Stock Market Forecasting using Ensemble Learning and Statistical Indicators

Anunay R. Bagga[*] , Harshita Patel[,**]

[*]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India-632014*

[**]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India-632014*

***Corresponding Author: hpatel.sati@gmail.com*

Submitted : 03-12-2021

Revised : 28-02-2022

Accepted : 21-03-2022

ABSTRACT

With a volume of 2 billion+ trades per day and a market capitalization of 2.56 trillion USD the national stock exchange (NSE), India is one of the largest stock exchanges in the world. Every day the value of stocks, commodities, bonds and futures fluctuate inducing volatility and forecasting these fluctuations to make money requires deep knowledge about the market and their historical data. Thus, a simple time series forecasting model is not enough to predict future movements as we need to know about the market sentiment, trend and industry fundamentals to bolster our stand of declaring a stock or commodity as bearish or bullish.

In this research, using machine learning forecasting models like Attention integrated Long Short Term Memory (LSTM) Model and a Reinforcement Learning agent coupled with statistical indicators and trading strategies like Auto Regression Integrated Moving Average (ARIMA), Prophet, Momentum trading and Pairwise trading to quantify the trend and market sentiment an approach to predict movements is devised. Using this approach increases the accuracy of stand-alone algorithms and helps in generating a cumulative analysis of the stock on the basis of itself and its stock universe data.

Keywords: Attention based Long Short Term Memory Network; Reinforcement Learning; Auto Regression Integrated Moving Average; Prophet; Stock Market; Time Series Prediction.

INTRODUCTION

In this day and age, the sudden boom in stock market has attracted people from almost all economical background. Due to the pandemic in 2020, markets had crashed significantly and when it corrected itself we saw the rise of many new millionaires. In this field, predicting the future plays a very important role and new strategies to do so using statistical and quantitative trading approaches are being developed rapidly. However, forecasting this heteroskedastic time series data requires two different approaches, i.e. Fundamental Analysis and Quantitative trading. Fundamental Analysis involve researching about the movement of the markets with respect to news, their sector movements and their quarterly reports showcasing the balance sheet and cash flow statements of the company. Metrics like Earnings per share (EPS) and Price to Earning (PE) ratio play an important role here as they help in calculating potential growth or losses the company may face later that year (AS, 2013; Roy, 2015). Contrarily, Quantitative analysis only rely on technical indicators, statistical models and graph reading. Traditionally, both these approaches were used separately to forecast market movements and algorithms like Linear Regression (Ashfaq, Nawaz, & Ilyas, 2021), ARIMA (Ariyo, Adewumi, & Ayo, 2014; Ullah et al., 2021), Rolling Mean and Bollinger Bands were used to estimate volatility. Although these methods worked initially, later on as more companies started listing and markets started diversifying, the accuracy of prediction these models had dropped sharply (Idrees, Alam, & Agarwal, 2019; Yang, Liu, Zhong, & Walid, 2020; Zou & Qu, 2020). Now, with the advent of Deep Learning models, industries have shifted to using AI and Machine Learning algorithms to forecast market movements and prediction in other research areas. In this Research, an approach to predict stock returns and generate signals to buy, sell or hold is developed using Attention-LSTM (Zou & Qu, 2020) and a Reinforcement Learning Agent (Li, Zheng, & Zheng, 2019; Yang et al., 2020) for quantitative analysis and their decisions are bolstered by using momentum trading, trend analysis and a pair wise behavioural strategy for quantifying the fundamentals and the current state of the market sentiment which a quantitative approach fails to account for. The input of this model is the Open, High, Low, Close (OHLC) time series data coupled with volume, stock coefficients and their adjusted close of multiple companies and it outputs the future price predicted and a signal to buy, hold or sell shares for each.

In this paper, firstly all algorithms and models used in the research are listed followed by the results obtained by using them as stand-alone use cases. At the end, the proposed model findings are mentioned. The objective of this paper is to:

a. Incorporate various algorithms towards building a model for accurately predicting market movements.
b. Use different point of views bought in by the different strategies constructively to bolster each model's strength.
c. To showcase the diversity of stock market prediction algorithms and their specific use cases.

RELATED WORK

In recent years, many models and algorithms have been developed for predicting stock prices and market behaviour. Many of these models focus on fundamental analysis of a company data for predicting movements (AS, 2013; Roy, 2015) whereas some only refer to the historical time series data and graph reading skills (Ariyo et al., 2014; Ashfaq et al., 2021; Idrees et al., 2019; Li et al., 2019; Pahwa, Khalfay, Soni, & Vora, 2017; Ray, Khandelwal, & Baranidharan, 2018; Yang et al., 2020; Zou & Qu, 2020). However, due to the ever-changing nature of the stock market all algorithms become inaccurate as more people discover their use so new models and algorithms take their place to keep up with this change. Presently, RNN and LSTMs are considered to predict prices most accurately (Ashfaq et al., 2021; Idrees et al., 2019; Ray et al., 2018; Zou & Qu, 2020) and reinforcement learning agents are considered as the future of stock prediction (Li et al., 2019; Yang et al., 2020). Although this is true, combining multiple different models and using their results to predict a pump or a dump in the prices generates much better results than any stand-alone model (Ashfaq et al., 2021; Roy, 2015; Ullah et al., 2021). Due to this, the use of ensemble learning techniques in the prediction models is increasing rapidly as all investors are looking to maximize their profits and cannot ignore either fundamental or technical analysis. Also ensemble learning has already proved to a worthy approach in different researches (Basha, Rajput, & Vandhan).

Other than these dedicated research work on stock prediction, some notable machine learning applications can also be considered as various prediction strategies that can further lead to predict stocks insights too. Starting from the basic data mining strategies (Basha & Rajput, 2018, 2019; H Patel & Rajput, 2011) to today's advanced machine learning approaches (Harshita Patel, Rajput, Stan, & Miclea, 2022; Harshita Patel et al., 2020; Tripathy, Parimala, & Reddy, 2021), information prediction got enough attention from research community and helped them to find important insights from various types of data(Harshita Patel & Thakur, 2019; Harshita Patel & Thakur). Some of these pre-approved approaches from other application can be applied in stock prediction as well (Alazab et al., 2020). The article mainly concerns about the latest approaches for stock market prediction.

DATASET AND FEATURES

The dataset used for this research includes companies listed in the NIFTY 50 index of the NSE stock exchange in India extracted from Alpha Vantage API Documentation (https://www.alphavantage.co/documentation). It consists of daily “Open”, “High”, “Low”, “Close”, “Volume”, “Adjusted Close”, “Split Coefficient”, “Dividend” and “Volume” for each company starting from the day it was listed on the exchange. As a part of pre-processing stage 2 more columns were added to this dataset which are “Return” and “Log Return” to further improve our models performance and get better evaluation metrics. Their calculation is as follows:

$$Return\ [day]\ =\ adj\_close[day + N] - adj\_close[day]. \tag{1}$$

$$LogReturn[day] = \ln\left(\frac{adj\_close[day+N]}{adj\_close[day]}\right). \tag{2}$$

The reason a 5-day period is selected (N = 5) in this research is because the evaluation is for calculating weekly returns rather than daily. Usage of logarithms is implemented because it scales the return between [-1,1] as our returns are mainly concerned with percentage growth so stocks with higher value per share will always show high numbers in comparison to other mid-caps or small-caps if log returns is not used. Another reason of using Logarithm is its additive property. If we need to calculate the returns of more than 1-time period, then summation of all log returns between start and end date can give the desired results because:

$$\ln\left(\frac{adj\_close(i+N)}{adj\_close(i)}\right) = \sum_{n=0}^{N-1} ln \cdot \left(\frac{adj\_close(i+n+1)}{adj\_close(i+n)}\right) . \qquad (3)$$

Where, i = Initial Day
N = Period till which Log Return to be calculated
A sample of the dataset used is shown in Table 1.

