STOCK TRADING APPLICATION

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

PARTHIK SHARMA (20BCB7005)
AUSTIN WILLIAM (20BCB7020)
GELLE HRUTHESH REDDY (20BCB7031)
ABDUL RAB BADRUDDIN (20BCB7045)

Under the Guidance of

PROF. KOLLURI RAJESH

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

DECEMBER 2023

CERTIFICATE

This is to certify that the Capstone Project work titled "STOCK TRADING APPLICATION" that is being submitted by **PARTHIK SHARMA(20BCB7005) AUSTIN WILLIAM (20BCB7020) GELLE HRUTHESH REDDY (20BCB7031) ABDUL RAB BADRUDDIN (20BCB7045)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of Bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have they been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

8/12/23

Prof. KOLLURI RAJESH
Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner 1

8/12/23

Internal Examiner 2

Approved by

8/12/2023

HoD, Networking and Security

School of Computer Science and Engineering

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. G. Viswanathan, Dr. S. V. Kota Reddy, Dr. Jagadish Chandra Mudiganti, Dr. Pradeep Reddy, SCOPE, for providing us with an opportunity to carry out this project during the tenure of the course.

We would also like to extend our guide, Dr. Kolluri Rajesh, for providing us with his constant motivation, understanding, and encouragement in doing this project. We learned many new things. We are truly thankful for his guidance throughout the project. This project would not have been completed without his continuous support.

We thank our friends who motivated us to complete our tasks. We would also like to thank people who have directly or indirectly helped us in the successful completion of our project.

Place: Amaravathi

Date: 8-12-2023

Parthik Sharma - 20BCB7005

Austin William – 20BCB7020

Gelle Hruthesh Reddy - 20BCB7031

Abdul Rab Badruddin - 20BCB7045

ABSTRACT

This report investigates the transformative landscape of financial markets characterized by a surge in individual investors and introduces the groundbreaking Stock Trend App. In response to the democratization of stock trading, this application integrates Artificial Intelligence (AI) and Machine Learning (ML) algorithms, revolutionizing stock market analysis. The Stock Trend App sets forth comprehensive objectives, including accurate predictive analytics, a user-friendly interface, personalized investment recommendations, real-time market updates, educational resources, and effective risk management. Positioned at the nexus of AI and finance, the app envisions an era where intelligent investing is accessible to a diverse user base. By offering a nuanced approach to individual investing, the Stock Trend App strives to bridge the gap between advanced analytics and diverse user profiles, reshaping the landscape of financial decision-making. Through its multifaceted features, the app aims to empower users with the knowledge and tools needed to navigate the dynamic world of stocks confidently. This report provides a detailed exploration of the Stock Trend App, highlighting its significance in the context of evolving financial markets and its potential to redefine the accessibility and intelligence of individual investing.

Keywords: ML, AI, finance, predictive analytics

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent years, the scene of the stock exchange and the venture has gone through a critical change, with a developing number of people effectively partaking in the monetary business sectors. The democratization of stock exchanging, worked with by online stages and the accessibility of data, has prompted an expanded revenue from a different scope of individuals, including prepared financial backers, novices, and those hoping to get their monetary future.

In the midst of this flood in-market support, the requirement for refined devices to explore the intricacies of stock patterns has become central. Perceiving this interest, a state-of-the-art Stock Pattern Application has been created, incorporating the force of Man-made brainpower (simulated intelligence) and AI (ML) calculations. This application addresses a weighty way to deal with financial exchange investigation, offering clients progressed experiences and prescient capacities to settle on informed venture choices.

The steadily developing scene of monetary business sectors, a phenomenal flood of revenue has flooded, as a rising number of people from different foundations effectively partake in stock exchange and speculation. This monetary renaissance, powered by open web-based stages and a hunger for monetary information, has catalyzed a groundbreaking period in venture rehearses.

As the elements of the stock exchange go through a change in outlook, a reference point of development arises - the Stock Pattern Application. Past being a simple application, it remains as a demonstration of the collaboration between Man-made reasoning (simulated intelligence) and the complex universe of financial exchanges. This earth-shattering application isn't simply an instrument for examination; it is a thorough arrangement intended to democratize admittance to cutting-edge market bits of knowledge and enable people on their monetary excursion.

The Rise of Individual Investors:

Lately, the monetary scene has seen a flood in the quantity of individual financial backers entering the securities exchange. The surge in individual investors marks a notable shift in the traditional landscape of stock trading. Fueled by accessible online platforms, educational resources, and a desire for financial autonomy, an increasing number of individuals from diverse backgrounds are actively engaging in the financial markets. This democratization of finance has dismantled barriers to entry, allowing both seasoned investors and newcomers to participate in the dynamic world of stocks.

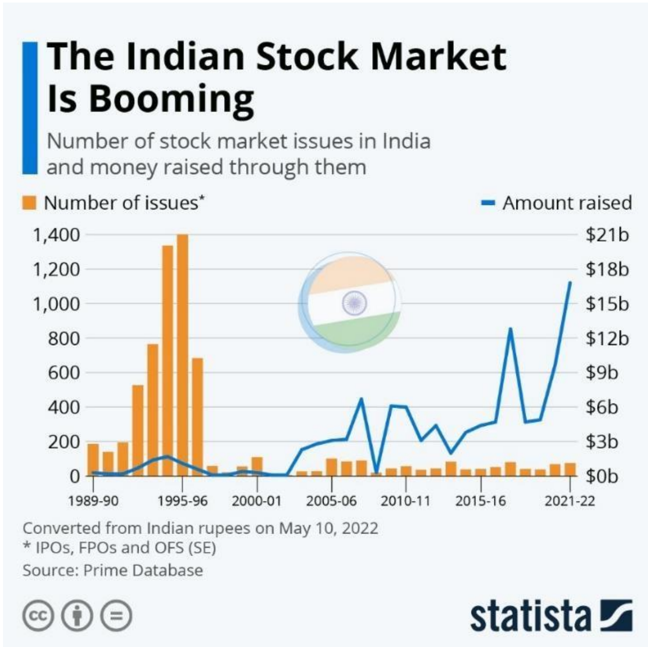


Figure 1.1 Stock trading trends of Indian Stock Market [13]

The rise of individual investors has been further accelerated by the availability of real-time market information, social media discussions, and the desire for alternative investment avenues beyond traditional asset classes. This trend signifies a transformative era, emphasizing the need for tools and platforms that cater to the diverse and evolving needs of individual investors. Energized by open web-based exchanging stages, instructive assets, and a craving for monetary freedom, individuals from different foundations are currently effectively captivated in stock exchanging and speculation.

1.1 OBJECTIVES

- **Risk Management and Portfolio Enhancement:**

  Consolidate highlights that help clients in overseeing speculation gambles, like devices for portfolio expansion and chance evaluation. Instruct clients on the significance of chance administration and give experiences in building even venture portfolios.

- **Consistent Improvement through Client Input:**

  Lay out an input circle to accumulate bits of knowledge from clients with respect to the application's presentation, ideas for upgrades, and extra elements they might see as important. Use client input to execute updates and improvements, guaranteeing that the application advances to meet the changing necessities and assumptions of its client base.

- **Ongoing Business sector Updates:**

  Coordinate ongoing information channels to keep clients informed about the most recent market patterns, news, and occasions that might influence stock costs. Guarantee that clients have convenient admittance to basic data, permitting them to answer quickly to advertise changes.

- **Precise Prescient Analysis:**

  Create and execute AI calculations to investigate authentic stock information and distinguish designs that add to exact forecasts of future stock patterns. Improve the application's prescient abilities to assist clients with settling on informed choices on purchasing, selling, or holding stocks.

- **Easy to understand Point of interaction and Openness:**

  Plan an instinctive and easy-to-understand connection point to guarantee that both beginner and experienced financial backers can without much of a stretch explore the application. Focus on availability highlights to make the application comprehensive, taking care of clients with shifting degrees of monetary aptitude.

- **Customized Investments Suggestions:**

  Execute a framework that tailors speculation suggestions in view of individual client profiles, considering factors like gamble resilience, venture objectives, and verifiable exchanging conduct. Furnish clients with altered procedures that line up with their exceptional monetary targets.

1.2 BACKGROUND AND LITERATURE SURVEY

A similar research paper and project was carried out by M K Ho et al.,[5] School of Mathematics, Technology Park Malaysia. In this paper, three forecasting machine learning models are used including ARIMA, NN model and LSTM models to predict Bursa Malaysia closing stock prices from 2/1/2020 to 19/1/2021. This range of time was chosen because it was the period when the number of COVID-19 cases increased dramatically in Malaysia, which also triggered an unpredictable movement of stock prices in 2020. To choose the best machine learning model, MAPE and RMSE are chosen. Among the three models, the LSTM model has the best performance in Bursa Malaysia stock price prediction because it has the smallest MAPE and RMSE values. For instance, the LSTM model is not only able to generate more than 90% accuracy but also able to explain the unpredictable movement of stock prices during this pandemic period. In the future, financial indicators can be included to improve the accuracy of prediction since these indicators may influence the movement of stock prices. It is also recommended to use sentiment analysis with the LSTM model to make a better prediction.

