

Stock Market Price Prediction and Analysis

Ajinkya Rajkar^[1], Aayush Kumaria^[2], Aniket Raut^[3], Nilima Kulkarni^[4]
Department of Computer Science and Engineering,
MIT School of Engineering, MIT Arts Design and Technology University,
Pune, 412201, India

Abstract—India's stock market is extremely variable and indeterministic, which has a limitless number of aspects that regulate the directions and trends of the stock market; therefore, predicting the uptrend and downtrend is a complicated process. This paper aims to demonstrate the use of recurrent neural networks in finance to predict the closing price of a selected stock and analyze sentiments around it in real-time. By combining both these techniques, the proposed model can give buy or sell recommendations.

The proposed system has been implemented as a web app using Django and React. The React Web App displays all live prices and news received from the self-built Django Server via web scraping. Additionally, the Django server serves as a bridge between the React frontend and the machine learning algorithm built with Keras and further enhanced with Tensorflow.

Keywords—Django Server, React Web-app, Recurrent Neural Networks, Stock Market Prediction, Time Series

I. INTRODUCTION

The Stock Market of India ranks 12th in the world in terms of market net worth. At the moment, the NSE India offers trading in 1659 companies. India's economy is based mainly on agricultural exports and related services such as software development and technical support. Regrettably, stock market trading accounts for only 4% of India's gross domestic product. It, therefore, is far less than the average for other developed countries like the USA is around 55%. This underutilized asset has the potential to be more effectively monetized to aid India's development.

This section discusses the shortcomings of conventional stock price prediction methods and the advantages of applying machine learning.

A. The conventional method for analyzing stocks

The stock market is highly variable and indeterministic due to various parameters impacting price movements in numerous sizes and layers. According to efficient market theory, the market corrects itself, meaning that the current share price represents the appropriate total combined price, which is neither excessively low nor excessively high. In simple terms, "the market is unbeatable," suggesting one cannot defeat the market, yet existing evidence shows otherwise. By monitoring stock movement patterns, it is possible to forecast market trends.

The conventional approach emphasizes technical and fundamental analysis to forecast the share market on a massive level, which very seldom transforms to low-level selected stocks forecasting. Some particular stocks, on the other hand, lead to overall price movements in the market.

a. Fundamental analysis

This strategy is primarily concerned with a company's prior success and reputation. Stocks that are prone to a positive price increase are filtered using performance criteria such as P/E ratios. It relies on the assumption that successful businesses will remain beneficial in the long run due to the market's rewarding nature.

b. Technical analysis

This strategy relies on forecasting future pricing using time-series analysis on historical trends. RSI, Bollinger Bands, VWAP, Moving averages are a few of the many indicators used to analyze and study the market trend and understand how the market is moving.

B. Stock market analysis with a modern perspective

Machine learning methods such as SVM, RNN, and EML, developed by computer scientists, can evaluate and conduct information discovery at large scales in a short period. RNN is used to forecast stock market prices in this research paper.

a. Qualitative Research

Newsfeeds about the stock market significantly impact the market trend, resulting in a downward trend in lousy sentiment and an upward trend when there are positive sentiments. As a result, media/social networks and the share price are correlated and uncertain. Therefore, news acts as an essential medium for analyzing how a stock would perform on a subsequent day. Stocks mimic each other in times of crisis, resulting in market collapse. News can give the public a view of how the company is performing and how it might perform in the future.

b. Quantitative research

Most markets now have historical information readily available. One can use these historical values to study and analyze how a particular stock has been performing. This data can also be studied using various Machine Learning Algorithms to analyze the upward and downward trends of the data to generate predictions for the future. Such algorithms prefer learning the movement of a single stock rather than many different stocks cumulatively as each stock moves and trades differently. Additionally, these models can also perform under a variety of situations and economic conditions.

II. RELATED WORK

In [2], V.K.S, Reddy mainly focused on Support Vector Machines (SVM). According to the author, the SVM algorithm works on large-scale data values gathered from the global economy. Overfitting is not a problem with SVM. The stock price is predicted using a variety of Machine Learning algorithms, but SVM proved to be the most accurate.

This article [6] estimated the share price using a deep learning algorithm of the Nikkei 225 and Nikkei 400 of Japan stock exchange index. Deep Neural Network, Backpropagation Neural Network, and Support Vector Machine Regression were the classification algorithms used to forecast the stock price.

