Algorithmic Trading of Financial Instruments

Dual Degree Project - Phase II

submitted in partial fulfillment
of degree in

Bachelors of Technology & Masters of Technology (Dual Degree)

By

Saurabh Kumar

Roll No. 16d100018

Under the guidance of

Prof. Alankar Alankar

Indian Institute of Technology Bombay

June 2021

Contents

List of Figures

List of Tables

1 Abstract

Making profit out of stock-market is a hectic and time-consuming process. In this work we have put our effort to automate the process of trading based on data analysis. Strategies are thoroughly back-tested to reduce the possibility of significant drawdown of the capital invested. ARIMA model has been used for forecasting future instance data, and a novel trading strategy utilising ARIMA(5,1,0) has been framed and tested out on reliance industries stock, capable of generating twice as much return than possible with passive investing in relative term and 29.69 % p.a. in absolute term for the analysis period. Framework for placing real-time orders from python scripting environment has been built using smartAPI provided by Angel Broking.

2 Introduction

Algorithmic trading is a new and growing field, used mainly by proprietary trading firms. Algorithmic trading refers to a set of principles where exchange takes place based on algorithms. By exchange, we mean the action of either buying or selling. Algorithmic trading in financial instruments imply letting algorithm exchange financial instruments to realize certain goal, as decided by the people who builds them. In general the goal is to make profit, which is realized by buying low and selling high on an average. Before diving into detail on algorithms that are popularly used to achieve the goal of making profit, let's first get a brief idea on financial instruments that algorithm can exchange.

Financial instruments refer to items that are actively traded online on stock exchanges, for example: NSE(National Stock Exchange) & BSE(Bombay Stock Exchange for India). Instnaces of financial instruments include:

1. Equities like RELIANCE, TCS, HDFCBANK, HINDUNILVR, etc
2. Commodities like gold and silver
3. Bonds(government & corporate) and debts
4. Derivatives like futures and options

Other options include managed funds like mutual funds, index funds, ETFs(Exchange Traded Funds), etc. For our analysis in our project we have focussed on equity segment only, particularly reliance stock.

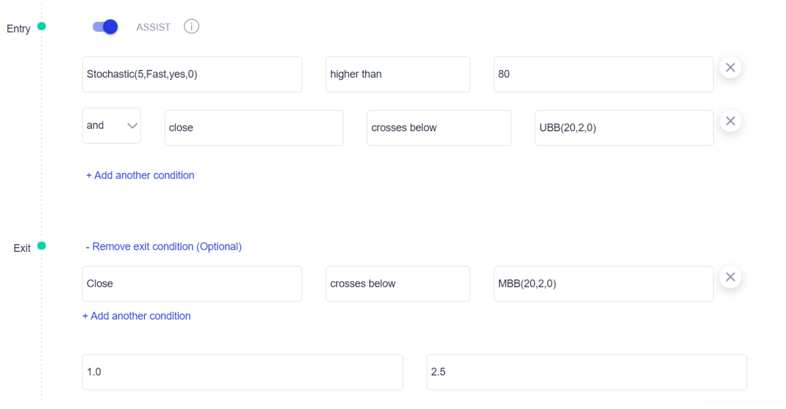
Coming on the algorithm part, popular strategies can be classified under broadly two categories:

1. **Statistics based**
   Statistics based strategies include mean-reversion, momentum based strategies utilising MACD, bollinger bands, different kinds of candle engulfing patterns, etc. These strategies are developed based on past data analysis and observing repeatable patterns where favourable trades generally tend to shape. Applying these strategies does not

require user to perform any kind of model training, these strategies could be hard coded and deployed to reap returns. Let's see condition of stochastic mean-reversion strategy for example:

- The mean reversion short strategy uses Stochastic and upper Bollinger band to gauge overselling in the market.
- Entry condition: BUY 100 shares when Close crosses below MBB(20,2,0) or at stop loss of 1% or target profit of 2.5% at 1 hour candle interval using candlestick chart.
- Exit condition: SELL 100 shares when Stochastic(5,Fast,yes,0) higher than 80 and close crosses below UBB(20,2,0) at 1 hour candle interval using candlestick chart. Enter trade between 09:00 to 23:59

Sample instance of above conditions in Streak platform is attached below:



Entry
Stochastic(5,Fast,yes,0) higher than 80
and close crosses below UBB(20,2,0)

Exit
Close crosses below MBB(20,2,0)

1.0 2.5

Figure 1: Stochastic Mean Reversion Strategy instance in Streak

UBB and MBB refers to Upper Bollinger Band and Middle Bollinger Band respectively.

2. **Machine Learning and Deep Learning based**
   ML & DL based strategies include time-series forecasting based models like ARIMA, neural networks model, reinforcement learning models, etc.

As part of this project, we have focussed on time-series based forecasting model, particularly ARIMA, identified optimal parameters p,d q and built a trading strategy utilising forecasting from the ARIMA model.

3 Literature Review

In this chapter, we have reviewed relevant papers where researchers have used the techniques of data-analysis to solve problems/situations in the domain of stocks trading.

3.1 Application of Box – Jenkins ARIMA (p, d, q) Model for Stock Price Forecasting and Detect Trend of S&P BSE Stock Index: An Evidence from Bombay Stock Exchange

In this paper, forecasting analysis has been performed on S&P BSE using ARIMA model. Trend analysis was conducted to find out the nature of time-series data before forecasting. Shape of the time series data was determined using skewness and kurtosis test. Stationarity of the data was checked using ADF(Augmented Dickey Fuller) & PP(Phillips – Perron) unit root test. AIC(Akaike's Information Criteria) & BIC(Bayesian Information Criteria) was utilised to check the performance of the model. MAPE(Mean Absolute Percentage Error) was used to test the forecast accuracy. [1]

3.2 Arima model in forecasting share prices

In this paper, time-series analysis using ARIMA model has been performed on past two year stock prices data of Star Cement Ltd. ADF test has been used to check for data stationarity. ACF(Autocorrelation function) and PACF(Partial Autocorrelation Function) has been used in conjunction to decide for the possible set of values of q(MA(q)) and p(AR(p)) respectively, once data is ensured to be stationary using ADF test. AIC(Alkaike's Information Criterion) has been used to shortlist the best parameter. Residuals from prediction made using the trained ARIMA model is shown to have mean value close to zero and near normal distribution around the mean. [2]

3.3 Forecasting stock market prices using mixed ARIMA model: a case study of Indian pharmaceutical companies

In this paper, time-series analysis using ARIMA model on time-series stock price data of pharmaceutical firms in NIFTY100 namely Sun Pharmaceutical Industries, Lupin Ltd and

Dr. Reddy Laboratories has been performed. Parameterd has been estimated using ADF test, and then PACF and ACF plot has been plotted on the 1st order differenced price to estimate the parameters p and q, respectively. AIC(Alkaike's Information Criterion), Schwartz Criterion, Volatility(SIGMASQ), R-sqaured abd adjsuted R-squared values have been considered to shortlist the better performing models. Authors claim that even the R-squared(almost always less than 0.65) and adjusted R-squared(less than 0.55) values are less than desired, the error between predicted values and actual values are less and the model can actually be used to make short term profits for the studied pharmaceutical companies in India. [3]

