IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

STOCK PRICE PREDICTION

[1]Abhishek Kumar, [2]Sheetal Pereira
[1]Student, [2]Assistant Professor
[1]Computer Engineering,
[1]KJ Somaiya College of Engineering, Mumbai, India

Abstract : Predicting Stock market is actually one of the major type of tasks. There are so many traditional techniques for predicting stock prices but they have achieved poor accuracy because they are unable to identify intricate patterns and nonlinear correlations in the data. By utilizing cutting-edge deep learning methods like LSTM and BiLSTM with Attention Mechanism, our model effectively captures nonlinear relationships and temporal dependencies in the data, allowing it to overcome these restrictions. This study uses deep learning (DL) and machine learning (ML) approaches to provide a thorough examination of stock price prediction. Yahoo Finance is used to gather daily historical stock data spanning the last five years. Training sets make up 80% of the dataset, whereas testing sets make up 20%. Metrics like Root Mean Squared Error (RMSE), Mean Squared Error (MSE) and R-squared (R2) are used to train and assess machine learning models like Support Vector Machine (SVM) Regressor and Random Forest Regressor as well as deep learning models like Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Stacked BiLSTM with Attention Mechanism. With the highest accuracy percentage, the Stacked BiLSTM model with Attention Mechanism is the best-performing model.

***Keywords* - LSTM, Bi-LSTM, DL, ML, SVM, RMSE, MSE, R2**

INTRODUCTION

The technique of projecting future changes in the values of stocks traded on financial markets is known as "stock price prediction" [1]. Predicting the direction and size of future price movements requires careful examination of past stock data, market patterns, and other pertinent variables. A crucial component of making informed investing decisions is stock price prediction, which aids traders and investors in spotting possible profit opportunities and predicting market moves.

MOTIVATION

This project's objective is to investigate and contrast the efficacy of different ML and DL models for stock price prediction. Through the use of cutting-edge methodologies and historical stock data from Yahoo Finance, we hope to create reliable models that can accurately represent the intricate workings of financial markets. Finding the best model to reliably anticipate stock prices and give market participants useful information is the ultimate objective.

LITERATURE SURVEY

[2] suggests combining convolutional neural networks (CNN), bi-directional long short-term memory (BiLSTM), and attention mechanisms (AM) to predict stock prices. BiLSTM forecasts closing prices for the following day, CNN extracts features, and AM records feature influences over time. With least MAE (21.952) and RMSE (31.694) and maximal R2 (0.9804), the technique surpasses seven others when tested on 1000 Shanghai Composite Index trading days. Investors have access to a dependable instrument for making decisions during market turbulence thanks to its accuracy and applicability for stock price prediction.

[3] examines papers from the DBLP database to conduct a survey of deep learning applications in stock and Forex prediction. We classify and compare many deep learning techniques, including Reinforcement Learning, CNN, LSTM, DNN, RNN, and others like Wave net and HAN. Datasets, variables, models, and performance indicators from various studies are examined in the survey. Promising results have been found in recent research that explores deep learning methodologies such as reinforcement learning and LSTM paired with DNN. The difficult part is having to choose the right models among the many available.

In recognition of the complexity of financial data, [4] suggests a DNN-based prediction model for financial product prices. Price series are reconstructed in a one-dimensional space using the PSR approach, which makes DNN and LSTM modelling possible for precise predictions. Capturing the time-dependent, nonlinear character of financial data among a multitude of associated components is a challenging task. Superior accuracy is shown by comparing the prediction model against other models. The suggested model, which makes use of the benefits of deep learning and neural networks, shows promise in terms of stock price prediction, supporting ongoing attempts to comprehend and forecast financial activity.

[5] compares the use of artificial neural networks (ANN), stochastic process-geometric Brownian motion (SP-GBM), and autoregressive integrated moving averages (ARIMA) in stock price prediction. The predictive models for each approach are constructed using historical stock data from Yahoo Finance. Model outputs and actual stock prices are compared in a comparative study. The results show that neural network models are not as effective at predicting the price of stocks one day from now as traditional statistical models like ARIMA and stochastic models like SP-GBM. Choosing the right modelling tools and mastering the intricacies of time series analysis present challenges. This research adds to our understanding of efficient predictive modelling for the analysis of daily stock data.

Four machine learning techniques are shown in [6] for stock index price prediction: Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and an attention-based neural network. In the past, established, less developed, and developing markets have been represented by data from the SP500, CSI300, and Nikkei225 indexes, respectively. Seven input variables are used, including technical indicators, macroeconomic variables, and trading data. In mature markets as opposed to emerging ones, all models perform better, but the attention-based model performs the best. Addressing erratic noise in index price predictions is one of the challenges. The results of this study provide valuable information for predictive modelling in a variety of financial markets.