The [5] objective of the authors of this paper is to use the linear regression technique to estimate stock price trends by training on historical data and testing on other data. Stocks are scored by dividing the regressed slope by the inverse mean squared errors.

The three aspects of this [4] project are preprocessing and data-extraction of the china Stock markets dataset, feature selection, and a price pattern prediction algorithm based on long short-term memory. They succeeded in developing a new algorithm component called feature extension. They employed feature expansion (FE) techniques in conjunction with recursive feature elimination (RFE) and principal component analysis (PCA) to achieve highly accurate results that outperformed the top methods in the majority of similar research.

The purpose of this [3] paper was to conduct a comprehensive comparative analysis of ensemble methods to predict the stock market based on four different stock market indexes (datasets). Three well-known machine learning algorithms were used in this study: decision trees, multilayer perceptron neural network, and support vector machines. They discovered that the substitution method of stacks for developing a classification model surpassed most other combinations compared to the Accuracy and loss statistics.

III. PROPOSED SYSTEM

This section outlines the methods, system architecture as displayed in Fig. 1, features, and algorithms we have used to create a product that can efficiently predict the closing price of a User-Selected Stock Query.

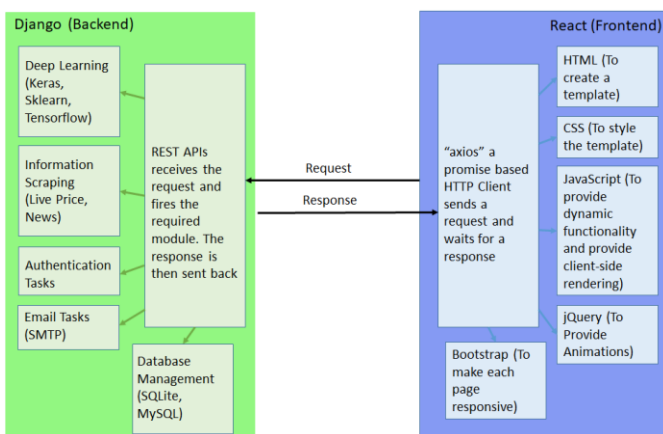


Fig. 1. The architecture of the Proposed System

A. Tech Stack

We have cumulatively used two different programming languages to create this product, two other databases, two frameworks, and one markdown Language, all categorized under two sections, the Frontend and the Backend.

a. The Frontend

The Frontend is a representation of what the user sees. It is the visual component, and frequently the section's sole purpose is to be visually appealing. The Frontend uses React to serve a blank HTML template onto which custom HTML is inserted and styled with CSS. Javascript makes the page dynamic and enables the user to navigate between pages and communicate with the server with ease.

Additionally, the Frontend makes the web app responsive via Bootstrap, which means it adapts to the screen size and resizes itself to look good on any device. The Frontend is enhanced with jQuery to provide a more animated and silky view of the page rather than a direct jump.

The primary reason for choosing React over any other Javascript library for frontend development is its fast rendering speed and SEO friendliness. Content delivery Speed acts as a significant factor when creating a web app, and deploying a React app is as easy as developing one.

b. The backend

The Backend speaks about what happens in the background. It is usually associated with handling and dealing with user requests. These requests are received and handled using REST APIs deployed via Django, which is programmed using Python. Python is responsible for communicating between various modules depending on the User's needs.

The Backend uses two databases, namely SQLite, to store user details when they create an account, and MySQL is used to store tabular data like a Stock and its current market price.

Python is also used to create a Recurrent Neural Network Model using Keras, built upon Tensorflow.

The primary reason for choosing Django over any other web framework like Flask is that Django allows us to create APIs using the REST architecture while maintaining an SQLite database. It will enable us to concentrate more on developing functions that we require rather than developing basic and repeatedly made tasks.

B. Account Creation and Use of SQLite

To use the proposed platform and all of its features, each User must create an account for themselves on the "InvestyX" platform. Details entered by the User are first validated in Javascript and then sent to Python for further Validation. In the first stage of Validation, which happens in Javascript, the data is checked using Regular Expressions. Data Like Email-ID, Phone Number, Pan Number, IFSC Code, to name a few, are all passed through the regular expression. If it passes the regular-expression test, the data is forwarded to Python; else, an error is thrown. Django enables us to quickly and efficiently check and add the User-Inputted data to the SQLite Database in Python. If the data entered previously exists in a database, an error is returned; else, the data is hashed and added.