3.4 Stock market analysis and prediction using time series analysis

In this paper, time-series daily stock price data of Reliance, HDFC, TCS, INFY and NIFTY index scrips have been considered. Their values with time has been plotted. Daily return defined as the value gained or lost by stock to the previous day has been analyzed to have a mean close to zero. Variance of daily return has been used to conclude that Reliance stocks was most volatile while HDFCBANK was least volatile. Box-plot has been used to conclude that INFY has a wider graph due to some outliers. Different categories of risk-adjusted return given by Sharpe ratio has been defined. Auto ARIMA model has been used to come up with the (p,d,q) value of (0,1,1), shortlisted by minimizing AIC(Alkaike's Information Criterion) value. MSE, MAPE, RMSE, MAPE value has been calculated for ARIMA(0,1,1). [4]

3.5 Stock Market Forecasting Technique using Arima Model

In this paper, key components of time-series data have been explained. Various tools and techniques used to analyze time-series data like ARIMA model, neural networks, exponential smoothing, simple moving average, etc have been briefly described. Decpomposition of data into seasonality(patterns which are repeated in particular seasons of year), trend(can be upward trend, stationary or downward) and cyclical(trend with no set repetition) part has been displayed and analyzed. Implementation of ARIMA model on NIFTY daily of NIFTY50 index have been performed. Predicted price using the ARIMA model has been very close to the actual values for the respective instances. .[5]

3.6 Using Reinforcement Learning in the Algorithmic Trading Problem

In this paper, authors have used the concept of reinforcement learning, and treated stock market as a game with Markov property comprising of states, actions & rewards. A trading system for exchanging fixed quantity of scrips is proposed using neural network architectures. Profitability of 110 %p.a. without accounting for transactional & brokerage charges and 66 % post accounting the charges has been achieved by the best performing model on RTX Index futures. [6]

3.7 An Application of Deep Reinforcement Learning to Algorithmic Trading

This paper discusses a deep reinforcement learning(DRL) based model to tackle algorithmic trading problem. To maximize the sharpe ratio(risk-adjusted return), a novel DRL trading strategy is proposed. Inspired from the DQN(Deep Q-Network) algorithm, the author proposed a TDQN(Trading DQN) algorithm specifically adapted to algorithmic trading arena. Model is trained based on generation of artificial trajectories from historical data of stock prices. [7]

3.8 Financial Trading as a Game: A Deep Reinforcement Learning Approach

In this paper, a MDP(Markov Decision Process) based model utilising the concepts of DRQN(Deep Recurrent Q-network) is proposed. Significantly small replay memory of the size of only a few hundreds were employed, compared to those millions in size used in modern DRL(Deep Reinforcement Learning). Utilised action augmented technique to incorporate greedy policy for model learning, resulting in improved empirical performance compared to epsilon greedy exploration. Reduced training time by a factor of T by sampling longer sequence for RNN(Recurrent Neural Network) training. Finally validated the model on spot foreign exchange market. [8]

3.9 Stock trading with cycles: A financial application of ANFIS and reinforcement learning

This paper utilises the concept of ANFIS(Adaptive Network Fuzzy Inference System). The model determines change in market trend, and dynamically suggests the periods for MA(moving average) and momentum by appropriately shifting ANFIS-RL cycle using RL frameworks. This helps find the suitable points to go long and short, thus helping user ride the waves of price cycle. The framework proposed beats the market by around 50 percentage points in 13 years analysis period. [9]

3.10 Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation

This paper utilises the concept of ANN(Artificial Neural Network) for prediction and decision making process. The paper analyzes data of three most traded currency pairs namely: GBP\USD, EUR\GBP, and EUR\USD. The tests confirmed that daily FOREX currency rates time-series data are not randomly distributed with a statistical significance of greater than 95 %. The implemented model reaches an prediction accuracy of 72.5 % and Annualized Net Return of 23.3 %. [10]

3.11 Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy

In this paper, a novel DRL(Deep Reinforcement Learning) based trading strategy is proposed. Different versions of virtual trading agents trading against each other has been implemented. Uses the concept of proximal policy optimization to improve decision making process. Risk-curiosity driven training acts as a reward function to learn relationships between actions & market-behaviours. The model has been tested against eight different stocks and the model delivers promising performances, and generally undertook less trades but with higher profit margins. [11]

3.12 Financial time series forecasting with deep learning : A systematic literature review: 2005–2019

This is a review paper on DL(deep learning) based time-series forecasting in financial domain. Extensive information on DBNs(Deep Belief Networks), CNNs(Convolutional Neural Networks), LSTM(Long-Short Term Memory) are provided from various papers. Possible opportunities and setbacks are highlighted to envision the future of the field. [12]

3.13 Deep Reinforcement Learning for Trading

In this paper, DRL(Deep Reinforcement Learning) based algorithms for eechanging continuous future contracts are considered. Both continuous and discrete action spaces are considered. They tested model performance on 50 future contracts and tested how they perform across different classes of equity indices, fixed income, foreign exchange markets and commodities. Finally they tested the model against classical time-series based momentum strategies and concluded that the model outperforms such baseline models despite heavy transactional costs. [13]

3.14 Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis

The paper proposes crypto trading model based on data extracted from twitter platform. Several supervised machine learning algorithms like logistic regression, Naive Bayes, SVM(Support Vector Machine), etc were incorporated to identify market movement of cryptocurrency. Error analysis were performed to ensure accurate inputs are utilised at each and every step of the model. The model achieves predction accuracy exceeding 90 % [14]

3.15 Sentiment and Knowledge Based Algorithmic Trading with Deep Reinforcement Learning

This paper utilises the concept of time-series analysis, sentimnet analysis and knowledge graphs. The authors combine time series stock price data with news headline sentiments using RL(Reinforcement Learning). The model parallely exploits knowledge graphs to learn news about implicit relationships. [15]

3.16 An Artificial Neural Network-based Stock Trading System Using Technical Analysis and Big Data Framework

In this paper MLP(Multi Layer Perceptron) based model is trained using technical indicators. FUsing technical indicators, time-series data is first converted into a series of buy, sell or hold signals and then a MLP model is trained on daily stock prices data for Dow30 stocks, ranging from year 1997 to 2007. Fine-tuning the technical indicators to be used helped improve performance of the model, and in turn the bottom line profits generated by the trading model. [16]

3.17 An Algorithmic Trading Agent based on a Neural Network Ensemble: a Case of Study in North American and Brazilian Stock Markets

In this paper, authors have tried and tested out a classification model instead of regression model commonly used as part of time-series analysis, where stock prices for future instances are forecasted based on previous values observed. Here, a neural network is trained just to predict whether the stock price is going to rise or fall, and the model was able to realize profit on both the datasets used by the author. [17]

3.18 Deep Robust Reinforcement Learning for Practical Algorithmic Trading

The paper proposes a DRL(Deep Reinforcement Kearning) based model. The authors utilised the concept of asynchronous advantage actor-critic(A3C) and extended the value based DQN(Deep Q-network) to improve the performance of the model. They further utilized LSTM(Long Short Term memory) and SDAEs(Stacked Denoising Autoencoders) frameworks to extract robust market representations and resolve financial time-series dependence. [18]

3.19 Advanced Markov-Based Machine Learning Framework for Making Adaptive Trading System

This paper proposes an implementation of ad-hoc machine learning algorithm for stock prices prediction based on historical data analysis. Forecasting accuracy was improved by using the concepts of Markov Stochastic Property and adaptive control. Authors confirm the effectiveness and robustness of the developed model by validating the results against various finacial instruments. [19]

3.20 Adversarial Attacks Against Reinforcement Learning-Based Portfolio Management Strategy

In this paper, an adversarial Reinforcement Learning framework is used. The authors proposed an extension of EIIE(Ensemble of the Identical Independent Evaluators) method. Improved portfolio performance was demonstrated to be achieved by enhanced EIIE compared to regular EIIE agent. Enhanced EIIE was then utilised by adversarial agent to learn how much and when to attack. Authors proposed that the developed model was 30 % more effective than the conventional attack mechanism. [20]

4 Objective

In this chapter, we have listed down the set of objectives that we have planned and achieved as part of phase-II of the project.