[7] provides an examination of methods for predicting stock prices, including graph-based, deep learning, conventional machine learning, and neural networks. The research explores strategies for forecasting both near- and long-term changes in the market. One of the challenges is efficiently navigating the uncertainty in equity market dynamics while combining statistical and AI algorithms. The study provides a thorough examination of the several methods used in stock price prediction, stressing both their advantages and disadvantages. It also talks about potential areas for future research in this area, highlighting the continuous development of equity market prediction models.

IMPLEMENTATION

3.1 Dataset

The dataset for this project comprises historical stock data collected from Yahoo Finance, spanning a period of five years. It encompasses daily stock metrics such as closing price, opening price, highest price, trading volume and lowest price. Each data point is associated with a specific date, providing a time-series aspect crucial for forecasting stock prices. Upon loading the dataset, initial pre-processing steps involve ensuring data integrity by handling missing values, outliers, and inconsistencies. Date Time feature extraction is performed to derive additional features such as day, month, and year, which can offer valuable insights into temporal patterns.

Data visualization techniques, leveraging libraries like matplotlib and seaborn, aid in uncovering trends, correlations, and anomalies within the dataset. Subsequently, time series forecasting necessitates specific pre-processing steps like window rolling to transform the data into a supervised learning format. To facilitate model training and evaluation, the dataset is split into training and testing sets, with an 80-20 ratio ensuring adequate training data for robust model performance assessment. The final dataset preparation step involves scaling the data using Min-Max normalization to ensure consistent ranges across features, a crucial prerequisite for effective model training across various algorithms.

3.2 Proposed Architecture

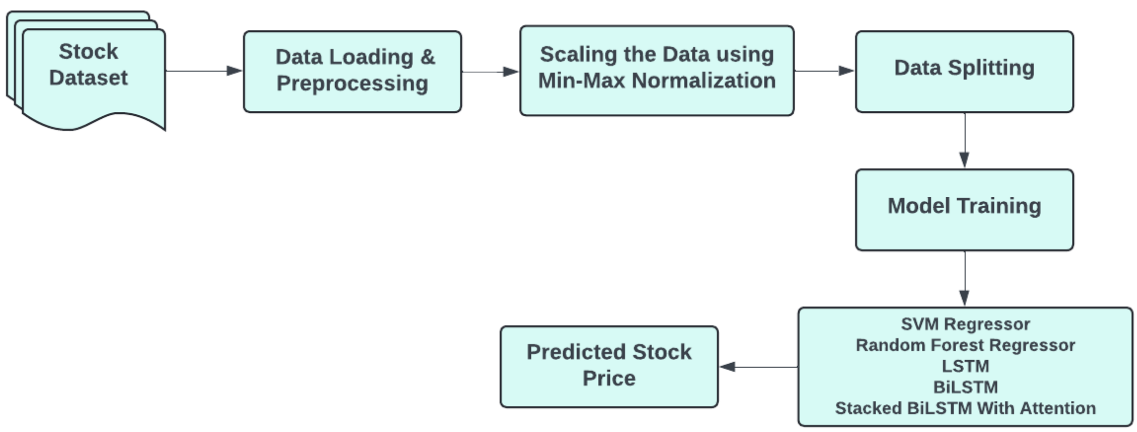

Figure 3.1: Architecture of proposed workflow system

1) **Data Collection:** The data collection process involves reading historical stock data from a CSV file using pandas in Python. The stock name is specified, and the corresponding CSV file containing the historical data is loaded into a pandas Data Frame. The file path is constructed dynamically using the specified stock name. This approach allows for flexible data retrieval and analysis for various stocks. Once the data is loaded, it can be further processed, cleaned, and analysed for model development and prediction purposes.

2) **Data Pre-processing and Cleaning:** Data pre-processing for time series prediction involves preparing the data in a format suitable for analysis and model training. In this scenario, since the data is already cleaned, the focus shifts to pre-processing steps specific to time series forecasting. Firstly, unnecessary columns are removed from the Data Frame to streamline the dataset, retaining only essential features like 'Date' and 'Close' price. The 'Date' column is then set as the index to facilitate time-based operations. Setting the 'Date' as the index is crucial for time series analysis, as it enables efficient time-based slicing and indexing of the data. This step ensures that the temporal aspect of the data is properly accounted for during analysis and model training. Visualizing the time series data using Plotly Express provides valuable insights into the underlying patterns and trends in the stock's closing price over time. This step helps in identifying any seasonality, trends, or irregularities present in the data, which can inform the selection of appropriate forecasting models and techniques. By completing these pre-processing steps, the data is now ready for further analysis and model development. The cleaned and pre-processed dataset serves as the foundation for building accurate and reliable time series forecasting models for predicting future stock prices.

