STOCK PRICE PREDICTION USING LSTM

A Course Project report was submitted

in partial fulfillment of the requirement for the award of the degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

Sai Teja Thatikonda	**(2103A52035)**
Reethu Varma Ugge	**(2103A52037)**
Mithrasri Kadarla	**(2103A52050)**
Chaithra Sri Polsani	**(2103A52027)**
Keerthan Juluri	**(2103A52019)**

Under the guidance of

Sallauddin Mohmmad

Assistant Professor, Department of CSE.

SR UNIVERSITY

Department of Computer Science and Artificial Intelligence

SR UNIVERSITY

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the project entitled **“STOCK PRICE PREDICTION USING LSTM”** is the bonafide work carried out by ***Sai Teja, Reethu Varma, Mithrasri, Chaithra sri, Keerthan*** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2024-2025 under our guidance and Supervision.

Mr. Sallauddin	**Dr. M.Sheshikala**
Asst. Professor,	Assoc. Prof. & HOD (CSE),
S R University,	S R University,
Ananthasagar, Warangal.	Ananthasagar, Warangal.

ACKNOWLEDGMENT

We express our thanks to Course co-coordinator **Mr. Sallauddin, Asst—Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to the Head of the department CS&AI, **Dr. M. Sheshikala, Associate Professor** for encouragement, support, and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us in the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance in completing this project in the institute.

Finally, we thank all the teaching and non-teaching staff of the department for their suggestions and timely support.

TABLE OF CONTENTS

ABSTRACT

A vital component of financial research is stock price prediction, This helps investors and traders make informed decisions about buying or selling stocks.This study proposes a model for predicting stock prices that predicts the closing price of Google stocks by using Recurrent Neural Networks and Long Short-Term Memory. The Google stock dataset is used to train the suggested model at first. The Yahoo Finance API was used to gather historical stock data over a 20-year span. To give trend insights, important parameters including percentage changes, adjusted closing prices, and moving averages for 100 and 250 days were calculated. Sequences of 100 days were utilized as input to forecast the following day's stock price following the dataset had been preprocessed by standardizing the values using MinMaxScaler.

The LSTM-based model, including two LSTM layers succeeded by dense layers was assessed using 30% of the dataset after being trained on 70% of it. The model demonstrated strong performance with a Root Mean Square Error (RMSE) of roughly 4.31, signifying its precision in predicting stock prices. The predictions were then inversely transformed to match the original scale of stock prices for validation.

To improve the prediction accuracy, we also employed a range of machine learning (ML) models, including Random Forest, Decision Tree, XGBoost, and Voting Classifier.

Furthermore, Streamlit was used to deploy the model in an intuitive interface that let users enter stock IDs and view projected prices in addition to moving averages. This study shows how well LSTM handles time-series data and how it may be used for real-time stock market forecasting and decision-making.

INTRODUCTION

In financial research, stock price prediction is a complex but important field, particularly in context of advancements in machine learning (ML) and artificial intelligence (AI), which have improved forecasting accuracy. Because of their volatile nature and wide range of influences, stock markets require predictive modeling. One method uses ensemble learning to combine several information sources, including search data, social media, and web news, to provide reliable predictions. The effectiveness of Long Short-Term Memory (LSTM) & Recurrent Neural Networks (RNN) across a variety of time horizons makes them widely used [1].

Although performance gains with machine learning techniques are also significant, research contrasting deep learning techniques like LSTM and Recurrent Neural Networks (RNN) with more conventional models like Decision Trees, Random Forest, and Support Vector Classifiers shows that deep learning typically performs better in binary classification scenarios [2].

Numerous research on stock prediction have demonstrated the adaptability of LSTM networks. To improve investment portfolios, for instance, LSTM-based predictive models have been created. These models have demonstrated encouraging outcomes in terms of forecast accuracy and possible financial benefit [3]. Furthermore, LSTM architectures have shown a significant degree of temporal modeling capability when integrated with stock market data, frequently surpassing traditional approaches, particularly when paired with advanced feature selection techniques [4][5]. In the meanwhile, Transformer models have outperformed other deep learning models in accuracy and net value analysis, demonstrating encouraging outcomes in financial time-series forecasting [6].

Sentiment analysis has grown to be an important part of stock prediction. The divisiveness of social networking and financial news can improve forecast accuracy and market volatility insights, as shown by studies that use sentiment-based techniques [7][8]. For example, hybrid models can better capture public sentiment dynamics and improve stock price projections by combining temporal sentiment analysis with cutting-edge methods like Off-policy Proximal Policy Optimization (PPO) [11].

Hybrid techniques and optimization algorithms influenced by nature have also been the subject of recent research on stock prediction. For example, forecasting accuracy has been greatly increased by combining deep LSTM models with optimization approaches [9]. Furthermore, hybrid models including LSTM-Deep Neural Network (DNN) architectures and spatial-temporal convolutional networks performed better than baseline approaches, demonstrating the benefits of multi-layered frameworks [10][14]. Using sophisticated architectures like Bi-LSTM and attention mechanisms to manage fast-paced data, blockchain-based real-time prediction models and reinforcement learning techniques have also demonstrated promise in real-time stock prediction [15][16].

For certain markets, like the Turkish stock market, studies also demonstrate the effectiveness of integrated systems that use methods like Bi-LSTM stacking, PCA, and deep learning to produce accurate forecasts in these situations [17][18]. The most successful methods for AI-based stock market forecasting are frequently Support Vector Machines, LSTM, and Neural Networks, according to systematic evaluations, showing wide application across financial sectors [19]. Finally, AI's adaptability to different financial contexts is demonstrated by frameworks such as Advisor Neural Network-based recommendations, which have been tested on stock and cryptocurrency markets and have demonstrated durability and financial benefits even during market downturns [20].

Collectively these studies highlight how stock price prediction techniques have evolved, with a focus on deep learning models, hybrid frameworks, and sentiment analysis integration for improved accuracy and useful applications in financial decision-making.

\

LITERATURE REVIEW

Fausto Ricchiuti and Giancarlo Sperli (2025) [1] – The study offers an Advisor Neural Network framework for providing investors with personalised stock recommendations using LSTM-based informative stock research. A selection of 417 stocks and 67 cryptocurrencies were used to evaluate this strategy over a three-year period on the stock and markets for cryptocurrency. Despite the NASDAQ market's downward trend during the quarter under review, they made a financial gain of more than 41%, and investment in the cryptocurrency market yielded a 39.38% return.

Chin Yang Lin, João Alexandre Lobo Marques(2024)[2] – This study carried out a thorough analysis of comprehensive reviews on AI-based stock market prediction. The study identified commonly employed methods like NN (including ANN, CNN, and RNN), SVM, and LSTM. It highlighted the potential of AI in improving stock market prediction accuracy, while also acknowledging the challenges and limitations.

Taner Uçkan(2024)[3]- The research provides an integrated approach to stock market forecasting that combines Principal Component Analysis and deep learning methods. The authors examine the Turkish stock market to assess the effectiveness of the suggested model. Conclusions demonstrate that market prediction performance is improved when deep learning models are combined with PCA-selected indicators. The PCA-LSTM-CNN model, one of the suggested models, was found to produce outstanding outcomes.

Ashkan Safari, Mohammad Ali Badamchizadeh (2024) [4]- The research proposes Deep Investing, a sequence-oriented Bi-LSTM stacking model for stock market prediction. The authors undertake a case study utilising an Amazon (AMZN) stock dataset to evaluate the performance of the suggested methodology. An further indicator of prediction accuracy is the RMSPE score, which varies between 0.020 for High to 0.343 for Adj Close. Low R2 values are 0.9901, whereas high R2 values are 0.9941.