Table 1: Sample Dataset of ONGC stock price

Timestamp	*Open*	*High*	*Low*	*Close*	*Adjusted Close*	*Volume*	*Dividend amount*	*Split Coefficient*	*Return*	*Log return*
2021-02-26	*115.9*	*118.4*	*110*	*111.15*	*111.15*	*3495120*	*0*	*1*	*3.75*	*0.03318*
2021-02-25	*115.6*	*120.5*	*115.4*	*119*	*119*	*6220965*	*0*	*1*	*-6.30*	*-0.0543*
2021-02-24	*113.8*	*115.5*	*101*	*113.65*	*113.65*	*6161383*	*0*	*1*	*0.3*	*0.00263*

ALGORITHMS AND TRADING STRATEGIES

Momentum Trading:

One of the most rudimentary trading strategies is momentum trading to generate trading signals. A trading signal is a sequence of trading actions that can be used to take trading actions. In this strategy, for each month a ranking of stocks is made using their previous returns and the top performing stocks are selected for investing. The bottom performing stocks can be selected to if needed to develop a short portfolio (Foltice & Langer, 2015).

Pair-wise Trading:

This trading strategy tries to find the most similar group of stocks and then assumes that all of them try to move towards the total mean price of the group (Diamond, 1990). Also known as mean reversion principle, this strategy keeps into account the market behaviour and sentiment towards the sector a stock belongs to since the most similar stocks is found in companies belonging to the same industry. If we see steep decline across multiple companies of a sector, chances of a fall of top performing companies in that group becomes very high.

In this project, a cosine similarity metric was used to find the most similar group of stocks. On the basis of how far they were from the mean, a score was allocated to each stock. A positive score signified that the value of stock is less than the mean of its similarly performing stocks so a growth is predicted whereas a negative score meant that the stock is already giving higher returns than expected and is hence predicted to fall in the coming future.

$$\text{cosine\_similarity}(a, b) = \frac{\sum(a \cdot b)}{\sqrt{\sum a^2} . \sqrt{\sum b^2}} . \qquad (4)$$

a, b are returns of stock A & B respectively

Figure 1 depicts the similar trends in ONGC, GAIL and IOC all three of which belong to Energy and Petroleum sector.

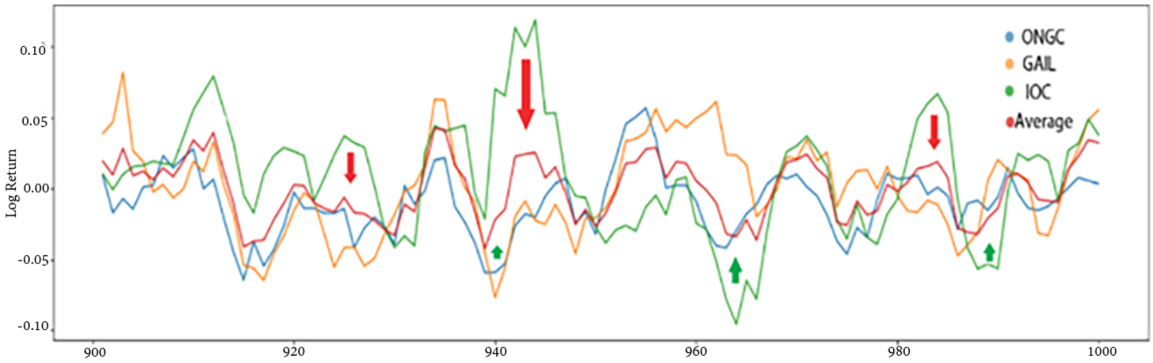


Figure 1: Data of ONGC, GAIL and IOC showing similar trend and mean reversion

Prophet Trend Analysis and Newton Forward Interpolation:

Prophet is a procedure for forecasting time series data based on an additive model which takes into account yearly, weekly and daily seasonality developed by Facebook (Taylor & Letham, 2018). They use a decomposable time series model following the equation:

$$y(t) = g(t) + s(t) + h(t) + \in \quad . \qquad (5)$$

Here,

- g(t) is the trend function which calculates the volatility in the time series in a non-periodic approach
- s(t) calculates volatility on a weekly or yearly seasonality.
- h(t) calculates the effect of holidays on the time series.
- ∈ is an error term. Since this research involves predicting weekly returns we only need trend analysis to work with as it is a very short time period.

The prophet model requires 2 parameters named ‘ds’ and ‘y’ which in this case will the ‘timestamp’ and Z-scaled ‘adjusted close’.

$$\text{Z} - \text{Scale} = \frac{\text{Stock[adj\_close]} - mean(\text{Stock[adj\_close]})}{\text{std(Stock[adj\_close])}} \quad . \qquad (6)$$

Using these we can get a simplified trend of the stock being studied as shown in Figure 2.

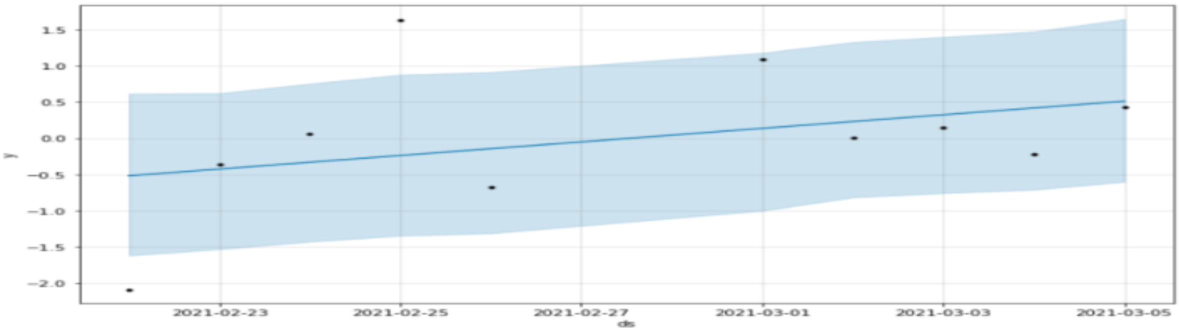


Figure 2: ONGC Trend for 20 days. Dark Blue line is the trend. Black dots are the Z-scaled adjusted close and the blue region shows the trend ± standard Deviation Zone.

Since the Prophet trend analysis algorithm outputs the coordinates of the trend curve, performing Newton Forward Interpolation at the last point can calculate the instantaneous slope of the equation helping in quantifying the momentum of the stock (Das & Chakrabarty, 2016). It can be used as follows:

$$y(t) = y_0 + p.\Delta y_0 + \frac{p(p-1).\Delta y_0^2}{2!} + \dots \frac{(p(p-1)..(p-n+1).\Delta y_0^n)}{n!} \quad . \qquad (7)$$

Here, y(t) will give us an equation of the trend line which later on is being differentiated and t substituted can give the slope which is a quantitative measure of the momentum of the stock. Slope for Figure 3 is -1.22. A positive slope shows a growing momentum and a negative one quantifies the fall.

