Customer Adaptive Automated Trading System with Capital Risk Analysis using Machine Learning

Harsh Agarwal, Aditi Kandoi, Bhavya Ahir, Sudhir Dhage
Department of Computer Engineering,
Sardar Patel Institute of Technology,
Mumbai, 400058
harshagarwal2703@gmail.com, kandoiaditi0@gmail.com, bhavya.ahir@spit.ac.in, sudhir_dhage@spit.ac.in

***Abstract*—Stock market plays a huge role in the economy of our country. Several attempts have been made to analyse and predict the stock market. While the existing systems try to exploit the patterns of stock prices using historical data, they do not take into the account the poor performance of the system. Moreover, there is no system which provides user specific trading strategies. The proposed solution explores filtration and different trading strategies using RoCE and Fuzzy Logic to solve the problem and predict the portfolio values. It also takes into consideration the sentiment aspect of trading using NLP and combines the two to efficiently to perform trading for even those users who have smattering knowledge about stock market thereby making it suitable for everyone.**

***Index Terms*—Automated System, Algorithmic Trading, Price Prediction, Data Analysis.**

I. Introduction

In India, prediction and analysis of the stock market are the most researched topics. The stock market acts as an interface for people to perform buying and selling of stocks. With the right investments at the right time, stocks can be the most effective place to invest for a person looking to earn money. Moreover, the country's economy is highly dependent on the stock market. However, due to the volatile behaviour of the market, no human can impeccably predict the direction of the stock market. As a result, it is a disadvantage for novice investors who have only a smattering knowledge in trading activities. Every human participating in trading activities must be at least aware of the variations taking place in the stock market. Therefore, there is a need for systems and techniques that address these demands, understand the market, and extend the scope of the market to novices.

The limitations by traditional trading have circumscribed the stock market penetration to less than five percent as stated in the article by financial express [1]. The scanting removal of these limitations has been possible by making the best use of technological development. Human activity has been replaced by automated trading systems which use predefined rules that allow computers to perform and monitor the trades. The biggest advantage of rule-based trading automation is that it can help the trading activities to function without the consideration of emotion. However, the existing automated systems often give rise to additional problems. These problems mainly include the inaccuracy in performance, high computation power etc. Hence, there is a need to improve such systems by the elimination of these problems and to make the system accessible to everyone.

Therefore, a novel solution is needed which can:

- Provide an efficient system that maximizes profits
- Provide an easy interface for amateur users who require no prior knowledge
- Perform trading in a robust manner with near to zero errors
- Provide custom trading strategies to users based on their risk score

The paper aims to develop a client-adaptive automated trading system which provides the privilege of simplicity in a way that it provides ease even to those, who do not understand its intricacy. The uniqueness of the proposed solution is that it introduces an initial filter based on the parameter: Return on Capital Employed (RoCE), to narrow down the scope of the stocks in order to reduce the computational time that leads to poor performance. Moreover, implementing user-specific trading strategies based on their risk score to increase profitability for each user and making the system entirely automated allows the system to make decisions regarding appropriate trades to generate maximum profits.

The content of the paper is divided into five sections. In Section II, the literature review of the existing automated trading systems and its limitations are summarized. Section III introduces the proposed solution which discusses the architecture of the proposed automated system and its flow. Section IV discusses the obtained results of the work. The paper concludes with Section V which details the conclusion.

II. Literature Survey

The literature review consists of various papers related to RoCE components, Sentiment Analysis, Price Prediction, Risk Analysis and Trading Bot. The features that are common across the previous works include study stock trends and prediction of stock prices.

Eli Amir and Itay Kama [2] examined the investor reaction to return on capital equity and its components during its quarterly earnings. The importance of net profit margin (NPM), asset turnover (ATO) and leverage (LEV) was considered relative to each other using Fama–MacBeth regression and portfolio analysis techniques. Using this, the beta and the risk values were found to determine the asset prices. It was observed that a high level of NPM led to a more positive market reaction as compared to the other components and vice versa. Any levels of the other components did not affect the market reaction.