4.1 Data Source

Identify sources from which we can retrieve relevant financial instrument data to train and test our data driven model.

4.2 Back-testing

Test the performance of our strategies and models on past data on aspects like returns that could have been realized and maximum drawdown of capital invested.

4.3 Trading simulation

Simulated trading is commonly referred as ”paper trading” in the world of financial instruments trading. Here, we wish to measure the returns in real-time, without putting the actual capital at stake.

4.4 Risk Management

Explore appropriate guards which can be put in place to safeguard our invested capital and not let it get blown away by some unusual/unexpected events or circumstances.

4.5 Live Trading

Here, our objective will be to build a framework using which we can execute orders in real-time based on buy-sell suggestions from the algorithm.

5 Methodology

To achieve the above set of objectives, following have been explored, tried and/or tested out.

5.1 Data Source

We need financial instrument data to train, test and then deploy them to execute favourable trades. Following data sources have been explored:

5.1.1 Kite Connect 3 API

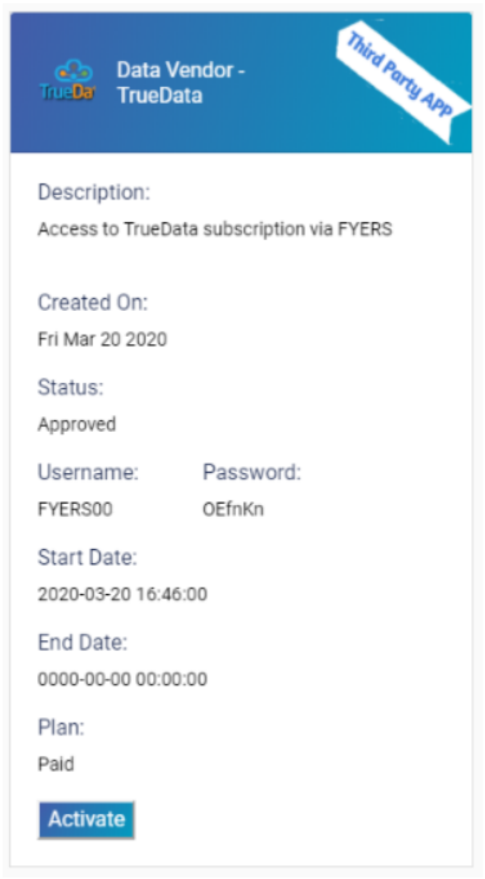
Kite Connect API(Application Programming Interface) lets user import data, both historical as well as real-time in scripting language. It also lets user place orders from the same environment if they wish to. Programming languages supported by the API are python, go, java, php, node js, .net, rust and R.
The parameters to be provided to retrieve the data are the scrip symbol(instrument_token), duration of data(from and to parameter) and the frequency, i.e. interval of data. The data for the asked period is returned in candle format including information on opening price(open), highest price during the interval(high), lowest price during the interval(low), closing price(close), number of transactions(volume) along with timestamp.

Kite Connect 3 Historical Candle data documentation Cost of using the service is ₹2000, recurring monthly.

5.1.2 TrueData API

Fyers in collaboration with TrueData provide both real-time data as well as historical data. This integrated API can be used with any programming language for building and executing trading strategies using the real-time data feed and also for the automatic placement of orders.

Fyers-TrueData collaborative Market Data API URI and Request parameters is similar to 5.1.1 Kite Connect 3 API, with small changes here and there. Cost of using the service is ₹1599 + GST(18%), recurring monthly



Data Vendor - TrueData

Description:
Access to TrueData subscription via FYERS

Created On:
Fri Mar 20 2020

Status:
Approved

Username: FYERS00
Password: OEfnKn

Start Date:
2020-03-20 16:46:00

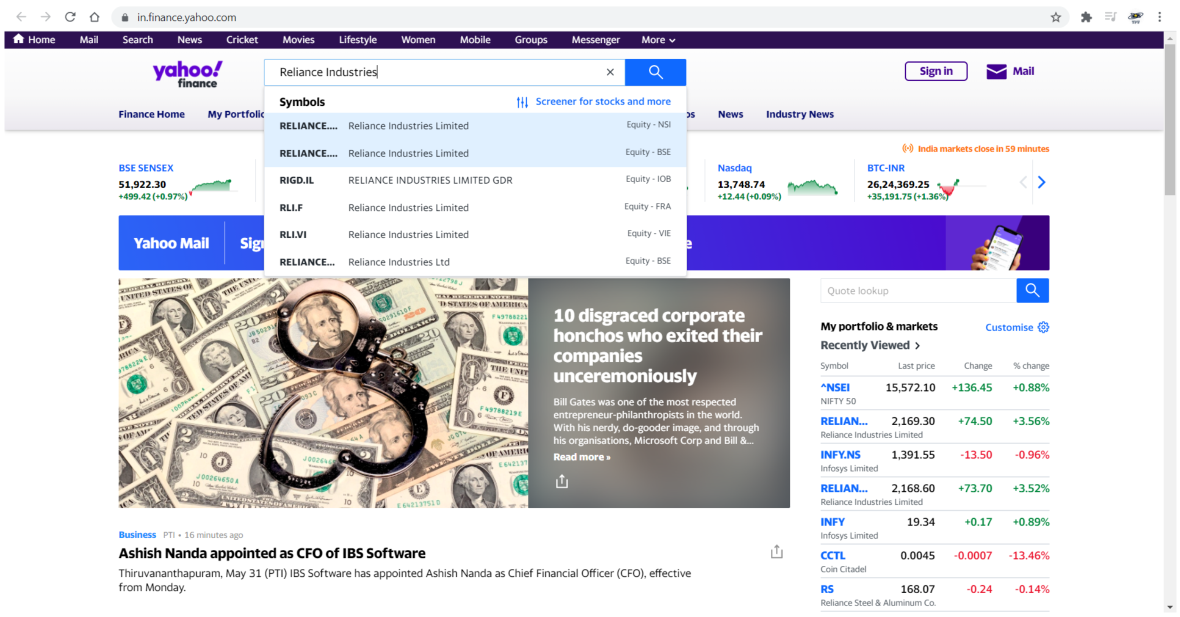
End Date:
0000-00-00 00:00:00

Plan:
Paid

Figure 2: TrueData app creation for API access

5.1.3 Yahoo Finance

Yahoo Finance provides free interface to view and download historical data. To get relevant data, just search for instrument in the search bar provided: Yahoo Finance



Reliance Industries

Symbols — Screener for stocks and more

RELIANCE....	Reliance Industries Limited	Equity - NSI
RELIANCE....	Reliance Industries Limited	Equity - BSE
RIGD.IL	RELIANCE INDUSTRIES LIMITED GDR	Equity - IOB
RLI.F	Reliance Industries Limited	Equity - FRA
RLI.VI	Reliance Industries Limited	Equity - VIE
RELIANCE...	Reliance Industries Ltd	Equity - BSE

India markets close in 59 minutes

BSE SENSEX 51,922.30 +499.42 (+0.97%)
Nasdaq 13,748.74 +12.44 (+0.09%)
BTC-INR 26,24,369.25 +35,191.75 (+1.36%)

10 disgraced corporate honchos who exited their companies unceremoniously

Bill Gates was one of the most respected entrepreneur-philanthropists in the world. With his nerdy, do-gooder image, and through his organisations, Microsoft Corp and Bill &... Read more »

Business PTI • 16 minutes ago

Ashish Nanda appointed as CFO of IBS Software

Thiruvananthapuram, May 31 (PTI) IBS Software has appointed Ashish Nanda as Chief Financial Officer (CFO), effective from Monday.