We also referred to the paper titled "Comparison of ARIMA, ANN and LSTM for Stock Price Prediction" by Qihang Ma [2]. Through the analysis of the establishment process and results of these three models, conclusions can be made. The ANN model is better than that of the ARIMA model, and the performance of the LSTM model may be more due to the ANN. And ARIMA-GARCH can further improve the accuracy of the ARIMA model by improving the white noise sequence. The disadvantage is that, as we all know, the fluctuation of stock prices is not only related to changes in time, but also related to economic factors, socio-political factors, and the listing of other stocks. Although the LSTM model introduces other variables than the other two models to distinguish market fluctuations and sudden changes, the three models are essentially deduced by using possible relationships in the time series without considering other external factors. This is also the direction in which future research can be further in-depth. Of course, the further development and use of the LSTM model in stock price prediction is also a subject of research value

To make the model more precise and error-prone, we referred to another paper whose authors are Haoran Wu et al., [4] titled “Comparison of ARIMA and LSTM for Stock Price Prediction”. This paper uses two forecasting machine learning models, ARIMA and LSTM models to predict Apple closing stock prices from 1/4/2016 to 9/21/2020. To determine the best machine learning model, RMSE is chosen. Of the two models, the LSTM model has the better performance in Apple stock price prediction because it has smaller RMSE values. With recent advances in the development of techniques based on complex machine learning, especially deep learning algorithms, 6the shortcomings of both models have come to light for instance, both the LSTM model and the ARIMA model essentially exploit relationships that may exist in time series without considering other external factors. The fluctuation of the stock price is not only related to the change of time but also affected by many factors including market factors, political factors, macroeconomic factors, industrial factors, and other external influences of the enterprise, as well as the management ability and organizational structure of the company. Although the LSTM model can introduce indicators like RSI and MSCD to better judge market fluctuations and sudden changes, it is suggested to combine the LSTM model with other models for better prediction.

1.3 ORGANIZATION OF THE REPORT

The remaining chapters of the report are given as follows:

Chapter 2: StockTrend Application

Chapter 3: Results and Discussions

Chapter 4: Conclusion

Chapter 5: Appendix

CHAPTER-2

STOCKTREND APPLICATION

2.1 PROPOSED SYSTEM:

Our proposed system, the Stock Trend App, seamlessly integrates cutting-edge Machine Learning algorithms with real-time stock data fetched from Yahoo servers. By providing personalized insights and recommendations, it empowers individual investors to navigate the complexities of the stock market with confidence and intelligence.

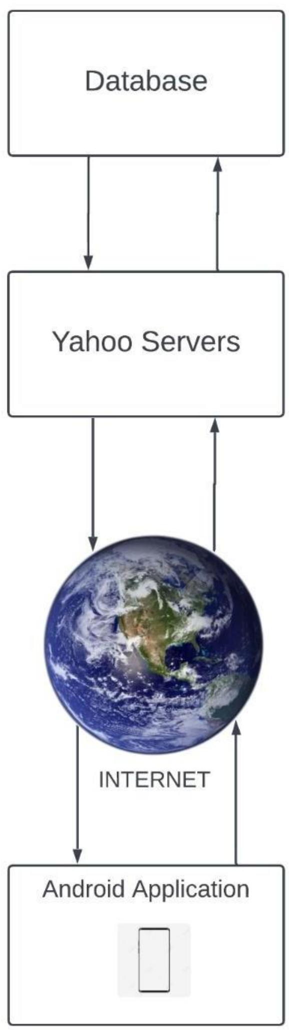


Figure 2.1 The proposed system of Application

The LSTM (Long Short-Term Memory) flowchart illustrates a recurrent neural network architecture, featuring specialized memory cells that allow for the retention and selective updating of information over extended sequences. It visually depicts the sequential flow of operations, including input, update, and output gates, facilitating the understanding of the model's capacity for capturing long-term dependencies in sequential data.

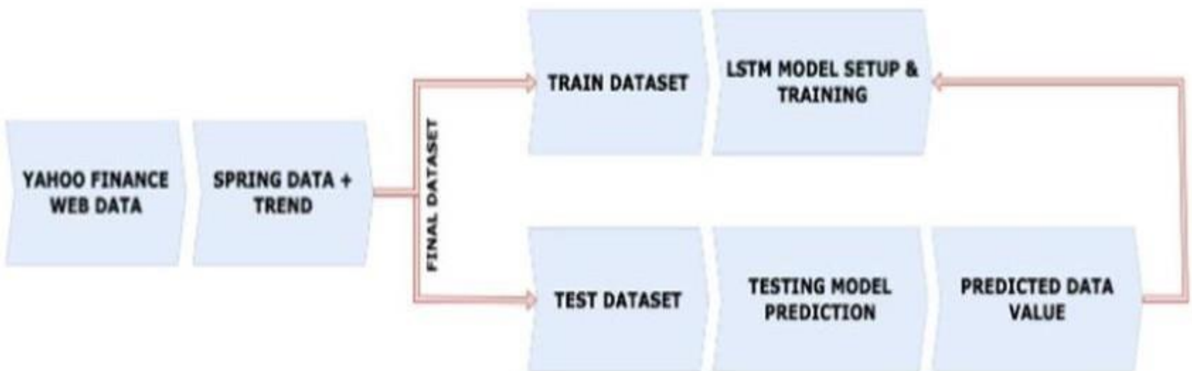


Figure 2.2 Sequence of activities in LSTM [6]

2.2 WORKING METHODOLOGY

Methodology for Stock Trend Analysis Using the Stock Trend App:

a. Data Collection:

Establish a connection with Yahoo servers to fetch real-time stock data for selected companies. Set up a database to store the collected stock values for historical reference and future analysis.

b. Data Preprocessing:

Clean the raw stock data by handling missing values, outliers, and any inconsistencies. Transform the data into a structured format suitable for machine learning, considering factors such as stock prices, trading volumes, and relevant financial indicators.

c. Feature Engineering:

Extract relevant features that can contribute to predicting stock trends, such as moving averages, historical volatility, and technical indicators. Engineer additional features that capture market sentiment, economic indicators, and other external factors that may influence stock prices.

d. Model Development:

Select appropriate machine learning algorithms for stock trend analysis, considering factors like historical performance, interpretability, and computational efficiency. Split the dataset into training and testing sets to evaluate the model's performance accurately.

e. Model Training:

Train the machine learning model on the historical stock data, utilizing the training set to learn patterns and relationships within the data.

f. Model Evaluation:

Assess the model's performance using the testing set, employing metrics such as accuracy, precision, recall, and F1 score. Fine-tune the model parameters to optimize its predictive capabilities.

g. Integration with Stock Trend App:

Integrate the trained machine learning model with the Stock Trend App, enabling it to generate predictions based on real-time stock data.

h. User Interface Design:

Develop an intuitive and user-friendly interface for the Stock Trend App, allowing users to input preferences, view recommendations, and access relevant information.

i. User Feedback Mechanism:

Implement a feedback loop within the app to gather user feedback on the accuracy and usefulness of recommendations.

j. Continuous Improvement:

Periodically update the machine learning model using new stock data and user feedback, ensuring that the Stock Trend App evolves to meet changing market dynamics and user needs.

By following this methodology, the Stock Trend App can effectively utilize machine learning algorithms to analyze stock trends, providing users with valuable insights to make informed decisions regarding the investability of various companies. The iterative nature of the methodology, with feedback mechanisms and continuous improvement, ensures the adaptability and reliability of the Stock Trend App over time.

2.3 SYSTEM REQUIREMENTS

Requirement analysis is a crucial component of product development, playing a pivotal role in assessing the viability of an application. It involves delineating the essential software and hardware prerequisites necessary for the creation of the product or application. This analysis encompasses software requirements, hardware requirements, and functional requirements.

Software requirements pertain to the functionalities that address end-user issues through the utilization of software. The steps involved in identifying software requirements include:

a. **Analysis:** Analysis is the process of comprehending customer needs logically and gaining a more precise understanding of their requirements.
b. **Specification:** Specification entails documenting requirements in the form of use cases, user stories, functional requirements, and visual analyses.
c. **Validation:** Validation is the verification of specified requirements to ensure their accuracy.
d. **Management:** Throughout the development phase, requirements undergo constant changes, necessitating thorough testing and updates to accommodate these changes.

Hardware requirements encompass the physical components essential for an application. Integration of software with hardware is indispensable for the culmination of the development process. The considered hardware requirements include:

a. Processor Cores and Threads
b. GPU Processing Power
c. Memory
d. Secondary Storage
e. Network Connectivity

2.4 SOFTWARE AND HARDWARE REQUIREMENTS

2.4.1 SOFTWARE REQUIREMENTS

Listed below are the software requirements for performing LSTM neural network on Yahoo finance data and making a Stock recommendation App:

a. **Operating System:** The operating system serves as the intermediary between user programs and the kernel. A prerequisite for this application is an operating system of Windows 8 or a later version (64-bit).
b. **Android Studio:** Android Studio, utilizing Kotlin language, is a powerful integrated development environment for efficiently building and testing Android applications.
c. **Jupyter Notebook:** Jupyter Notebook is an open-source web application enabling the creation and sharing of documents featuring live code, equations, visualizations, and narrative text. It serves various purposes, including data cleaning, numerical simulation, statistical modeling, data visualization, and machine learning.
d. **Dataset:** The dataset utilized in this project was sourced from Yahoo Finance through their API, providing reliable and up-to-date financial data for analysis and modeling.