Joshi et al. [3] proposed a solution to discover future trends of a stock using news articles about a company with the help of fundamental analysis techniques. The sentiment analysis along with the determination of the relationship between stock price and the news articles was done using supervised machine learning and text mining techniques. Polarity was assigned to an article based on the stop words mentioned in a dictionary. They compared three classification models: Random Forest which gave the highest accuracy ranging from 88% to 92%, SVM which worked well giving an accuracy of 86% and Naive Bayes having 83% accuracy. However, the dictionary created consisted of a limited financial terms which resulted in poor performance by the models. Unlike Joshi, Guangyu Ding and Liangxi Qin [4] proposed a solution to predict multiple value outputs using deep recurrent neural network having multiple inputs and outputs using a long short term memory (LSTM) network. The backtesting was done using multiple data sets and then compared with the two models individually. Since, the network gave multiple outputs, the model could predict open price, lowest price and highest price of any stock at the same time thereby reaching an accuracy of over 95%. However, this model did not take into account the relationship between the user and their loss handling capability which made the system less precise.

The paper [5] aimed to analyze the risk and return values of the banking by considering Nifty Index and compared fifty stocks. They found the relationship between returns and volatility with Standard Deviation. As a result, it was observed that all the trending and top stocks had a positive beta value based on the market index and the less volatile stocks had a low beta value. However, the system did not successfully prove to be accurate because it did not analyse the market continuously to select the top stocks.

In [6], trading decision methods were proposed based on a multiple classifier system and candlestick patterns. The trading strategies were evaluated using Bollinger Bands and Parabolic SAR indicators. The performance of the proposed decision strategies was tested by developing a web application which used the end of day stock prices, and the stocks were used to represent the up and down trends. The best strategy for up trend stocks gave around 17% profit and around 1% for sideway market trend. Similarly, in down trend stocks, the loss was minimized to 2.62%. The paper [7] compared the use of three famous trading strategies: SMA, MACD and PIVOT. The strategies were backtested using the d Backtest PS method. The systems ran for 1.5 years and closing price of the data was considered. They concluded that MACD>PIVOT>SMA in terms of the profits generated by each.

Due to the unpredictable events that completely change the trend of the market, most algorithmic traders opt for High Frequency Trading (HFT). In this, the brokers or the traders execute a large number of trades in a short period of time to extract profits from low changes in prices. The paper [8] proposed the use of Grid Trading Algorithms on FOREX markets to execute such trades. The system placed orders in regular time intervals to generate steady profits from these deals. Regression network and trend classifier were used to predict which buying and selling of a particular stock. The paper achieved a good Return on Investment(ROI) of 13.76%.

L. Chen and Q. Gao [9] used Deep Q-network (DQN) and Deep Recurrent Q-network (DRQN) for decision making in stock market. The two methods were compared and it was observed that DQN easily learned profitable patterns from the historical stock market data. However, introduction of recurrence to DQN led to an improvement in the performed in the trading actions. Akhil Raj et al. [10] proposed the use of Deep Reinforcement Learning model in an automated trading system. RCNN was used to predict sentiment from the stock market news data which was fed into the reinforcement learning neural network model. This model predicted the trend and made Buy/Sell decisions to maximize the profit. The model having an accuracy of 96%, was extremely efficient in predicting the sentiment and the trend of the market.

In [11], Prasetijo et al. proposed trading strategies using Bollinger Bands and Parabolic SAR indicators. They also built a web application to perform backtesting of the strategies. By taking into consideration the end-of-day stock price for different LQ-45 stocks, the system categorized into up, down and sideway trends. Bollinger Bands generated profit 17.06% for up-trending stocks while it gave profit of 1.19% for sideway market trend. Parabolic resulted in losses for all the trends hence, it worked the worst in comparison to Bollinger Bands. C. N. W. Tan [12] simulated a trading based on the two years NYSE stocks data using an artificial neural network based financial trading system. The system fit quite well giving an accuracy of 83% with a tolerance of 5%. The reliability of the system is poor as it did not consider several parameters affecting the neural network.

Most of the previous research on automated trading system focus on predicting stock prices. These systems perform trading by considering the whole stock market without categorizing and filtering out non-profitable stocks. Moreover, a generalized trading strategy is used for all kinds of user portfolios which result in inefficient performance due to the ignorance of the risk handling capabilities of users. Hence, the proposed system resolves this by introducing a filter stage to remove the non required stocks in carrying out trading. Additionally, a fuzzy logic controller has also been implemented to suggest users different trading strategies based on their portfolio.