3) **Datetime Feature Extraction:** In this section, datetime feature extraction is performed on the Data Frame's 'Date' column to derive additional temporal features, enhancing the dataset's richness for time series analysis and forecasting. Firstly, the 'Date' column is converted to datetime format using the `pd.to_datetime()` function, ensuring consistency and compatibility for subsequent operations. Next, three new columns are added to the Data Frame: 'Year', 'Month', and 'Day'. These columns capture the year, month, and day components extracted from the 'Date' column using the `dt.year`, `dt.month`, and `dt.day` accessors, respectively. This transformation enables the representation of each data point with granular temporal information, allowing the model to capture potential seasonal and cyclical patterns in the data. By incorporating these datetime features, the dataset becomes enriched with additional contextual information, facilitating more nuanced analysis and modelling. These features provide valuable insights into how stock prices vary across different time periods, aiding in the development of more accurate and robust forecasting models. Ultimately, this datetime feature extraction step enhances the dataset's predictive power and ensures that the model can effectively capture the underlying dynamics of the stock price data.

4) **Scaling Data with Min-Max Normalization:** Min-Max normalization is applied to scale the data between a specified range, commonly between 0 and 1, to ensure uniformity and consistency across features. In this section, the `MinMaxScaler` from scikit-learn is utilized, specifying the feature range as (0, 1). The scaler is instantiated, and then the `fit_transform` method is applied to the data after reshaping it into a 2D array to comply with the scaler's input format requirements. This transformation scales the data such that the maximum value is mapped to 1 and the minimum value is mapped to 0, while preserving the relative distances between data points. Normalization is essential for ensuring that all features contribute equally to model training, preventing features with larger magnitudes from dominating the learning process. By scaling the data using min-max normalization, the dataset is prepared for training machine learning and deep learning models, promoting stable and efficient convergence during the optimization process. This pre-processing step enhances the model's performance and aids in achieving more accurate predictions by mitigating the impact of varying feature scales on model training.

5) **Training various Machine Learning/Deep Learning models:** In the training phase, both machine learning (ML) and deep learning (DL) models are employed to predict stock prices effectively. For ML models, Support Vector Machine (SVM) Regressor and Random Forest Regressor are utilized. SVM Regressor aims to find the hyperplane that best fits the data points in the feature space, while Random Forest Regressor constructs multiple decision trees and averages their predictions to improve accuracy. Moving to DL models, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), are employed for their ability to capture temporal dependencies in sequential data like stock prices. Additionally, Bidirectional LSTM (BiLSTM) networks enhance model performance by processing input data in both forward and backward directions, capturing information from past and future timestamps simultaneously. Furthermore, a more sophisticated approach involves stacking BiLSTM layers with an Attention Mechanism. This mechanism allows the model to focus on relevant parts of the input sequence, dynamically adjusting its attention weights during training to emphasize important temporal features.

RESULTS

4.1 Data Visualization

This figure 4.1 shows pie chart which is based on year wise average price of Stock. In year 2024 there is 22.3%, 2023 there is 18.8%, 2022 there is 18.1%, 2021 there is 19.8%, 2020 there is 11.7% and at last in 2019 there is 9.29%.

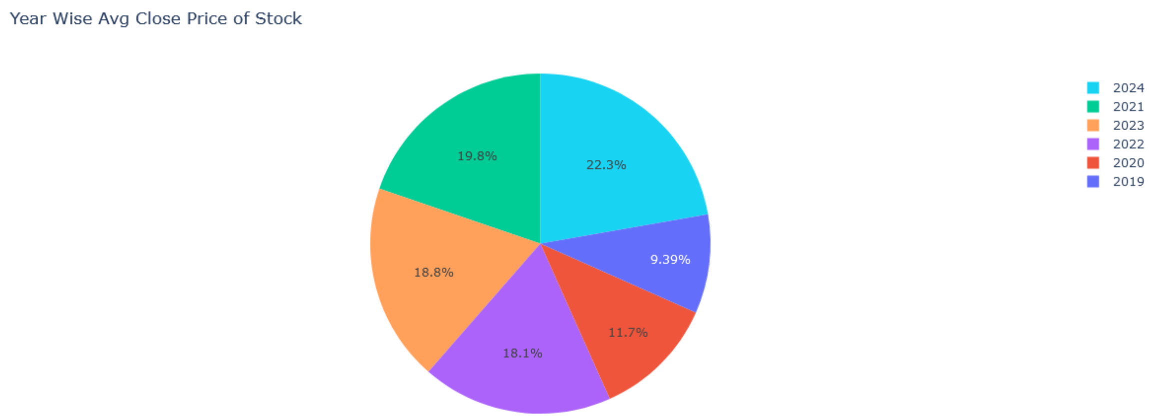

Figure 4.1: Pie Chart

This Figure 4.2 shows Month wise average close price of stock. In month=1, close is 97.8362, in month=6, close is 97.7729 and month=12, close is 106.1068.