2.4.2 HARDWARE USED

a. **Processor:** Intel i5 2.5 Ghz upto 3.5Ghz (or AMD equivalent)
b. **GPU (preferred):** dedicated GPU from NVIDIA or AMD with 4GB VRAM
c. **Memory:** minimum 8GB RAM
d. **Secondary Storage:** minimum 128GB SSD or HDD
e. **Network Connectivity:** bandwidth ~ 10 Mbps 3 75 Mbps

2.5 FUNCTIONAL AND NON–FUNCTIONAL REQUIREMENTS

2.5.1 FUNCTIONAL REQUIREMENTS

Functional requirements are those specifying the system's functions, covering aspects like real-time stock data retrieval, recommendation algorithms, news integration, and other technical operations.

a. **Stock Data Retrieval:** Integrate APIs to fetch real-time stock market data for various stocks.
b. **News Integration:** Integrate financial news feeds to keep users informed about market events influencing stock prices.
c. **Data Pre-processing:** This process involves cleaning, transforming, and reducing data to convert raw data in a useful manner.
d. **Training:** Initially, the system must train based on the data set given. The training period is when the system learns how to perform the required task based on the inputs given through the dataset.
e. **Forecasting:** Forecasting involves predicting future outcomes through the analysis of past and present data, typically focusing on stock trend analysis.
f. **Evaluation:** To know whether the system is working efficiently, the generated stock data is reintroduced into the classifier, determining the type of stock forecast produced and ensuring the effectiveness of the overall forecasting system.

2.5.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the particulars that serve as measurable criteria for assessing the performance of the system. These include:

a. **Accuracy:** This refers to the number of correct outputs to the total number of outputs.
b. **Openness:** The system should work efficiently for a certain time.
c. **Portability:** Portability is a key consideration, requiring the system to be designed in a manner that is platform-independent, facilitating its operation across multiple systems without necessitating extensive modifications.
d. **Reliability:** The system must produce fast and accurate results.

2.6 SYSTEM DETAILS

This section describes the software details of the system:

2.6.1 SOFTWARE DETAILS:

Android Studio and Jupyter Notebook are used.

a. **Android Application:**

The Android application is built on the platform called Android Studio where Kotlin language is used to write code for the app. It encompasses a powerful code editor, intuitive UI design tools, an emulator for testing, robust build and debugging features, and seamless integration with version control systems.

Developing Mobile Application:

- Start by setting up a new Android Studio project, and selecting the appropriate templates and configurations.
- After setting up a new project in Android Studio, the next crucial step is crafting the layout of your stock recommendation app. Begin by designing an intuitive and visually appealing user interface using Android Studio's layout editor.

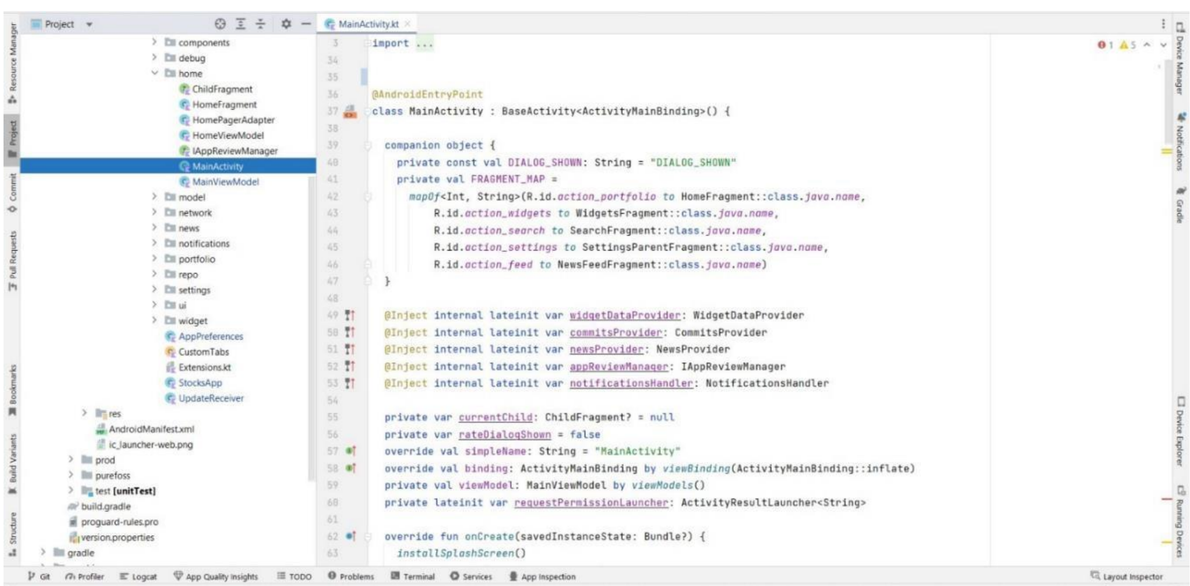


Figure 2.3 Code for Homepage

- Add layout for homepage “Watchlist”, enabling users to monitor selected stocks, providing key metrics and timely updates for informed decision-making.

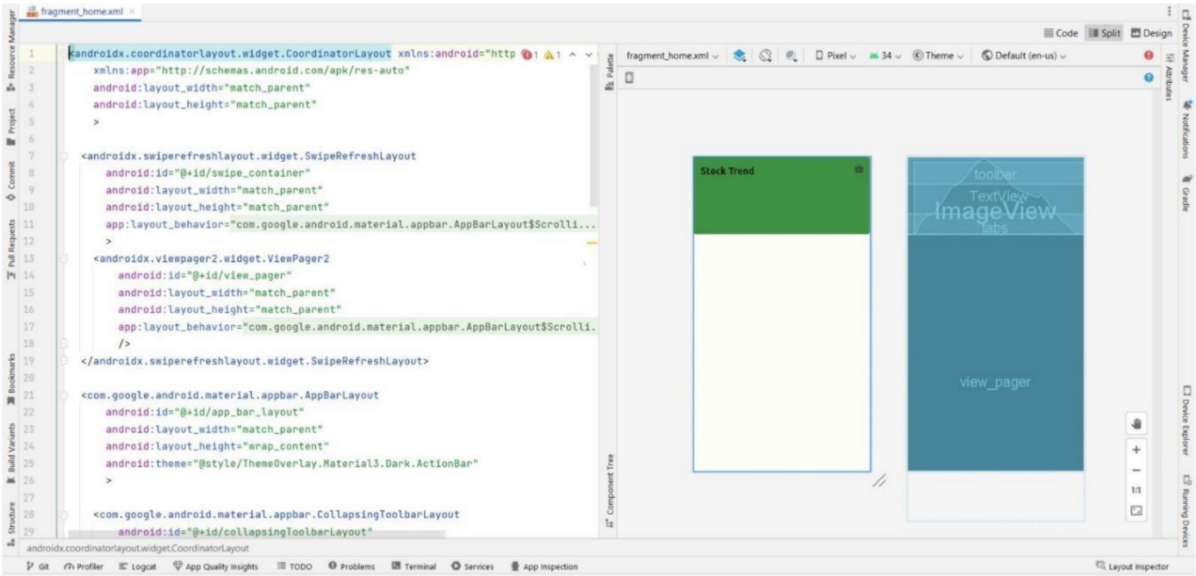


Figure 2.4 Code for Watchlist Page

- Likewise add layouts for each tab such as incorporating "Widgets" functionality for displaying key financial metrics or dynamic charts, enhancing the user experience with at-a-glance insights.

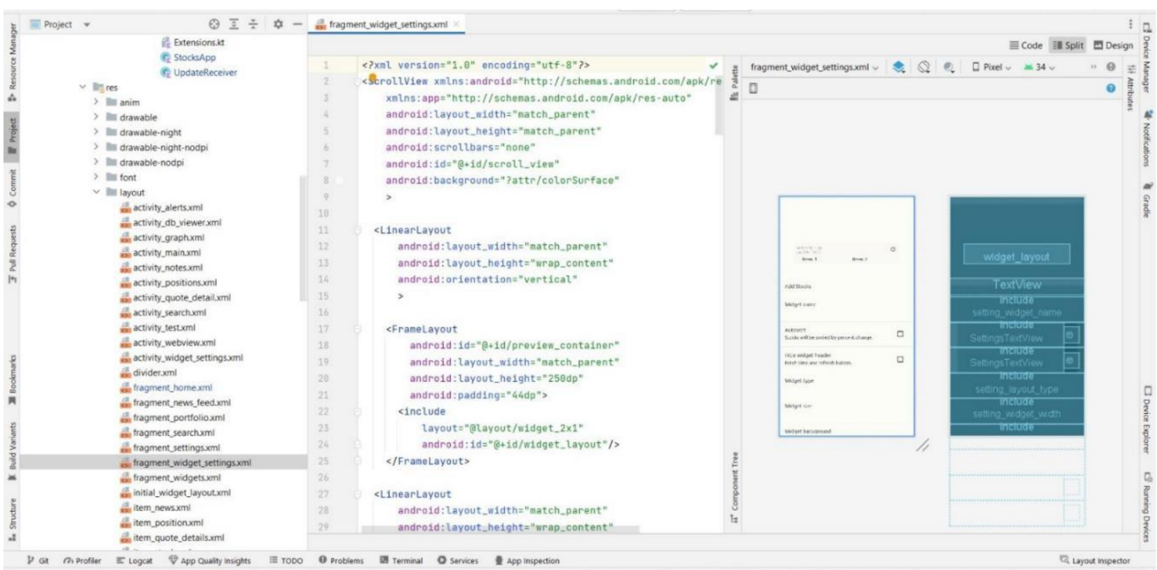


Figure 2.5 Code for Widget

- Likewise add layout for “Search” functionality to enable users to explore and find specific stocks of interest efficiently.