III. Methodology

A. Architecture Model Description

The heart of the research involves providing a customer adaptive automated trading system to extend the scope of the stock market in India by making it friendly for everyone. The aim is to eliminate the limitations in the present automated trading systems like high computational power and time. Therefore, a filtering stage is included in the system to narrow down the scope of the stocks. User-specific trading strategies are also suggested based on their risk handling capabilities. Data from market data API and news articles on stocks enter the filter stage. News and articles' data is used by evaluating the sentiment for all the articles related to stocks. The filtration is performed by calculating the Return on Capital Employed for all the available stocks. The filtered stocks are stored in the System Bot database which consists of the users' transactions and the portfolio details as illustrated in Figure 1.

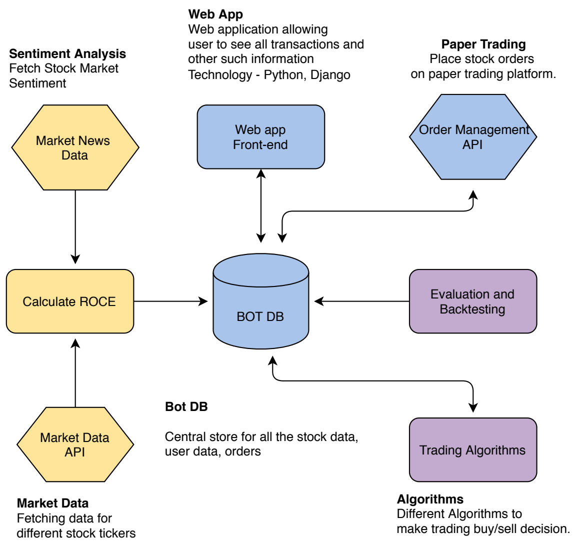


Fig. 1: System Architecture for Trading Bot

Information regarding the users' transactions and portfolio is extracted from the database and sent to the Django server. It is displayed on the web portal by the users. The system comprises of six different trading algorithms of which one is performed using Order Management API for a user based on his requirements. The evaluation and backtesting is done to assess the viability of the strategies implemented.

B. Automated Trading System

The proposed system comprises of six modules as shown in Figure 2. Each module is explained below.

*1) **Data Scraping Module**:* All the financial data affect the volatility of the stock market. The National Stock Exchange currently consists of 1837 listed companies, and manually acquiring data for all the companies is difficult. Hence, the Data Scarping Module in the system is used to collect the data. The script web-scrapes data from reliable sources to generate a database. Two types of data are collected using this script:

(i) Financial Records of the Companies

The financial records of a company give details of its stocks which help in predicting and trading stock. It consists of the quarterly bank sheet, cash flow statement, and the income statement of all the stocks.

The extraction of the financial records is performed using Beautiful Soup with Selenium and YahooFinancials. YahooFinancials is a python library that scrapes the yahoo finance website to generate financial records. There are 1466 companies available on yahoo which were gathered in batches of ten to prevent from blocking the IP.

The remaining companies were scraped from sources like moneycontrol.com, Finviz India or Screener using BeautifulSoup with Selenium. BeautifulSoup parses through the HTML page to extract IDs which consist of the required data while Selenium repeats the task over different threads to get the stock ticker list. A total of **1573** companies' financial data was collected and stored.

(ii) Latest News Data

The news articles related to the stock market play a major role in determining the trending stocks. The data obtained from news articles is required to precisely performing trading of stocks.

The live news articles are scraped from 'Finviz India' website using BeautifulSoup. The data consists of the headlines of the news articles, its date and time. This is then stored in the form of data frames using the python library 'Pandas'.

*2) **Data Filtering Module**:* The large size of the data results in increasing computational time and power of the system. Because of this, the system becomes less efficient. Hence, the proposed system uses a unique filter - Return on Capital Employed (RoCE). This parameter is a ratio which indicates how profitable a company is and how effective its investments are. The ratio is given by:

$$\text{RoCE} = \frac{\text{Earning Before Interest and Tax (EBIT)}}{\text{Total Assets - Current Liabilities}} \tag{1}$$

Based on the latest research, companies having RoCE values greater than 20 are considered profitable. The high value of RoCE means high profitability which in turn can be considered a good investment. A company's RoCE can be calculated from its financial records using Equation 1. From the financial records of companies extracted in the Data Scraping Module, a list of thirty companies is generated after passing through the Data Filtering Module.