My portfolio & markets — Customise

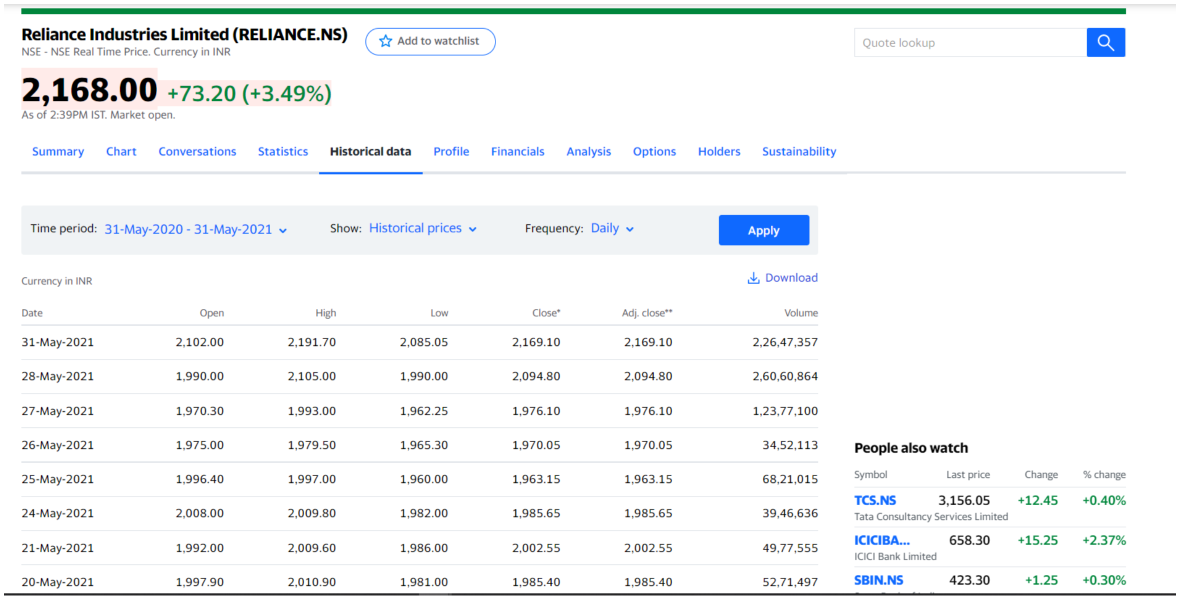
Recently Viewed

Symbol	Last price	Change	% change
^NSEI NIFTY 50	15,572.10	+136.45	+0.88%
RELIAN... Reliance Industries Limited	2,169.30	+74.50	+3.56%
INFY.NS Infosys Limited	1,391.55	-13.50	-0.96%
RELIAN... Reliance Industries Limited	2,168.60	+73.70	+3.52%
INFY Infosys Limited	19.34	+0.17	+0.89%
CCTL Coin Citadel	0.0045	-0.0007	-13.46%
RS Reliance Steel & Aluminum Co.	168.07	-0.24	-0.14%

Figure 3: Yahoo Finance search instrument

Select the exchange, either "BSE" or "NSE", based on whose exchabge data we want to retrieve.

This provides us with summary of the instrument on the selected exchange. Navigate to "Historical data" tab and select the "Time period" and "frequency" as per the requirement.



Reliance Industries Limited (RELIANCE.NS)
NSE - NSE Real Time Price. Currency in INR

2,168.00 +73.20 (+3.49%)
As of 2:39PM IST. Market open.

Time period: 31-May-2020 - 31-May-2021 Show: Historical prices Frequency: Daily

Currency in INR

Date	Open	High	Low	Close*	Adj. close**	Volume
31-May-2021	2,102.00	2,191.70	2,085.05	2,169.10	2,169.10	2,26,47,357
28-May-2021	1,990.00	2,105.00	1,990.00	2,094.80	2,094.80	2,60,60,864
27-May-2021	1,970.30	1,993.00	1,962.25	1,976.10	1,976.10	1,23,77,100
26-May-2021	1,975.00	1,979.50	1,965.30	1,970.05	1,970.05	34,52,113
25-May-2021	1,996.40	1,997.00	1,960.00	1,963.15	1,963.15	68,21,015
24-May-2021	2,008.00	2,009.80	1,982.00	1,985.65	1,985.65	39,46,636
21-May-2021	1,992.00	2,009.60	1,986.00	2,002.55	2,002.55	49,77,555
20-May-2021	1,997.90	2,010.90	1,981.00	1,985.40	1,985.40	52,71,497

People also watch

Symbol	Last price	Change	% change
TCS.NS Tata Consultancy Services Limited	3,156.05	+12.45	+0.40%
ICICIBA... ICICI Bank Limited	658.30	+15.25	+2.37%
SBIN.NS	423.30	+1.25	+0.30%

Figure 4: Reliance Industries past 1 year daily data

Reliance Historical Data This interface let's user download historical data in .csv format for free of any listed instrument starting from 1st Jan 1996 till date with frequency option of daily, monthly or weekly.

5.1.4 SmartAPI

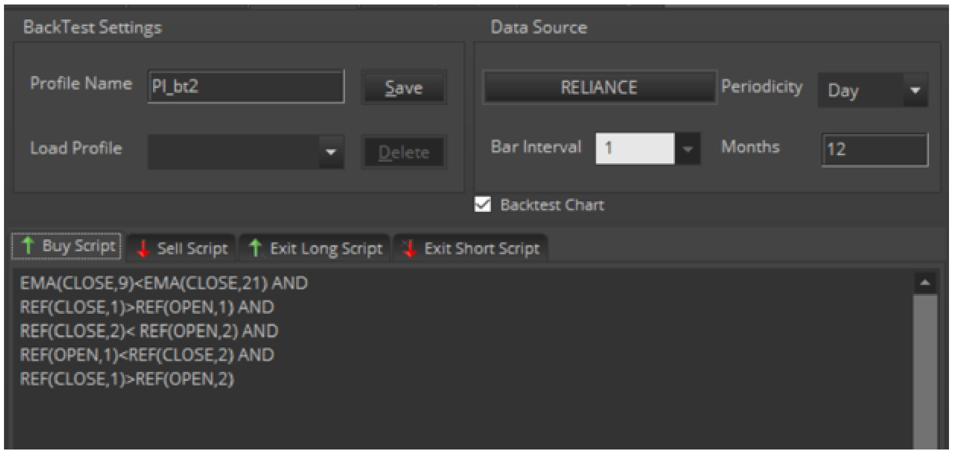
This API is being provided by Angel Broking and lets user access stock-exchange data in real-time, the best thing about this api is that the service is currently free of cost. This helped us build our live trading framework, using which we can place orders to exchange from scripting environment. More details about this has been mentioned on "Data Model" section.