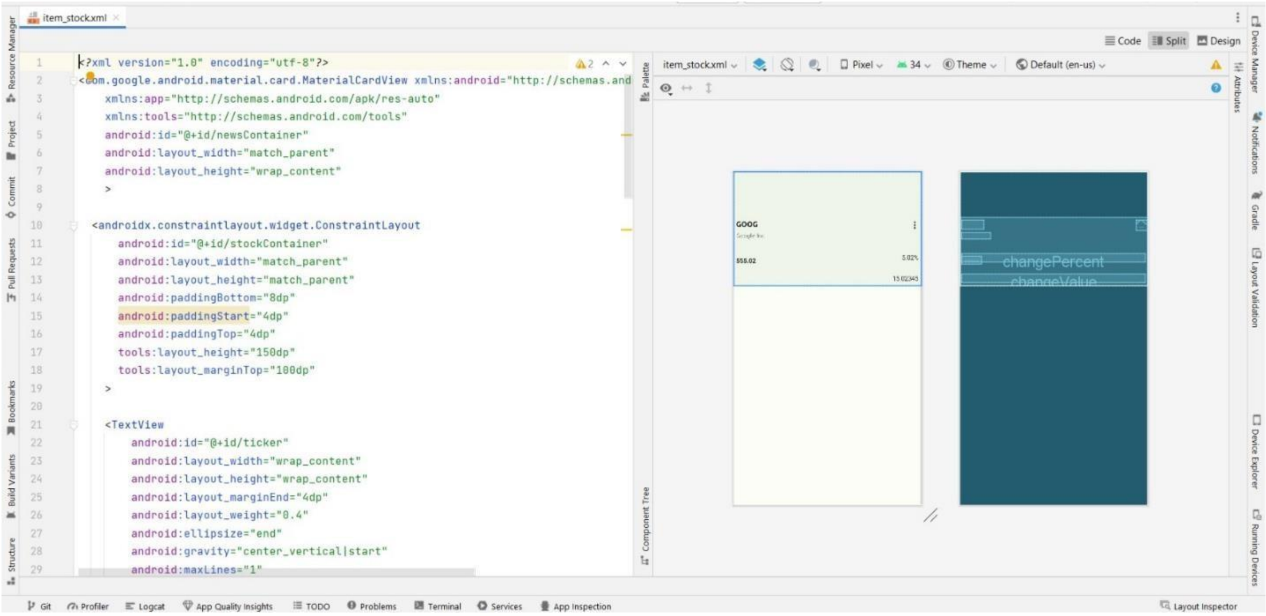


Figure 2.6 Code for Search Page

b. Algorithm

- **Long Short-Term Memory Network:** Long Short-Term Memory Networks, commonly known as LSTMs, represent a specialized type of Recurrent Neural Networks (RNNs) crafted to grasp long-term dependencies in data.
- LSTMs excel across diverse problem domains due to their inherent capability to circumvent the challenges associated with long-term dependency issues. Unlike traditional RNNs, LSTMs are explicitly engineered to seamlessly retain information over extended periods, making them particularly adept at handling tasks where sustained memory is crucial.
- The integral element of the LSTM architecture lies in its memory cell and gates, including the forget gate and input gate. The inner content of the memory cell is influenced by these gates, where the input gates and forget gates regulate the modification of the cell's contents.

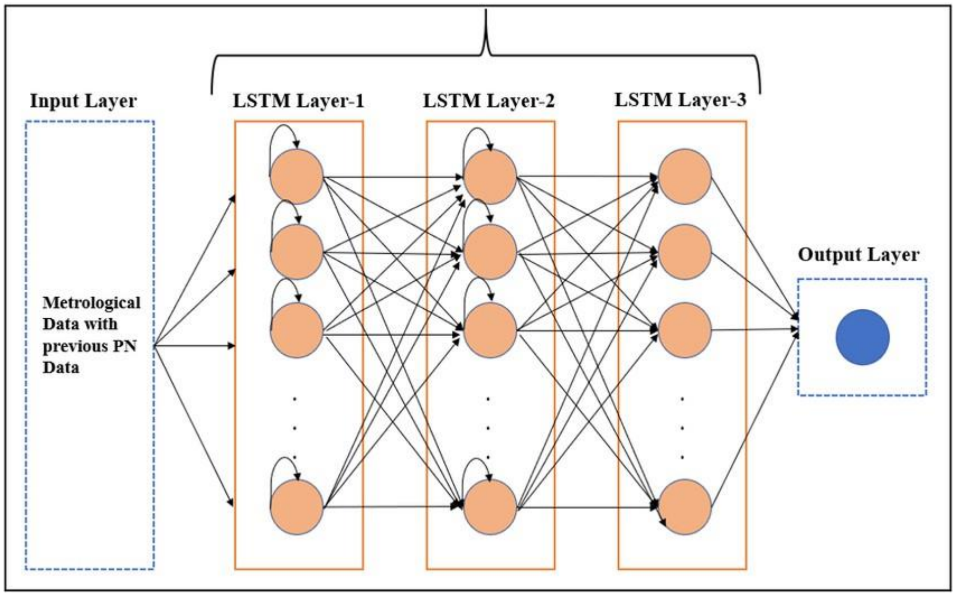


Figure 2.7 LSTM network [8]

- When both gates are closed, the memory cell contents remain unchanged from one time step to the next, enabling the LSTM's unique gating structure to retain information across multiple time steps. This characteristic addresses the issue of vanishing gradients commonly encountered in many Recurrent Neural Network models.

Architecture of Long Short-Term Memory Network:

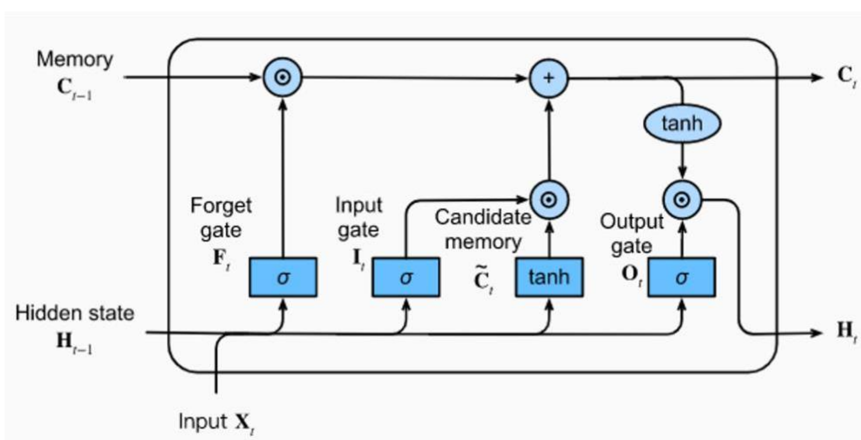


Figure 2.8 Architecture of LSTM network [8]

The architecture of a Long Short-Term Memory (LSTM) network comprises several essential elements:

- **Memory Cell:** The central component that preserves information over time, possessing an internal state subject to modification by various gates.
- **Input Gate:** Determines the amount of new information to be stored in the memory cell, regulating the input to the memory cell.
- **Forget Gate:** Manages the extent to which previously stored information in the memory cell should be discarded, deciding on the relevance of existing data.
- **Output Gate:** Controls the information output from the memory cell to the next time step, shaping the next hidden state based on the current input and memory cell content.
- **Hidden State (or Output):** Represents the information forwarded to the subsequent time step, serving as a filtered version of the memory cell content determined by the output gate.

c. Comparison Between LSTM vs ARIMA vs SVM:

Table 2.1 Comparison Between LSTM vs ARIMA vs SVM

LSTM	ARIMA	SVM
LSTM is a type of recurrent neural network (RNN) designed for sequential data, making it effective for time series forecasting in stock prediction.	ARIMA decomposes time series data into autoregressive, differencing, and moving average components, making it suitable for capturing trends and seasonality.	SVM can handle non-linear relationships in data, making it versatile for capturing intricate patterns in stock price movements
Utilizes a memory cell and gating mechanisms to capture long-term dependencies in data, enabling it to learn intricate patterns in stock trends.	ARIMA models are relatively simple and provide interpretable results, making them useful for understanding underlying patterns in stock data.	SVM models can become computationally complex, especially with large datasets and high-dimensional feature spaces.
Suitable for complex and non-linear relationships in data, making it well-suited for capturing the dynamic nature of stock market fluctuations.	ARIMA models have limited memory, which may restrict their ability to capture long-term dependencies in complex stock market data	This complexity may lead to increased training times and resource requirements. SVMs may face scalability challenges when dealing with large datasets.
Training an LSTM model can be computationally intensive, especially with large datasets, but its ability to capture complex patterns often justifies the computational cost.	ARIMA assumes stationarity in the data, which can be a limitation when dealing with non-stationary stock price movements.	While SVMs provide powerful predictive capabilities, interpreting the resulting model and understanding the contribution of each feature can be complex, especially in non-linear kernel settings.

d. Why LSTM?

1. Temporal Dependency Mastery:

LSTM's prowess in capturing intricate temporal dependencies is pivotal for accurately predicting market trends, setting it apart from traditional models.

2. Flexibility in Non-Linear Modeling:

The recurrent architecture of LSTM allows for flexible adaptation to non-linear financial behaviors, surpassing SVM and addressing ARIMA's limitations in handling non-linear trends.

3. Proficiency in Handling Prolonged Trends:

LSTM's sequence proficiency excels in stock prediction by effectively capturing prolonged trends, overcoming challenges faced by ARIMA and Random Forest in this aspect.

4. Automatic Feature Extraction:

LSTM's automatic feature extraction capability eliminates the need for manual engineering, distinguishing it from SVM and Random Forest, which rely on predefined features.

5. Dynamic Adaptability:

LSTM's adaptability to changing market dynamics and evolving patterns contributes to its effectiveness in forecasting, providing a dynamic edge over static models like SVM.

6. Robust Handling of Sequential Data:

The inherent design of LSTM is tailored for sequential data processing, making it particularly robust in handling the time-sensitive nature of stock prices, outperforming models like Random Forest.

7. Long-Term Memory Retention:

LSTM's ability to retain information over extended sequences allows it to capture subtle trends and patterns over time, offering an advantage over models with limited memory capabilities such as traditional SVM.