Also, the module removes unnecessary data from real-time market data. The redundant values and abnormalities are cleaned from the data followed by preprocessing. It is done by converting the data to DateTime format and normalizing the stock prices using MinMaxScaler from Scikit-Learn preprocessing library. The normalized values of the portfolio are required in the LSTM neural net module to predict the user's portfolio. The feature_range is set between 0 and 1. These scaled values improve the performance of the prediction module hence providing better accuracy in forecasting.

*3) **Natural Language Processing Module**:* The stock market's volatile nature is due to the fluctuations in demand and supply. The relation between demand and supply is extremely sensitive to the news articles at that instant. The analysis of the sentiments of news articles gives information on trending stocks that help in a profitable trade. Hence, a Natural Language Processing module is implemented that assigns a sentimental value to the news articles extracted in the Data Scraping module and gives a list of the trending stocks.

The news articles' data obtained is cleaned initially to remove the stop words which do not play any role in assignment of sentiment to the article. Natural Learning ToolKit Vader is used to express sentiment polarity and intensity to the news articles. Once the data is cleaned, the lexicon approach and rules come into picture to scan the data to find lexicons. It then finds the sentiment features and calculates the news article's compound score which ranges from -1 to 1. Higher the value of the score, more positive is the sentiment of the article and vice versa.

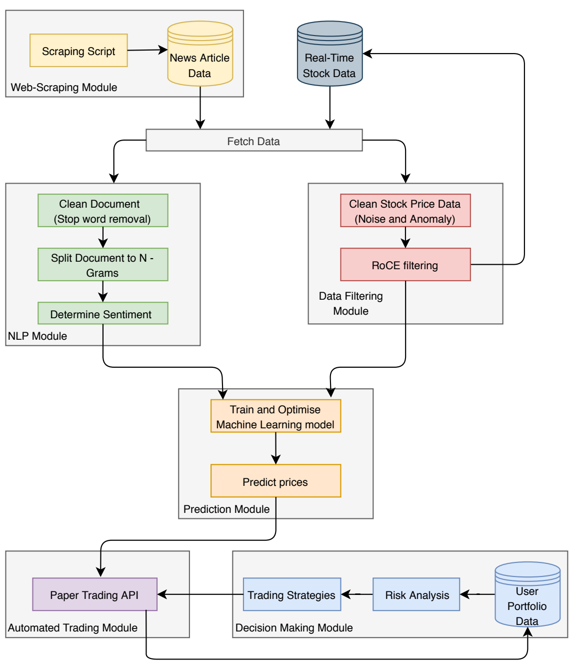


Fig. 2: Trading System with Different Modules

*4) **Prediction Module**:* The proposed system allows users to view the predictive performance of stocks based on their trading strategies. The prediction is performed using the LSTM (Long-Short Memory Model) neural net models.

The model is trained on the data of the user's portfolio based on a particular trading strategy. To attain accuracy, the model used the data obtained from backtesting the trading strategies. It consists of the trading activities for 676 days from 1st January, 2018 to 1st October, 2020 for training and the last 60 days for testing. The model used the ReLU activation function with Adam optimizer to optimize the output and ran for 150 epochs. Finally, the results of the graphs are viewed on the dashboard of the website by the users.

*5) **Decision Making Module**:* The decision making module is the most important module of the proposed system. It is responsible for determining user specific strategies and making buy and sell decisions. The module is divided into three parts: User Portfolio Data which acts as a storage unit for user data, Trading Strategies which help to execute the trading decisions and Risk Analysis which determines which strategy is suitable for a particular user.

User Portfolio Data: The user portfolio is a central database that stores data such as the information about all the stocks held by users, transactions made by the system, risk analysis and evaluation of the trading strategies. Every financial day, the database is updated with the current trending stock tickers from the Natural Language Processing module. These tickers are used with the user's trading strategies for analysis and to carry out buy/sell decisions.

Trading Strategies: Algorithmic trading strategies use a set of predefined rules and indicators to analyze the market in order to perform trades. The strategies use different parameters like price variations, stock volume, market trend, historical data, etc. for making effective trading decisions. There are six trading strategies which are stored in the system.

(i) **Simple Moving Average (SMA) Strategy**
Simple Moving Average is calculated by adding the closing prices and dividing the sum by the number of time periods in the average. This helps in analyzing a bull or bear trend.