Dingjun Yao and Kai Yan (2024) [5] – The study presents DLWR-LSTM, an innovative deep learning framework for stock market index time series forecasting. The authors use deep learning and wavelet transforms to improve the stock's accuracy market index predictions. The findings show that sample time series variations have no effect on the DLWR-LSTM algorithm's prediction accuracy, and that the model's final prediction result stays constant at about 1% irrespective of time series variance.

Abhay Kumar Yadav and Virendra P. Vishwakarma (2024) [6] – The study presents a blockchain-based real-time stock price prediction model that incorporates Bi-LSTM, CNN and AM (Attention Mechanism) approaches. The authors develop a hybrid model that takes advantage of each technique's characteristics to improve real-time stock price predictions. The suggested prediction algorithm collects data from four distinct stocks: Uber, Nike, Facebook, and Apple. Computation time and RMSE are used to assess performance. As seen by the average values of 0.091 & 11.57 seconds, respectively, it outperformed benchmark models.

Khorshed Alam et al. (2024) [7] – The paper describes a strong LSTM-DNN model for improving stock market prediction. The authors created a combination of models that combines the advantages of deep neural networks (DNNs) and LSTM networks in order to improve the performance of stock market predictions. By doing thorough evaluations on twenty six stock datasets, the study confirms the model's scalability and robustness, yielding an average R2 value of 0.98606, Mean Absolute Error - 0.0210, and MSE – 0.00111.

Manjusha Pandey, Sinkon Nayak (2023) [8]- This study suggests using Twitter data to analyse sentiment in order to anticipate the stock market. The paper talks about how sentiment analysis might be used to forecast stock market performance. The results suggest that sentiment research can offer valuable insights into how the general public feels about particular stocks or sectors. However, the quality of the data gathered, the precision of sentiment analysis algorithm (SAA) used, and the ability to connect the sentiment information with stock market action all affect how effective stock anticipation is.

Shuzhen Wang (2023) [9] – A new stock price prediction technique based on an improved Transformer model and BiLSTM is presented in the paper. Additionally, 14 Shanghai and Shenzhen companies and five index equities were used to test the efficacy of the strategy. Out of the stock dataset, this method has the greatest R2 at 85.7%. RMSE falls to 93.5%, a decrease of 24.3% while R2 rises by 0.3% to 15.6.The results show that the BiLSTM-MTRAN-TCN technique predicts stock prices more accurately and with more generalisation capacity.

Guangyu Mu, Nan Gao(2023)[10]- The research provides a unique method for forecasting stock prices that integrates sentiment analysis from social media and stock market data. The authors use the Off-policy PPO technique to solve the issues of imbalanced sentiment classification, and a TLSTM model to combine temporal dynamics of stock prices with sentiment analysis. The implemented model has an RMSE of 2.147 and 82.19, whereas the semantic analysis model has an F-measure of 89%. The technique gives a more in-depth understanding of market dynamics as well as practical advice to investors and policymakers. The proposed method enhances the accuracy of stock price projections.

Changhai Wang, Jiaxi Ren, Hui Liang(2023)[11]- The paper introduced a spatial-temporal convolutional network that is localised to enhance stock market index forecasting, outperforming traditional methods in terms of various evaluation metrics. The experimental comparison included three assessment metrics: rise rate regression, however, further trend classification, and backtesting of the stock market. According to results of studies, our solution outperforms conventional methods by 4.2%, 3.1%, and 40%, respectively.

Burak Gülmez (2023) [12] – The research provides a novel technique to stock price prediction that combines an used a synthetic rabbits optimisation technique to tune a deep LSTM network. The author investigates the use of artificial rabbits optimisation, a nature-inspired approach, to improve the hyperparameters of a deep LSTM system for forecasting stock prices. The study shows how the suggested method increases stock price forecast accuracy.

Xu Zhang; Li Zhang(2022)[13]- In this study, the LSTM approach of deep learning is used to examine the factors that affect stock market prediction. Using the past performance of the Shanghai A-share the index, the previous US NASDAQ index, and the occurrences of the terms "raise position" and "reduce position" on Weibo, the authors find three types of important characteristics. The experiment employs data spanning from January 1, 2018 to December 31, 2020, and three metrics—mean absolute error (MAE), the root mean square error (RMSE), and the mean absolute percentage error (MAPE)—are used to assess the prediction findings. According to the results, the prediction accuracy of the single-factor group was higher than that of the multiple-factor group.

Raju Kumar (2022) [14]- The paper explores the difficulties of stock market forecasting and presents a technique for stock forecasting values according to news and social media research. Investigated the relationship between news polarity and stock market volatility, effectively predicting stock prices using data from multiple sources, including financial news and social media.

Chaojie Wang, Yuanyuan Chen (2022) [15]- The research assesses the Transformer model's effectiveness in stock market prediction, revealing that it outperforms typical deep learning algorithms and the purchase and hold approach in terms of net value analysis and forecast accuracy. The Transformer model produces promising results in forecasting financial time series. Deep learning models can help investors generate bigger returns.

Vuong, P. H., Dat,(2022) [16] – In this paper, a deep learning and advanced machine learning methodology to improve the performance of a stock-price forecasting (SPF) system is proposed. Employed a deep LSTM network and XGBoost for feature selection to improve stock price forecasting, outperforming the ARIMA model in terms of various evaluation metrics. The suggested strategy beats the baseline autoregressive integrated moving average technique in terms of mean absolute error, mean squared error, and root-mean-square error, according to experimental results.

Hum Nath Bhandari et al. (2022) [17] – This research presents an LSTM-based framework for the stock market index prediction. To predict the value of stocks in the future, the authors create an LSTM network architecture using previous stock market data. The study shows how the suggested strategy works to increase the precision of stock market forecasts. In comparison to The experimental findings show that a single layer LSTM framework provides advantages over multilayer LSTM models a better fit and excellent prediction accuracy.

Akhter Mohiuddin Rather (2021) [18] – An The article presents an LSTM-based deep neural network model for stock prediction and predictive optimisation. The author uses a new portfolio optimisation methodology to construct an investment portfolio using an LSTM deep neural networks and a regression model based on the projected values. The study shows how the suggested strategy works well for optimising investment portfolios and raising the accuracy of stock price forecasts.

Mojtaba Nabipour et al. (2020) [19] – s. In this study, nine machine learning models—Decision Tree, a Random Forest, adaptive boost (Adaboost), an eXtreme Gradient Boosting (XGBoost), the SVC, a Naïve Bayes, K-nearest-neighbors (KNN), Logistic Regression (LR), and Artificial Neural Network (ANN)—are compared with two powerful deep learning techniques: Recurrent Neural Network (RNN) Long Short Term Memory (LSTM).Deep learning models outperformed traditional methods, especially for improved model performance.

Naadun Sirimevan, I.G. U. H. Mamalgaha(2019) [20]- The study suggests a new approach to stock market price prediction by analysing a number of information sources, including search engine queries, user-generated content, web news, and historical stock data. suggested an ensemble model that combines weighted average, differential evolution, and LSTM-RNN to predict stock prices from a variety of data sources with good short-term forecast accuracy.

