# A Comprehensive Approach for Stock Price Prediction Using Technical Analysis Based LSTM Model

Dara Rajesh Babu[1], Prof. Bachala Sathyanarayana[2]

[1]Research Scholar, Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, Andhra Pradesh-515003, India.

[2]Professor, Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, Andhra Pradesh-515003, India.

**Abstract:** The rapid progress in artificial intelligence and machine learning, coupled with the availability of extensive data and enhanced computational capabilities, has paved the way for the development of sophisticated methods in predicting stock prices. Simultaneously, the easy accessibility to investment opportunities has added complexity and volatility to the stock market. There is a global quest for an accurate and reliable predictive model capable of comprehensively capturing the market's highly volatile and nonlinear behavior. This study introduces the use of long short-term memory (LSTM), a specific neural network architecture. In this paper, we propose the design and implementation of a stock price prediction algorithm that captures market behavior based on a well-balanced combination of nine predictors, carefully constructed using fundamental market data and technical indicators to assess the stock market's movement accurately. Regression models and LSTM models are developed with selected input variables, analyzing stock trends to determine optimal buying and selling points. The performance of these models is compared using standard assessment metrics—Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared error.

**Keywords:** Stock market**,** Regression, LSTM, Machine learning, Prediction, Deep learning.

**1. Introduction:** The stock exchange comprises various marketplaces and exchanges facilitating routine transactions involving the buying, selling, and issuing of publicly traded company shares. These financial transactions occur either through formal exchanges or over-the-counter markets, each adhering to a set of regulations. The primary objective of participating in stocks is to augment earnings by investing in shares of companies anticipated to generate profits, and stock prices are governed by the principles of demand and supply. The stock market is intricately connected to the economy, with key performance indicators heavily reliant on fluctuations in equity prices. A judicious price estimation can be derived with well-informed analysis.

Regression analysis [5] encompasses a collection of machine learning techniques enabling the prediction of a continuous outcome variable (y) using one or multiple predictor variables (x). In essence, the objective of a regression model is to formulate a mathematical equation that expresses y as a function of the x variables. Subsequently, the equation y = b0 + b*x + e can be employed to forecast the outcome (y) by inputting new values for the predictor variables (x).

Within Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) [1] stands as an algorithm employed for stock market price prediction. This algorithm incorporates a memory cell that integrates complex network gates involving operations such as tanh, addition (+), and subtraction (-) to facilitate predictions. These memory cells effectively store past values associated with the data, enhancing the model's ability to generate more accurate forecasts.

In this paper, we have formulated, modeled, and forecasted the stock returns of APPLE based on data gathered from Yahoo Finance. The dataset was subsequently utilized for both training and testing phases of the model. Section 2 delves into a comprehensive literature review of related works in this domain. Section 3 provides an in-depth exploration of the technical terms associated with each process. Section 4 outlines the design approach, while Section 5 presents results and discussions, incorporating graphical representations of the model. Finally, Section 6 encapsulates the conclusion and offers recommendations.

**2. Literature Review:** In the course of composing this research paper, extensive references have been consulted. Reference [1] employs LSTM to predict Apple stock prices over a 10-year horizon and scrutinizes day-to-day fluctuations. Reference [2] highlights LSTM's ability to rapidly process entire data series and introduces a memory cell for efficient remote linking of memories and feedback. Reference [3] centers on forecasting stock prices using the ARIMA model for a span of 2 years and employs Random Forest and LSTM for predicting the next day's stock price. Reference [4] implements LSTM and Generative Adversarial Network (GAN) for daily frequency stock price prediction. Additionally, it investigates the impact of the model parameter update cycle on stock forecast performance through rolling segmentation on both training and testing sets of raw data.