(ii) **Exponential Moving Average (EMA) Strategy**
Exponential Moving Average analyzes the predominance of a trend in the market. It is a line on a price chart that smoothens the price action by using the mathematical formulae. It puts emphasis on the recent price which makes it more reliable as it reacts faster to the latest changes in the price data.

$$\text{Initial SMA} = 20 - \text{Period Sum}/20 \tag{2}$$

$$\text{Multiplier } (k) = \frac{2}{\text{Time period+1}} \tag{3}$$

$$\text{EMA} = \text{Close Price} * k + \text{EMA}_n * (1 - k) \tag{4}$$

where,
n = Previous day

(iii) **Double Exponential Moving Average (DEMA) Strategy**
The Double Exponential Moving Average reduces the amount of lag present in the traditional moving averages. A change of trend is observed when the prices crosses the average. Hence, it confirms the uptrends when the

price moves above average and downtrends when the price moves below average.

$$\mathrm{DEMA} = (2 * \mathrm{EMA_n}) - (\mathrm{EMA\ of\ EMA_n}) \quad (5)$$

where,
n = Look-back period

(iv) **Triple Exponential Moving Average (TEMA) Strategy** Triple Exponential Moving Average reduces the lag time further as compared to DEMA. It has a complex formula which involves calculating EMA of an EMA of an EMA. Like DEMA, it also helps to find the trend changes by comparing with the average.

$$\mathrm{TEMA} = (3 * \mathrm{EMA_1}) - (3 * \mathrm{EMA_2}) + \mathrm{EMA_3} \quad (6)$$

$$\mathrm{EMA_2} = \mathrm{EMA\ of\ EMA_1} \quad (7)$$

$$\mathrm{EMA_3} = \mathrm{EMA\ of\ EMA_2} \quad (8)$$

(v) **Bollinger Bands Strategy** Bollinger Bands are trendlines which are plotted at two stand deviation levels above and below a simple moving average (SMA) of a price. A middle band which is the simple moving average, an upper band and a lower band are the three lines of the Bollinger Bands. These can be modified based on the user preferences. The main parameters are period and the standard deviation which help to predict the prices on a relative basis.

$$\mathrm{BOLU} = \mathrm{MA(TP,n)} + m * \delta[\mathrm{TP,n}] \quad (9)$$

$$\mathrm{BOLD} = \mathrm{MA(TP,n)} - m * \delta[\mathrm{TP,n}] \quad (10)$$

where,
BOLU = Upper Bollinger Band
BOLD = Lower Bollinger Band
MA = Moving average
TP (typical price) = (High+Low+Close)÷3
n = Number of days in smoothing period
m = Number of standard deviations
δ[TP,n] = Standard Deviation over last n periods of TP

(vi) **Commodity Channel Index (CCI) Strategy** CCI calculates the difference between the historical average and the current price. It also helps to determine the overbought and oversold levels for an asset. If the CCI is above zero then the price is higher than the average and if it is below zero then the price is lower than the historical average.

$$\mathrm{CCI} = \frac{\mathrm{Last\ price} - \mathrm{SMA\ of\ last\ n\ prices}}{\mathrm{Impact\ factor} * \delta} \quad (11)$$

$$\delta = \frac{\sum_{i=k}^{k+n} \mathrm{Price}(i) - \mathrm{SMA\ of\ last\ n\ prices}}{\mathrm{n}} \quad (12)$$

where,
δ = Mean Deviation
n = Period
k >= 0

Risk Analysis: The analysis of the risk associated with the trading decision that is made by the system is calculated on the stored historical data. Every user has a specific trading strategy which is determined by using Fuzzy Logic Controller. The user enters his risk handling capabilities and expected returns. The inputs for fuzzification are the Sortino ratio, Sharpe ratio and the Return on investment. The rule base derives the estimated risk and return range which maps to each of the trading strategies. By taking into consideration the users capabilities, a specific trading strategy is assigned to the user. On the assigned trading strategy, analysis is performed for a period of two years.