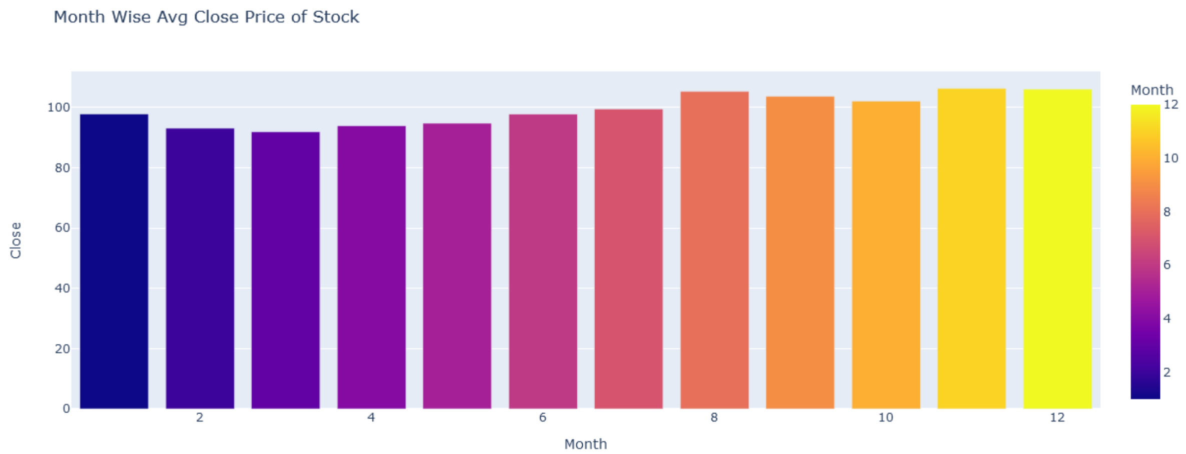

Figure 4.2: Bar Graph

The original stock price data is displayed alongside a rolling average calculated over a predetermined window size in the window rolling graph. In this instance, the rolling mean is computed using a window size of 15 days. The daily variations in stock prices over time are represented by the original price data, which is shown in orange. This line demonstrates the noise and short-term volatility present in financial markets. On the other hand, the rolling average, represented in green, averages prices across successive 15-day intervals, smoothing out the initial price volatility. By eliminating short-term noise and highlighting longer-term patterns or trends in the movement of the stock price, this smoothing effect aids in the visualization of the underlying trend in the data. It facilitates the identification of trends, cycles, and changes in market sentiment by offering insights on the general directionality of the stock's price movement over time. The rolling average minimizes the effect of daily variations while capturing the overall trend of the stock price, as seen by the comparison between the original price and the rolling average. In addition to helping to understand the general behaviour of the stock price, this representation can be helpful for trend research, spotting possible reversal points, and helping investors make well-informed decisions based on longer-term, more stable patterns.

Figure 4.3: Window Rolling Graph

4.2 Support Vector Regressor

With an average squared difference between expected and actual stock prices of 0.0045, the Support Vector Regressor (SVR) model produced this result. The average deviation between projected and actual prices is measured by the Root Mean Squared Error (RMSE), which is equal to 0.067 as the square root of the mean squared error (MSE). The SVR model's ability to explain variation in stock prices is indicated by its R-squared (R2) score of 0.852, which shows that the model has a strong predictive capacity and can account for 85.2% of the variance in the data.

Figure 4.4: Actual vs Predicted Graph

4.3 Random Forest Regressor

With a mean squared difference between expected and actual stock prices of 0.00196, the Random Forest Regressor produced this result. The average difference between actual and expected prices which has been represented by the RMSE, which is 0.0442. The model explains roughly 93.6% of the variance in the stock prices, with an R2 score of 0.936. This indicates the model's good predictive ability and capacity to capture a sizable amount of the underlying variability in the data.