METHODOLOGY

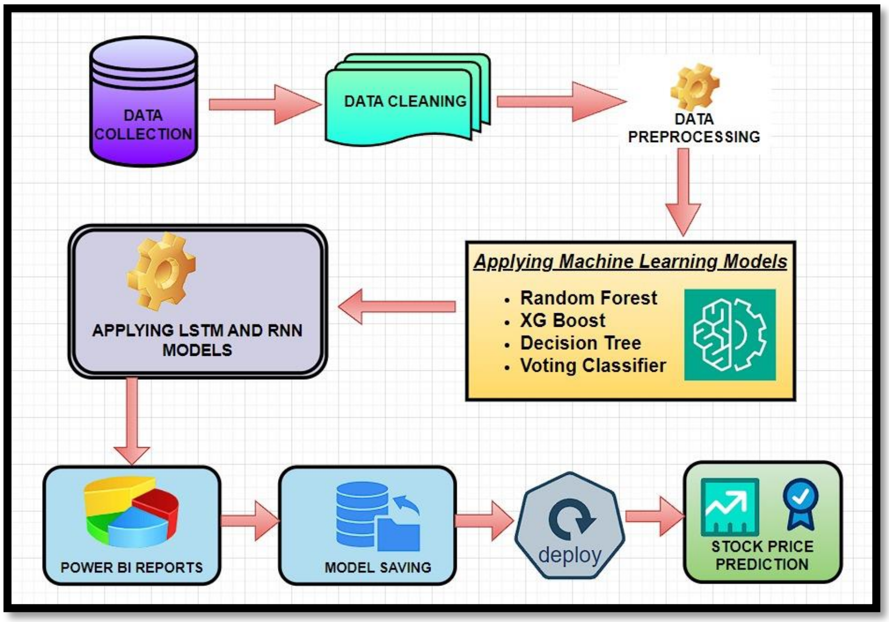


Fig.1. Architecture of proposed model

1. Data Acquisition

The first step involves gathering historical stock data for Google (GOOG) using the yfinance library. Data is downloaded from Yahoo Finance, covering approximately 20 years (from 2004 to the present date) The dataset includes six main columns: Open, High, Low, Close, Adj Close, and Volume, with a datetime index.

2. Data Exploration and Cleaning

The code begins by exploring the dataset's shape and basic statistics, such as mean, standard deviation, minimum, and maximum values for each column. The info() and isna() functions are used to ensure there are no missing values in any columns, confirming data integrity for modeling.

3. Data Visualization

To analyze and understand the data trends, the code visualizes each column in the dataset using a plot_graph function that takes the figure size, column values, and column name. Special emphasis is placed on the Adj Close price to observe overall stock price movements over time.

4. Feature Engineering: Moving Averages

The code calculates moving averages for periods like 100 days and 250 days, adding these columns to the dataset and visualizing them alongside the adjusted closing price.
The code calculates rolling moving averages to smooth out stock price fluctuations and reveal the underlying trend over different periods:

- ✓ **100-day Moving Average**: Captures shorter-term trends.

- ✓ **250-day Moving Average**: Shows longer-term trends, providing insights into how the stock price moves over the course of a trading year. The moving averages are calculated using:

These averages are added as columns to the dataset, and visualizations help identify patterns over time.

5. Feature Engineering: Percentage Change

The daily percentage change in the adjusted close price is computed to capture volatility. This additional feature helps in understanding the degree of daily price fluctuations:

6. Data Normalization

LSTM models require scaled data for optimal performance, as they are sensitive to the magnitude of inputs. The MinMaxScaler from the sklearn.preprocessing library is applied to scale Adj Close prices between 0 and 1.

7. Sequence Data Preparation

The code prepares the data into sequences suitable for LSTM:

- ✓ Each input sequence (x_data) consists of 100 consecutive days of the scaled adjusted close prices.
- ✓ The target output (y_data) is the price on the following day.
- ✓ For example, the first x_data entry contains data from days 1 to 100, with y_data holding the price on day 101. The code creates input-output pairs for the model to learn time dependencies using a sliding window approach.

8. Data Splitting

The dataset is split into training and testing sets:

- ✓ **Training Set**: 70% of the data, used for model training.
- ✓ **Testing Set**: 30% of the data, used for model evaluation.

9. LSTM Model Construction

A sequential model with LSTM layers is defined and trained on the x_data and y_data sets, using a mean-squared-error loss function and the Adam optimizer.

The model is built using Keras' Sequential API with three layers:

- ✓ **Layer 1**: An LSTM layer with 128 units and return_sequences=True, allowing the next LSTM layer to take sequences as input.
- ✓ **Layer 2**: Another LSTM layer with 64 units, return_sequences=False, producing a vector for the Dense layers.
- ✓ **Output Layer**: Two Dense layers, where the first has 25 units and the final Dense layer outputs a single prediction value. The model is compiled with the Adam optimizer and mean squared error (MSE) loss function for regression tasks. The model is then trained for two epochs.

10. Model Predictions

The trained model is used to predict values for the test dataset. These predictions are inverse-transformed to original values to provide a realistic comparison against actual stock prices.

11. Model Evaluation

The model's performance is assessed using the Root Mean Square Error (RMSE), calculated as: This metric quantifies the average difference between predicted and actual values in the same unit as the original stock prices.

12. Visualization of Predictions

Finally, the code visualizes the model's performance by plotting actual versus predicted Adj Close values on both test data and the entire dataset, giving a visual sense of the model's effectiveness in tracking stock trends.

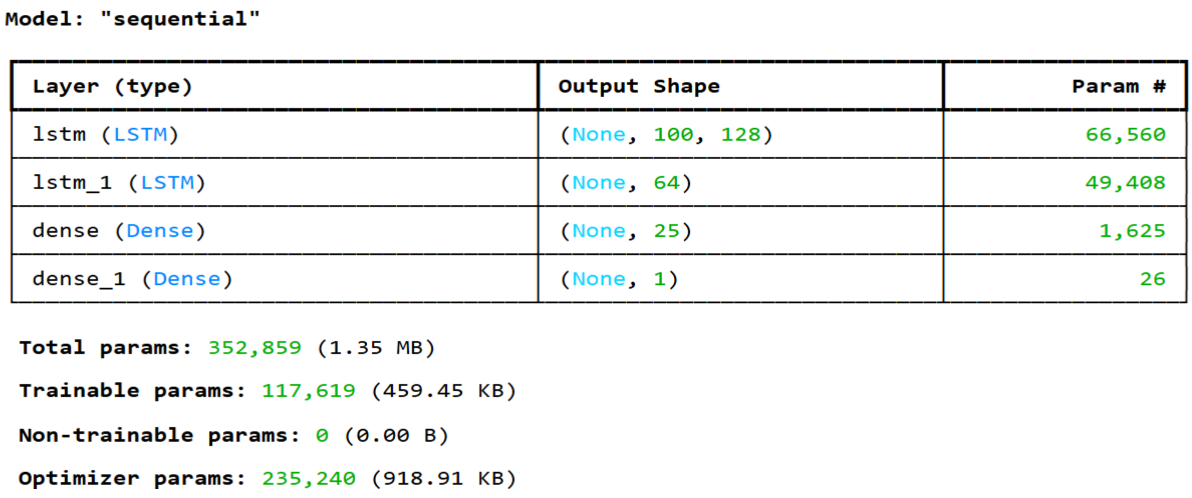
13. Model Saving

The trained LSTM model is saved for future use as a .keras file:
This step ensures the model can be loaded later for predictions without retraining.

14. Streamlit Deployment for Real-Time Prediction

The model is saved and later loaded into a Streamlit app for user interaction, where users can input a stock ticker to retrieve data, visualize moving averages, and display real-time predictions.

MODEL ARCHITECTURE



```
Model: "sequential"

┌──────────────────────────────┬───────────────────────────┬─────────────────┐
│ Layer (type)                 │ Output Shape              │         Param # │
├──────────────────────────────┼───────────────────────────┼─────────────────┤
│ lstm (LSTM)                  │ (None, 100, 128)          │          66,560 │
├──────────────────────────────┼───────────────────────────┼─────────────────┤
│ lstm_1 (LSTM)                │ (None, 64)                │          49,408 │
├──────────────────────────────┼───────────────────────────┼─────────────────┤
│ dense (Dense)                │ (None, 25)                │           1,625 │
├──────────────────────────────┼───────────────────────────┼─────────────────┤
│ dense_1 (Dense)              │ (None, 1)                 │              26 │
└──────────────────────────────┴───────────────────────────┴─────────────────┘

 Total params: 352,859 (1.35 MB)

 Trainable params: 117,619 (459.45 KB)

 Non-trainable params: 0 (0.00 B)

 Optimizer params: 235,240 (918.91 KB)
```

Fig. 2. **Model Architecture**

Model Type:
LSTM (Long Short-Term Memory) Neural Network.