5.2 Back-testing

Backtesting refers to a process where user checks the performance of his/her strategy on past data of financial instruments. Although, there is almost no guarantee that we can expect similar trends/behaviour in future but this does provide a quantitative framework to quantify the returns that would have been generated had this strategy been put during that(past) time. We have checked various platforms like PI, Streak, etc which lets backtest trading startegies with their own designed dashboards, where users are just supposed to enter the strategy and financial instrument on which the strategy should be tested on, along with the time interval and data frequency.

5.2.1 PI

Sample snapshot of PI dashboard is attached below:



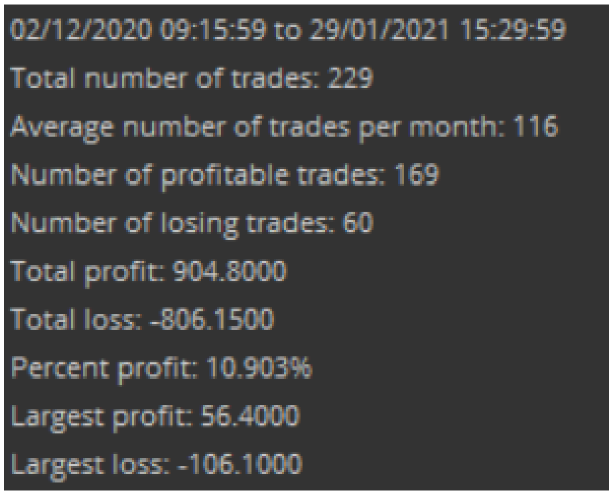
Profile Name: PI_bt2
Load Profile:
RELIANCE
Periodicity: Day
Bar Interval: 1
Months: 12
✓ Backtest Chart

```
EMA(CLOSE,9)<EMA(CLOSE,21) AND
REF(CLOSE,1)>REF(OPEN,1) AND
REF(CLOSE,2)< REF(OPEN,2) AND
REF(OPEN,1)<REF(CLOSE,2) AND
REF(CLOSE,1)>REF(OPEN,2)
```

Figure 5: PI backtest interface

Here, we have the interval of selecting scrib(currently selected RELIANCE), periodicity(Day: frequency with which favourable condition for the strategy should be tested), Bar Interval(1: 1 candle for daily data) and Months(12: last 12 months data). Along with this, the most important aspect we have the option of writing trading strategy, buy and sell condition.

Sample instance of the backtest result provided by the interface is attached below:



```
02/12/2020 09:15:59 to 29/01/2021 15:29:59
Total number of trades: 229
Average number of trades per month: 116
Number of profitable trades: 169
Number of losing trades: 60
Total profit: 904.8000
Total loss: -806.1500
Percent profit: 10.903%
Largest profit: 56.4000
Largest loss: -106.1000
```

Figure 6: PI backtest summary

Here, we have details on profit and number of trades executed along with breakdown of trades making profit and trades making loss out of the total executed.

Below is an attached snapshot of buy and sell points for each of the 229 trades.

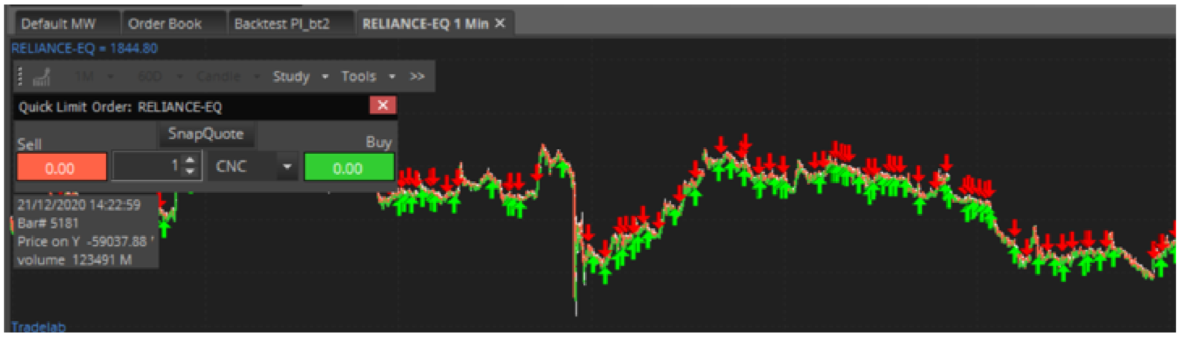
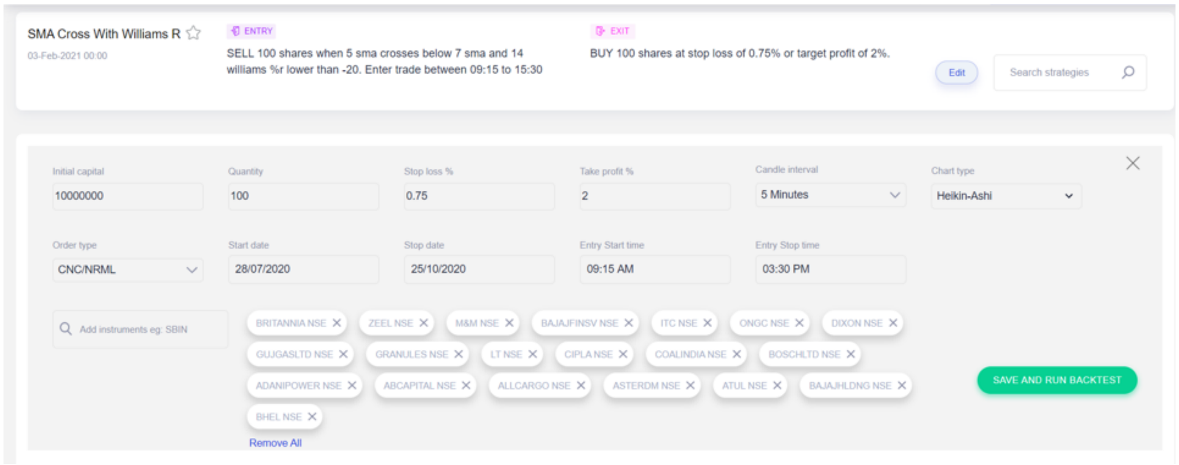


Figure 7: PI backtest buy-sell points

Red downward arrow represents sell signal and green upward arrow represents a buy signal. Link of the interface: PI

5.2.2 Streak

Streak is another dashboard based platform where we have the option of backtesting trading strategies. This platform let's user write trading strategies and see how they perform on chosen financial instruments. Sample snapshot of the interface is attached below:


SMA Cross With Williams R

03-Feb-2021 00:00

ENTRY

SELL 100 shares when 5 sma crosses below 7 sma and 14 williams %r lower than -20. Enter trade between 09:15 to 15:30

