AUTOMATED STOCK TRADING ALGORITHM

[1]Ritwik Budhiraja, [2]Nikki Goel, [3]Parul Bhardwaj, [4]Ashish Sharma

[1]Student, [2]Student, [3]Student, [4]Assistant Proferssor
Computer Science and Technology,
Maharaja Agrasen Institute of Technology, Delhi, India.

Abstract - Indicators have always been essential to stock trading. They can be used to predict the behaviour of the stock market. Our goal is to maximize cumulative profit after brokerage using trading indicators in a fully automated algorithm. This model uses the concept of moving averages in which cross-overs are used as indicators to generate profit. By applying these indicators to the past three months' data, we formulated an algorithm with optimum parameters. The algorithm takes price averages at every minute for the past nine and twenty-one minutes, based on which a decision is made: to buy or to sell a particular stock at a particular time and price. The trading done at any moment is strictly limited to a volume equal to 1. The total sum of cumulative profit after brokerage for the period 1st June to 1st September for the company, Reliance, was calculated to be ₹93.80.

Keywords- **Stock market; moving average; stop loss; buffer value; crossover; golden cross; dead cross; automated.**

I. INTRODUCTION

The stock market is a complex, non-stationary, chaotic, and non-linear dynamic system. "Forecasting stock market, currency exchange rate, bankruptcies, understanding and managing financial risk, trading futures, credit rating, loan management, bank customer profiling, and money laundering analyses are the core challenging tasks to be considered".[1]

"Day traders need continuous feedback on short-term price action to make lightning-fast buy and sell decisions. Intraday bars, wrapped in multiple moving averages, serve this purpose, allowing quick analysis that highlights current risks as well as the most advantageous entries and exits".[2] These averages work as macro filters as well, telling the observant trader the best times to stand aside and wait for more favourable conditions. "Choosing the right moving averages adds reliability to all technically based day trading strategies, while poor or misaligned settings undermine otherwise profitable approaches".[2]

The simple moving average (SMA) and the exponential moving average (EMA) are the two most common types of indicators. The SMA is a basic average of price over the specified timeframe. For example, if one plots a 20-period SMA onto a chart, it will add up the previous 20 closing prices and divide by the number of periods (20) to determine what the current value of the SMA should be. The series of various points are joined together to form a line. "Given this particular market is in an overall uptrend, the moving average is positively sloped, being reflective of the price. Moreover, the price will tend to be above moving averages in uptrends as various lower prices will be baked into the reading from earlier in the trend".[6]

Our model is an automated trading algorithm that works for the day trading system to generate the maximum possible cumulative profit after brokerage. Our model chooses the best fit for the stop loss and buffer value and based on the cross over strategy, wherein the automated algorithm makes decisions based on intersection points of smaller and bigger simple moving averages. When the smaller moving average line crosses over the bigger moving average line in a graph, while rising, the phenomenon is defined as a Golden cross. On the other hand, when the bigger moving average line crosses over the smaller moving average line in a graph, while declining, the phenomenon is defined as a Dead cross [5]. To get a better estimation of profits earned by trading, we also calculated brokerage from an online calculator known as Zerodha[3]. This brokerage was then subtracted from cumulative profits to give us the value of real profit earned.

Our model is completely automated and needs no human intervention for stock trading. Our model studies the data and patterns for the last 3 months of that stock to make trading decisions: whether to buy or sell, when to buy or sell. Stop loss and buffer value are also chosen based on the data from the past 3 months. Our model focuses not only on generating profit but also minimizing losses while maintaining consistent profits.

The main contributions of this paper are:
- The main objective of this paper was to achieve profits in intraday stock trading for Reliance, which we achieved by developing this automated algorithm for stock trading.
- Our algorithm focuses on minimizing losses and maintaining consistent profits.