Reference [5] concentrates on two distinct methods: Regression and Classification. In the regression approach, the system anticipates the closing price of a company's stock, while in the classification method, it predicts whether the closing price will increase or decrease the next day. Reference [6] establishes a model utilizing LSTM to forecast China stock returns (Chen, Zhou, & Dai, 2015). The historical data is transformed into 30-day long sequences with ten learning features and a 3-day learning rate labeling. The authors assert that the model enhances accuracy from 14.3% to 27.2% compared to random prediction methods. Reference [7] by Roondiwala et al. discusses the LSTM model for NIFTY 50 data spanning from 2011 to 2016 (Roondiwala, Patel, & Varma, 2017). Fundamental data (open, close, low, and high) is utilized without incorporating macroeconomic and technical indicators to predict the closing price. Reference [8] by Fischer and Krauss explores LSTM networks for the classification problem, predicting directional movements for constituent stocks of S&P 500 from 1992 to 2015 (Fischer & Krauss, 2018). The authors conclude that LSTM networks effectively extract meaningful information from financial time series data, outperforming random forests, standard deep networks, and logistic regression based on prediction accuracy and daily returns after transaction costs. Reference [9] by Kara et al. suggests support vector machine (SVM) and artificial neural network methods to predict movement in the daily Istanbul Stock Exchange National 100 Index from 1997 to 2007 (Kara et al., 2011). The authors employ ten technical indicators as input for their model.

The experimental outcomes demonstrated that the artificial neural network model exhibited significantly superior performance compared to the SVM model. Karmiani et al. conducted a comparison of LSTM, Backpropagation, SVM, and Kalman filter for stock price prediction Reference [10]. The study focused on nine selected companies for prediction, with LSTM emerging as the optimal choice in terms of prediction accuracy and low variance and integrated a phase-space reconstruction method for time series analysis with an LSTM model to forecast stock prices.

**3. Scientific Terms:** In this study, we encompassed a range of scientific topics, delving into technical terms and the design approach. The section on technical terms and the design approach elucidates the mathematical and computer science foundations employed in constructing the model. These fields are intricately intertwined, emphasizing the crucial role of mathematics and statistics in deep learning, where diverse fundamental and advanced computer science concepts are addressed.

**3.1 Artificial Intelligence:** Artificial intelligence fundamentally involves simulating the human brain or intellect within a machine. The machine endeavors to solve problems and engage in reasoning similar to the human mind. Despite its intricate nature, this concept is actively applied across various quantitative fields, particularly in finance, for analytical purposes and effective problem-solving.

**3.2 Machine Learning:** Machine learning, commonly referred to as ML, is a field of study that empowers computers to learn without explicit programming, as articulated by Arthur Samuel, a prominent figure in the AI domain. ML encompasses a substantial amount of mathematics and applied statistics, which are utilized for learning and training the model. The techniques within machine learning include the following. Supervised Learning, Unsupervised Learning, Reinforcement learning. In this project, we leverage a supervised machine learning algorithm, specifically the training of a pre-informed dataset. This allows us to make inferences based on the algorithm's learning from the provided dataset. In supervised learning, the algorithm can evaluate itself against the intended output, refining its predictive capabilities. Machine learning, with its capacity to analyze vast amounts of data, delivers fast and accurate results. This capability enables us to discern between informative changes and potential risks, the former being insightful and the latter potentially dangerous. Constructing the model appropriately may require a certain amount of time.

**3.3 Regression:** Regression serves as a method to comprehend the correlation between independent variables or features and a dependent variable or outcome. Once the relationship between these variables is estimated, predictions for outcomes can be made. In the realm of statistics, regression is a pivotal component of forecast models within machine learning. It is employed as an approach to forecast continuous outcomes in predictive modeling, offering utility in predicting and forecasting outcomes from data. In machine learning regression, the typical practice involves plotting a line of best fit through the data points, minimizing the distance between each point and the line to achieve the optimal fit. Regression is instrumental in identifying patterns and relationships within datasets, which can then be extrapolated to new and unseen data. This establishes regression as a crucial element in machine learning applications within finance, often utilized for forecasting portfolio performance, stock costs, and trends. Models can be trained to discern the relationship between various features and the desired outcome.

**3.4 Neural Networks:** A subset of machine learning is neural networks, where a computer performs computations by intricately analyzing labeled examples. Modeled after the human brain, a neural network consists of millions of interconnected, simple processing nodes. Typically organized into layers, these networks are feed forward, meaning they operate in a unidirectional manner. Each node in a layer can connect to multiple nodes in the layer below and send/receive data to/from various nodes in the layer above. During the initial stages of neural network training, threshold values and weights are initialized to random values. The training data is inputted into the first layer (input layer) and passed through subsequent layers, undergoing complex combinations of addition and multiplication until reaching the output layer, where it undergoes radical changes. Throughout the training process, weights and threshold values are continually adjusted to ensure consistent outputs for training data with the same labels.