Input Layer:
The input layer takes a 3-dimensional input shape of (100, 1), where 100 represents the sequence length of previous time steps (historical prices) used to predict the next price, and 1 is the feature dimension, which corresponds to the adjusted closing price.

LSTM Layers:
There are two LSTM layers in this model:

- ✓ **Layer 1**: A thick LSTM layer with 128 units and return_sequences=True to allow the LSTM to output a sequence for each time step, feeding data into the next LSTM layer.
- ✓ **Layer 2**: Another thick LSTM layer with 64 units and return_sequences=False, designed to output only the last hidden state to proceed with the dense layers.

Dense Layers:
Following the LSTM layers, two dense layers refine the prediction:

- ✓ **Dense Layer 1**: A dense layer with 25 units, acting as an intermediary layer for feature processing.
- ✓ **Output Layer**: The final dense layer with one unit to provide a single continuous output, representing the next predicted adjusted closing price.

Model Compilation:
The model is compiled with the following specifications:

- ✓ **Loss Function**: Mean Squared Error (MSE), suitable for continuous-valued regression tasks.
- ✓ **Optimizer**: Adam, which is an adaptive gradient descent optimization algorithm.

- ✓ **Evaluation Metrics**: During training and validation, model performance is evaluated based on the MSE loss.

Model Training:

The fit function is used to train the model with the following parameters:

- ✓ **Training Data**: The model is trained on the preprocessed sequences of stock prices (x_train) and their next-day prices (y_train).
- ✓ **Batch Size**: The model processes 1 sample per batch for detailed parameter updating.
- ✓ **Epochs**: The model is trained over 2 epochs (could be increased in practice for better accuracy).
- ✓ **Validation Data**: A separate test set (x_test, y_test) is used for evaluation after training.

Model Evaluation:

After training, the model's performance on the test set is measured using RMSE (Root Mean Squared Error) as an indicator of prediction accuracy.

Predictions:

A predict function is employed to generate price predictions on the test set (x_test). The predictions are inverse-transformed to the original price scale using MinMaxScaler for better interpretability and are compared to actual prices to evaluate model accuracy visually.

Summary:

This LSTM-based deep learning model is specifically designed for time series forecasting on historical stock prices. It leverages multiple LSTM layers for capturing long-term dependencies, followed by dense layers for additional processing, using MSE to ensure accurate stock price predictions over time.

PARAMETERS

1. Root Mean Squared Error (RMSE):

Definition: RMSE is a widely used metric to measure the accuracy of a model's predictions, particularly in regression tasks. It quantifies the difference between the predicted values and the actual values in the dataset. RMSE is calculated using the following formula

Formula: RMSE=$[(1/p)\sum(y_i - y_i`)^2]^{(1/2)}$

Where:

- i= 1 to p.
- n is the number of observations (data points),
- y_i is the actual value,
- y_i` is the predicted value.

In your case, the RMSE is approximately **11.04**, indicating that, on average, the predictions deviate from the actual values by about 11.04 units.

2.Mean Squared Error (MSE):

MSE is another measure of the average squared differences between predicted and observed values. It emphasizes larger errors more than smaller ones since the errors are squared.

Formula: MSE= $(1/p)\sum(y_i - y_i`)^2$

Where:

- i= 1 to p.
- n = number of observations
- y_i= actual value
- y_i`= predicted value

Your MSE is approximately **121.88**, indicating the average of the squared differences between predicted and actual values.

3. Mean Absolute Error (MAE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It provides a straightforward measure of prediction accuracy.

Formula: MAE= $(1/p)\sum|y_i - y_i`|$

Where:

- i= 1 to p.
- n = number of observations
- y_i= actual value
- y_i`= predicted value

In this instance, the MAE is approximately **8.74**, indicating that, on average, the absolute difference between the predicted and actual values is about 8.74 units.

4. Coefficient of Determination (R²):

R² indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. An R² of 1 indicates perfect predictions, while an R² of 0 indicates no predictive power.

Formula: $R^2 = 1-(ss_{res}/ss_{tot})$

Where:

- ss_{res} = $\sum$(yi−yi`)2 (the residual sum of squares)

- $ss_{tot} = \sum(yi-\bar{y})2$ (the total sum of squares)
- i= 1 to p
- $\bar{y}$ = mean of the actual values

The R² value of approximately **0.91** suggests that about 90.67% of the variance in the actual stock prices can be explained by the model's predictions, indicating a strong fit.

PARAMETERS LSTM	
RMSE	11.0399
MSE	121.879
MAE	8.7361
R^2	0.9066

Table 1: Performance analysis

DATASET

This dataset is derived from Yahoo Finance (YFinance) and captures the historical stock prices of Google (GOOG) over a 20-year period, from October 11, 2004, to October 8, 2024. The dataset comprises **5,033 rows and 6 columns**.

Using Yahoo Finance's API, you can extend this to collect datasets for multiple companies for stock price prediction during user interface. (Apple Inc. (AAPL), Microsoft Corporation (MSFT), Amazon.com, Inc. (AMZN), Tesla, Inc. (TSLA), Meta Platforms, Inc. (META) – formerly Facebook, Alphabet Inc. (GOOG) – Google's parent company, NVIDIA Corporation (NVDA), Johnson & Johnson (JNJ), Procter & Gamble Co. (PG), Coca-Cola Co. (KO)…..etc).

Here are the key details of each column:

- **Open**: The opening price of the stock for each trading day.
- **High**: The highest price of the stock during the trading day.
- **Low**: The lowest price of the stock during the trading day.
- **Close**: The closing price of the stock for the day.
- **Adj Close**: The adjusted closing price, accounting for stock splits, dividends, and other corporate actions.
- **Volume**: The number of shares traded on the given day.

Table 2: MSFT(Microsoft)

Date	Open	High	Low	Close	AdjClose	Volume
2004-10-28 00:00:00	28.11	28.54	27.9	28.01	17.4749	63,059,6000
2004-10-29 00:00:00	28.12	28.15	27.8	27.97	17.45	80,010,100
2004-11-01 00:00:00	28.16	28.28	27.96	28.08	17.5186	72,930,900
2004-11-02 00:00:00	28.26	28.47	28.03	28.24	17.6184	89,417,100
2004-11-03 00:00:00	28.65	28.65	28.31	28.47	17.7619	79,666,700
2004-11-04 00:00:00	28.38	29	28.38	29	18.0926	87,867,700
2004-11-05 00:00:00	29.21	29.36	29.03	29.31	18.286	95,337,700
2004-11-06 00:00:00	29.18	29.18	29.13	29.28	18.2673	112,802,100

Table 3: Reliance(Rs)

Date	Open	High	Low	Close	AdjClose	Volume
2004-10-28 00:00:00	17.465	17.47	16.81	16.94	12.2546	63,059,600
2004-10-29 00:00:00	16.965	17.185	16.81	17.16	12.4137	80,010,100
2004-11-01 00:00:00	17.21	17.23	16.8	17.105	12.3739	72,930,900
2004-11-02 00:00:00	17.435	17.925	17.375	17.725	12.8224	89,417,100
2004-11-03 00:00:00	18	18.37	17.915	18.205	13.1697	79,666,700
2004-11-04 00:00:00	18.15	18.485	18	18.47	13.3614	87,867,700
2004-11-05 00:00:00	18.49	18.7	18.235	18.325	13.2565	95,337,700
2004-11-06 00:00:00	18.45	18.61	18.16	18.375	13.2927	112,802,100