The rest of the paper is organised as follows: In Section 2, we introduce some information and related work on our algorithm for stock trading; in Section 3, we describe our methodology for training our model for intra-day trading for stocks of reliance and compare it with other various already existing techniques. This section also includes information about the algorithm used as well as the dataset; in Section 4, we described our evaluation methods, which were used to experiment with to get results for maximizing the profits. In Section 5, we present our research and discuss the outcomes of using our algorithm for trading over other methods. In Section 6, we conclude the paper.

II. RELATED WORK

Many efforts are made daily by traders to maximize their profit, for which they make use of numerous indicators to predict the behaviour of the stock market. Our model proposes to perform transactions automatically, without any human intervention, in which we use cross-overs of moving averages 9 and 21.

Manual indicators are comparatively error-prone and sometimes may get difficult to apprehend. In this paper, we aim to introduce a new method for stock trading; that is automating the transaction process(decision making) by using the cross-over strategy[7]. This method not only makes the whole process easier but also reduces the chances of a manual error.

III. METHODOLOGY

A. DATASET

We crawled historical data till date from Zerodha[4]. It included the following columns:

id: It displays the unique ids assigned to different stocks.
scrip_name: It displays the names of different stocks.
timestamp: It displays the timestamp at any particular moment with a 1-minute separation between 2 timestamps.
close_price: It displays the final price value of stocks in that particular minute.
high_price: It displays the highest price value of stocks in that particular minute.
low_price: It displays the lowest price value of stocks in that particular minute.
volume: It displays the number of shares transacted in that particular minute.

id	scrip__name	timestamp	close_price	high_price	low_price	volume
140411	RELIANCE	2020-08-04 03:45:00+00:00	2019.10	2021.90	2009.65	508731
140412	RELIANCE	2020-08-04 03:46:00+00:00	2016.00	2020.00	2000.80	503753
140413	RELIANCE	2020-08-04 03:47:00+00:00	2022.00	2022.60	2015.80	179432
140414	RELIANCE	2020-08-04 03:48:00+00:00	2023.75	2025.00	2021.10	174483
140415	RELIANCE	2020-08-04 03:49:00+00:00	2023.30	2025.00	2022.00	184504
140416	RELIANCE	2020-08-04 03:50:00+00:00	2021.10	2023.70	2020.00	119961
140417	RELIANCE	2020-08-04 03:51:00+00:00	2018.25	2020.95	2018.00	77020
140418	RELIANCE	2020-08-04 03:52:00+00:00	2016.00	2019.00	2015.65	87179
140419	RELIANCE	2020-08-04 03:53:00+00:00	2019.10	2019.40	2015.50	63604
140420	RELIANCE	2020-08-04 03:54:00+00:00	2018.50	2019.90	2018.00	39786
140421	RELIANCE	2020-08-04 03:55:00+00:00	2023.25	2027.00	2018.30	154095
140422	RELIANCE	2020-08-04 03:56:00+00:00	2024.40	2025.00	2022.20	69895
140423	RELIANCE	2020-08-04 03:57:00+00:00	2020.65	2025.00	2020.15	85987
140424	RELIANCE	2020-08-04 03:58:00+00:00	2021.05	2022.00	2020.05	51012
140425	RELIANCE	2020-08-04 03:59:00+00:00	2021.35	2022.70	2020.20	69688
140426	RELIANCE	2020-08-04 04:00:00+00:00	2019.05	2021.80	2019.05	68017
140427	RELIANCE	2020-08-04 04:01:00+00:00	2017.30	2019.75	2017.00	59444
140428	RELIANCE	2020-08-04 04:02:00+00:00	2019.80	2020.00	2017.35	40895

fig. 1. stock price dataset.