8. Reduced Sensitivity to Noisy Data:

LSTM's architecture exhibits a higher tolerance for noisy data, making it more resilient to fluctuations in stock prices and enhancing its reliability in the presence of market uncertainties compared to SVM.

9. Improved Accuracy in Time-Series Forecasting:

LSTM's success in time-series forecasting, a critical aspect of stock prediction, makes it a preferable choice over SVM, which may not inherently excel in capturing the sequential nature of financial data.

10. Enhanced Learning from Historical Data:

LSTM's deep learning capabilities enable it to learn complex patterns from historical stock data, offering superior insights compared to traditional models like ARIMA, which may struggle with the depth of learning required in financial time series.

e. Packages/Libraries Used:

Keras: Keras stands out as a high-level neural network application programming interface (API) implemented in Python. Designed for simplicity and user-friendly interaction, Keras functions as a layer of abstraction compatible with multiple deep learning frameworks like TensorFlow, Theano, or Microsoft Cognitive Toolkit. Its user-friendly approach makes it accessible to both novices and seasoned researchers, providing a modular and intuitive platform for constructing neural networks efficiently. Supporting various neural network architectures, including convolutional networks (CNNs) and recurrent networks (RNNs), Keras has become a widely embraced and flexible tool for diverse deep learning applications, particularly with its integration into TensorFlow since version 2.0.

NumPy: NumPy is a robust numerical computing library for Python that enables efficient handling of large, multi-dimensional arrays and matrices. It offers a comprehensive set of mathematical functions tailored for operations on these arrays. Renowned for its speed and effectiveness, NumPy is extensively employed in scientific computing, data analysis, and machine learning, playing a foundational role in numerous Python libraries within the realms of data science and machine learning.

Pandas: Pandas are a widely used Python library for data manipulation and analysis. It offers flexible data structures, including Data Frames, designed for efficient handling and analysis of structured data. With robust capabilities for cleaning, transforming, and exploring datasets, pandas are a key tool in data science, machine learning, and various fields that involve the manipulation and analysis of tabular data. Recognized for its ease of use and effectiveness, it has become a foundational component in the Python ecosystem for tasks related to data management.

Matplot: matplotlib.pyplot module, a popular library for creating diverse static, animated, and interactive visualizations. matplotlib is widely employed for generating various types of plots, including line plots, bar charts, scatter plots, and histograms. By using the shorthand plt as an alias for pyplot, this library provides a convenient interface for creating compelling data visualizations in scientific computing, data analysis, and other fields that require effective data representation.

Yahoo Finance: This library is commonly employed to retrieve financial data, particularly related to the stock market, from Yahoo Finance. Its interface provides a straightforward means of accessing historical market data, stock quotes, and other financial information utilizing the Yahoo Finance API.

Date Time: Python imports the datetime class from the datetime module. This class is crucial for handling date and time operations, providing functionalities for creating, modifying, and formatting date and time objects.

CHAPTER 3

RESULTS AND DISCUSSIONS

3.1 APPLICATION FOR STOCK RECOMMENDATION

- The main page of our application is the "Watchlist," showcasing significant stocks chosen by the viewer.

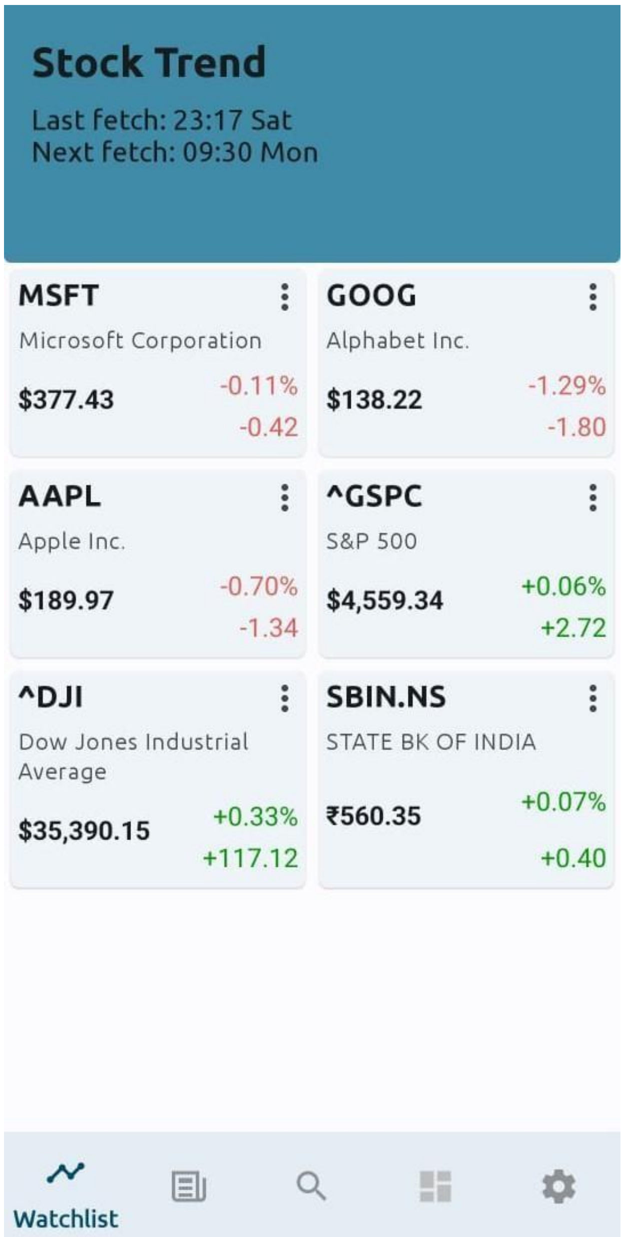


Figure 3.1 Watchlist page

- The second tab features trending stocks, presenting a display of all currently trending stock options, and it also fetches trending market news from Google, Investopedia and Yahoo.

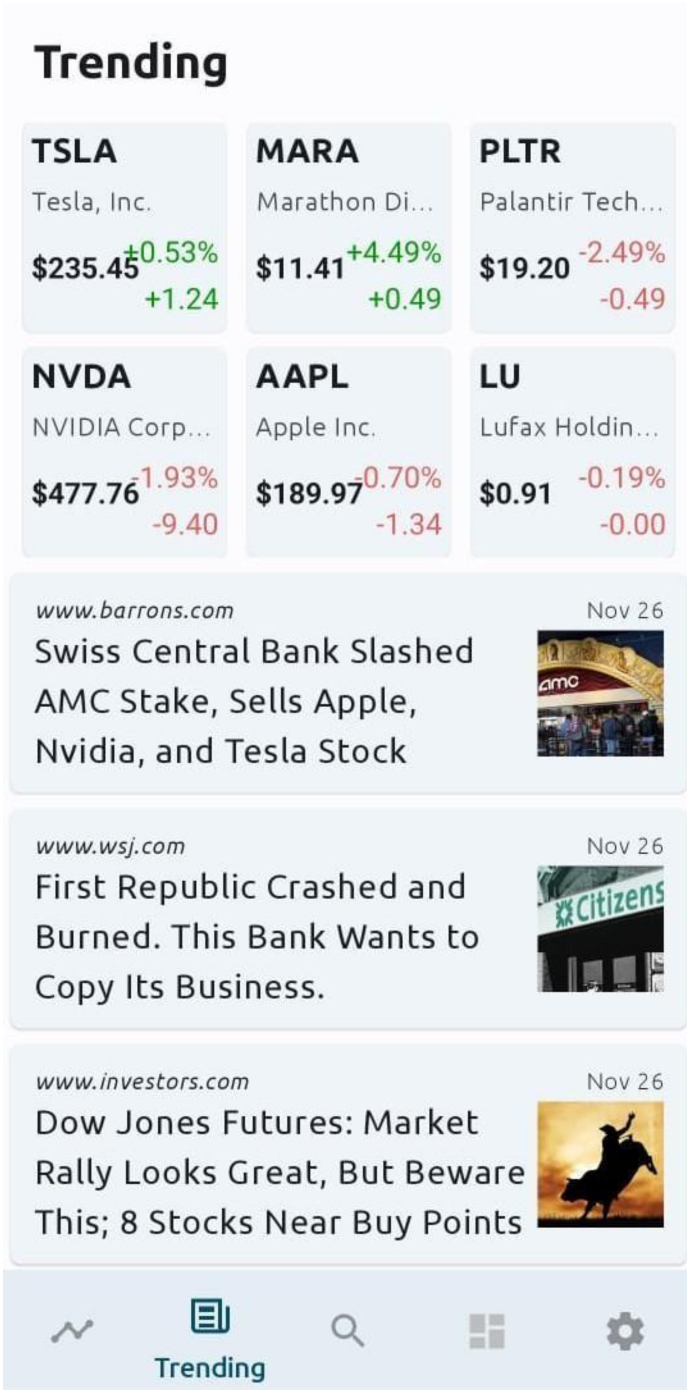


Figure 3.2 Trending page

- The third tab is dedicated to searching, allowing users to explore all stocks available in the Yahoo Finance API.