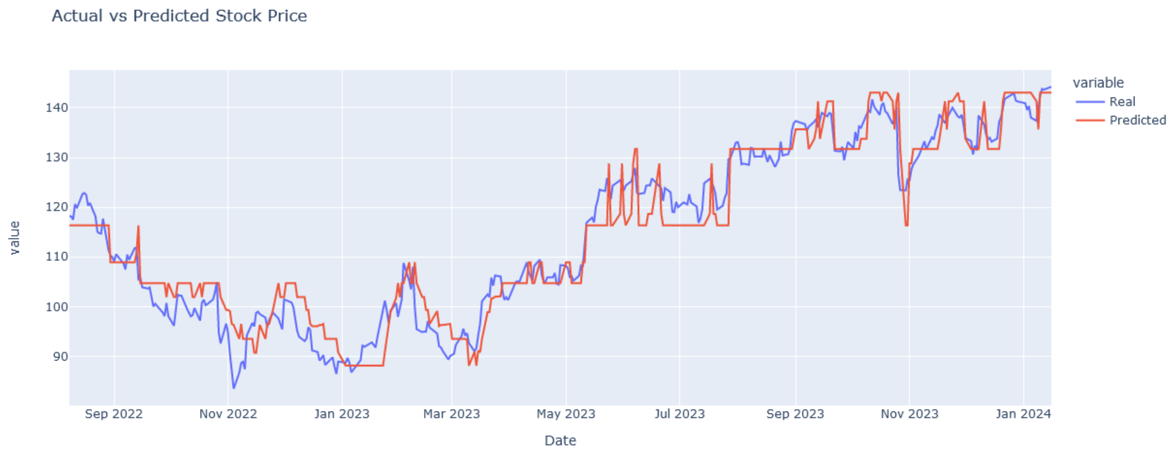

Figure 4.5: Actual vs Predicted Graph

4.4 LSTM Model

With an average squared difference between projected and actual stock prices of 0.00634, the LSTM (Long Short-Term Memory) model produced this result. The average difference between actual and expected prices is shown by the RMSE, which is 0.0796. This is the square root of the mean square error (MSE). With a 0.791 R-squared (R2) score, the model accounts for about 79.1% of the variation in stock prices. Although the LSTM model shows promise in terms of prediction, it only explains a small fraction of the variability in the data when compared to other models, indicating potential for improvement.