B. DATA ANALYSIS AND VISUALIZATION

Here two columns were made: one for moving average of time period 9 minutes(ma9) and the other one for moving average of time period 21 minutes(ma21). These two moving averages have been used throughout our research and experiment. Then we made a function to split the data frame into sub-data frames based on frequency. For example, frequency = 1 would divide the data frame into a list of sub-data frames, each representing the total time of 1 day.

id	scrip__name	timestamp	close_price	high_price	low_price	volume	MA_9	MA_21
140411	RELIANCE	2020-08-04 09:15:00	2019.10	2021.90	2009.65	508731	NaN	NaN
140412	RELIANCE	2020-08-04 09:16:00	2016.00	2020.00	2000.80	503753	NaN	NaN
140413	RELIANCE	2020-08-04 09:17:00	2022.00	2022.60	2015.80	179432	NaN	NaN
140414	RELIANCE	2020-08-04 09:18:00	2023.75	2025.00	2021.10	174483	NaN	NaN
140415	RELIANCE	2020-08-04 09:19:00	2023.30	2025.00	2022.00	184504	NaN	NaN
140416	RELIANCE	2020-08-04 09:20:00	2021.10	2023.70	2020.00	119961	NaN	NaN
140417	RELIANCE	2020-08-04 09:21:00	2018.25	2020.95	2018.00	77020	NaN	NaN
140418	RELIANCE	2020-08-04 09:22:00	2016.00	2019.00	2015.65	87179	NaN	NaN
140419	RELIANCE	2020-08-04 09:23:00	2019.10	2019.40	2015.50	63604	2019.844444	NaN
140420	RELIANCE	2020-08-04 09:24:00	2018.50	2019.90	2018.00	39786	2019.777778	NaN
140421	RELIANCE	2020-08-04 09:25:00	2023.25	2027.00	2018.30	154095	2020.583333	NaN
140422	RELIANCE	2020-08-04 09:26:00	2024.40	2025.00	2022.20	69895	2020.850000	NaN
140423	RELIANCE	2020-08-04 09:27:00	2020.65	2025.00	2020.15	85987	2020.505556	NaN
140424	RELIANCE	2020-08-04 09:28:00	2021.05	2022.00	2020.05	51012	2020.255556	NaN
140425	RELIANCE	2020-08-04 09:29:00	2021.35	2022.70	2020.20	69688	2020.283333	NaN
140426	RELIANCE	2020-08-04 09:30:00	2019.05	2021.80	2019.05	68017	2020.372222	NaN
140427	RELIANCE	2020-08-04 09:31:00	2017.30	2019.75	2017.00	59444	2020.516667	NaN
140428	RELIANCE	2020-08-04 09:32:00	2019.80	2020.00	2017.35	40895	2020.594444	NaN

fig. 2. stock price dataset analysis.

After that, a graph was used to display closing price, ma9, and ma21 values for a given time period to visualise the trends between the three.
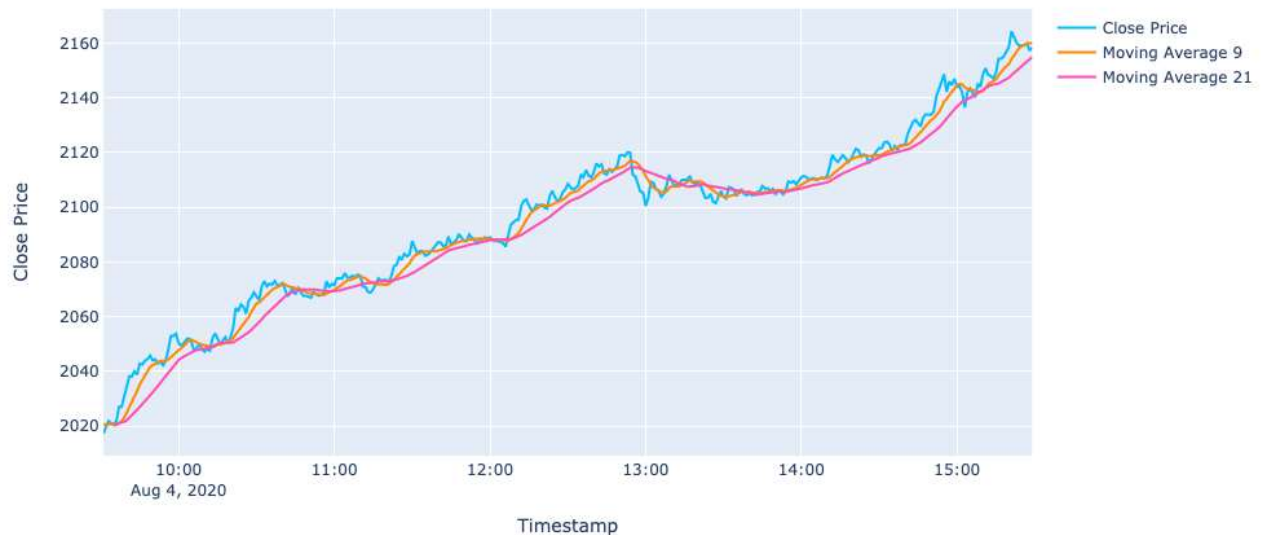
fig. 3. graph with stock price, moving average 9 and moving average 21.