Sensitive information such as Aadhar Number, Email ID, Password, Account Number, to name a few, are all hashed using the "SHA-256 hashing algorithm. In the SHA-256 hashing algorithm, the data acts as an input and is passed through a cryptographic hash function which outputs a 64-bit alphanumeric text.

```
Investyx1 ->
f64d0ed7490a6efaf9e1cc37c5d98d90466457bb8f4897f25d16
d575f0e10461

Investyx2 ->
b32c7b2cf5ce953ba6774019b44c820cca22d811319a310003f1
d296b5e8d6e5
```

Along with Hashing the Sensitive Details, a mandatory 2 Factor Login is triggered. An email is sent to the User, which provides a unique six-digit alphanumeric Usercode and a link to create a Pin for that User Code. Additionally, the Pin and User Code are hashed upon creation to ensure that user information is protected in the event of a security breach.

C. Scraping News and Live Indices Prices

The dashboard of our application displays live prices, percentage changes, and value changes for the four most popular indices, namely the NIFTY50, the NIFTY100, the NIFTY BANK, and the SENSEX, which are updated every second when the market is open. This data is retrieved from Yahoo Finance via our server and passed to the frontend.

The news section of the dashboard displays the five most recent headlines about the stock market, which are updated every fifteen seconds. Economic Times was used to compile this information because their News is updated the most frequently compared to other news-related websites such as MoneyControl and Yahoo Finance.

D. Autocomplete Searching Stock Names

We manually entered information about over 5500 stocks listed on the NSE and BSE into a Google Sheet and stored it on Google Drive. These details include the Stock Code, the Company's Name, the Industry, and the exchange on which the stock is listed. We could access and use this database in our Backend via the Google Cloud connection.

Given the possibility that the user may search for either the Company or the Stock Code, we needed to develop a search algorithm that takes both of these parameters into account and returns the four best predictions.

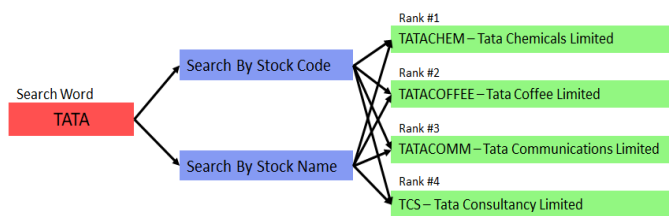


Fig. 2. Working of the Search Algorithm

E. Volume Movement Market Sentiments

We use the RSI (Relative Strength Index) to determine whether the stock is in an Overbought or Underbought Zone over 14 days to determine whether the index Nifty 50 is in an Overbought or Underbought Zone. We have set the upper and lower bounds of our RSI value to 60 and 40, respectively, whereas the traditional approach is 70/30. When we set it to 60/40, we get a more precise output. If the RSI value is less than 40, it is in the Underbought Zone, implying that the market is bullish; if the RSI value is greater than 60, it is in the Overbought Zone, implying a dip is imminent.

When this RSI value is entered into a condition statement, we get four possible outcomes:

- Extreme Fear (which is an excellent time to open new positions)
- Fear (which is a reasonable time to open new positions)
- Greed (which is not a good time to open new positions)
- Extreme Greed (Worst Possible Time to Open New Positions)

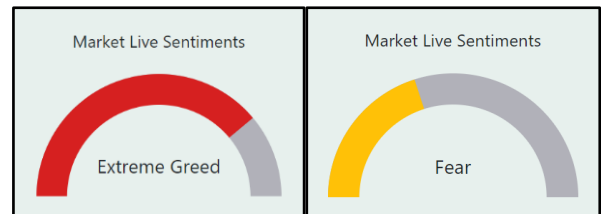


Fig. 3. Market Live Sentiments

F. Watchlist with MySQL

The watchlist feature in our application allows the user to add their favorite stocks to the watchlist and observe the changes. After adding the stock to the watchlist, it gets added to the MySQL database, where each user has a unique table. The maximum number of columns in the table varies from user to user, depending on their chosen plan. When adding the stock to the watchlist, the StockCode and Company Name are captured along with its Live Price. When displaying the watchlist, the stock name, stock company, value change, and percent change are displayed. Along with this, the live price recorded is used along with the current live price to determine the Unrealised Profit/Loss, which would mean that if we had bought the stock, how much of a change would we have.