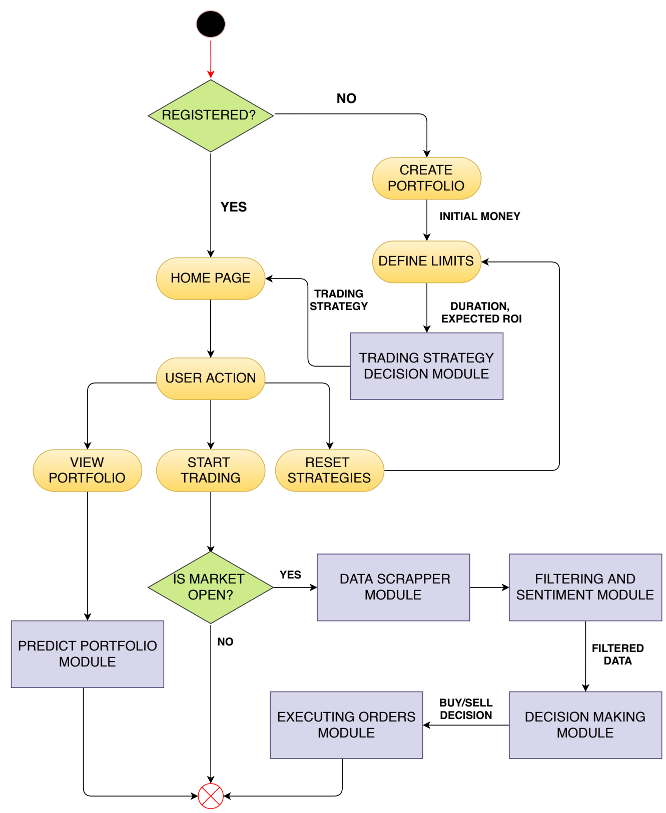


Fig. 3: User Interaction Diagram

*6) **Trade Execution Module**:* Trade Execution module is the final module where actual trading activities take place. Once the trading strategy is assigned to the user and the trading decision is made by the Decision-Making module, the stock orders are placed in the Trade Execution module. For this purpose, a paper trading platform is used which is a simulated trade that allows people to perform buying and selling actions without the involvement of real money.

C. System Operation

The fundamental purpose of the proposed system is to make stock market trading easy for users. The interaction of the user with the system is explained in Figure 3. It is an automated system which only requires the users to set the limits for performing trading actions in order to assign a trading strategy based on their requirement. It allows them to view their portfolio, start trading or reset their input to change the strategy. By viewing the portfolio, a user can either check their current worth based on their transactions or ask the system to run a prediction to check their future portfolio. Prediction module helps the user to get an estimated profit made using the current trading strategy. By starting the trading, the user gives a signal to the system to start the execution of trades and by resetting the input, users can redefine the limits and parameters.

IV. Results and Discussion

This research aims to fulfill the objectives mentioned earlier. To achieve those objectives, a system consisting of different modules is put together and following results with the respect to the mentioned objectives were achieved.

A. Sentiment Analysis on News Articles

News articles play an important role in predicting the stock prices and performing buy and sell actions in trading. By using the Vader analyzer, the compound scores of all tickers were calculated. The compound scores for three Indian stocks: HDFC Bank, Infosys and Tata Motors were compared based on their news articles for a period of four days.

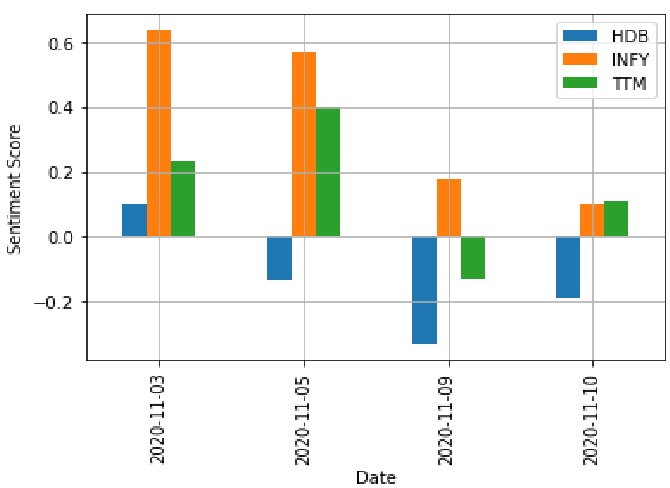


Fig. 4: Comparison of Tata Motors, HDFC Bank and Infosys Stocks

It is observed from Figure 4 that Infosys had a very high score on November 3, 2020, this means that there were several positive news articles about that company. Hence, it can be inferred that the stock price of INFY would be high based on its demand. Similarly, it can also be viewed that Tata Motors and HDFC Bank has negative score on November 9, 2020. This shows that there were negative sentimental articles on those companies which contributed to the low prices of their stocks.