Table 4: **HDFC**

Date	Open	High	Low	Close	Adj Close	Volume
2004-10-28 00:00:00	3.394	3.411	3.33	3.359	2.9685	917,000
2004-10-29 00:00:00	3.369	3.518	3.368	3.517	3.1081	1,417,000
2004-11-01 00:00:00	3.523	3.524	3.385	3.405	3.0091	1,135,000
2004-11-02 00:00:00	3.455	3.595	3.455	3.475	3.071	608,000
2004-11-03 00:00:00	3.55	3.624	3.54	3.54	3.1284	534,000
2004-11-04 00:00:00	3.54	3.611	3.5	3.57	3.1549	989,000
2004-11-05 00:00:00	3.775	3.89	3.66	3.711	3.2795	1,726,000
2004-11-08 00:00:00	3.811	3.84	3.75	3.75	3.314	2,155,000

Table 5: **Google**

Date	Open	High	Low	Close	AdjClose	Volume
2004-10-28 00:00:00	24.7599	25.181	24.6378	24.4894	6.7879	88,602,149
2004-10-29 00:00:00	24.861	24.861	24.0525	24.2378	6.6369	4,313,288
2004-11-01 00:00:00	24.5157	24.8273	23.9894	24.701	6.7637	70,674,688
2004-11-02 00:00:00	24.8357	24.8452	24.5999	24.6463	6.7487	43,878,763
2004-11-03 00:00:00	24.7599	25.2316	24.6336	25.1684	6.8917	52,326,793
2004-11-04 00:00:00	25.2653	25.4253	24.8147	24.8947	6.8167	47,893,902
2004-11-05 00:00:00	24.9705	25.3495	24.9115	25.1221	6.879	38,258,162
2004-11-08 00:00:00	25.2568	25.6779	24.181	25.4048	6.9574	55,050,870

DATA ANALYSIS (Visualizations):

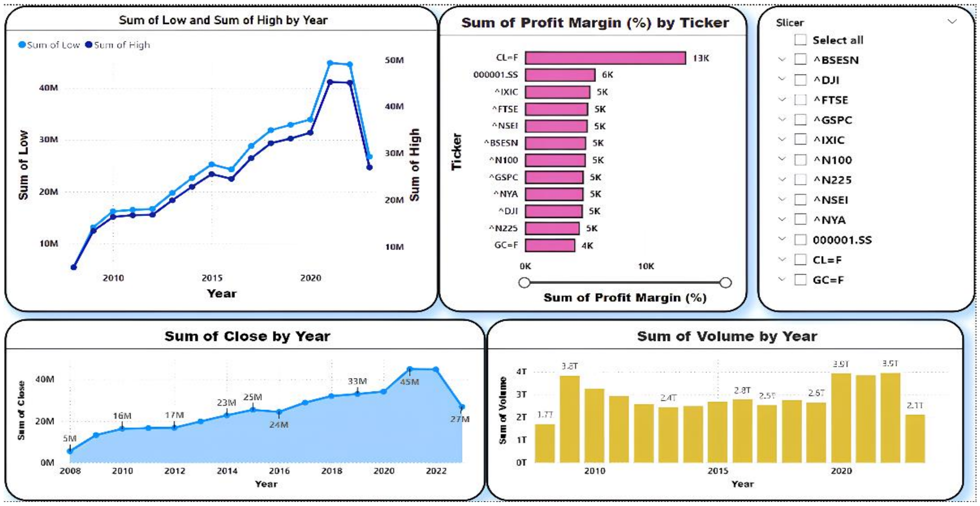


Fig.3. Dataset Visualization chart-I

Chart 1: Sum of Low and Sum of High by Year

This line chart presents the annual "Sum of Low" and "Sum of High" values for an asset or portfolio, depicted by two lines. The blue line represents the yearly "Sum of Low" values, while the orange line represents the "Sum of High" values. The chart illustrates a general upward trend in both lines, suggesting an increase in the asset's value over time. Additionally, the widening gap between the two lines may indicate a rise in price volatility, as the range between lows and highs becomes more pronounced each year.

Chart 2: Sum of Profit Margin (%) by Ticker

This bar chart visualizes the "Sum of Profit Margin" across various tickers, likely representing different stocks or funds. The chart allows for a comparison of profitability among different assets, with the length of each bar indicating the relative profitability—longer bars signify higher profit margins. The color-coding of the bars may represent different asset classes or sectors, adding another layer of comparison. A "Slicer" feature on the right enables users to filter data by specific tickers, making the chart interactive and customizable for a more targeted analysis.

Chart 3: Sum of Close by Year

In this line chart, the "Sum of Close" values for each year are tracked to illustrate the overall closing value trend for the asset or portfolio over time. The line generally trends upward, suggesting an increase in the asset's value throughout the years. However, there are fluctuations within this upward movement, indicating phases of both growth and decline that reflect the asset's dynamic performance across different periods.

Chart 4: Sum of Volume by Year

This bar chart displays the "Sum of Volume" for each year, showcasing trading activity or volume for the asset or portfolio. The bars exhibit an overall upward trend, implying an increase in trading volume over time. Notably, there are fluctuations in the bar heights, which suggest periods of higher and lower trading activity. This variability might be indicative of changes in investor interest or market conditions affecting the volume traded in certain years.

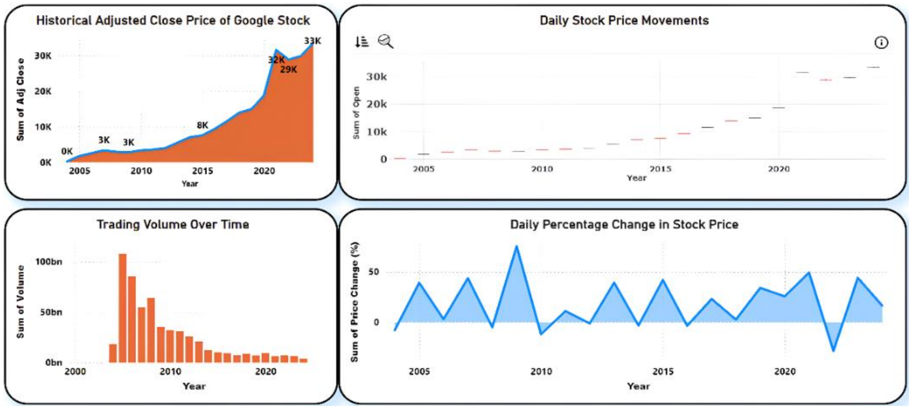


Fig.4. Dataset Visualization chart-II

Chart 1: Historical Adjusted Close Price of Google Stock

This line chart tracks the historical adjusted closing price of Google's stock, highlighting the stock's long-term trend. Since 2005, Google's stock price has demonstrated a strong upward trajectory, showing robust growth over time. Notably, this growth has accelerated in recent years, with the increase becoming especially pronounced after 2015. This sustained upward trend likely reflects positive market sentiment and strong company performance, marking a clear appreciation in Google's stock value.

Chart 2: Daily Stock Price Movements

This bar chart illustrates daily fluctuations in Google's stock price, emphasizing short-term price volatility. The bars reveal that the stock has experienced both gains and losses on a daily basis, showing dynamic price behavior. Over time, the magnitude of these daily movements appears to have increased, indicating a rise in volatility and larger price swings. This increasing volatility may be attributed to various market factors, including changes in investor behavior, economic conditions, and company-specific news or events.

Chart 3: Trading Volume Over Time

Displayed as a bar chart, this chart showcases the trading volume for Google's stock over the years, providing insights into the trading activity. Since 2005, the trading volume has shown a general decrease, suggesting a potential stabilization in the stock price. Lower trading volume may reflect reduced speculative interest, as the stock price becomes more stable and as the stock matures in the market. This trend in volume could also indicate a shift towards longer-term investors holding onto the stock rather than actively trading it.