EXIT

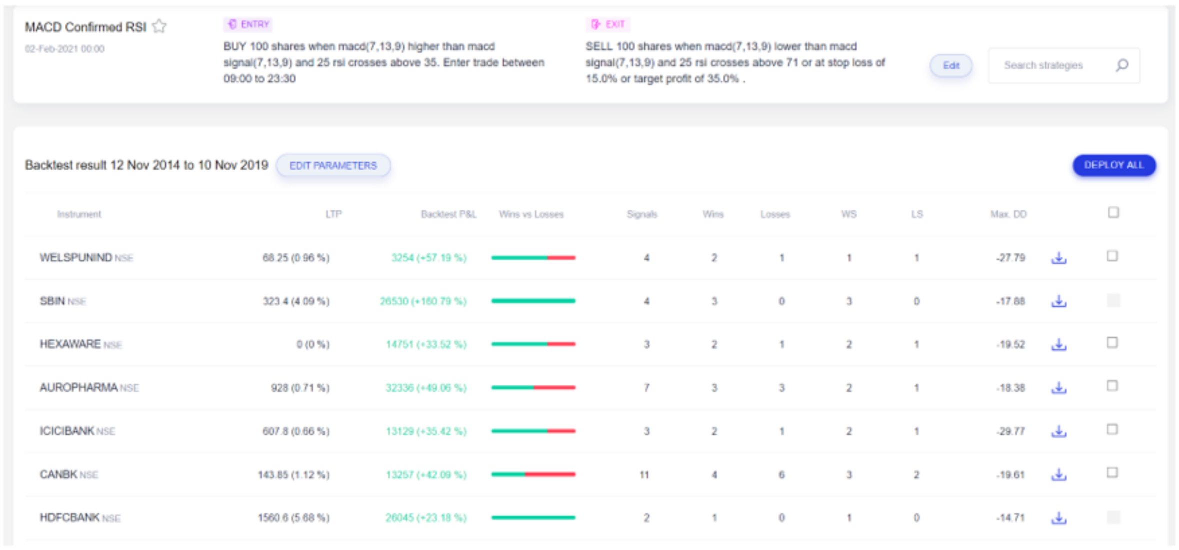
BUY 100 shares at stop loss of 0.75% or target profit of 2%.

Initial capital	Quantity	Stop loss %	Take profit %	Candle interval	Chart type
10000000	100	0.75	2	5 Minutes	Heikin-Ashi
Order type	Start date	Stop date	Entry Start time	Entry Stop time	
CNC/NRML	28/07/2020	25/10/2020	09:15 AM	03:30 PM	

BRITANNIA NSE, ZEEL NSE, M&M NSE, BAJAJFINSV NSE, ITC NSE, ONGC NSE, DIXON NSE, GUJGASLTD NSE, GRANULES NSE, LT NSE, CIPLA NSE, COALINDIA NSE, BOSCHLTD NSE, ADANIPOWER NSE, ABCAPITAL NSE, ALLCARGO NSE, ASTERDM NSE, ATUL NSE, BAJAJHLDNG NSE, BHEL NSE

Figure 8: Streak backtest interface

It lets user select the scrib, time-interval of testing, candle interval, stop-loss, initial-capital and number of instruments to be exchanged if favourable condition occurs.



MACD Confirmed RSI

02-Feb-2021 00:00

BUY 100 shares when macd(7,13,9) higher than macd signal(7,13,9) and 25 rsi crosses above 35. Enter trade between 09:00 to 23:30

SELL 100 shares when macd(7,13,9) lower than macd signal(7,13,9) and 25 rsi crosses above 71 or at stop loss of 15.0% or target profit of 35.0% .

Backtest result 12 Nov 2014 to 10 Nov 2019

Instrument	LTP	Backtest P&L	Signals	Wins	Losses	WS	LS	Max. DD
WELSPUNIND NSE	68.25 (0.96 %)	3254 (+57.19 %)	4	2	1	1	1	-27.79
SBIN NSE	323.4 (4.09 %)	26530 (+160.79 %)	4	3	0	3	0	-17.88
HEXAWARE NSE	0 (0 %)	14751 (+33.52 %)	3	2	1	2	1	-19.52
AUROPHARMA NSE	928 (0.71 %)	32336 (+49.06 %)	7	3	3	2	1	-18.38
ICICIBANK NSE	607.8 (0.66 %)	13129 (+35.42 %)	3	2	1	2	1	-29.77
CANBK NSE	143.85 (1.12 %)	13257 (+42.09 %)	11	4	6	3	2	-19.61
HDFCBANK NSE	1560.6 (5.68 %)	26045 (+23.18 %)	2	1	0	1	0	-14.71

Figure 9: Streak backtest summary

Below is an attached summary of backtest result provided by the interface:

The result include details like profit/loss made by the strategy(Baktest P&L), number of winnning trades(Wins) and number of losing trades(Losses) out of the total number of trades executed(Signals). It also includes details like winning streak(WS) and loosing streak(LS) along with maximum drawdown(Max. DD). Maximum drawdawn refers to the maximum downturn in capital value, in terms of percentage during the time-interval had the strategy been put in place.

5.2.3 From scratch in python

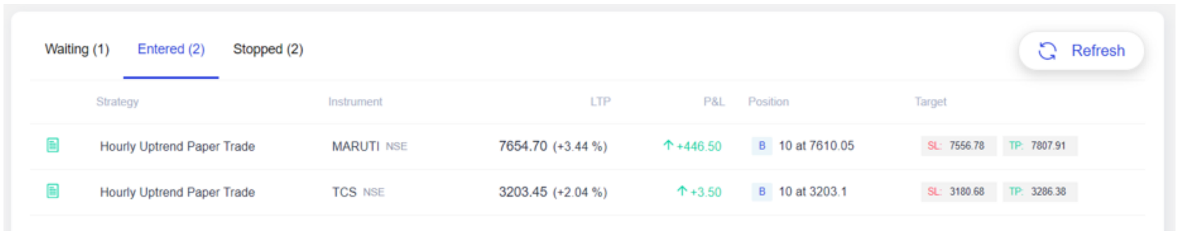
This is by far the best method for users who are comfortable writing scripts, as this provides them with enough freedom to try things on their own terms and the way they want the values to used and processed further. Here, the idea is to import historical data in scripting interface(used python for the purpose), and then build and test the strategy based on aspects like return that would have been realised and how does that compare with the return had someone just invested their capital and performed no active management during the interval, results are included in return analysis section.

5.3 Trading simulation

Trading simulation refers to paper trading, which in other words mean participating in trade without using capital. It is popularly used to test how the strategy would perform in real-time environment. Streak and Smart API can be used for paper-trading, both of which are described below:

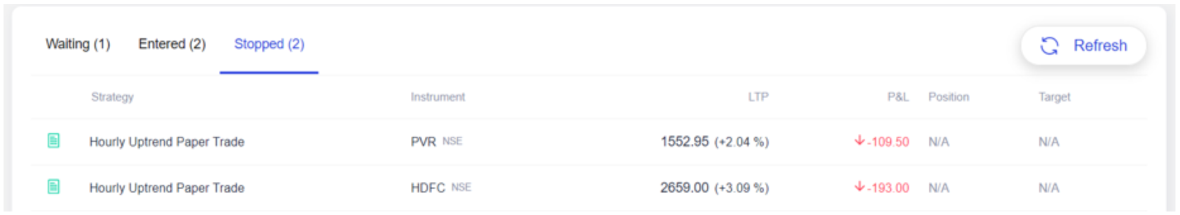
5.3.1 Streak

As we have seen above, Streak is a dashboard based platform, that help us in backtesting. The same platform allows user to simulate trading as well. Users just have to select whether they want to deploy the strategy live or in paper-trading mode. Snapshot of paper-trading instances has been attached below:



Strategy	Instrument	LTP	P&L	Position	Target
Hourly Uptrend Paper Trade	MARUTI NSE	7654.70 (+3.44 %)	+446.50	B 10 at 7610.05	SL: 7556.78 TP: 7807.91
Hourly Uptrend Paper Trade	TCS NSE	3203.45 (+2.04 %)	+3.50	B 10 at 3203.1	SL: 3180.68 TP: 3286.38

Figure 10: Streak papertrading towards target



Strategy	Instrument	LTP	P&L	Position	Target
Hourly Uptrend Paper Trade	PVR NSE	1552.95 (+2.04 %)	-109.50	N/A	N/A
Hourly Uptrend Paper Trade	HDFC NSE	2659.00 (+3.09 %)	-193.00	N/A	N/A

Figure 11: Streak papertrading stoploss hit

The interface provides details about P&L(profit/loss) relaised by the profit and stoploss & target levels decided by the algorithm along with ltp(last traded price). As can be see in Figure 9, the strategy is making profits on Maruti and TCS scrips and stoploss has been hit for PVR and HDFC scrips.