C. THE ALGORITHM

We chose Reliance as our scrip and the time period from 1st June 2020 to 1st September 2020.

First, we made an algorithm in which we found the optimum values of stop-loss, a value used to reduce the amount of any loss incurred, and buffer, a value used to prevent losses because of the volatility of the market, for the previous day. Using these values, we found out the cumulative profit after brokerage for the current day. This algorithm was then made to run on three months of data to calculate the total sum of cumulative after brokerage for the same time period.

Unsatisfied with the results, we made another algorithm in which we found the optimum values of stop loss and buffer for three months. The algorithm, along with those values of stop loss and buffer, was then made to iterate through that data frame, and all the decisions were made on the same data frame only. Moving averages of time period 9 minutes and 21 minutes were used. First, on encountering a 'Crossover' of these two moving averages, it was checked whether the value of moving average 9 was increasing than moving average 21 after intersection and quantity in stock was not equal to 1 or the value of moving average 9 was decreasing than moving average 21 after intersection and quantity in stock was not equal to -1. If yes, it was followed by checking whether the area between moving average 9 was greater than the decided buffer value. Based on this, the decision to 'buy' or 'sell' was made respectively. A condition was added to check at the end of the day that whether the quantity in stock was zero or not. If not zero, then the respective buy or sell decision was made to balance it out.
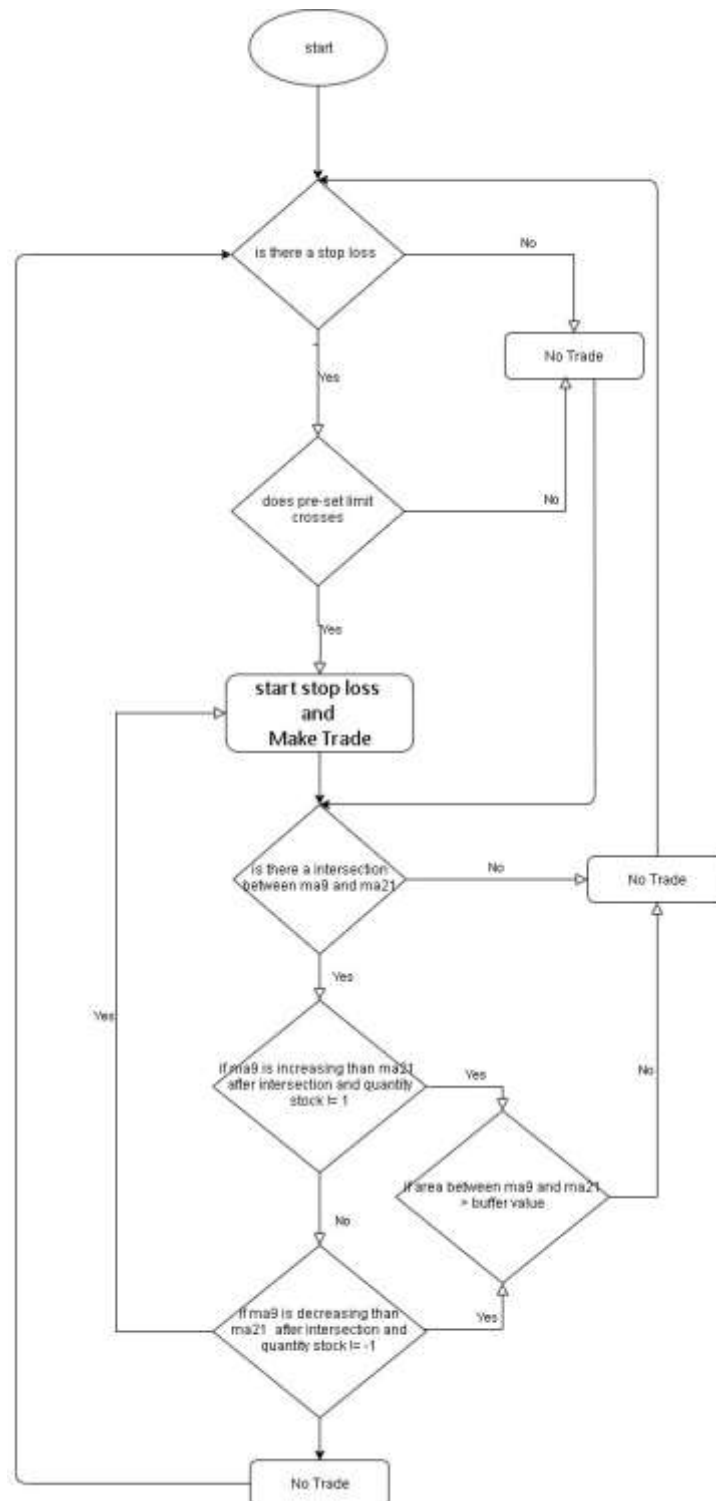
fig. 4. flowchart of the model.