Chart 4: Daily Percentage Change in Stock Price

This line chart captures the daily percentage change in Google's stock price, focusing on the magnitude of price fluctuations. The daily percentage changes indicate notable short-term volatility, with significant day-to-day fluctuations observed in the stock price. Over time, there seems to be an increase in these percentage-based price swings, reinforcing the idea of heightened volatility as the stock grows in value and as market conditions shift.

RESULT AND DISCUSSIONS

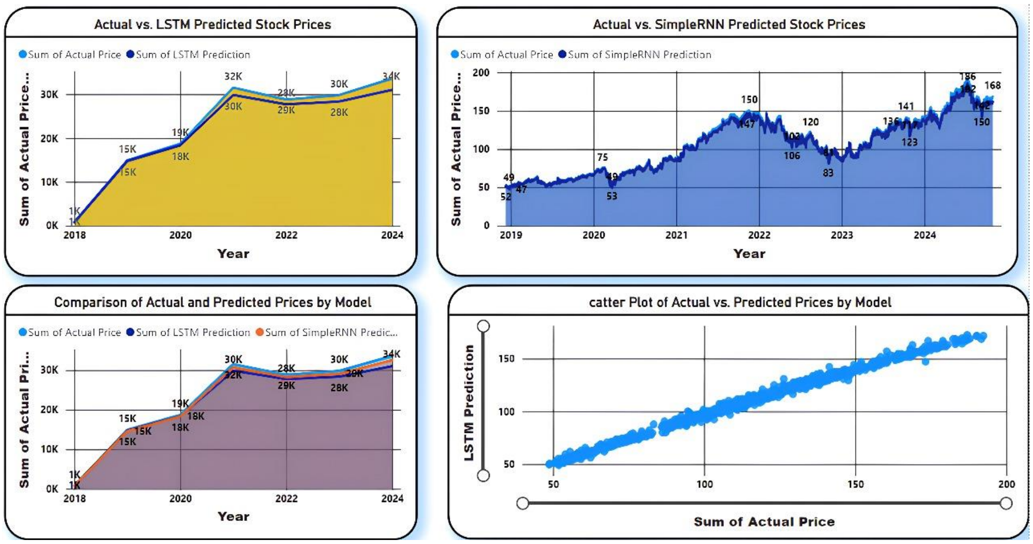


Fig.5. Actual vs. LSTM and RNN Predicted Stock Prices

Chart 1: Actual vs. LSTM Predicted Stock Prices

This line chart compares the actual stock prices with the prices predicted by an LSTM (Long Short-Term Memory) model from 2018 to 2024. The blue line represents the actual prices, while the orange line represents the LSTM model's predictions. The two lines closely follow each other, indicating that the LSTM model is able to capture the general trend of the stock prices over the years. The values increase significantly from 2018 to around 2021, reaching around 32K, with some fluctuations afterward. By 2024, both the actual and predicted prices appear to level off, reaching close to 34K. This alignment suggests that the LSTM model is reasonably accurate in predicting the overall trend.

Chart 2: Actual vs. SimpleRNN Predicted Stock Prices

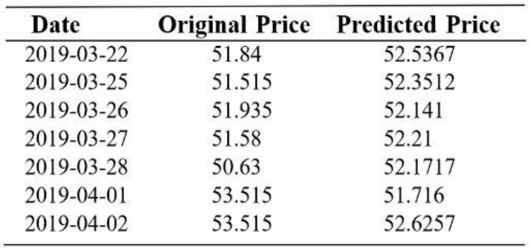
This chart also compares actual stock prices with predictions, but using a SimpleRNN (Recurrent Neural Network) model from 2019 to 2024. Here, the blue line represents the actual stock prices, and the darker overlay area represents the predictions. The SimpleRNN model predictions follow the actual prices relatively closely, though with slightly less precision than the LSTM model shown in Chart 1. The prices show fluctuations over the years, increasing from around 50 in 2019 to nearly 200 by 2024. There are noticeable deviations between the predicted and actual prices at certain points, suggesting that the SimpleRNN model may be less capable of capturing more nuanced price movements compared to the LSTM model.

Chart 3: Comparison of Actual and Predicted Prices by Model

This area chart compares the actual prices with predictions from both the LSTM and SimpleRNN models from 2018 to 2024. The blue line represents actual prices, the orange line represents LSTM predictions, and the red line represents SimpleRNN predictions. Both LSTM and SimpleRNN predictions closely follow the actual price trend from 2018 to 2024, though the LSTM model appears to be more accurate. The area between the lines highlights the slight differences between the actual and predicted values, with the SimpleRNN showing more variation. The actual prices rise significantly until around 2021, then level off, with the models capturing this trend well. This chart provides a side-by-side comparison, showing that the LSTM model offers a closer fit to the actual data.

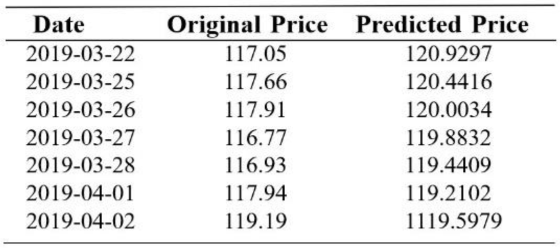
Chart 4: Scatter Plot of Actual vs. Predicted Prices by Model
This scatter plot compares actual prices on the x-axis with LSTM model predictions on the y-axis, representing how well the LSTM model aligns with the actual data. The points form a nearly diagonal line from the bottom left to the top right, suggesting a strong positive correlation between the actual and predicted values. The alignment of points along this diagonal line indicates that the LSTM model is generally effective in predicting the actual prices, as the predicted values are close to the actual ones. Any deviations from this line would represent instances where the model's predictions diverge from the actual values, but the tight clustering indicates good model performance.

These charts collectively demonstrate that both the LSTM and SimpleRNN models are capable of capturing the overall trend in stock prices, with the LSTM model showing higher accuracy. The upward trend in stock prices is clearly depicted, along with increased stability in later years. The scatter plot further confirms the LSTM model's effectiveness, as its predictions closely align with actual prices.



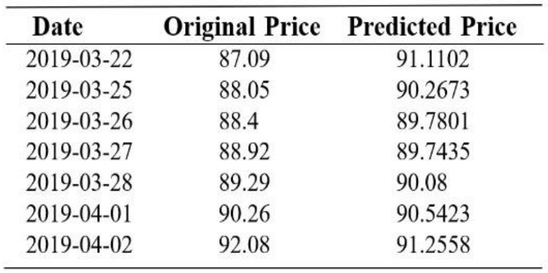
Date	Original Price	Predicted Price
2019-03-22	51.84	52.5367
2019-03-25	51.515	52.3512
2019-03-26	51.935	52.141
2019-03-27	51.58	52.21
2019-03-28	50.63	52.1717
2019-04-01	53.515	51.716
2019-04-02	53.515	52.6257

Table 6. Actual vs Predicted in GOOGLE



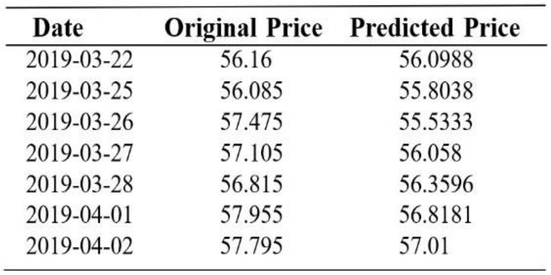
Date	Original Price	Predicted Price
2019-03-22	117.05	120.9297
2019-03-25	117.66	120.4416
2019-03-26	117.91	120.0034
2019-03-27	116.77	119.8832
2019-03-28	116.93	119.4409
2019-04-01	117.94	119.2102
2019-04-02	119.19	1119.5979