5.3.2 Smart API

This API provides user with real-time data of financial instruments, which can be used to simulate trading by writing scripts. Further details about this has been mentioned in analysis section.

5.4 Risk Management

Risk management is an important aspect in production stage of algorithmic trading deployment. It helps safeguard the capital from significant downturn. There are three popular principles followed to mitigate the risk part of trading:

5.4.1 Max. Drawdown guard

In this particular practice, a percentage is decided based on risk capability of traders, say some number x. When the sum of leftover capital and value of stocks held, if falls below $100 - x$ percent, then auto sqaure-off should be initiated to prevent the trader from further loosing the capital.

5.4.2 Deciding Stop Loss and Target beforehand

This measure should be followed on each of individual trades. Traders should exit from a trade which has delivered pre-decided target and not wait to get some additional return, as the trend may sometimes reverse and price may fall below the purchased level as well. Similarly, appropriate bottom level should also be decided beforehand, that is price fall below that level, then that trade should be squared-off without waiting any further in recovering the losses, as the losses may further increase.

5.4.3 Diversification across different financial instruments

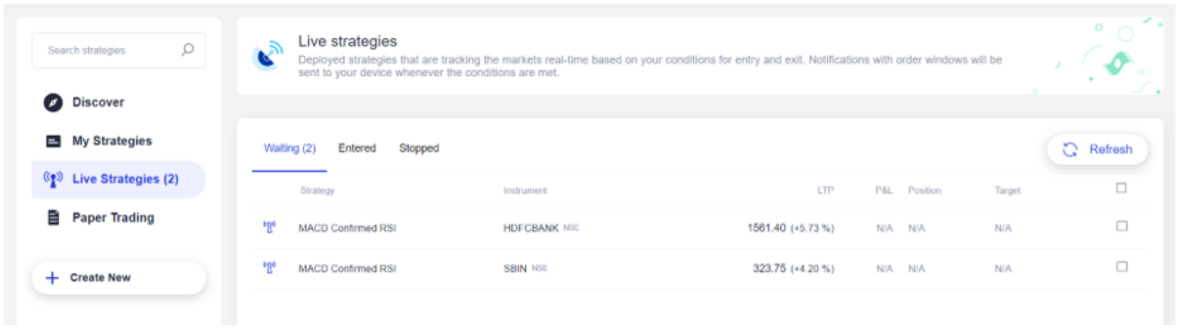
This follows from the principle that all eggs should not be kept in single basket. Traders should invest their capital across varied scrips in different industries say finance, pharma, auto, etc. It's a good practice to not invest greater than 10% of capital on any one scrip, unless that seems very lucrative.

5.5 Live Trading

Here, we plan to execute live order based on signals received by the strategy. For this purpose, we have explored two platforms:

5.5.1 Streak

As discussed earlier, Streak platform allows backtesting and simulated trading(paper trading). This platform also allows user to deploy the strategy live. Sample instance of the live trading interface is attached below:



Live strategies

Deployed strategies that are tracking the markets real-time based on your conditions for entry and exit. Notifications with order windows will be sent to your device whenever the conditions are met.

Strategy	Instrument	LTP	P&L	Position	Target
MACD Confirmed RSI	HDFCBANK NSE	1561.40 (+5.73 %)	N/A	N/A	N/A
MACD Confirmed RSI	SBIN NSE	323.75 (+4.20 %)	N/A	N/A	N/A

Figure 12: Streak strategy live deployment

In the above instance, strategy has been deployed live on HDFCBANK and SBIN scrips. The instance created scans for favourable conditions every minute and if the conditions are met, then it creates a notification on which user has to act to actually place the order. The platform do not allow the option to place the order on user's behalf, and works only in semi-automated mode. This semi-automated aspect of the platform makes this choice less favourable than api based service as the prices may not be that lucrative when the user actually responds to the notification, which happens quite often if the price is significantly changing quite frequently, i.e. if the market is volatile for the selected scrip on a particular trading day.

5.5.2 Smart API

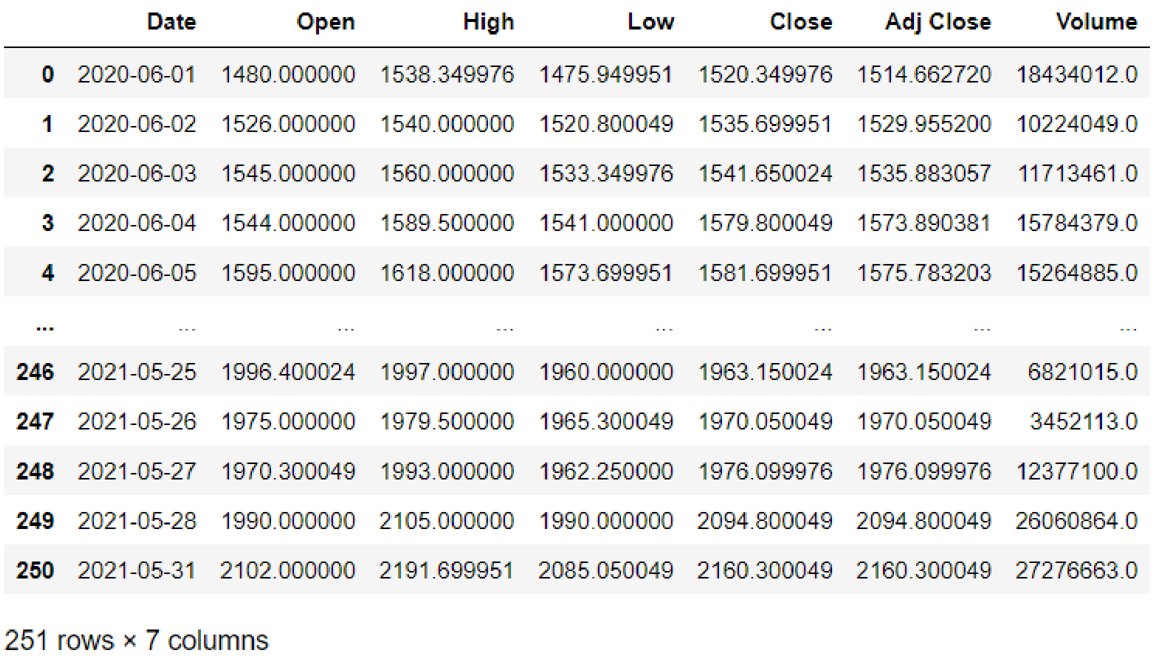
As discussed earlier, this api provides user with real-time market data, using which simulated trading can be performed. On top of it, it also allows to place orders from scripting environment directly, thus helping users to place orders when the right conditions based on buy-sell signal received by the developed trading strategy. Details about the strategy, and its return aspect of it has been described in "Analysis" section.