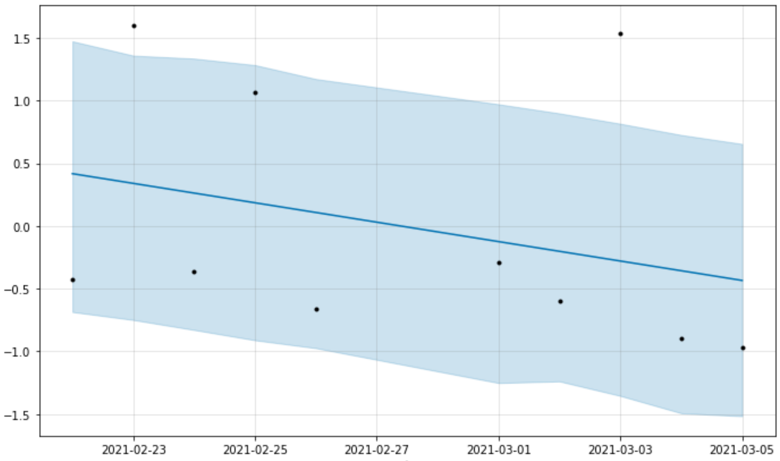


Figure 3: Trend Analysis of M&M for 20 days

Auto Regressive Integrated Moving Average (ARIMA):

ARIMA model is a statistical Time Series analysis and forecasting method in which data is differentiated at least once to make it stationary and solve the autocorrelation problem and then passed on to an Auto Regression and Moving Average integrated equation (Ariyo et al., 2014; Ray et al., 2018). Auto Regression model derives a linear equation to predict the next time series data by using p lags of the dataset whereas Moving Average model derives a linear equation using q lagged forecast errors.

Auto Regression:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots \beta_p Y_{t-p} + \varepsilon_t. \qquad (8)$$

Moving Average:

$$Y_t = \alpha + \varepsilon_t + \gamma_1 \varepsilon_{t-1} + \gamma_2 \varepsilon_{t-2} + \cdots \gamma_q \varepsilon_{t-q}. \qquad (9)$$

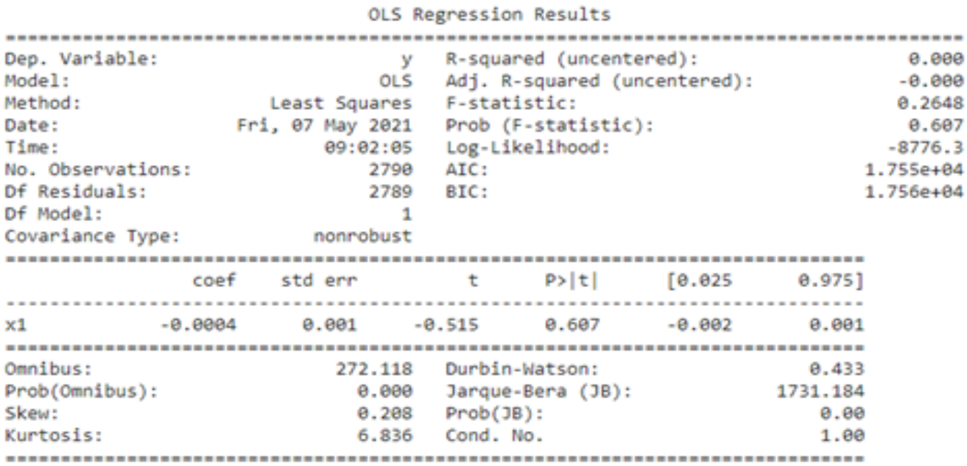
ARIMA:

$$Y_t = \alpha + AutoRegression(Y_t) + MovingAverage(Y_t)\ . \qquad (10)$$

For calculating the parameters (p,q,d) where d is the order of differentiation, the steps followed were:

- Conduct an Ordinary Least Square Regression Test (OLSR test) to analyse the relation between each data point and check for autocorrelation problem.
- Using the Durbin – Watson test coupled with ORS to identify whether an autocorrelation problem exists or not using this method.

After conducting the test on ONGC stock, a strong positive autocorrelation was found with a Durbin-Watson value of 0.433 since the underlying data is a time series data and linear regression models fail in such cases.



```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                      y   R-squared (uncentered):                   0.000
Model:                            OLS   Adj. R-squared (uncentered):             -0.000
Method:                 Least Squares   F-statistic:                             0.2648
Date:                Fri, 07 May 2021   Prob (F-statistic):                       0.607
Time:                        09:02:05   Log-Likelihood:                         -8776.3
No. Observations:                2790   AIC:                                  1.755e+04
Df Residuals:                    2789   BIC:                                  1.756e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.0004      0.001     -0.515      0.607      -0.002       0.001
==============================================================================
Omnibus:                      272.118   Durbin-Watson:                   0.433
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1731.184
Skew:                           0.208   Prob(JB):                         0.00
Kurtosis:                       6.836   Cond. No.                         1.00
==============================================================================
```

Figure 4: OLS Regression test results for undifferentiated data

- Therefore, after double differentiating the data the problem of autocorrelation is resolved.
- For choosing p, an autocorrelation vs. lag graph was plotted for a double differentiated dataset and a point giving minimal autocorrelation is selected. For ONGC a lag of 4 was selected as per the graphs 5a and 5b below.

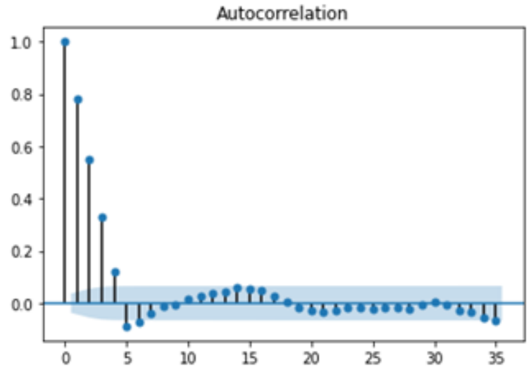


Figure 5a: Single Differential graph of autocorrelation vs lag plot. As observable, in the range [0,4] and strong positive correlation is present and after a wave like pattern emerges

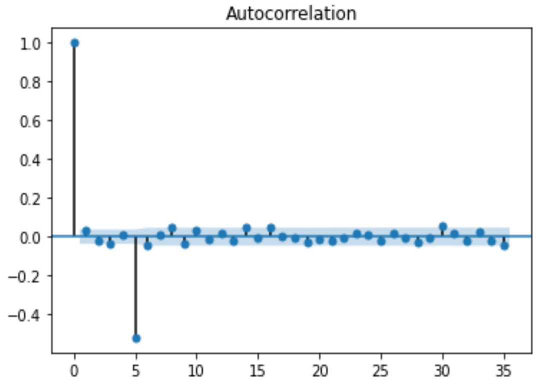


Figure 5b: Double Differential graph of autocorrelation vs lag plot. At 4 a 0 autocorrelation is found after which a deep negative correlation occurs at 5 since a week ends after 5 days in the stock universe so a deviation is expected here.

- q is selected as 2 for all as per Holt's linear method with additive errors model also called as double exponential smoothing.
-

Stacked Attention Based Long Short Term Memory (LSTM):

LSTM introduce the concept of memory to classical Neural Networks. It does so by passing the results of previously iterated data through the network and processes current and previous data calculation in 4 different stages or gates which are Forget Gate, Input Gate and Output Gate. This 3 stage calculation method helps forecasting time series data easily and efficiently.

The Forget Gate (f_t) determines which information to store and which is no longer needed and is forgotten. It uses previous time step (h_{t-1}) and current step (x_t) and gives a value ranging between [0,1] to show how much memory should be discarded. (0 implies no memory is forgotten and 1 implies complete deletion)

The Input Gate determines how much of the information from previous states is relevant to the current input state. It involves getting the state of the current cell (i_t) then creates a new cell state ($\bar{C}_t$) and finally updates the current cell with the new state values (C_t)

The Output Gate filters the memory in the new cell and the resultant value (O_t) is processed with the updated cell state to get the time step of current cell (h_t).

Figure 6 (Varsamopoulos, Bertels, & Almudever, 2019) below includes all calculations used to get the fields mentioned above.