**3.4.1 Recurrent Neural Networks (RNN):** Recurrent Neural Networks (RNNs) are a type of neural network that enables the utilization of previous outputs as inputs while maintaining hidden states. The advantages of employing Recurrent Neural Networks include:

1. Accommodation of processing data inputs of variable lengths.
2. Inclusion of historical values in the prediction process.
3. Sharing of weights across different points in time.

**3.4.2 Long Short Term Memory (LSTM):** "LSTM," commonly referred to as Long Short-Term Memory Networks, represents a specialized variant of Recurrent Neural Networks (RNNs) known for their ability to capture long-term dependencies, as illustrated in Fig 1. Particularly effective across a diverse range of problems, LSTM finds widespread application in handling long-term dependency challenges. The unique design of LSTM allows it to remember and retain pertinent information over extended periods, preventing the risk of forgetting previously learned patterns. In contrast to standard recurrent neural networks, which have a basic repeating module, LSTM's distinctive architecture enhances its capacity for preserving and utilizing relevant information over time.

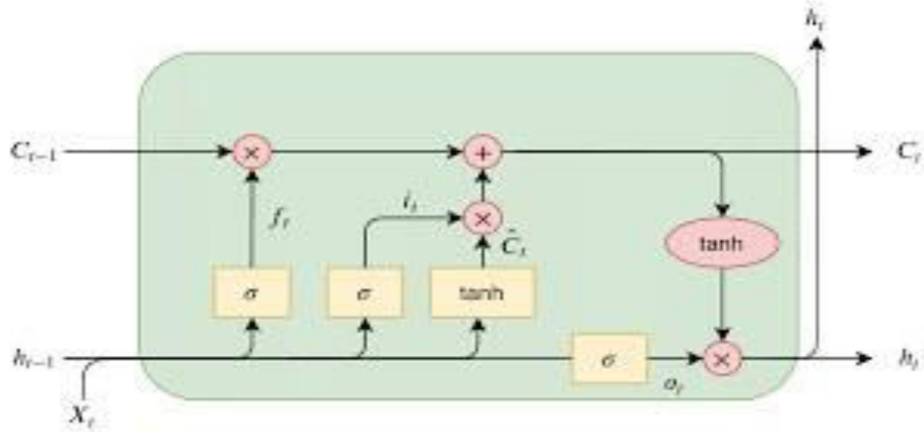


**Fig 1:** Long short-term memory (LSTM) architecture.

Fig. 1 depicts the LSTM architecture at time $t$, designed to model sequential input. Specifically, the illustration showcases four gates - output, change, input, and forget - along with their operations at time $t$. For a given input sequence $\{x_1, x_2, ..., x_n\}$, $x_t \in R^{k*1}$ is the input sequence at time $t$, the memory cell $ct$ undergoes updates through three gates: input gate $i_t$, forget gate $f_t$, and change gate $c_t$. The hidden state $h_t$ is then updated using the output gate $o_t$ and the memory cell $c_t$.

At time t, the respective gates and layers compute the following functions:

$f_t = \sigma(W_f\, x_t + W_{hf}\, h_{t-1} + b_f)$,
$i_t = \sigma(W_i x_t + W_{hi} h_{t-1} + b_i)$,
$o_t = \sigma(W_o x_t + W_{h0} h_{t-1} + b_0)$,˜
$c_t = \tanh(W_c x_t + W_{hc} h_{t-1} + b_c)$,
$h_t = o_t \times \tanh(c_t)$