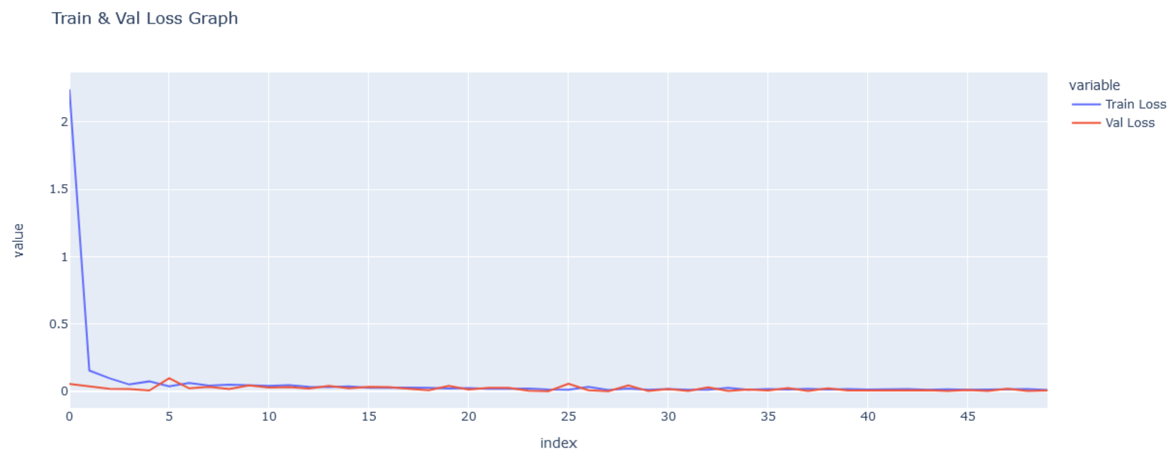

Figure 4.6: Train and Val Loss Graph

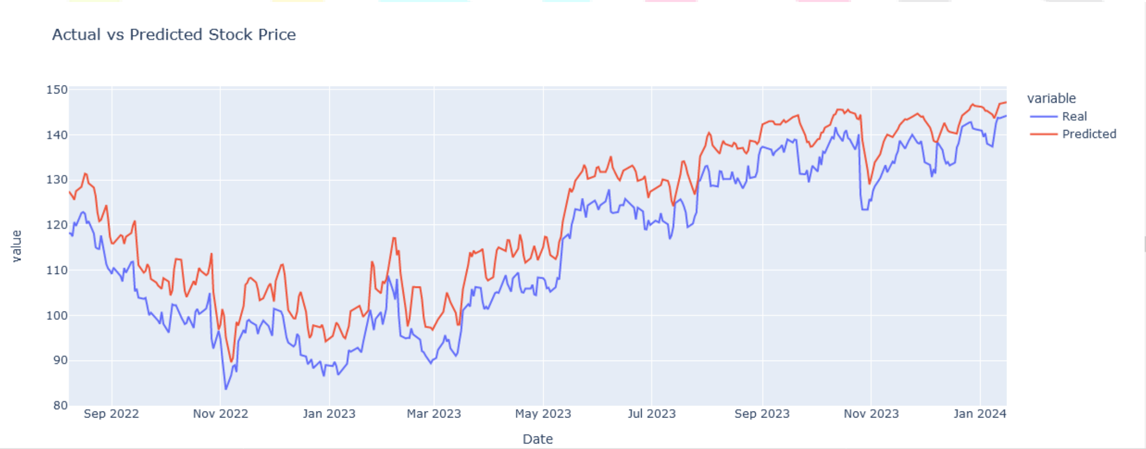

Figure 4.7: Actual vs Predicted Graph

4.5 BILSTM Model

The average squared difference between the predicted and actual stock prices was 0.00216, which is the result of the BiLSTM (Bidirectional Long Short-Term Memory) model. The average difference between actual and expected prices is displayed by the RMSE, which is 0.0465. With a 0.929 R2 score, the model accounts for almost 92.9% of the variation in stock prices. The BiLSTM's great prediction ability and capacity to capture a sizable amount of the underlying variability in the data are demonstrated by its high R-squared score.

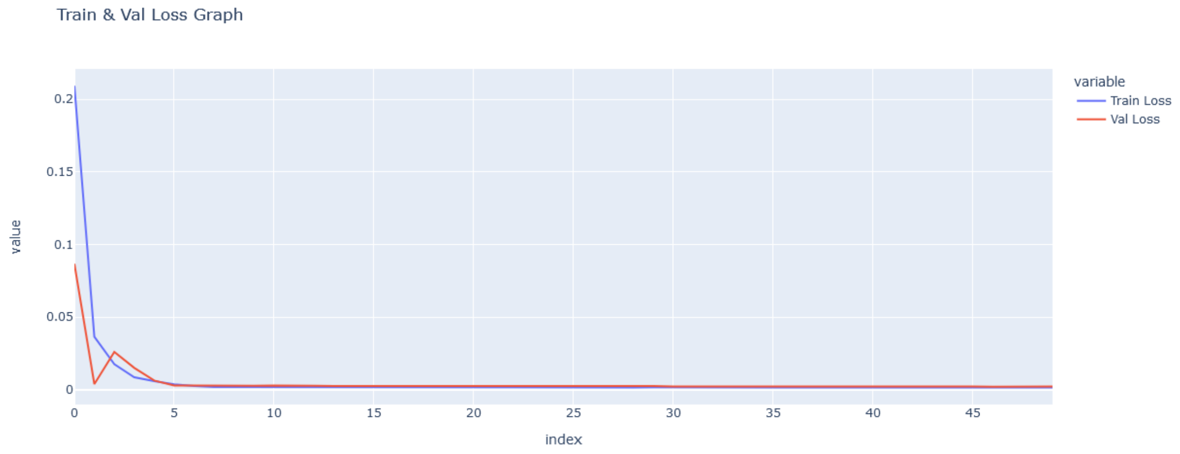

Figure 4.8: Train and Val Loss Graph

Figure 4.9: Actual vs Predicted Graph

4.6 Stacked BILSTM Model with Attention Layer

Using an Attention Layer, the Stacked BiLSTM model performs better than other models with higher accuracy metrics. It shows a small variation between the expected and actual stock values, with an MSE of 0.00109 and a RMSE of 0.0331. The outstanding capacity of the model to explain nearly 96.4% of the variance in stock prices is shown in its high R2 score of 0.964. The attention mechanism and bidirectional LSTM layers in the model's design allow it to capture complex temporal dependencies in the data, leading to unmatched prediction performance and robustness.