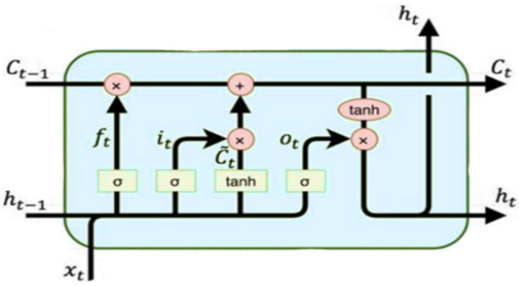


Figure 6: LSTM Cell

In this research, a stacked LSTM model is used which means there are multiple layers of LSTM cells incorporated into the network and Attention Model is also integrated to this. An Attention mechanism helps the model to concentrate on only the important aspects of its memory states and gives more weightage to them. This method is viable for stock markets too as all the parameters in a stock dataset are of different importance levels and some have a higher impact on stocks volatility than the others (Ashfaq et al., 2021; Pahwa et al., 2017; Ray et al., 2018; Zou & Qu, 2020). An Attention vector can be calculated using Attention weights and a context vector as follows:

$$\alpha_t = \frac{e^{(h_t, h_s)}}{\sum_{s=1}^{S} e^{score(h_t, h_{s'})}} \quad \text{[Attention Weights].} \quad (11)$$

$$c_t = \sum \alpha h_s \quad \text{[Context Vector].} \quad (12)$$

$$\alpha_t = \tanh\left(W_c[c_t; h_t]\right) \quad \text{[Attention Vector].} \quad (13)$$

Here W_c is the weight matrix and is a trainable parameter. The emphasized information is marked by the Attention vector. The Attention weights weigh in input and replace it with the new weighted input sequence.

This updated sequence can improve the LSTM model by signifying the specific key features and ignore the redundant ones.

The model's description used in this research is given in Table 2 below.

Table 2: Stacked LSTM with Attention Integrated model used in this research

Layer (type)	Output Shape	Number of Parameters
LSTM_	(None, 20, 64)	16896
Dropout	(None, 20, 64)	0
LSTM	(None, 20, 32)	12416
Dropout	(None, 20, 32)	0
Last_hidden_state (Lambda)	(None, 32)	0
Dense Attention score vector	(None, 20, 32)	1024
Attention score (Dot)	(None, 20)	0
Attention Weight Activation	(None, 20)	0
Context Vector	(None, 32)	0
Attention output Concatenate	(None, 64)	0
Dense Attention Vector	(None, 128)	8192
Dense Layer	(None, 1)	129

Reinforcement Learning Agent:

A Q-Learning based agent is used to generate buy, hold and sell signals (Li et al., 2019; Yang et al., 2020). A 4 layered CNN with ReLU activation function is developed and states with a window of 30 days adjusted closing price data is passed through this neural network to generate the 3 trading signals. All buy signals store the current price of a stock in a queue and with each sell signal a dequeue operation is conducted and a profit or loss is calculated. A reward score is then calculated depending on how accurate the signals were to generate a profit. By doing so, the current state, trading signal, reward and next state is stored in the agent's memory and the weights are re-evaluated for each iteration.

PROPOSED METHOD

To predict the highly volatile and heteroskedastic movements in a stock market, relying on a singular algorithm for forecasting trends and investing money is never a smart idea. Instead what this research proposes is using multiple Deep Learning and Statistical algorithms to make an informed decision . Using Stacked LSTM and ARIMA for predicting the closing price of the next 5 days of the week and then bolstering that decision with the help of trend analysis algorithms i.e. Momentum trading, Pair-wise trading and Prophet Algorithm helps increase the accuracy of the time series prediction. Adding a reinforcement learning agent to this equation also has a supplementary effect on the results as the buy/hold/sell signals pass through another filter of scrutiny before final results. Figure 7 below is a diagrammatic representation of the method used in this research and shows how all algorithms are being used to get our final trading signals.

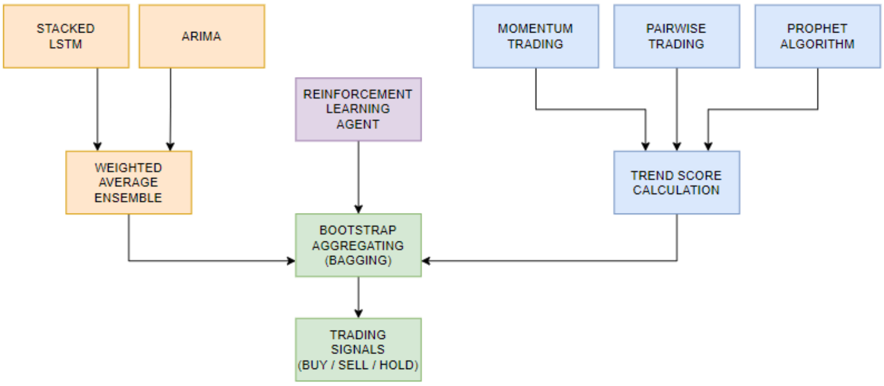

STACKED LSTM
ARIMA
MOMENTUM TRADING
PAIRWISE TRADING
PROPHET ALGORITHM
REINFORCEMENT LEARNING AGENT
WEIGHTED AVERAGE ENSEMBLE
TREND SCORE CALCULATION
BOOTSTRAP AGGREGATING (BAGGING)
TRADING SIGNALS (BUY / SELL / HOLD)

Figure 7: Proposed Model

Weighted Average Ensemble as the name suggests assigns weights to the forecasted values generated from both Stacked LSTMs and ARIMA models and multiplies them to their respective results to get a weighted result which takes both of these algorithms into account. The weights are in the range of [0, 1] and can be changed according to the user. The equation is as follows:

$$\mathrm{P_t} = w_1 \,.\, P_{t_1} + \, w_2 \,.\, P_{t_2} + \cdots \,. \tag{14}$$

Here w$_1$, w$_2$ are the weights and P_{t_1}, P_{t_2} are the predicted prices on day t generated by model 1 and 2 respectively. Since the output is the predicted price on day t its value ranges depending on the stock and won't be a uniform metric to compare with others, hence log returns is used instead of closing prices and then they are passed on for Bagging.

Trend Score Calculation also follows the same logic and generates a singular metric recapitulating all trend analysis algorithms used. The value output of this method lies in the range of [-1, 1] where values closer to -1 mean sell and 1 indicate buy. Hold signals can be assigned as 0 or values within a certain range with 0 as their mean. For this research Hold signal was interpreted if trend analysis had an output in the range of [-0.3, 0.3]. Since, both Momentum trading and Pair wise generate values between [-1, 1] but Prophet Algorithm generates values in the range of $[-\infty, \infty]$ a MinMax Scaling Algorithm is used on it to scale down the data to the acceptable range of [-1, 1] and then weights are assigned to all three of them to get final trend scores.

RESULTS & DISCUSSION

Momentum Trading:

The momentum of individual stocks in the stock universe is as follows in table 3.