Here, $\sigma$ and tanh represent the sigmoid and hyperbolic tangent functions, respectively. The operator × denotes the element-wise product. The weight matrices are denoted as $W \in R^{d*k}$, $W_h \in R^{d*d}$ and the bias vector as $b \in R^{d*1}$. Additionally, $n$, $k$, $d$ represent the sequence length, the number of features, and the hidden size, respectively. The LSTM cell incorporates three essential pieces of information: the current input sequence $x_t$, the short-term memory from the previous cell $h_{t-1}$, and the long-term memory from the previous cell state $c_{t-1}$at time $t$. The forget gate utilizes information from $x_t$ and $h_{t-1}$, producing an output between 0 and 1 through the sigmoid layer. It then determines which information to discard from the previous cell state $c_{t-1}$. A value of 1 signifies the retention of all information, while a value of 0 implies the omission of all information from the previous cell state. Similarly, the input gate identifies which information to update from the change gate. The output gate determines the information to be taken as the output from the current cell state.

### 3.4.3 Proposed Technical Analysis based LSTM structure:

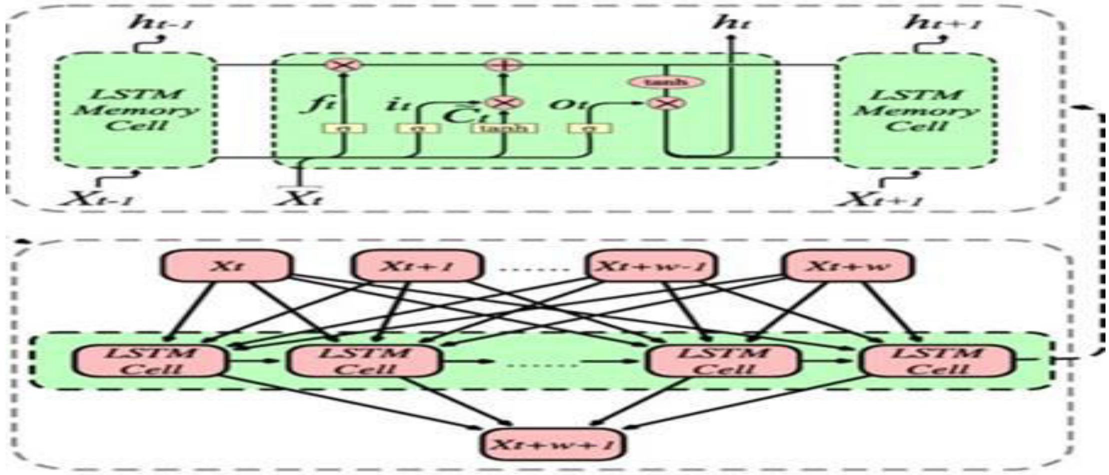


**Fig 2:** Technical Analysis based LSTM architecture.

Given a multivariate financial time series data collected from various sources, the objective of the proposed model is to predict the next day's closing price using a multivariate sequence of input features. The LSTM implementation procedures for this task are as follows:

Starting from the original dataset $X = (x_1, x_2, \ldots, x_n)$ of size $k \times n$, two sequences are created: $\{x_1, x_2, \ldots, x_{n-1}\}$ and $\{y_1, y, \ldots, y_{n-1}\}$, where $x_t \in R^{k*1}$ is the input sequence and $y_t \in \mathrm{R}$ is the next day's closing price at time t. Here, $k$ is the number of input features, and $n$ is the total number of observations.

To align with the required dimension of the LSTM architecture, the input sequence $x_t$ is formed by considering $m$ continuous sequence $x_t : x_{t+m-1}$, which results in a matrix of shape $k \times m$ for $t \in \{1, 2, \ldots, n - m - 1\}$.

The output $ht$ of the LSTM represents a feature representation for the input sequence $Xt$ at time $t$. Mathematically, $ht$ can be expressed as follows: $h_t = LS(X_t, h_{t-1}, c_{t-1}, w)$

Here, $w$ denotes all learnable parameters. As the final hidden state $h_f$ encapsulates the most information from the input sequence, it undergoes conversion to a vector using a dense layer.