Table 7. Actual vs Predicted in RELIANCE



Date	Original Price	Predicted Price
2019-03-22	87.09	91.1102
2019-03-25	88.05	90.2673
2019-03-26	88.4	89.7801
2019-03-27	88.92	89.7435
2019-03-28	89.29	90.08
2019-04-01	90.26	90.5423
2019-04-02	92.08	91.2558

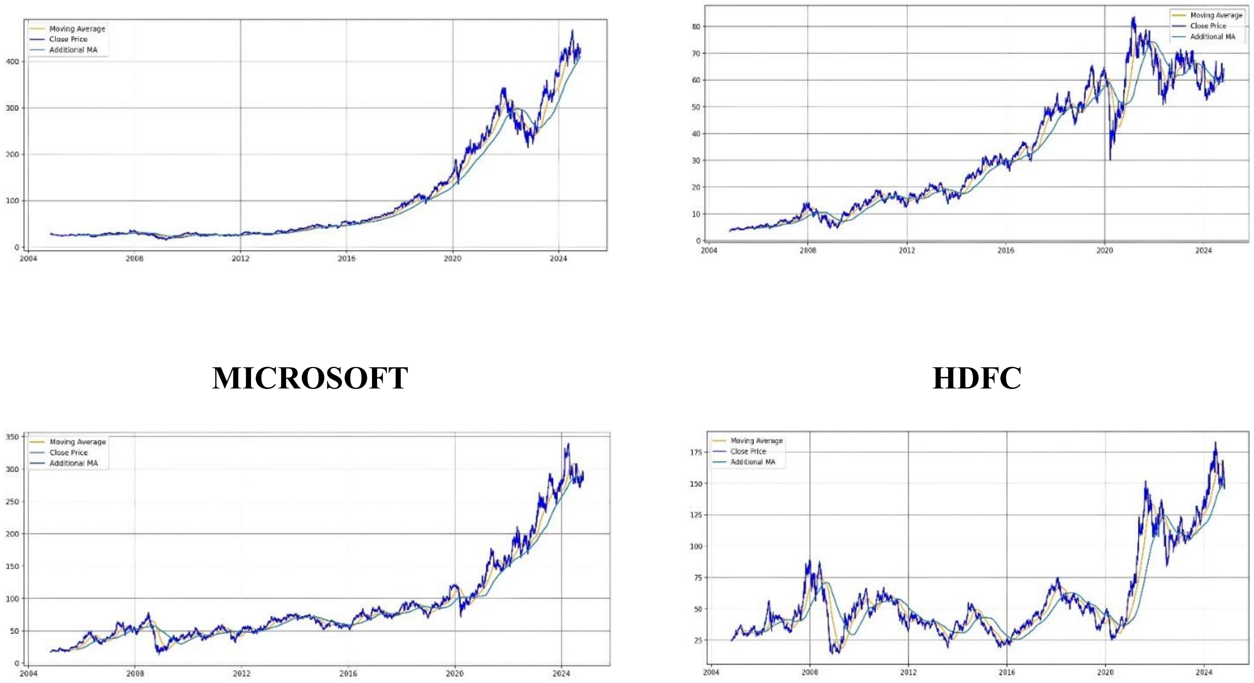
Table 8. Actual vs Predicted in MICOSOFT



Date	Original Price	Predicted Price
2019-03-22	56.16	56.0988
2019-03-25	56.085	55.8038
2019-03-26	57.475	55.5333
2019-03-27	57.105	56.058
2019-03-28	56.815	56.3596
2019-04-01	57.955	56.8181
2019-04-02	57.795	57.01

Table 9. Actual vs Predicted in HDFC

The tables comparing actual and predicted stock prices for Microsoft, HDFC, Reliance, and Google serve to evaluate the accuracy of predictive models in forecasting these stocks' closing prices. Each table includes columns for the date, actual closing price (Original), and predicted closing price (Predicted), with the purpose of assessing model performance and identifying areas for improvement. Generally, the predicted values closely match the actual values, indicating reasonable model accuracy. However, occasional deviations suggest that the models may struggle during periods of high volatility, where price movements are harder to predict. These tables are valuable for identifying trends in stock prices. For example, when the predicted prices align with an upward or stable trend, it confirms that the models capture overall market sentiment. Moreover, they provide actionable insights: potential buying opportunities arise when the predicted price is lower than the actual price, and potential selling opportunities emerge when the predicted price is higher. By monitoring these predictions alongside actual prices, investors can make more informed decisions about entering or exiting positions in Microsoft, HDFC, Reliance, and Google.



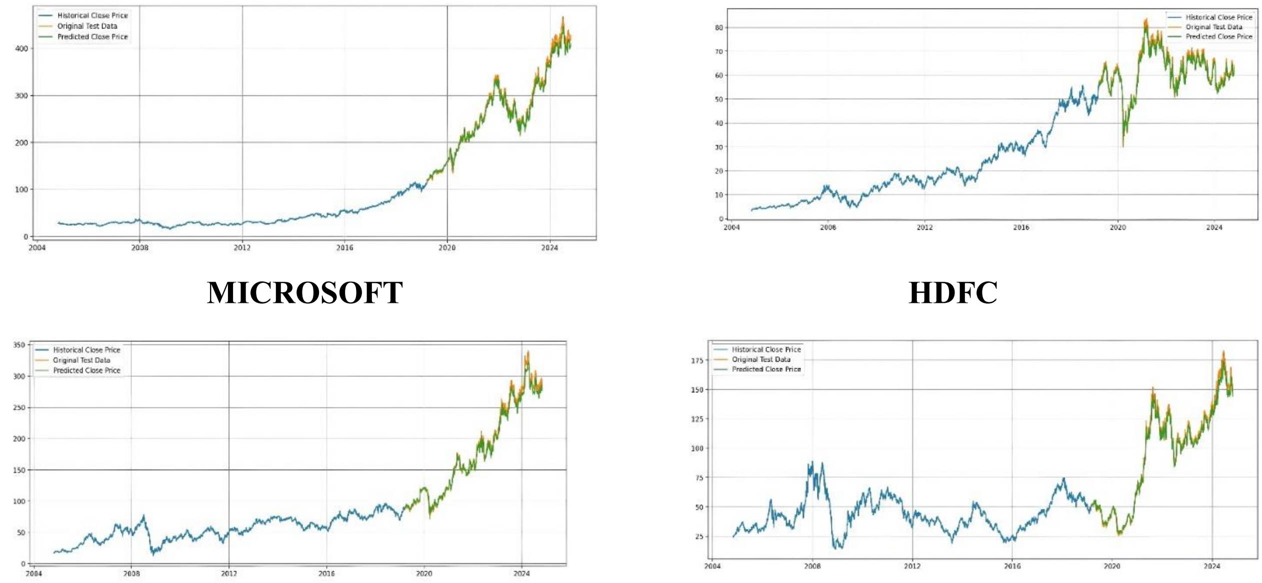
MICROSOFT **HDFC**

RELIENCE **GOOGLE**

Fig.6. Original Close Price with MA for 100 days ,200 & 250 days

The graphs for Microsoft, HDFC, Reliance, and Google stock prices provide a comprehensive view of each company's historical price trends, along with three moving averages (MAs) representing different timeframes: 100 days (orange line), 200 days (blue line), and 250 days (green line). The Moving averages smooth out short-term price fluctuations and highlight longer-term trends, which helps investors identify potential buy or sell signals. It helps identify key trends: an upward trend when the stock price is above all MAs, a downward trend when it is below all MAs, and a sideways trend when it fluctuates between them. The 200-day and 250-day MAs often act as support and resistance levels, where prices may either bounce up (support) or face selling pressure (resistance). Additionally, crossover signals like the "Golden Cross" (50-day MA crossing above the 200-day MA) indicate a bullish trend, while the "Death Cross" (50-day MA crossing below the 200-day MA) suggests a bearish trend. Analyzing these indicators enables more informed trading decisions about buying and selling stocks.