Stock Name	Price	Change	Unrealised P/L
ADANIEN NSE	1302.6	-1.75% -23.2	12.5
HDFCBANK NSE	1503.45	1.4% 20.8	-24.5
ICICIBANK NSE	643.05	-1.41% -9.2	7.25

Fig. 4. The output of the Executed Watchlist

G. Live Scraping Search Price

When a user searches a stock, the Stock Name and its respective Stock Code are sent to the server for more details about the stock. The server uses Yahoo Finance to get the fundamentals of the company. This is especially useful for stock traders who prefer to buy for a more extended period. The information is extracted every 5 seconds in which the Current Market Price, Value Change, Percent Change, Statistics, Information, and much more are extracted, which is then sent to the Frontend.

Using "nsepy," we get the historical price for one year starting from the current date that gets passed to the Frontend with its respective date. Since our Frontend is created using React, we use "recharts," a React Chart Library to create fluid charts, which get updated every day once the market closes.

Additionally, the user can add the searched stock to their watchlist or forecast how the stock will close during the next trading session.

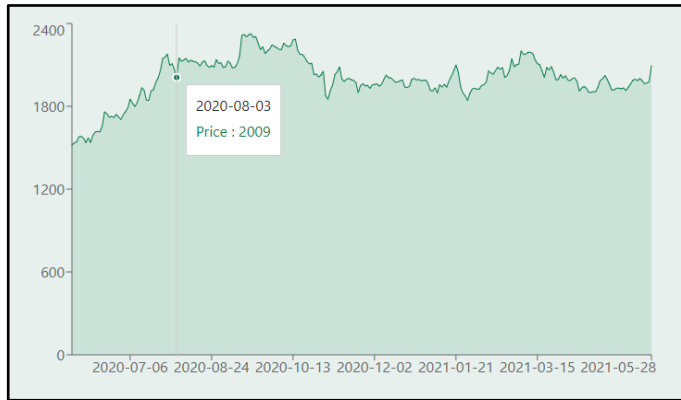


Fig. 5. Example of a Stock Chart for One Year on one-day time frame

H. Prediction Algorithm

To forecast how the stock would close in the subsequent trading session, we chose to use Long Short Term Memory, also known as LSTM, as used in [1], a Deep Learning-based Recurrent Neural Network Model. The rationale for selecting this architecture is that "History Repeats Itself," and thus LSTM fits perfectly into the picture. The memory cell in the LSTM enables us to build a model in which the closing price of a stock over "x" consecutive days accurately predicts the closing price on the "x+1" day. LSTM is constructed using Keras, which is in turn built upon Tensorflow.

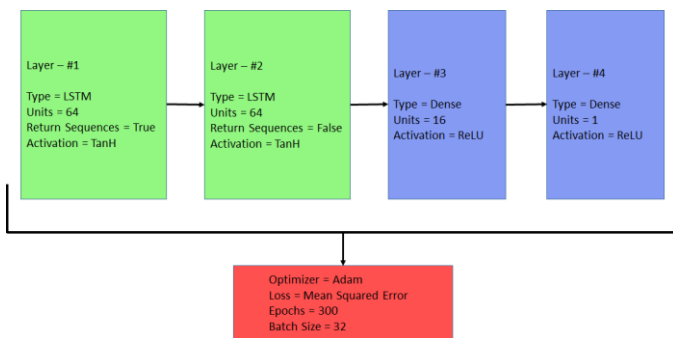


Fig. 6. The architecture of the Proposed Recurrent Neural Network

In the preprocessing stage, two years of historical data are scaled using "MinMax Scaling" and then overlapped with the premise that seven days of data predict the eighth day. After this, the model is then trained using a four-layer LSTM model with 64 units on the first layer and 64 units on the second layer. The model is then flattened into a Dense Layer of 16 nodes, followed by one output node. The Adam optimizer is used to optimize this, and the loss is calculated using the "mean squared error." It has been trained for 300 epochs with a 32-batch size. After each training session, the approximate loss is "5.243e-04". The model is then used to predict how the stock will close on the following trading day, yielding the predicted closing price.