LSTM excels at learning dependencies across extended time intervals, addressing the vanishing gradients problem encountered by conventional RNNs. This is achieved by replacing the ordinary neuron with a sophisticated architecture known as the LSTM unit or block. The functionality of LSTM-TA can be succinctly summarized through the following set of equations.

$$z_t = \tanh(W^z x_t + R^z h_{t-1} + b^z)$$
$$i_t = \sigma(W^i x_t + R^i h_{t-1} + b^i)$$
$$f_t = \sigma(W^f x_t + R^f h_{t-1} + b^f)$$
$$o_t = \sigma(W^o x_t + R^o h_{t-1} + b^o)$$
$$s_t = z_t \cdot i_t + s_{t-1} \cdot f_t$$

$h_t = \tanh(s_t) \cdot o_t$

Here, $i_t$ denotes the input gate, $o_t$ denotes the output gate, $f_t$ denotes the forget gate, $s_t$ denotes the memory cell, and $h_t$ denotes the hidden state. The sigmoid σ and hyperbolic tangent (tanh) functions are defined in equations 2 and 3, respectively.

$\sigma(z) = \frac{1}{1+e^{-z}}$

$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

**4 Design Approach:** To forecast the price of APPLE stock, both Regression and LSTM models were implemented using the Keras framework. These models were trained on the Apple dataset, which contains information about the stock's price. Visualization was performed on the close price data with corresponding dates, providing an overview of the forecasted stock price in relation to its current value.

**4.1 Software and Libraries used:** The project's hardware prerequisites encompass a Windows 10 operating system with an i3 processor. The software suite incorporates Python (version 3.7), and various Python libraries were utilized in the development process, including Pandas, Numpy, Scikit-learn (Sklearn), TensorFlow (as the backend), Keras (as the frontend), Matplotlib, Anaconda, Jupyter Notebook, and Excel.

**4.2 Data Collection:** Data for Apple stock was gathered from Yahoo Finance and stored as 'APPLE' in a CSV-formatted document using MS Excel, spanning 2517 days. The dataset encompasses fundamental or historical data, providing essential information for stock trading. It includes open, high, low, and close prices, where the open price denotes the initial transaction price upon market opening, the high and low prices represent the day's highest and lowest traded prices, and the closing price reflects the last traded price on that day.

In addition to fundamental data, technical indicators such as Moving Average Convergence Divergence (MACD), Exponential Moving Averages (EMA), and Momentum were incorporated. Technical indicators involve mathematical calculations on variables like price and are widely utilized by active traders for short-term price movement analysis. The dataset covers values from January 2011 to December 2020. Close prices were visualized along with their corresponding dates using the Matplotlib plot() function.

**4.3 Scaling and Training the Data:** To facilitate data training, it is imperative to scale the data using the MinMaxScaler() from the Sklearn.preprocessing library. This step involves normalizing the data, enhancing the predictive performance of the model. The estimator, MinMaxScaler(), plays a crucial role in scaling and translating each feature independently to lie within a specified range, typically 0 to 1, on the training set. Subsequently, training and testing are conducted using this normalized data, where a portion of the data is used for training the model, followed by testing its performance.

- Train value = 0.8

- Test value = 0.2

This signifies that 80% of the data was allocated for training purposes, and the remaining 20% was designated for predicting the stock price.

**4.4 Technical Analysis based LSTM:** In the subsequent phase, the neural network is provided with the data and trained to predict, initializing random biases and weights. Our Time Series Analysis (TA) based Long Short-Term Memory (LSTM) model comprises an input layer, sequentially followed by an LSTM layer and then a dense layer, imported from the Keras model and Keras layers, respectively. The activation function used in the dense layer is linear, and the loss function is computed using Mean Squared Error (MSE). The optimizer employed for prediction is 'Adam,' a deep neural network algorithm that computes distinct learning rates based on various parameters.

The Adam optimizer formula is represented as: $m_n = E[X^n]$

Epochs are utilized to test the data in stages, denoting an integer value where the number of epochs determines the iterations for the test subject. Batch size represents the number of samples per gradient update. For the current model, Epochs = 150, Batch size = 64, Activation function is tanh.

**5. Results and Discussion:** Prediction of Test Data the figure below Figure 3 depict the comparison between stock close prices and dates for LSTM-FA and LSTM-TA models. The blue line represents the actual stock prices, while the orange line corresponds to the predicted prices for the LSTM-TA model. Additionally, the red line represents the predicted prices for the LSTM-FA model. The model's performance was evaluated using the R-square error formula, aiming for a value closer to 1, which is favorable for both training and testing. Although the predicted prices may not be entirely accurate, the visual representation serves the purpose of day-to-day predictions.