Table 3: Momentum scores of all stocks

Stock	Momentum Value
M&M	0.2986
HINDALCO	0.2877
BAJFINANCE	0.2438
TATASTEEL	0.2410
JSWSTEEL	0.2383
SBIN	0.2301
INFY	0.2274
RELIANCE	0.2137
TCS	0.2137
HCLTECH	0.2109
HEROMOTOCO	0.2027
UPL	0.2027
ADANIPORTS	0.1918
GAIL	0.1863
SHREECEM	0.1808
ULTRACEMCO	0.1671
BPCL	0.1644
POWERGRID	0.1644
ONGC	0.1616
NESTLEIND	0.1616
MARUTI	0.1534
HINDUNILVR	0.1472
LT	0.1425
HDFC	0.1342
NTPC	0.1233
IOC	0.1095

According to this data, the top performing stocks are M&M, HINDALCO, BAJFINANCE, TATASTEEL and JSWSTEEL so it's a signal to buy these stocks. The top dropping stocks are IOC, NTPC, HDFC, LT and HINDUNILVR which indicates a plummet in their values so a sell or short signal is generated for them. According to this if a long only strategy is followed, a mean return of the stock universe comes out as -70.561 INR for 5 days which means if we buy 1 share in all of the stocks listed we will make a 70 INR average loss for each share however if only the top performing stocks are invested in, the mean return is equal to 67.303 INR.

Although this result seems promising, upon conducting a one tailed t-test on the results, a p-value of 0.176 is calculated. Since p-value > 0.05, the hypothesis that the portfolio mean is not significantly greater than global mean stands therefore momentum as a stand-alone strategy for trading fails. Also the accuracy of this algorithm is only 55.61% when tested on 1825 days.

Pair-Wise Trading:

Stocks deemed to be similar (using cosine similarity) to a selected stock and it's calculated score predicting a price increase or decrease is shown in Table 4. Since log returns is used in this case as we are comparing a stock value to others and a need for standardization rises the mean return of the stock universe is 0.24 and the mean return of our predicted to increase stocks i.e. BPCL, NTPC, TCS, HCLTECH, SBIN, SHREECE, HINDUNILVR, M&M and LT is 0.261. Although these numbers are small the difference between them is huge as inverse log of 0.261 gives 1.296 which means a 29.6% profit on total investment. Since these are log return values, the method of mean return calculation changes, here instead of buying 1 share in each stock we are buying a number of shares with total monetary value equal to the greatest valued share in the stock universe. This means if 100 INR is the highest valued stock in the universe we will be buying 100 shares of a 1 INR stock instead of only 1 share as done previously. Hence, upon conducting a one-tailed t-test on this data, a p-value of 0.07 is calculated. Though this value is still greater than 0.05 and null hypothesis still holds but it is far better than momentum trading and an accuracy of 67% is also seen when pair wise trading had been tested for 2600 days.

Table 4: Pairwise Similarity Table

Stock	Similar Stocks	Score
ONGC	GAIL, TATASTEEL, IOC	0.0116
GAIL	ONGC, HINDALCO, TATASTEEL	-0.0103
MARUTI	JSWSTEEL, LT, ULTRACEMCO	0.0054
HEROMOTOCO	MARUTI, JSWSTEEL, HCLTECH	-0.0144
NESTLEIND	HINDUNILVR, LT, RELIANCE	-0.0072
ULTRACEMCO	SHREECEM, LT, JSWSTEEL	-0.061
HINDUNILVR	NESTLEIND, MARUTI, ULTRACEMCO	0.0489
IOC	BPCL, ONGC, GAIL	0.0061
LT	ULTRACEMCO, BAJFINANCE, SHREECEM	0.0864
RELIANCE	HCLTECH, UPL, TCS	0.0879
BPCL	IOC, LT, GAIL	0.015
ADANIPORTS	LT, ULTRACEMCO, HINDALCO	-0.0626
BAJFINANCE	LT, HDFC, SBIN	-0.0337
M&M	GAIL, TATASTEEL, ONGC	0.0504
INFY	HCLTECH, TCS, RELIANCE	0.0007
TCS	HCLTECH, INFY, UPL	0.0196
NTPC	POWERGRID, LT, IOC	0.0187
HDFC	SBIN, BAJFINANCE, LT	0.005
JSWSTEEL	HINDALCO, TATASTEEL, ULTRACEMCO	0.0106
HCLTECH	INFY, TCS, JSWSTEEL	0.02
POWERGRID	NTPC, ONGC, IOC	0.0067
SBIN	HDFC, TATASTEEL, HINDALCO	0.0226
SHREECEM	ULTRACEMCO, LT, HDFC	0.0279
TATASTEEL	HINDALCO, JSWSTEEL, GAIL	-0.0112

HINDALCO	JSWSTEEL, TATASTEEL, GAIL	0.0494
UPL	LT, JSWSTEEL, ADANIPORTS	0.0003

Modified Prophet Algorithm:

The following graphs in figure 8 plot daily seasonality and trend obtained for a stock using Facebook's prophet algorithm.

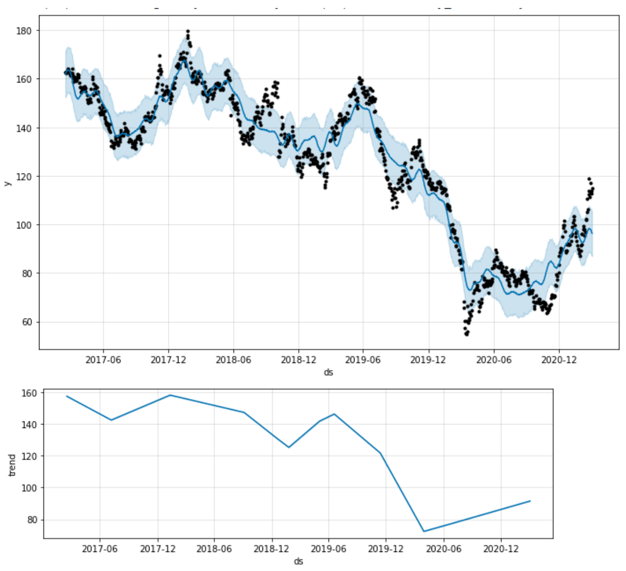


Figure 8: Facebook Prophet Algorithm trend analysis for ONGC

For the dates 22nd Feb. 2021 to 5th Mar. 2021 (10 days) window the trend values for 6th Mar. 2021 is as mentioned in the table 5 below:

Table 5: Prophet Analysis Table.

Stock	Trend Value	Inference
ONGC	1.481	Buy
GAIL	0.122	Hold
MARUTI	2.762	Buy
HEROMOTOCO	0.750	Hold
NESTLEIND	3.488	Buy
ULTRACEMCO	3.171	Buy
HINDUNILVR	1.895	Buy
IOC	3.425	Buy
LT	-0.651	Sell
RELIANCE	3.408	Buy
BPCL	3.321	Buy
ADANIPORTS	3.620	Buy
BAJFINANCE	0.256	Hold
M&M	-1.226	Sell
INFY	3.215	Buy
TCS	2.250	Buy
NTPC	3.385	Buy
HDFC	-2.223	Sell
JSWSTEEL	0.934	Buy
HCLTECH	3.287	Buy
POWERGRID	-0.046	Hold
SBIN	-0.906	Sell
SHREECEM	2.582	Buy
TATASTEEL	2.563	Buy

HINDALCO	2.498	Buy
UPL	3.306	Buy

In this, all stocks with values in range [-0.75,0.75] are marked as Hold, others with positive values are given Buy signals and remaining negative valued stocks have a Sell Signal. Similar to Pairwise Trading, using log returns will give more accurate test results hence if we take only the top performing stocks i.e. BPCL, NTPC, RELIANCE, IOC, NESTLEIND and ADANIPORTS, portfolio mean return equals 0.395 whereas the stock universe mean return remains the same at 0.24. These numbers are already much better than both momentum and pairwise trading and conducting a one tailed t-test on this only reinforces the statement by giving a p-value of 0.035 which means null hypothesis is rejected and portfolio returns are significantly greater than stock universe. The model also gives an accuracy of 84.6% when tested for 1000 data points.