B. Prediction of stock prices

Future values of the user portfolio was calculated using LSTM neural net model. Figure 5 represents the user's portfolio over time. Blue line indicates the training data for the model. Yellow and orange lines indicate the training and validation over the period of 2 months. As seen in Figure 6, it can be clearly inferred that the model is accurate in predicting results in the future and has a train loss of 0.001 and validation loss of 0.02. The model predicted that the user's portfolio will increase by 12% over the next week.

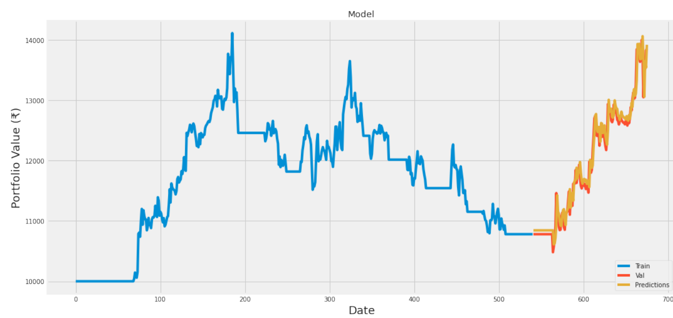


Fig. 5: User Portfolio Prediction

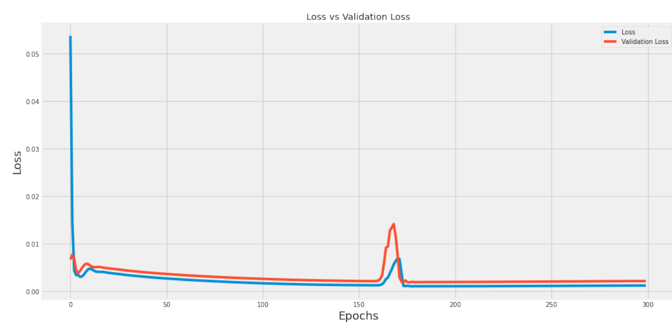


Fig. 6: Loss and Validation Loss

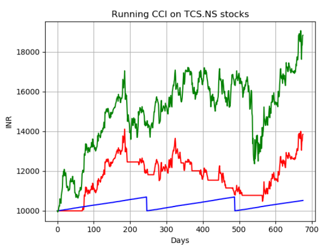
C. User specific Trading Strategies

One of the primary objectives was to implement different trading strategies and find the best fit according to the risk associated with those strategies. Bollinger bands, CCI, Moving momentum and various moving averages based Trading Strategies were implemented in the system. Backtesting was done on these trading strategies by initialising the amount at hand as Rs 10,000 and for a period of 2 years for stock TCS. Figure 7 compares backtesting of various trading strategies. This comparison shows how the strategies will react to the change in price. Green line indicates the company's stock price, red line indicates the user's portfolio and the blue line is the risk free return which is set to 7% as the standard market rate.

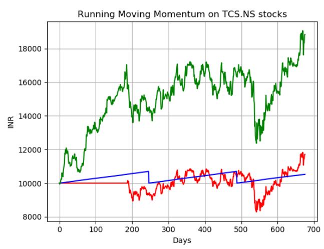
TABLE I: PROFIT GENERATED AFTER BACKTESTING

Strategy Name	Average Profit	Total Profit	Backtesting Period
CCI	5.5	3851	700
Momentum	2.4	1710	700
SMA	4.7	3350	700
EMA	7.5	5252	700
DEMA	7.9	5573	700
TEMA	8.6	6068	700

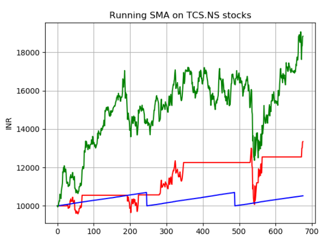
As seen in Table I, TEMA yields the best result for the period of time taken into consideration. This results might alter with change in duration and other influencing factors that affect the stock market.