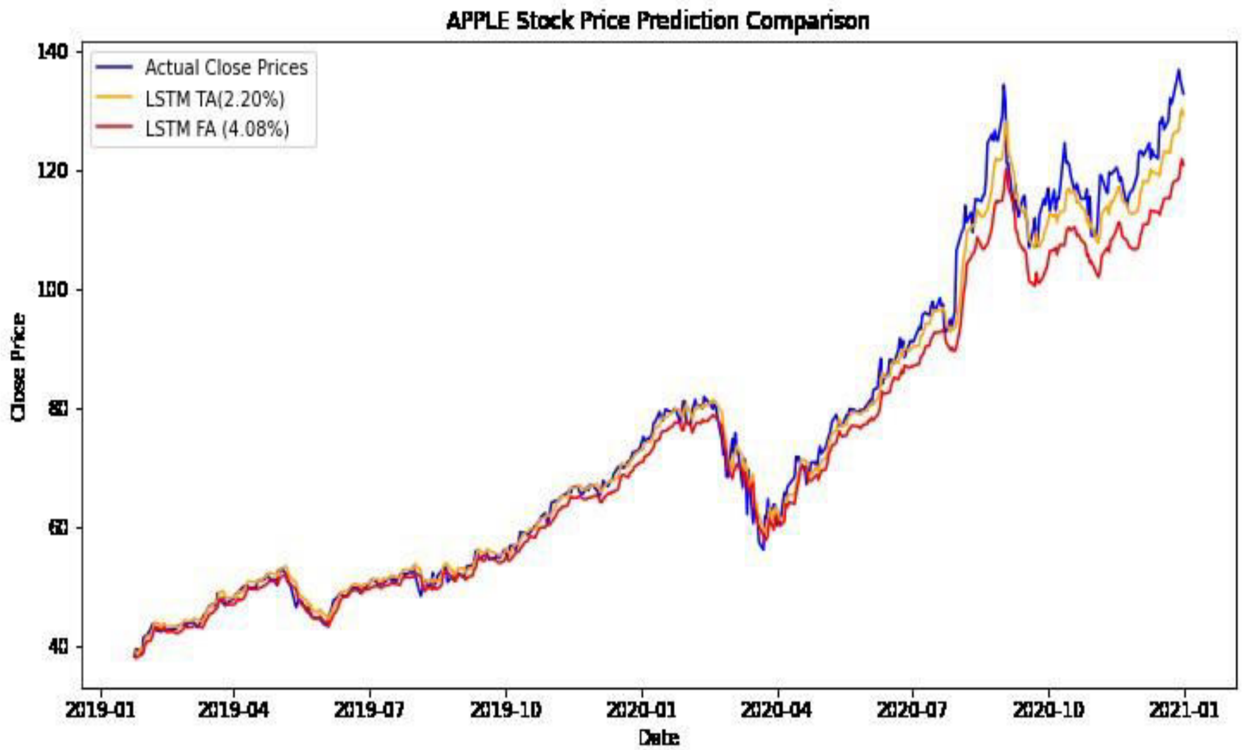


Fig 3: Analysis between actual close price vs predicted prices of two different models

Evaluation metrics Root Mean Square Error (RMSE) estimates how close the observed data points are to the model predicted values, Mean Absolute Error (MAE) measure how accurate our predictions are and what is the amount of deviation from the actual values, whereas R Square error gives the relationship between dependent and independent variables.