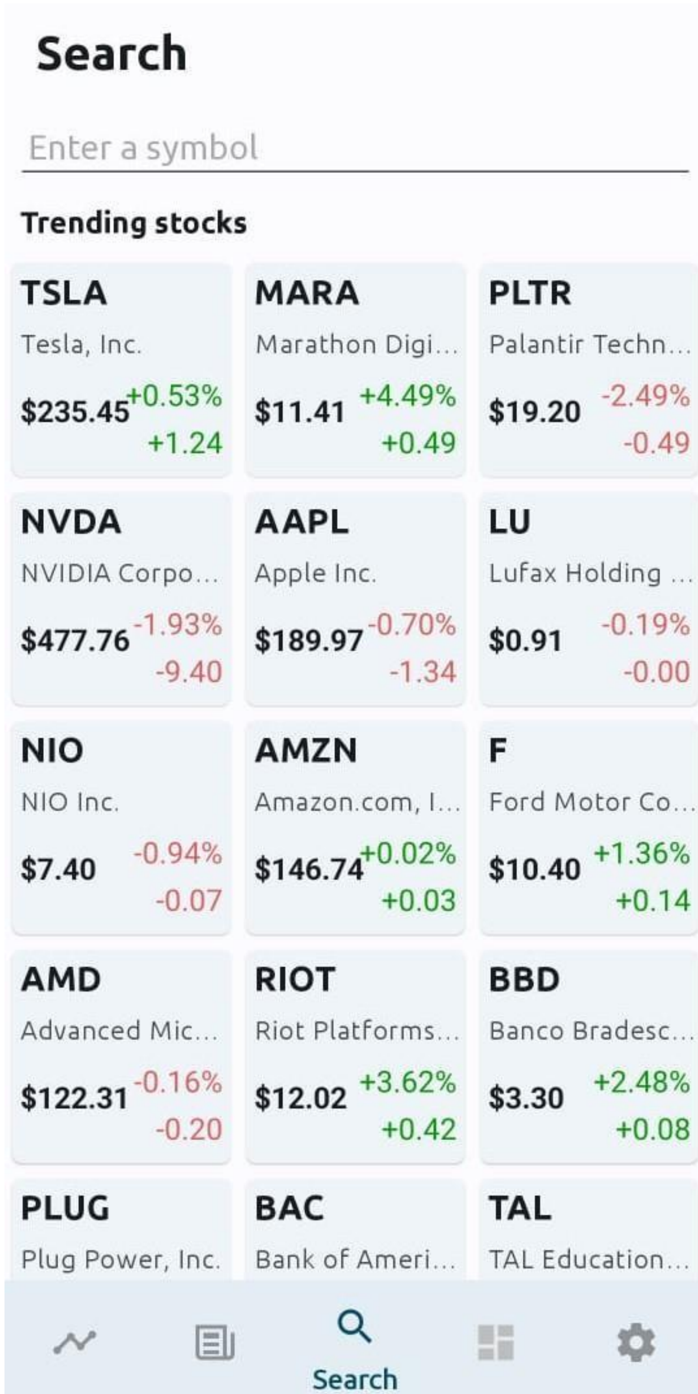


Figure 3.3 Search page

- In this section, details of the chosen company's stock are presented, encompassing metrics like opening prices, PE ratio, and Market cap. Additionally, a graphical representation illustrates the stock's trend, providing a visual indicator of whether it is ascending or descending.

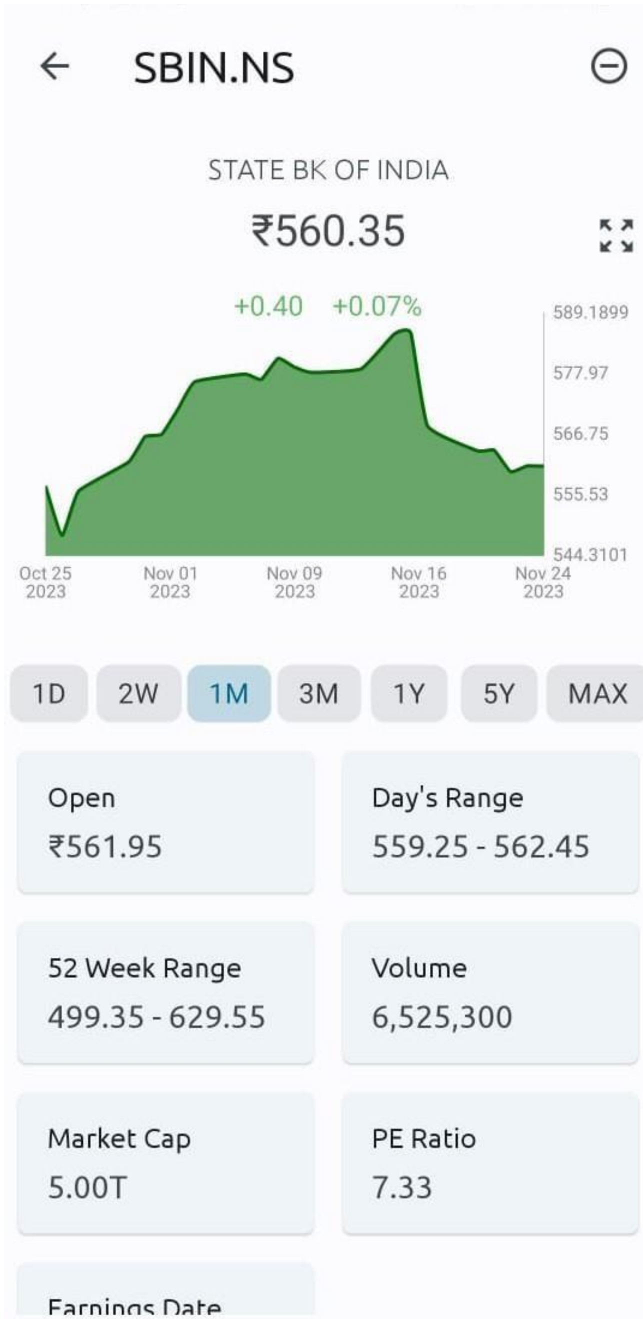


Figure 3.4 Company details page

- The fourth tab is designated for widgets, emphasizing tabs related to invested stocks.

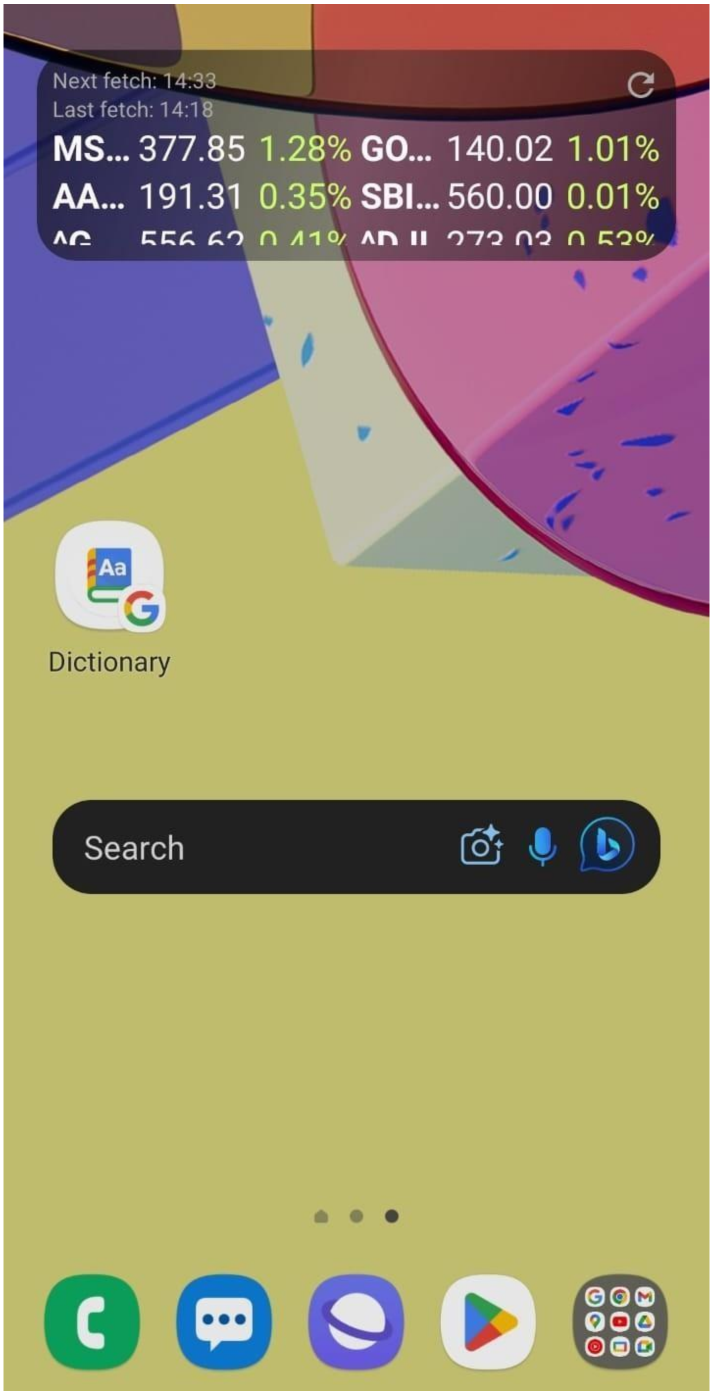


Figure 3.5 Application widget

3.2 LONG SHORT-TERM MEMORY (LSTM)

- The Yahoo Finance dataset for LSTM has undergone rigorous preprocessing, including data cleaning, normalization, and sequence structuring. Feature selection, target variable definition, and a train-test split were implemented, optimizing the dataset for LSTM model training. These efforts enhance the model's ability to discern meaningful temporal patterns in stock market data.

```
[ ]  from sklearn.preprocessing import MinMaxScaler
     scaler=MinMaxScaler(feature_range=(0,1))
     train_data_transform=scaler.fit_transform(train_data)
     train_data_transform

     array([[0.02191227],
            [0.02174975],
            [0.01668473],
            ...,
            [0.57749184],
            [0.59371613],
            [0.55205845]])
```