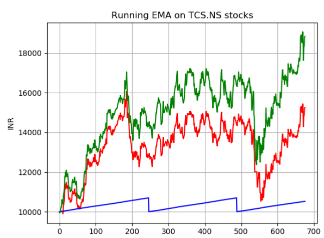
(a) CCI



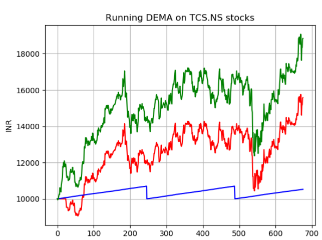
(b) Momentum



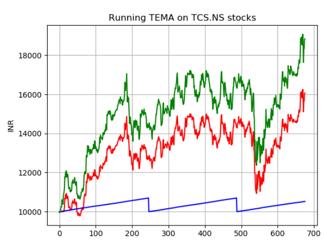
(c) SMA



(d) EMA



(e) DEMA



(f) TEMA

Fig. 7: Evaluation graphs for Trading Strategies

V. Conclusion

Stock market is an important aspect in the financial industry. A lot of research has been carried out in the past to automate the trading system. With recent advancements in the technology sector, there is an increasing impetus to optimise automated system and make it user adaptive. The proposed system uses both stock market data and news articles related to them to perform trading. From this, using RoCE filter, the stocks that are non-profitable and unnecessary are removed. The sentiment analysis of stocks is also performed using NLP to generate the trending stocks. Trending stocks are the ones whose news articles have high positive sentiment values. One of the several trading strategies is assigned to a user using a Fuzzy Logic Controller by taking into consideration the Sharpe ratio, Sortino ratio and the Return on Investment, and the user defined limits. Using the combined filtered data, the system performs trading system using the user-specific trading strategy. With the help of this, the solution can help to perform trading easily and efficiently. In addition, the system also predicts future value of users' portfolios using LSTM. The validation loss of the model was 0.02, hence, it fit well with the system. These novel features make the proposed system robust, user-friendly and one of a kind application which is available for all masses.

References

[1] P. Gagdani, “Stock markets in 2019: How mobile apps, internet growth may sharply shore up number of equity investors,” The Financial Express, 31-Dec-2018. [Online]. Available: https://www.financialexpress.com/market/stock-markets-in-2019-how-mobile-apps-internet-growth-may-sharply-shore-up-number-of-equity-investors/1429416/. [Accessed: 6-Jan-2021].

[2] Amir, Eli & Kama, Itay. (2005). The Market Reaction to ROCE and ROCE Components. SSRN Electronic Journal. 10.2139/ssrn.623161.

[3] Joshi, Kalyani & N, Bharathi & Rao, Jyothi. (2016). Stock Trend Prediction Using News Sentiment Analysis. International Journal of Computer Science and Information Technology. 8. 67-76. 10.5121/ijcsit.2016.8306.

[4] Ding, Guangyu & Qin, Liangxi. (2019). Study on the prediction of stock price based on the associated network model of LSTM. International Journal of Machine Learning and Cybernetics. 10.1007/s13042-019-01041-1.

[5] Prabhu, Ruchi. (2019). Risk & return analysis of nifty stock in Indian capital market. 10.13140/RG.2.2.26421.01767.

[6] F. Yang and J. Zhang, ”Bullish-bearish-based neural network stock trading decision supportano its application in Hong Kong stock market,” 2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Guangzhou, 2015, pp. 179-184, doi: 10.1109/ICWAPR.2015.7295947.

[7] D. Th. Vezeris and C. J. Schinas, ”Performance Comparison of Three Automated Trading Systems (MACD, PIVOT and SMA) by Means of the d-Backtest PS Implementation,” International Journal of Trade, Economics and Finance vol.9, no.4, pp. 170-173, 2018.

[8] Rundo, Francesco & Battiato, Sebastiano & Trenta, Francesca & Stallo, Agatino. (2019). Grid Trading System Robot (GTSbot): A Novel Mathematical Algorithm for trading FX Market. Applied Sciences. 9. 1796. 10.3390/app9091796.

[9] L. Chen and Q. Gao, ”Application of Deep Reinforcement Learning on Automated Stock Trading,” 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2019, pp. 29-33, doi: 10.1109/ ICSESS47205.2019. 9040728.

[10] Azhikodan A.R., Bhat A.G.K., Jadhav M.V. (2019) Stock Trading Bot Using Deep Reinforcement Learning. In: Saini H., Sayal R., Govardhan A., Buyya R. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 32. Springer, Singapore.

[11] A. B. Prasetijo, T. A. Saputro, I. P. Windasari and Y. E. Windarto, ”Buy/sell signal detection in stock trading with bollinger bands and parabolic SAR: With web application for proofing trading strategy,” 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, 2017, pp. 41-44, doi: 10.1109/ICITACEE.2017.8257672.

[12] C. N. W. Tan, ”Trading a NYSE-stock with a simple artificial neural network-based financial trading system,” Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, 1993, pp. 294-295, doi: 10.1109/ANNES.1993.323022.