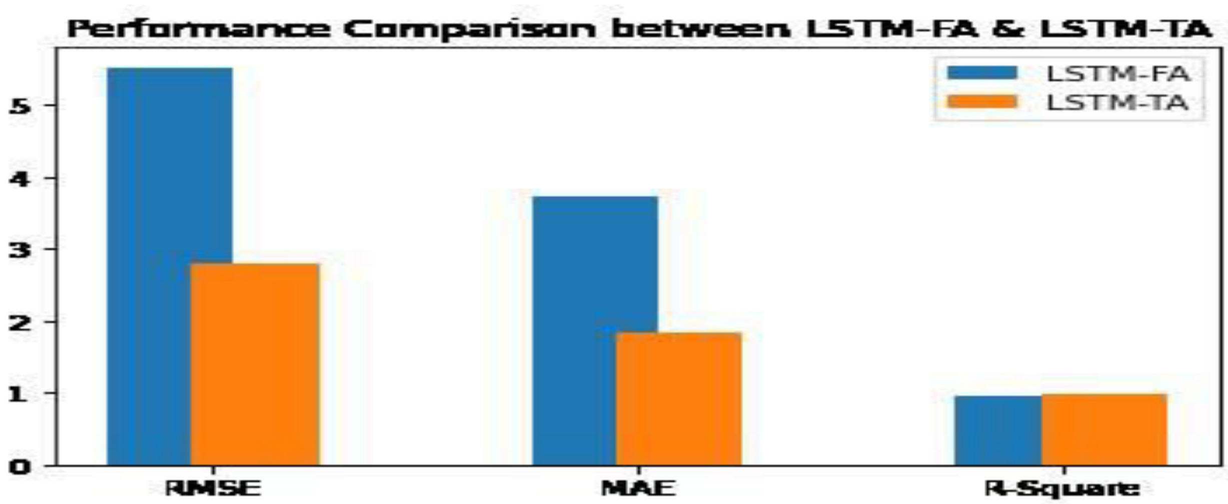


Fig 4: Evaluation metrics obtained for two different models

**6. Conclusion and Future Work:** We propose a formalization for stock price prediction based on deep learning. Our findings indicate that deep neural network architectures excel in capturing hidden dynamics and making accurate predictions. The model was trained using Apple's data, successfully predicting the stock price. This demonstrates the system's ability to identify interrelations within the data. The results highlight that the LSTM-TA architecture is effective in recognizing changes in trends. Among the proposed methodologies, LSTM-TA stands out as the best-performing model, leveraging technical indicators for prediction. While other models also utilize technical indicators, they do not surpass the performance of LSTM-TA in this context. The LSTM-TA architecture's capability to model both long-term and short-term data, capturing sudden changes in stock markets, contributes to its superior performance. Analyzing such information necessitates networks like LSTM-TA, which rely on both current and longer-term data. Future work in this study could explore additional methods, such as building a BiLSTM model that encodes the sequence in both forward and backward directions, concatenating the results for a more comprehensive analysis.

**References**


1. Archit Kapoor, varun Rastogi, Nirbhay Kashyap (2020). Forecasting daily close prices of stock indices using LSTM. 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp.10-14.
2. S.Sarvesh, Sidharth, Vaishnav (2021). A hybrid model for stock price prediction using machine learning techniques with CNN. 5th International Conference on Information Systems and Computer Networks (ISCON) GLA University, Mathura, India, pp.1632-1638
3. Pratheeth S, Vishnu Prasad R (2021). Stock price prediction using machine learning and deep learning. IEEE Mysore Sub Section International Conference (MysuruCon), pp.660-664.
4. Ayan Maiti, Pushparaj Shetty D (2020). Indian stock market prediction using deep learning. IEEE REGION 10 CONFERENCE (TENCON) Osaka, Japan, pp.1215-1220.
5. Bhawana Panwar, G. Dhuriya, Prashanth (2021). Stock market prediction using linear regression and SVM. International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp.629-631.
6. Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of china stock market. 2015 IEEE international conference on Big Data, pp. 2823-2824.

7. Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. International Journal of Science and Research (IJSR), 6(4), pp.1754–1756.
8. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), pp.654–669.
9. Karmiani, D., Kazi, R., Nambisan, A., Shah, A., & Kamble, V. (2019). Comparison of predictive algorithms: Backpropagation, SVM, LSTM and Kalman filter for stock market. In 2019 amity international conference on artificial intelligence (AICAI) pp. 228–234.
10. Yu, P., & Yan, X. (2019). Stock price prediction based on deep neural networks. Neural Computing and Applications, 32, pp.1609–1628.