this dataset, the technical indicators such as the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and its signal line exhibit weak correlations with the target variable (closing prices). The correlation coefficients for these indicators are all below 0.5, indicating a low to negligible relationship between these features and the stock's closing price. This suggests that while these indicators may provide some insights into market trends and momentum, they do not strongly influence the prediction of the stock's closing price in this particular dataset. Therefore, relying solely on these features may not be sufficient for building a robust predictive model.



IV. Experiments and Results

A. Metrics

We implement single layer and multilayer LSTM architecture to predict the closing price. Within each of these models, several options are considered with different number of neurons. Prediction accuracy and reliability of these models are assessed by calculating three different performance metrics —RMSE, MAPE, and R. The analytical form of these metrics are defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2},$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_i}}{y_i}\right|,$$

$$R = \frac{\sum_{i=1}^{N}(y_i - \bar{y_i})(\widehat{y_i} - \bar{\widehat{y_i}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y_i})^2 (\widehat{y_i} - \bar{\widehat{y_i}})^2}}$$

where,

y_i
: Original time series,

$\bar{y_i}$
: The average value of the original time series,

$\widehat{y_i}$
: Predicted time series computed from the model,

$\bar{\widehat{y_i}}$
: Average value of the predicted time series,

N
: Number of observations.

Among three prediction metrics, RMSE measures the square root of the mean square error of the actual values and estimated values, MAPE estimates the size of the error computed as the relative average of the error, and R determines the linear correlation between actual and predicted values. Smaller the values of RMSE and MAPE, better the performance of the model. On the contrary, larger value of R indicates the similarity between predicted and actual series. Moreover, performance scores are calculated after applying the inverse transformation in the predictions obtained from the normalized data. Each model are executed multiple times independently in order to address the stochastic behavior. Average RMSE score obtained from these multiple replicates is considered as a primary model selection criteria followed by average MAPE and R scores. A model with the smallest RMSE and MAPE along with the greatest possible R would be considered as the best model.

B. RESULTS

Mean Squared Error (MSE):	559.80
Root Mean Squared Error (RMSE):	0.08
R-squared (R2):	0.95

The predictive model for Tata Motors stock prices demonstrates strong accuracy, with a Mean Squared Error (MSE) of 559.80 and a Root Mean Squared Error (RMSE) of 0.08, indicating minimal deviation between predicted and actual stock prices. Furthermore, an R-squared (R^2) value of 0.95 shows that the model explains 95% of the variance in the stock price data, highlighting its ability to capture underlying market trends effectively. These results suggest the model's robustness and potential utility for stock price forecasting and investment decision-making. However, further evaluation under diverse market conditions is recommended to validate its long-term reliability and adaptability.

The graph illustrates the performance of the predictive model for Tata Motors stock prices, with the predicted prices closely aligning with the original prices over time. The model captures major price trends and transitions effectively, as evidenced by the overlapping of the red (predicted prices) and green (original prices) lines. However, some deviations are noticeable, particularly during periods of sharp price changes, suggesting potential areas for refinement in capturing volatility. Overall, the model demonstrates strong predictive accuracy and reliability, reinforcing its suitability for stock price forecasting applications.

V. Conclusion

VI. References