Auto Regressive Integrated Moving Average(ARIMA):

The scores generated depending on how much the prices are expected to move in the span of 100 days for each stock are as Table 6:

Table 6: ARIMA Results

Stock	ARIMA Score
SHREECEM	26.401
BAJFINANCE	4.015
POWERGRID	2.283
M&M	0.402
ULTRACEMCO	0.166
IOC	0.011
BPCL	-1.139
SBIN	-1.527
HINDUNILVR	-1.869
RELIANCE	-1.978
TCS	-2.116
HCLTECH	-2.246
GAIL	-2.658
INFY	-2.954
ONGC	-3.056
JSWSTEEL	-3.134
NTPC	-3.346
LT	-3.779
HDFC	-5.166
ADANIPORTS	-7.132
HINDALCO	-8.169
TATASTEEL	-17.252
HEROMOTOCO	-17.957
NESTLEIND	-20.927
MARUTI	-31.054

By considering the top 6 performing stocks in this case which are SHREECEM, BAJFINANCE, POWERGRID, M&M, ULTRACEMCO and IOC as our portfolio the mean return for after a total of 100 days of back testing is 2.55 and the global mean for stock universe after 100 days' equals 1.77. This implies a profit is generated and by conducting a one tailed t-test a p-value of 0.03 is calculated which means that the mean return of the portfolio is significantly greater than the stock universe.

If the forecasting accuracy is analysed instead of the scores, the following metrics are observed when comparing predicted returns and actual returns:

Table 7: ARIMA model Analysis

Evaluation Metric	Value
Mean Absolute Percentage Error	0.0344
Mean Error	-0.047
Mean Absolute Error	4.056
Root Mean Square Error	8.901
MinMax Error	1.01
AutoCorrelation Function	0.78
Correlation	0.957

A plot of the actual returns vs back tested data of ONGC since it was listed on 3rd January 2005 is shown in figure 9.

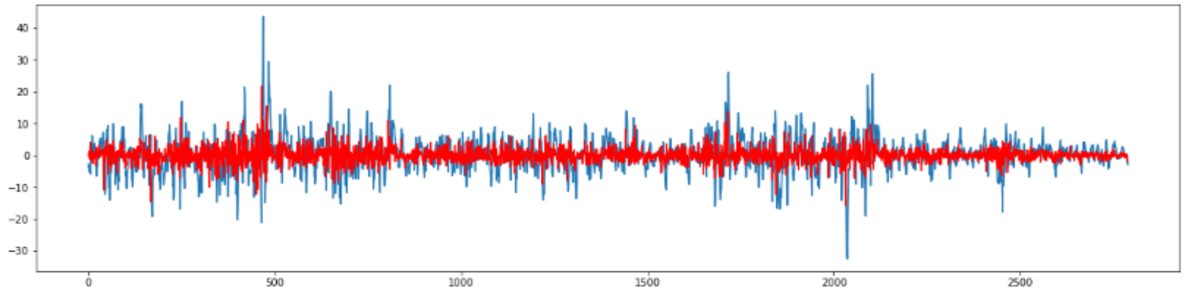


Figure 9: ONGC Stock 5 day return plotted in blue and the ARIMA forecasted return calculated for each day plotted in red

Stacked Attention based Long Short Term Memory(LSTM):

To build a robust and accurate LSTM model, standardization is a crucial step during pre-processing. This is because having different valued features between each stock value will cause the network to assign the weights based on a false prioritization manner. Stocks having large market capitalization will be influencing the model more as compared to mid-caps and small-caps, hence using log returns and further adding a standard scaler to the pre-processing pipeline is beneficial. The dataset after this stage is standardized and is shown in figure 10.

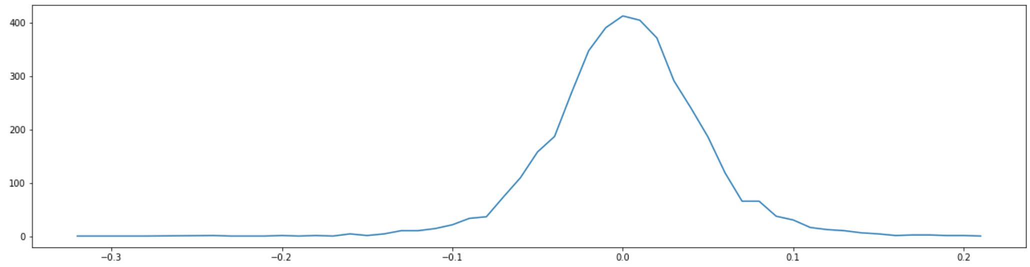


Figure 10: Standardized log return vs frequency graph of GAIL

When this data is passed through the LSTM, a Mean Squared Error (MSE) of only $5.9 * 10^{-3}$ on training data and $6.2 * 10^{-3}$ on valuation data is observed after training the model for 50 epochs per stock. Clearly so far the Attention based LSTM generated the best results and figure 11 accurately represents how the model performs for the GAIL stock.

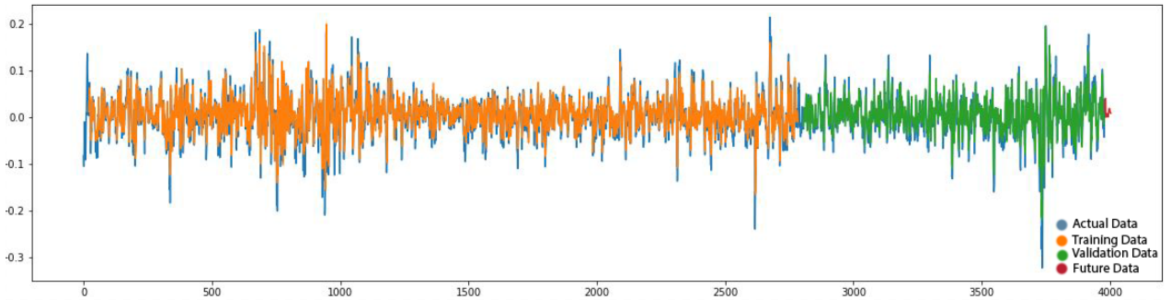


Figure 11: LSTM model output for GAIL

If zoomed in on the future data part of the plot which means 25 days in future from the last day present in the dataset, it is observed that after 5 days the curve starts to flatten out as shown in figure 12. This is because our model is trained on 5 days' period log returns so even the scope of the model is for predicting weekly prices of a stock rather than monthly stock volatility.

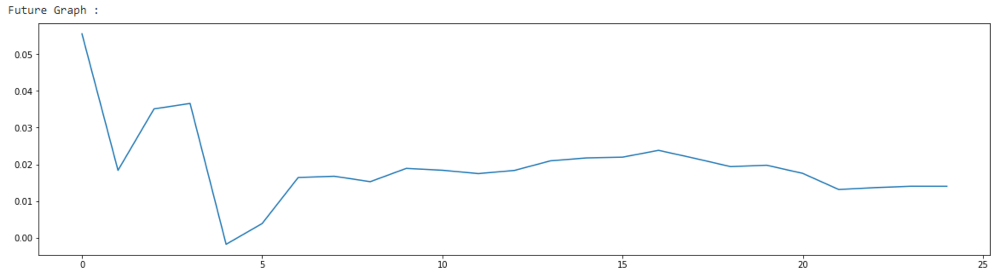
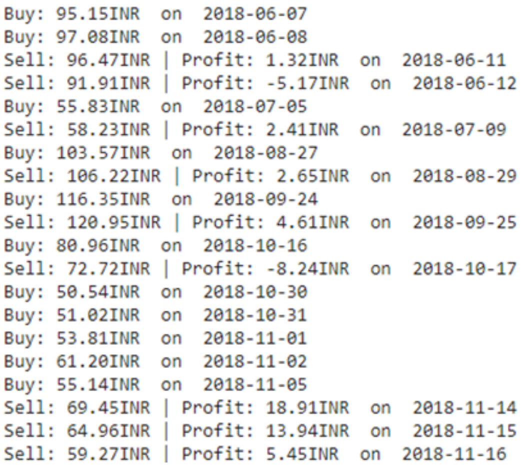