A data frame was made by us to record the resultant transactions. It included the following columns:

scrip: It displays the names of different stocks.
timestamp: It displays the timestamp at any particular moment with a 1-minute separation between 2 timestamps.
buy/sell: It displays the decision taken at any particular moment.
price: It displays the close price value of stocks in that particular minute when the decision was made.
brokerage: It displays the small amount charged by brokers for providing their services.
qtystock: It displays the quantity of stocks currently in stack. 1 denotes that a stock is currently in 'buy' decision and needs to be sold to break even. -1 denotes that a stock is currently in 'sell' decision and needs to be bought to break even.
profit: It displays the profit earned during 1 complete transaction.
cum_profit: It displays the cumulative profit over a series of transactions.
pab: It displays the profit after brokerage earned during 1 complete transaction.
cum_pab: It displays the cumulative profit after brokerage over a series of transactions.

```
  scrip           timestamp buy/sell    price  brokerage qtystock profit cum_profit       pab    cum_pab
RELIANCE 2020-08-04 09:42:00      buy  2038.10   0.862116        1      0          0 -0.862116  -0.862116
RELIANCE 2020-08-04 13:03:00     sell  2109.00   1.356087        0   70.9       70.9 69.543913  68.681797
RELIANCE 2020-08-04 14:49:00      buy  2133.90   0.902640        1      0       70.9 -0.902640  67.779157
RELIANCE 2020-08-04 15:29:00     sell  2158.45   1.387883        0  24.55      95.45 23.162117  90.941274
```

fig. 4. example of transaction data frame.

IV. RESULTS AND DISCUSSION

In this paper, we wanted to prove the efficacy of cross over automated algorithm[6] for maximizing cumulative profit after brokerage. This is a complex task; the model needs to be robust enough to fluctuations in the stock market in the real-world environment. The dataset was complicated; it contained the past 6-months real-life data of various companies that were used to study the stock market in a diverse set of environments and backgrounds. Cross over strategy was chosen as it is good at detecting even small fluctuations in the trend of the stock market; it is fast and accurate, as a result of which, the algorithm can make decisions that lead to the maximum cumulative profit after brokerage. Our results indicate that the cross over strategy automated algorithm is indeed a trustworthy tool when it comes to making decisions in the stock trading world, as you will see in this result section below.