Figure 3.6 LSTM patterns

- Following the meticulous preprocessing steps, the LSTM model is created to harness the optimized dataset. Leveraging its sequential learning capabilities, the LSTM model is designed to effectively capture temporal dependencies and intricate patterns in the preprocessed Yahoo Finance dataset, laying the foundation for accurate stock price predictions.

```
model.add(LSTM(units=80,activation='relu',return_sequences=True))
[ ] model.add(Dropout(0.4))

model.add(LSTM(units=120,activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(units=1))
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 dropout (Dropout)           (None, 100, 50)           0

 lstm_1 (LSTM)               (None, 100, 60)           26640

 dropout_1 (Dropout)         (None, 100, 60)           0

 lstm_2 (LSTM)               (None, 100, 80)           45120

 dropout_2 (Dropout)         (None, 100, 80)           0

 lstm_3 (LSTM)               (None, 120)               96480

 dropout_3 (Dropout)         (None, 120)               0

 dense (Dense)               (None, 1)                 121

=================================================================
Total params: 178,761
Trainable params: 178,761
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.7 Model Summary of LSTM

- After training and testing the LSTM model with the preprocessed dataset, the outcome is a set of predicted stock trends. Leveraging the learned patterns and temporal dependencies, the model provides forecasts for stock price movements, offering valuable insights into potential market trends and aiding in informed decision-making.

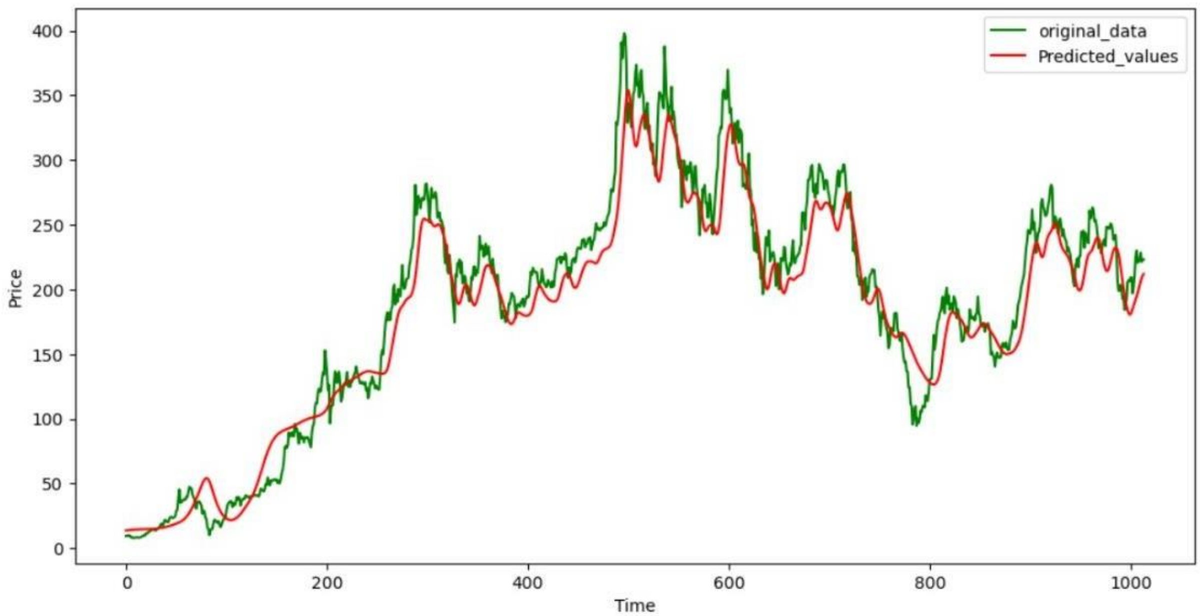


Figure 3.8 Graph comparing original data vs predicted data

- After predicting, the model's performance is quantified using key metrics. The Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) values are assessed. These metrics offer a comprehensive evaluation of the model's accuracy, providing valuable insights into the predictive capabilities and reliability of the LSTM model in forecasting stock trends.

```
mape = np.mean(np.abs(y_pred - y_test)/np.abs(y_test))
print('MAPE: '+str(mape))
```

```
MSE: 459.28192507416196
MAE: 16.774389197530738
RMSE: 21.430863843395628
MAPE: 4.016343777348667
```

Figure 3.9 Model performance

3.3 COMPARISON OF PREDICTED VALUES OF LSTM WITH ARIMA, SVM AND RANDOM FOREST

- To visualize the predictions made by the ARIMA model for stock prices, a plot is generated. This graphical representation illustrates the forecasted stock values against the actual observed values over a specific time. The plot aids in assessing the model's accuracy and its ability to capture trends and fluctuations in the stock market.

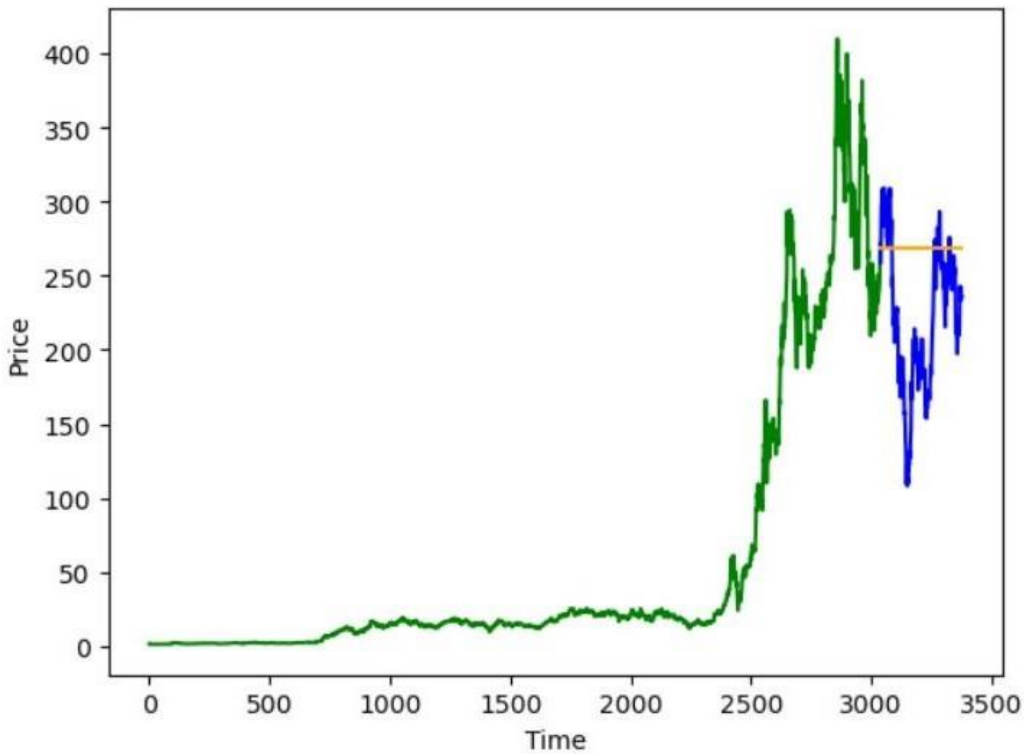
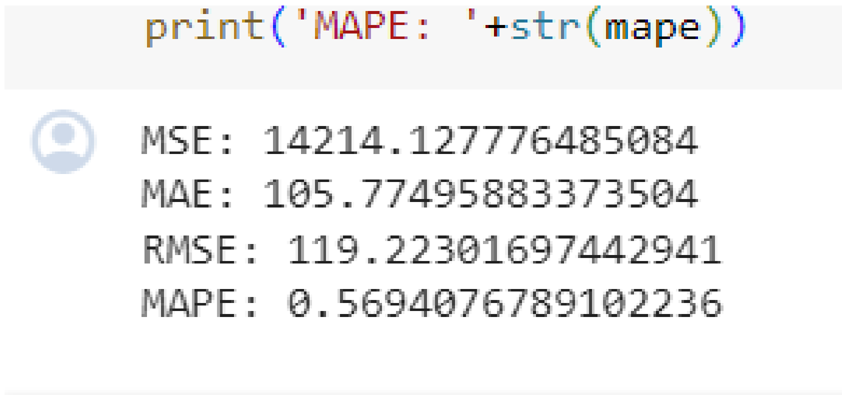


Figure 3.10 Graph of ARIMA model

- The Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) values are assessed. These metrics offer a comprehensive evaluation of the model's accuracy.