Figure 12: Zoomed In plot of Future for GAIL

Reinforcement Learning Agent:

The RL agent generated a profit of 130% after iterating through 50 episodes when back tested on ONGC for the period of 2016-2019. The reason 2020 and 2021 were not used is that the Covid-19 pandemic which struck then crashed the markets and created an outlier phenomenon, hence considering them in this study of the RL agent would have given inaccurate results. The output of the agent is a series of trading signals being generated daily and in the back-end a queue stores the data of these signals and calculates the total profit as well as profit per sell order. Figure 13 shows how these signals are being used and how profits are being calculated.



```
Buy: 95.15INR  on  2018-06-07
Buy: 97.08INR  on  2018-06-08
Sell: 96.47INR | Profit: 1.32INR  on  2018-06-11
Sell: 91.91INR | Profit: -5.17INR  on  2018-06-12
Buy: 55.83INR  on  2018-07-05
Sell: 58.23INR | Profit: 2.41INR  on  2018-07-09
Buy: 103.57INR  on  2018-08-27
Sell: 106.22INR | Profit: 2.65INR  on  2018-08-29
Buy: 116.35INR  on  2018-09-24
Sell: 120.95INR | Profit: 4.61INR  on  2018-09-25
Buy: 80.96INR  on  2018-10-16
Sell: 72.72INR | Profit: -8.24INR  on  2018-10-17
Buy: 50.54INR  on  2018-10-30
Buy: 51.02INR  on  2018-10-31
Buy: 53.81INR  on  2018-11-01
Buy: 61.20INR  on  2018-11-02
Buy: 55.14INR  on  2018-11-05
Sell: 69.45INR | Profit: 18.91INR  on  2018-11-14
Sell: 64.96INR | Profit: 13.94INR  on  2018-11-15
Sell: 59.27INR | Profit: 5.45INR  on  2018-11-16
```

Figure 13: RL Agent Output

Ensembled Model:

When compared, all the models individually can be ranked as the most accurate (i.e. LSTM) to the least accurate one (i.e. Momentum Trading) however using the ensemble model this research has further increased the accuracy observed by any of the mentioned stand-alone algorithms generating a profit of approximately 250% of the initial value invested over the span of 5 years with an accuracy of 91%. The weights used for each model is as follows:

Forecasting Algorithms:

LSTM:	80%
ARIMA:	20%

Trend Analysis Algorithms (Trend Score Calculation):

FB-Prophet Algorithm:	45%
Pair-Wise Trading:	30%
Momentum Trading:	25%

Final Weights (Bagging):

Forecasting Algorithms:	55%
Trend Analysis:	35%
RL Agent:	10%

According to the ensemble model, the top performing stocks are: SHREECEM and BAJAJFINANCE and these stocks generated the highest returns as well over the period of 2010 to 2020 which is evident in figure 14a and figure 14b. The worst performer according to our model came out to be ONGC and the net profit generated by this stock in our portfolio was only 5% of the total profit earned. This is also evident by looking at the OHLC data of ONGC graphically represented in figure 14c.

In figure 14a and 14b, the trend of SHREECEM and BAJAJFINANCE is an upward slope so buying at an early stage and selling at the end stage was the most profitable decision. The ensemble model generated more buy signals in the years 2010 – 2013 and generated more sell signals in the period 2018-2020 for both of these stocks. Conversely, for ONGC (Figure 14c) the period between 2015-2017 had more sell signals which prompted the trading model to go into negative portfolio returns on ONGC.

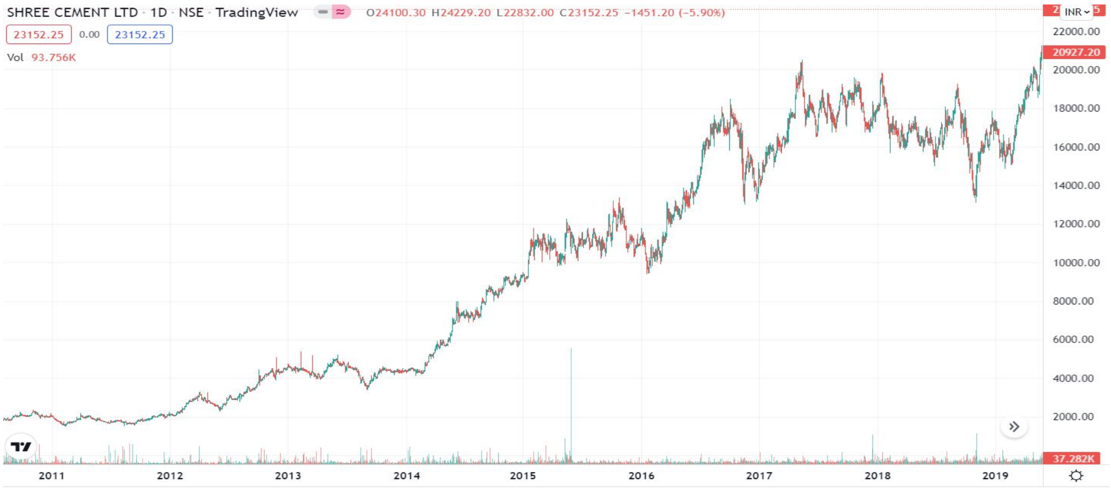


Figure 14a: SHREECEM OHLC History (TradingView)

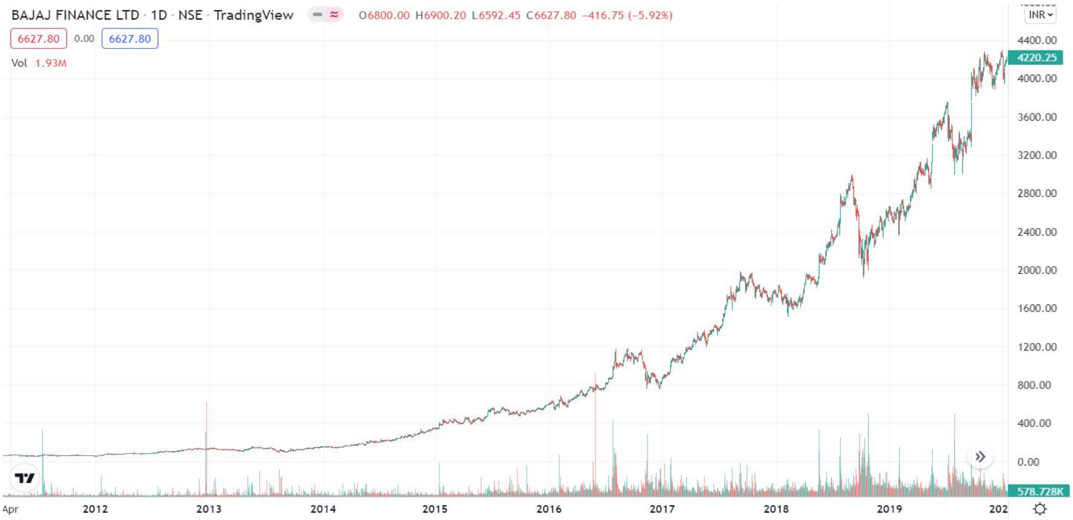


Figure 14b: BAJAJFINANCE OHLC History (TradingView)