The data was scraped from Zerodha[4] API; Zerodha is a trading platform. Then a new file was made for this data and imported to postgress to get the data in RDBMS format.

The data was then further accessed by python jupyter notebook. A range of values was taken for stop loss and buffer; buffer value is the area between moving average 9 and moving average 21 at a particular timestamp. We then iterated over this range to find out the optimum values for both stop loss and buffer. These optimum values were then further used in the algorithm to find out the maximum total sum of cumulative profit after brokerage.

	Stop loss	Buffer	Total cumulative PAB	Average cumulative PAB	Average percentage cumulative PAB
37	0.025	0.0022	93.808229	1.839377	0.105479
38	0.025	0.0024	86.178797	1.795392	0.104630
47	0.030	0.0022	79.611070	1.561001	0.090034
27	0.020	0.0022	73.244808	1.436173	0.091576
48	0.030	0.0024	71.981638	1.499617	0.088229
28	0.020	0.0024	67.166031	1.399292	0.091818

fig. 5. resultant cumulative profit after brokerage for various combinations of stop loss and buffer value constants.
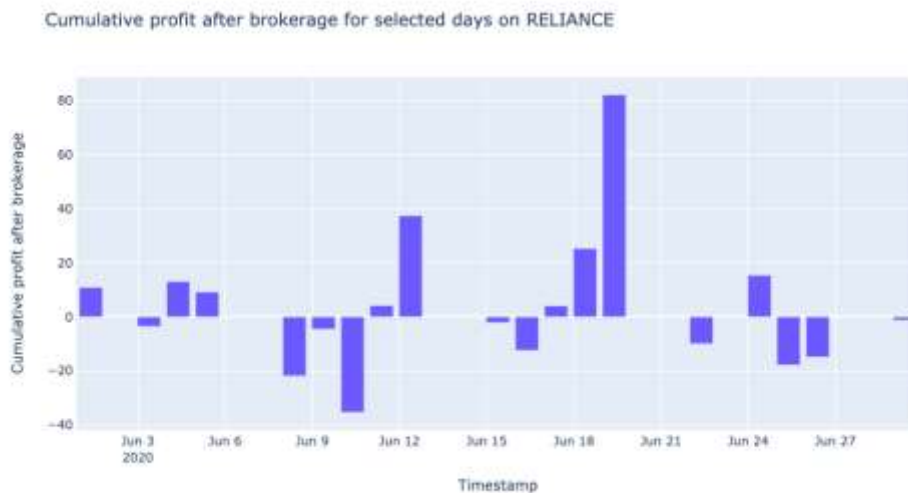

fig. 6. cumulative profit after brokerage of each open day for a month.

fig. 7. example of transactions for one day.

V. CONCLUSION

The main motive of our experiment is to prove the capability of our automated intra-day trading algorithm. We evaluated and implemented various strategies in our algorithm and came to the conclusion that an automated algorithm for intra-day trading is a good choice to be applied on the stocks of the company Reliance to make decisions about buying or selling. Automated algorithms are comparatively error-less and easier to implement than manual indicators. Our algorithm delivers a substantial profit of ₹93.80. We believe the reason behind this is that our model makes use of a dataset comprised of past 3 month data from reliance to understand the stock trading platform better. According to this study, our model outperforms manual indicators in the above-mentioned factors and provides an accurate method for maximizing profit.

REFERENCES

[1] B.UmaDevi, D.Sundar, Dr.P.Alli "*A Study on Stock Market Analysis for Stock Selection –Naïve Investors' Perspective using Data Mining Technique"*.

[2] https://www.investopedia.com/articles/active-trading/010116/perfect-moving-averages-day-trading.asp

[3] https://zerodha.com/brokerage-calculator#tab-equities

[4] https://zerodha.com/

[5] Kotaro Miwa, Kazuhiro Ueda "*The Influence of Investor Sentiment on the Formation of Golden-cross and Dead-cross"* the Association for the Advancement of Artificial Intelligence (AAAI).

[6] https://www.daytrading.com/moving-averages