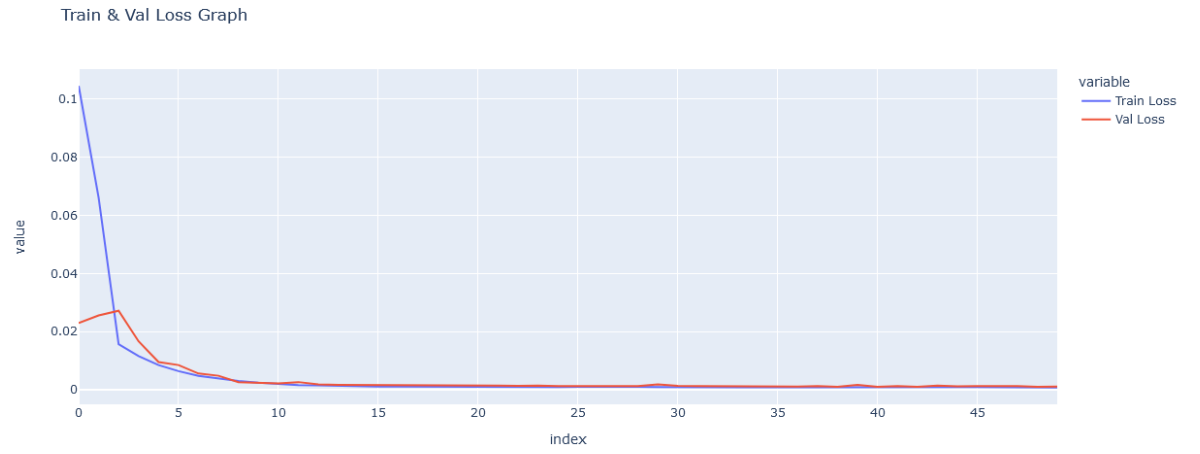

Figure 4.10: Train and Val Loss Graph

Figure 4.11: Actual vs Predicted Graph

Table 4.1: Comparison of ML/DL Models for Stock Price Prediction

Model	**Mean Squared Error (MSE)**	**Root Mean Squared Error (RMSE)**	**R-squared (R2Score)**
SVM Regressor	0.00449	0.0670	0.8521
Random Forest Regressor	0.00196	0.0442	0.9356
LSTM	0.00634	0.0796	0.7910
BiLSTM	0.00216	0.0465	0.9287
Stacked BiLSTM with Attention Layer	0.00109	0.0331	0.9639

4.7 Graphical User Interface

Figure 4.12 is showing home screen of stock price prediction.

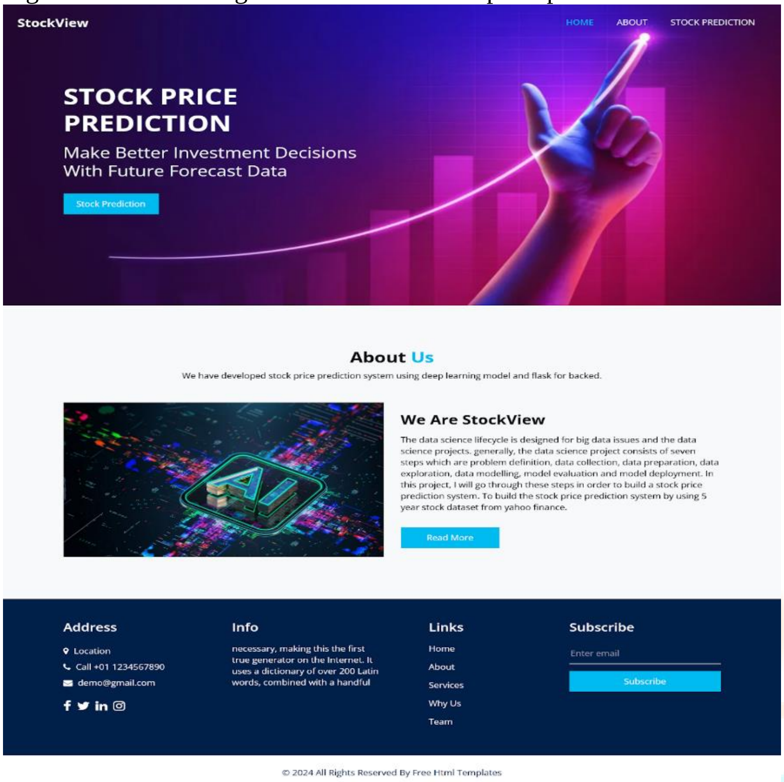

Figure 4.12: Home screen

Figure 4.13 is showing an interface in which first user will select stock by having lots of option and then select date and then click predict stock price in this web app.

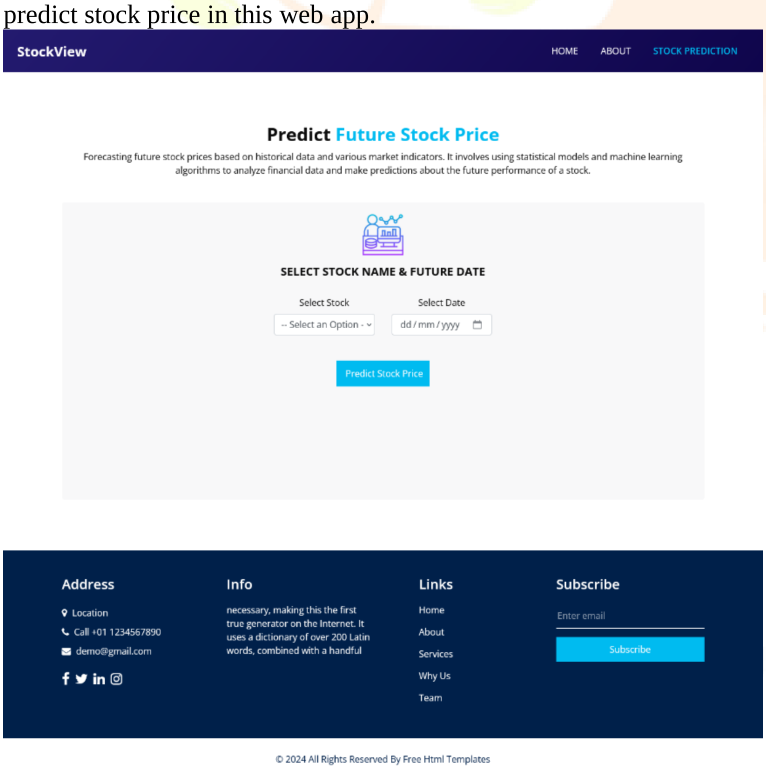

Figure 4.13: Predict stock price

Figure 4.14 is showing that predicted price is something around 131.266.