Additionally, Live News is scraped and input into a point-based system, where the presence of words such as "Dip, Bear, Sell" indicates that the stock will decline, resulting in a lower sentiment value. On the other hand, if terms such as "Buy, Bull, Rise" are present, there is a greater likelihood that the stock will rise, resulting in a higher sentiment value. The sentiment value and predicted value change are then inserted into a conditional statement determining whether now is the optimal time to buy or sell the stock.

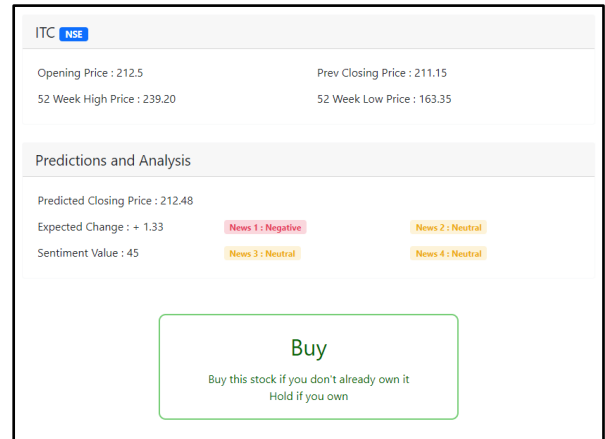


Fig. 7. Screenshot of the Final Output

IV. RESULTS AND DISCUSSIONS

This section will discuss the outcomes of the proposed system.

Our model gives a real number as an output which is the predicted closing price for T+1 day. This model can only be compared with the actual closing price to see how accurately it measures the performance of selected stock. Table 1. showcases the difference between the predicted day change and actual change for 28th May 2021.

TABLE I. COMPARISON BETWEEN PREDICTED AND ACTUAL CLOSING

Stock Name	Last Closing Price	Predicted Closing Price	Actual Closing Price	Difference
POWERGRID	228.3	222.27	226	-1.63%
SBIN	425.2	417.86	422	-0.97%
HINDUNILVR	2,326.40	2317.77	2,328.30	-0.45%
INFY	1,402.25	1405.86	1,407.70	-0.13%
NESTLEIND	17,746.70	17474.28	17,545.00	-0.40%
BAJAJ-AUTO	4,246.10	4186.67	4,203.00	-0.38%
HDFCBANK	1,482.65	1462.66	1,507.75	-3.04%
INFY	1,402.25	1405.86	1,407.70	-0.13%
BRITANNIA	3,414.65	3433.87	3,428.70	0.15%
TITAN	1,594.25	1594.81	1,578.20	1.04%
ITC	211.15	213.27	212.9	0.18%
ADANI PORTS	751.4	779.89	777	0.38%
QUESS	680.85	671.57	689	-2.56%
HINDZINC	328.75	329.13	326.65	0.75%
RELIANCE	1,976.10	2111.16	2,094.45	0.85%
TATASTEEL	1096.65	1113.61	1,104.00	0.88%
GRANULES	315.9	322.81	316.7	1.93%
SUNPHARMA	699.5	678.95	672.65	0.90%
ASIANPAINT	2,949.35	2964.87	2,935.70	0.99%
SHREECEM	28,066.05	27993.64	27,599.00	1.41%

REFERENCES

- [1] POTHUGANTI, K. A. R. U. N. A. K. A. R. (2021). Long Short-Term Memory (LSTM) Algorithm Based Prediction of Stock Market Exchange. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.3770184>
- [2] Reddy, V. K. (2018). Stock Market Prediction Using Machine Learning. International Research Journal of Engineering and Technology (IRJET), Vol. 5.
- [3] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data, 7(1). <https://doi.org/10.1186/s40537-020-00299-5>
- [4] Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data, 7(1). <https://doi.org/10.1186/s40537-020-00333-6>
- [5] Sandhu, O. (2021). Stock market trend prediction using regression errors. <https://doi.org/10.32920/ryerson.14645169>
- [6] Harahap, L. A., Lipikorn, R., & Kitamoto, A. (2020b). Nikkei Stock Market Price Index Prediction Using Machine Learning. Journal of Physics: Conference Series, 1566, 012043. <https://doi.org/10.1088/1742-6596/1566/1/012043>