```
print('MAPE: '+str(mape))

MSE: 14214.127776485084
MAE: 105.77495883373504
RMSE: 119.22301697442941
MAPE: 0.5694076789102236
```

Figure 3.11 ARIMA model performance

- After training and testing, the following is the output for the SVM model:

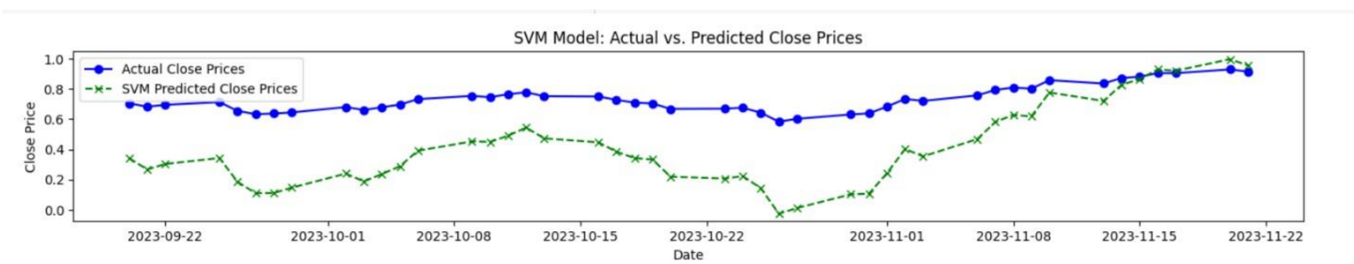


Figure 3.12 Graph for SVM model

- After training and testing, the following is the output for the Random Forest Regressor model:

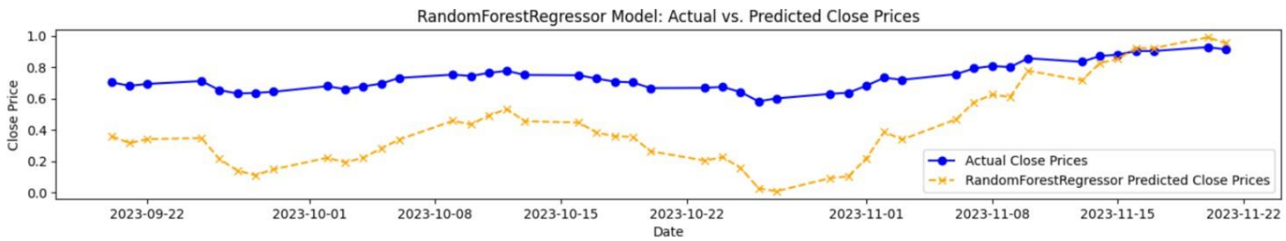


Figure 3.13 Graph of Random Forest

- After training and testing, the following is the output for the Ensemble model.

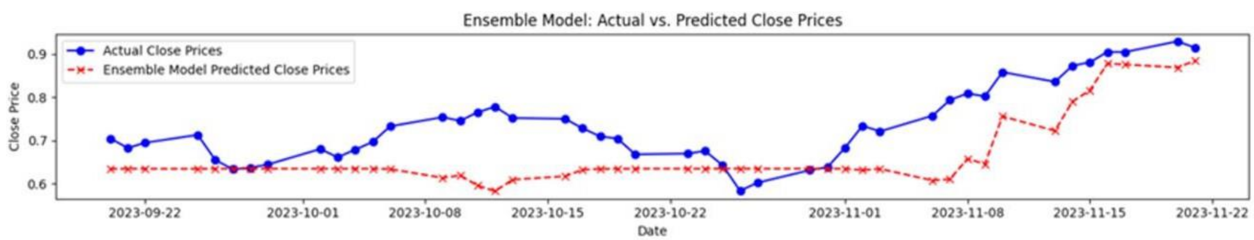


Figure 3.14 Graph of Ensemble Model

- These are the final predictions of SVM, Random Forest Regressor, and Ensemble mode on Closing prices:

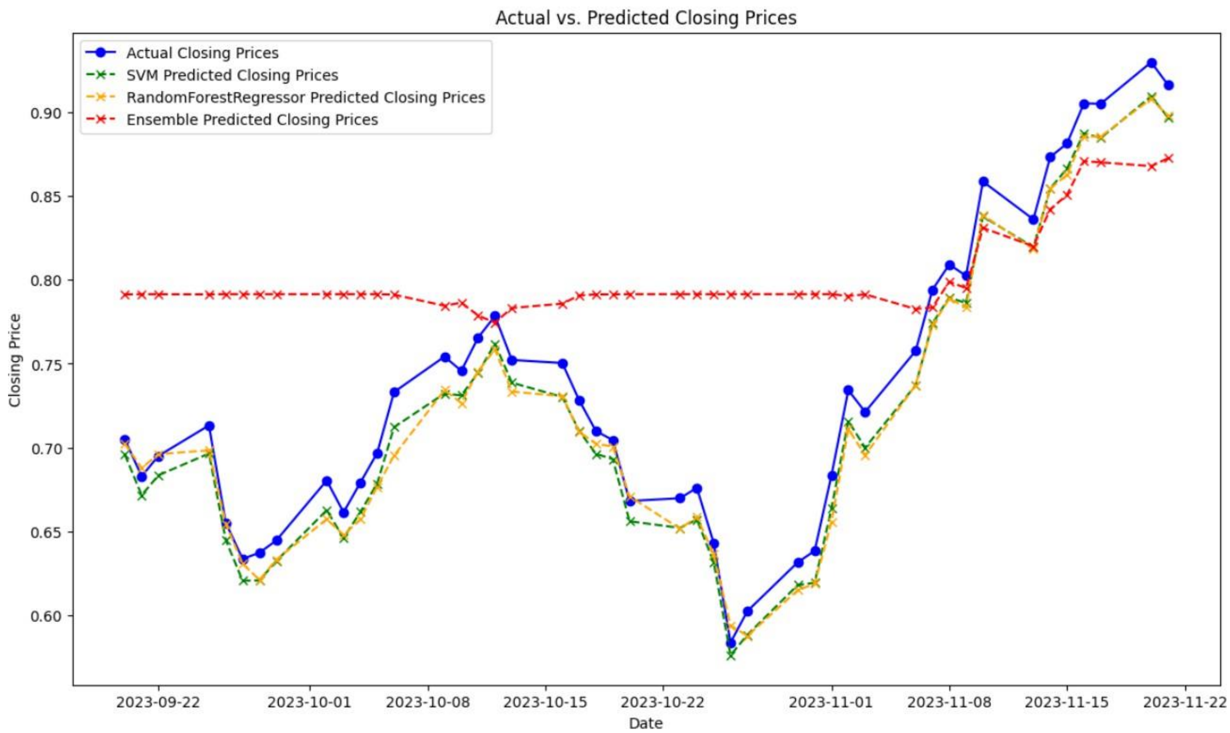


Figure 3.15 Comparison of all the 3 models

CHAPTER 4

CONCLUSION

In conclusion, our application offers comprehensive insights into stock market dynamics by furnishing key metrics such as opening prices, PE ratio, and Market cap for selected companies. The graphical representation aids users in intuitively understanding stock trends, empowering them to make informed investment decisions. While our LSTM algorithm forms the backbone of our predictive capabilities, it's imperative to note that its usage in real-world scenarios carries inherent risks. The dynamic nature of financial markets and the potential for unforeseen events make any predictive model, including LSTM, subject to uncertainties. Hence, while our app serves as a valuable tool for stock analysis, users are advised to exercise prudence and consider multiple factors before making investment choices. It's crucial to recognize that no algorithm can eliminate market risks entirely, and a well-informed, cautious approach remains key in navigating the complexities of the stock market.

Users can leverage the information presented to make data-driven decisions, but it's vital to supplement algorithmic predictions with a thorough analysis of market conditions, economic indicators, and external factors. The LSTM algorithm, while powerful, must be used with caution, considering its sensitivity to varying market conditions. Continuous monitoring and adaptation to market changes are essential for maintaining the effectiveness of our predictive features. Ultimately, our app serves as a valuable tool in the investor's toolkit, but the collaborative integration of algorithmic insights with comprehensive market knowledge is key for navigating the complexities of the stock market.

CHAPTER 5

APPENDIX

5. REFERENCES

[1] Olah, Christopher. "Understanding LSTM Networks." *Github.io*, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

[2] K. J, H. E, M. S. Jacob, and D. R, "Stock Price Prediction Based on LSTM Deep Learning Model," *IEEE Xplore*, Jul. 01, 2021. https://ieeexplore.ieee.org/document/9526491

[3] K. Pawar, R. S. Jalem, and V. Tiwari, "Stock Market Price Prediction Using LSTM RNN," *Advances in Intelligent Systems and Computing*, pp. 493–503, Nov. 2018, doi: https://doi.org/10.1007/978-981-13-2285-3_58.

[4] "Stock market's price movement prediction with LSTM neural networks | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/abstract/document/7966019?casa_token=Psi7AkxLbHMAAAAA:i99Yd40FgTmH2PAXD7aLJyZH_OIHNHT2cKs6UlQSeTGwgHIslG14vsRYN92eNwwkDHmcoWJlXujhkg (accessed Nov. 27, 2023).

[5] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-Based Model to Forecast Stock Prices," *Complexity*, vol. 2020, pp. 1–10, Nov. 2020, doi: https://doi.org/10.1155/2020/6622927.

[6] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020, doi: https://doi.org/10.1016/j.procs.2020.03.049.

[7] "Prediction of Stock Price Based on LSTM Neural Network | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/abstract/document/8950831?casa_token=FGsyGLyBnDwAAAAA:fSAJ-MipW-yL9kuGXd_qqaXGCQsQAiEOsIIIG8tWskivS4WswkaBpCOYhWq17glPIYY4WSs32T-aqw (accessed Nov. 27, 2023).

[8] "Stock Price Prediction using LSTM and ARIMA | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/abstract/document/9532655?casa_token=TdT-0dx1KJYAAAAA:ks-nMOKmCJvceVe2mQU9UX6BerqrBu2K6Gp5rKCX-GDO0qp8qgw4EjYiocAoiRb5XFDv9XDngvPd0Q (accessed Nov. 27, 2023).

[9] S. Verma, Satya Prakash Sahu, and Tirath Prasad Sahu, "Ensemble Approach for Stock Market Forecasting Using ARIMA and LSTM Model," pp. 65–80, Jan. 2022, doi: https://doi.org/10.1007/978-981-16-7330-6_6.

[10] “Comparison of ARIMA and LSTM for Stock Price Prediction,” *Financial Engineering and Risk Management*, vol. 6, no. 1, 2023, doi: https://doi.org/10.23977/ferm.2023.060101.

[11] “Infographic: The Indian Stock Market Is Booming,” Statista Daily Data, May 11, 2022. https://www.statista.com/chart/27421/number-of-stock-market-issues-in-india-money-raised/#:~:text=The%20sale%20also%20brought%20in (accessed Nov. 27, 2023).