In all four companies, the actual closing price (black line) is displayed alongside the three MAs, which highlight various price trends. The stock price trend for all four companies shows an overall upward trajectory, with the values of all three MAs generally increasing over time, indicating a bullish outlook.



MICROSOFT

HDFC

RELIENCE

GOOGLE

Fig.7.Original Close Price vs Predicted Close Price

The graphs of actual vs. predicted closing prices for Microsoft, HDFC, Reliance, and Google provide valuable insights into stock price trends, model accuracy, and potential trading opportunities. In each graph, the blue line represents the historical closing price, the orange line shows the original test data (a subset of historical data used for testing), and the green line represents the predicted closing price generated by the model. The primary purpose of these graphs is to visualize historical stock performance, evaluate the model's accuracy in forecasting prices, and identify trends and patterns that may guide investment decisions.

Across all companies, an overall upward trend in stock prices is observed, reflecting growth and stability over time. The model generally captures this trend, as the predicted values closely follow the actual prices, indicating that it effectively models the overall direction of each stock's price. However, some discrepancies between the predicted and actual prices appear during periods of high volatility, highlighting the model's sensitivity to short-term fluctuations and the challenges of predicting turning points.

By analyzing these graphs, investors can use the predicted prices to identify potential buying opportunities when the model forecasts a lower price than the actual value and selling opportunities when the predicted price is higher. Together, these graphs and the underlying predictive models provide a visual and data-driven approach to making informed stock trading decisions.

CONCLUSION

This Stock Market Price Prediction project is designed to forecast stock prices based on historical data. The app’s user-friendly interface allows users to input a stock ticker (e.g., "GOOG") and select a date range, after which the app fetches relevant historical data from Yahoo Finance. A pre-trained LSTM model, known for capturing sequential dependencies in time-series data, is loaded to make predictions. The data is prepared by scaling and windowing, ensuring compatibility with the model’s requirements, and transformed back to the original scale for interpretable predictions. Additionally, the application features visualizations of moving averages over 100, 200, and 250 days, providing users with insights into both short and long term stock price trends. The app displays actual vs. predicted stock prices side by side, both in a table and on a line graph, giving users a clear view of the model’s performance. The LSTM model effectively captures overall price trends, though some deviations suggest areas for potential refinement. For a more comprehensive analysis, including error metrics like MAE & RMSE could help users better understand prediction accuracy. This project illustrates the power of combining machine learning with real-time data visualization to assist users in making informed investment decisions, with future potential to integrate additional models and real-time analysis capabilities to enhance accuracy and usability.

Overall, this project highlights the capability of deep learning models to handle complex time-series data in the financial domain, making them powerful tools for real-time forecasting and analysis.

FUTURE SCOPE

This project shows promising results in stock price prediction using LSTM, but there are various areas where future improvements can be made. One direction is to incorporate additional features like macroeconomic indicators, sector-specific metrics, or even sentiment data from news and social media, which can influence stock prices. Advanced architectures, such as Transformers or hybrid models combining LSTM with attention mechanisms, could be explored to capture complex market patterns, especially during volatile periods. Additionally, implementing reinforcement learning could allow the model to not only predict prices but also optimize trading strategies by learning when to buy or sell based on predicted movements.

Further, enhancing the deployment and usability of the model could broaden its application. Currently deployed via Streamlit, it could be expanded into a mobile or web app for greater accessibility, with potential integration to broker APIs for automated trading. Real-time performance could be improved by leveraging cloud services optimized for low-latency applications, making it viable for high-frequency trading. Backtesting features would allow users to evaluate the model’s performance under historical market conditions, helping refine predictions. Ensemble approaches combining LSTM with other models like XGBoost or an ensemble of deep learning models could further enhance prediction accuracy, providing a more robust and comprehensive tool for investors and institutions alike.

REFERENCE

[1] Ricchiuti, F., & Sperlí, G. (2024). An Advisor Neural Network framework using LSTM-based Informative Stock Analysis. Expert Systems with Applications, 125299.

[2] Lin, C. Y., & Marques, J. A. L. (2024). Stock market prediction using Artificial Intelligence: A systematic review of Systematic Reviews. Social Sciences & Humanities Open, 9, 100864.

[3] Uçkan, T. (2024). Integrating PCA with deep learning models for stock market Forecasting: An analysis of Turkish stocks markets. Journal of King Saud University-Computer and Information Sciences, 36(8), 102162.

[4] Safari, A., & Badamchizadeh, M. A. (2024). DeepInvest: Stock Market Predictions with a Sequence-Oriented BiLSTM Stacked Model–A Dataset Case Study of AMZN. Intelligent Systems with Applications, 200439.

[5] Yao, D., & Yan, K. (2024). Time series forecasting of stock market indices based on DLWR-LSTM model. Finance Research Letters, 68, 105821.

[6] Yadav, A. K., & Vishwakarma, V. P. (2024). An Integrated Blockchain Based Real Time Stock Price Prediction Model by CNN, Bi LSTM and AM. Procedia Computer Science, 235, 2630-2640.

[7] Alam, K., Bhuiyan, M. H., ul Haque, I., Monir, M. F., & Ahmed, T. (2024). Enhancing Stock Market Prediction: A Robust LSTM-DNN Model Analysis on 26 Real-Life Datasets. IEEE Access.

[8] Pandey, M., Nayak, S., & Rautaray, S. S. (2023, December). An Analysis on Sentiment Analysis and Stock Market Price Prediction. In 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 367-370). IEEE.

[9] Wang, S. (2023). A stock price prediction method based on BiLSTM and improved transformer. IEEE Access.

[10] Mu, G., Gao, N., Wang, Y., & Dai, L. (2023). A stock price prediction model based on investor sentiment and optimized deep learning. Ieee Access, 11, 51353-51367.

[11] Wang, C., Ren, J., Liang, H., Gong, J., & Wang, B. (2023). Conducting stock market index prediction via the localized spatial–temporal convolutional network. Computers and Electrical Engineering, 108, 108687.

[12] Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Systems with Applications, 227, 120346.

[13] [13] X. Zhang, L. Zhang, L. Xu and Y. Jiang, "Research on Influential Factors in Stock Market Prediction with LSTM," 2022 7th International Conference on Big Data Analytics (ICBDA), Guangzhou, China, 2022

[14] Kumar, R., Sharma, C. M., Chariar, V. M., Hooda, S., & Beri, R. (2022, May). Emotion analysis of news and social media text for stock price prediction using svm-lstm-gru composite model. In 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) (pp. 329-333). IEEE.

[15] Wang, C., Chen, Y., Zhang, S., & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. Expert Systems with Applications, 208, 118128.

[16] Vuong, P. H., Dat, T. T., Mai, T. K., & Uyen, P. H. (2022). Stock-price forecasting based on XGBoost and LSTM. Computer Systems Science & Engineering, 40(1).

[17] Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. (2022). Predicting stock market index using LSTM. Machine Learning with Applications, 9, 100320.

[18] Rather, A. M. (2021). Lstm-based deep learning model for stock prediction and predictive optimization model. EURO Journal on Decision Processes, 9, 100001.

[19] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. Ieee Access, 8, 150199-150212.

[20] Sirimevan, N., Mamalgaha, I. G. U. H., Jayasekara, C., Mayuran, Y. S., & Jayawardena, C. (2019, December). Stock market prediction using machine learning techniques. In 2019 International Conference on Advancements in Computing (ICAC) (pp. 192-197). IEEE.