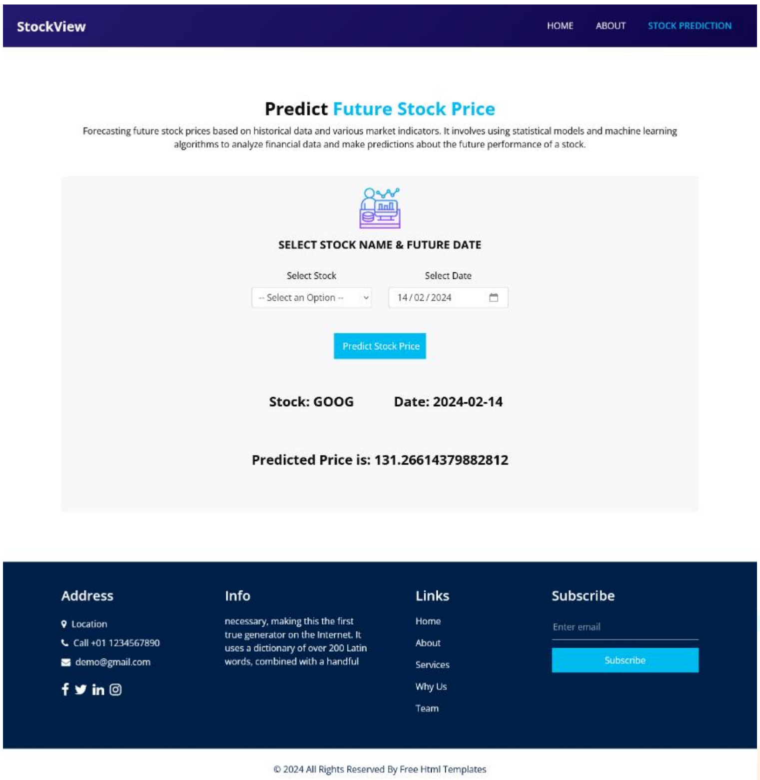

Figure 4.14: Predicted price in web app

CONCLUSION

For this project, Yahoo Finance provides a 5-year collection of daily historical stock data that includes all of the necessary parameters for analysis. Following the import of the required libraries, the data is loaded from a CSV file and cleaned to guarantee its integrity for further analysis. By extracting the day, month, and year components from the dataset, Date Time feature extraction improves it. Perceptive visuals facilitate comprehension of the properties of the data. Window rolling techniques are used in time series forecasting, and the data is evenly scaled using min-max normalization afterward. Model evaluation is made possible by dividing the dataset into training and testing sets. MSE, RMSE, and R-square metrics are used in the training and performance evaluation of deep learning models such as LSTM, BiLSTM, and Stacked BiLSTM with Attention Mechanism, as well as machine learning models such as Random Forest Regressor and SVM Regressor. With the highest accuracy percentage, the Stacked BiLSTM model with Attention Mechanism is the top performer.

REFERENCES

[1] Kumar, G., Jain, S. and Singh, U.P., 2021. Stock market forecasting using computational intelligence: A survey. *Archives of computational methods in engineering*, *28*(3), pp.1069-1101.

[2] Lu, W., Li, J., Wang, J. and Qin, L., 2021. A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, *33*(10), pp.4741-4753.

[3] Hu, Z., Zhao, Y. and Khushi, M., 2021. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, *4*(1), p.9.

[4] Yu, P. and Yan, X., 2020. Stock price prediction based on deep neural networks. *Neural Computing and Applications*, *32*(6), pp.1609-1628.

[5] Islam, M.R. and Nguyen, N., 2020. Comparison of financial models for stock price prediction. *Journal of Risk and Financial Management*, *13*(8), p.181.

[6] Gao, P., Zhang, R. and Yang, X., 2020. The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, *25*(3), p.53.

[7] Soni, P., Tewari, Y. and Krishnan, D., 2022. Machine Learning approaches in stock price prediction: A systematic review. In *Journal of Physics: Conference Series* (Vol. 2161, No. 1, p. 012065). IOP Publishing.