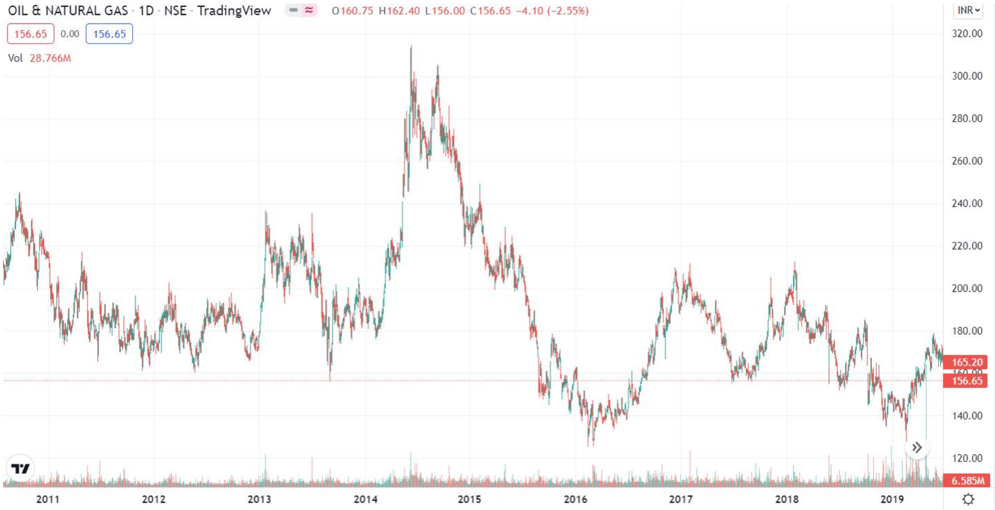


Figure 14c: ONGC OHLC History (TradingView)

CONCLUSION AND FUTURE WORK

This paper analyses and ensembles multiple trading strategies and tries to bridge the differences between fundamental and quantitative trading using statistical and deep learning models. Statistical Models include Momentum Trading, Pair-wise Trading, Facebook's Prophet Algorithm and ARIMA whereas Deep Learning Models used were Stacked Attention based LSTMs and a Q-Learning based RL trading agent. Since each and every model mentioned here has its own set of pros and cons, using a model which tries to handle all of these different approaches and give a cumulative singular result, drastically reduces an investor's uncertainty about which models to follow and when one model should be preferred over the other. This approach also increases the total accuracy of the trade price forecasting system and aids the investors to make their informed decision. The higher accuracy results in the proposed model, LSTM and Modified Prophet Algorithm further bolsters this statement. Moreover, this research also shows how traditional strategies like momentum based trading and pair wise trading are inferior to the current algorithms used to predict market fluctuations and also how the deep learning algorithms are outperforming statistical models.

The results from these models can further be improved by using more complex LSTMs and also training a better RL agent for more number of episodes which due to computation limitations could not be done in this research. Also, incorporating Natural Language Processing for analysing news, government policies and company financial reports and Computer Vision Algorithms to scope out potential growth of a sector (For example, using satellite images to find growing cities helps in predicting a growth within the construction and housing sector of that region) and graph reading would give even robust results as they include more information about a stock than a simple OHLS dataset.

REFERENCES

Alazab, M., Khan, S., Krishnan, S. S. R., Pham, Q.-V., Reddy, M. P. K., & Gadekallu, T. R. 2020. A multidirectional LSTM model for predicting the stability of a smart grid. IEEE Access, 8, 85454-85463.

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. 2014. Stock price prediction using the ARIMA model. Paper presented at the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation.

AS, S. 2013. A study on fundamental and technical analysis. International Journal of Marketing, Financial Services & Management Research, 2(5), 44-59.

Ashfaq, N., Nawaz, Z., & Ilyas, M. 2021. A comparative study of Different Machine Learning Regressors For Stock Market Prediction. arXiv preprint arXiv:2104.07469.

Basha, S. M., & Rajput, D. S. 2018. A supervised aspect level sentiment model to predict overall sentiment on tweeter documents. International Journal of Metadata, Semantics and Ontologies, 13(1), 33-41.

Basha, S. M., & Rajput, D. S. 2019. Survey on Evaluating the Performance of Machine Learning Algorithms: Past Contributions and Future Roadmap Deep Learning and Parallel Computing Environment for Bioengineering Systems (pp. 153-164): Elsevier.

Basha, S. M., Rajput, D. S., & Vandhan, V. 2018. Impact of gradient ascent and boosting algorithm in classification. International Journal of Intelligent Engineering and Systems 11 (1), 41-49

Das, B., & Chakrabarty, D. 2016. Newton's forward interpolation: representation of numerical data by a polynomial curve. International Journal of Statistics and Applied Mathematics, 1(2), 36-41.

Diamond, P. 1990. Pairwise credit in search equilibrium. The Quarterly Journal of Economics, 105(2), 285-319.

Foltice, B., & Langer, T. 2015. Profitable momentum trading strategies for individual investors. Financial Markets and Portfolio Management, 29(2), 85-113.

Idrees, S. M., Alam, M. A., & Agarwal, P. 2019. A prediction approach for stock market volatility based on time series data. IEEE Access, 7, 17287-17298.

Li, Y., Zheng, W., & Zheng, Z. 2019. Deep robust reinforcement learning for practical algorithmic trading. IEEE Access, 7, 108014-108022.

Pahwa, N., Khalfay, N., Soni, V., & Vora, D. 2017. Stock prediction using machine learning a review paper. International Journal of Computer Applications, 163(5), 36-43.

Patel, H., Rajput, D. S., Stan, O. P., & Miclea, L. C. 2022. A New Fuzzy Adaptive Algorithm to Classify Imbalanced Data. Computers, Materials \& Continua, 70(1), 73--89.

Patel, H., & Rajput, D. 2011. Data mining applications in present scenario: a review. International Journal of Soft Computing, 6(4), 136-142.

Patel, H., Singh Rajput, D., Thippa Reddy, G., Iwendi, C., Kashif Bashir, A., & Jo, O. 2020. A review on classification of imbalanced data for wireless sensor networks. International Journal of Distributed Sensor Networks, 16(4), 1550147720916404.

Patel, H., & Thakur, G. S. 2019. An improved fuzzy k-nearest neighbor algorithm for imbalanced data using adaptive approach. IETE Journal of Research, 65(6), 780-789.

Patel, H., & Thakur, G. S. 2017. Improved fuzzy-optimally weighted nearest neighbor strategy to classify imbalanced data. International Journal of Intelligent Engineering and Systems, 10(2), 156-162.

Ray, R., Khandelwal, P., & Baranidharan, B. 2018. A survey on stock market prediction using artificial intelligence techniques. Paper presented at the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT).

Roy, S. G. (2015). Equity research: Fundamental and technical analysis. International Journal of Science and Research, 4(9), 272-275.

Taylor, S. J., & Letham, B. 2018. Forecasting at scale. The American Statistician, 72(1), 37-45.

Tripathy, B., Parimala, M., & Reddy, G. T. 2021. Innovative classification, regression model for predicting various diseases Data Analytics in Biomedical Engineering and Healthcare (pp. 179-203): Elsevier.

Ullah, A., Imtiaz, F., Ihsan, M. U. M., Alam, M., Rabiul, G., & Majumdar, M. 2021. Combining machine learning classifiers for stock trading with effective feature extraction. arXiv preprint arXiv:2107.13148.

Varsamopoulos, S., Bertels, K., & Almudever, C. G. 2019. Comparing neural network based decoders for the surface code. IEEE Transactions on Computers, 69(2), 300-311.

Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. Available at SSRN.

Zou, Z., & Qu, Z. 2020. Using LSTM in Stock prediction and Quantitative Trading. CS230: Deep Learning, Winter.