6 Data Model

We have used two different data sources to make the trading model. For training and testing of the ARTIMA model, financial instrument data of Reliance insdutries stock price has been procured from yahoo finance and for placing the orders for stock purchase in real-time SmartAPI provided by Angel Broking has been used.

6.1 Data model for ARIMA model training & testing

Reliance stock data ranging from period 01-06-2020 to 31-05-2021 with frequency of daily has been used. The below is an attached snapshot of the data imported in python:



	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-06-01	1480.000000	1538.349976	1475.949951	1520.349976	1514.662720	18434012.0
1	2020-06-02	1526.000000	1540.000000	1520.800049	1535.699951	1529.955200	10224049.0
2	2020-06-03	1545.000000	1560.000000	1533.349976	1541.650024	1535.883057	11713461.0
3	2020-06-04	1544.000000	1589.500000	1541.000000	1579.800049	1573.890381	15784379.0
4	2020-06-05	1595.000000	1618.000000	1573.699951	1581.699951	1575.783203	15264885.0
...	...	...	...	...	...	...	...
246	2021-05-25	1996.400024	1997.000000	1960.000000	1963.150024	1963.150024	6821015.0
247	2021-05-26	1975.000000	1979.500000	1965.300049	1970.050049	1970.050049	3452113.0
248	2021-05-27	1970.300049	1993.000000	1962.250000	1976.099976	1976.099976	12377100.0
249	2021-05-28	1990.000000	2105.000000	1990.000000	2094.800049	2094.800049	26060864.0
250	2021-05-31	2102.000000	2191.699951	2085.050049	2160.300049	2160.300049	27276663.0

251 rows × 7 columns

Figure 13: Reliance Historical Data

This data contains information of its daily price movement, various fields of which are described below:

1. **Date**: This field represents the time aspect of the time-series data, particularly date on which that particular data has been recorded for the reliance stock. This field is serially arranged, as to be used by the ARIMA model. The earlier date comes

first and the later one comes later. This field contains all the date for which trading has happened(excludes holidays and any national holidays in general) for the period 01-06-2020 to 31-05-2021.

2. **Open**: This field represents the price at which first transaction executed for the reliance stock, i.e. say some seller,s sold the stock to some buyer, b at this price for the first time on that particular trading day.

3. **High**: This field represents the highest price at which the instrument was exchanged for on that particular day.

4. **Low**: This field represents the lowest price at which the instrument was exchanged for on that particular day.

5. **Close**: This field represents the execution price of the last transaction that happened on that particular day.

6. **Adj. Close**: This field represents the closing price of the instrument after accounting for corporate actions.

7. **Volume**: This field represents the amount of transactions that did get executed on that particular date for the instrument.

For our analysis, "Close" field data has been used. Reference to "Close" field data should be made for any mention of data henceforth.

Line plot of the data has been appended below:

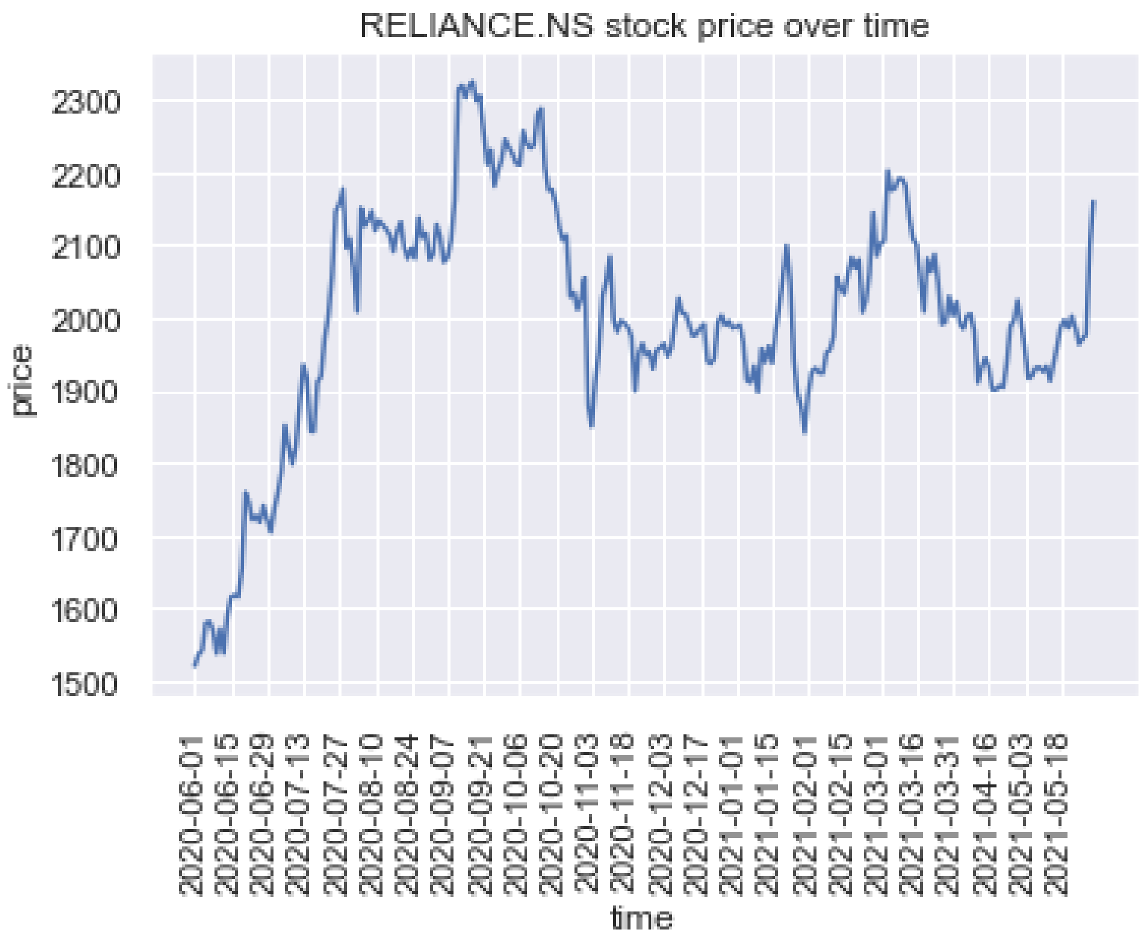


Figure 14: Reliance Historical Data plot

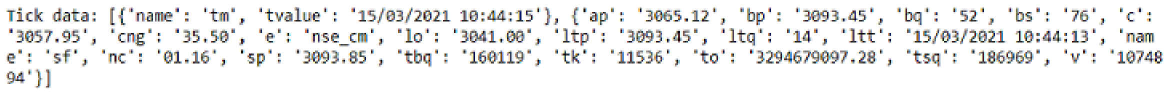
For training and testing purpose, 80-20 split has been considered. So, out of the total 250 instances, 200 has been used for training and the rest 50 has been used for testing purpose.

Table 1: **Data Details**

	Date Range	**No. of Instances**
Original Set	01-06-2020 to 31-05-2021	250
Training Set	01-06-2020 to 12-03-2021	200
Testing Set	15-03-2021 to 31-05-2021	50

6.2 Data model for placing real time orders

SmartAPI provided by Angel Broking has been used to feed real-time data into ARIMA model and place orders when price-action seems to be favourable. Sample data instance received from the api has been appended below:



```
Tick data: [{'name': 'tm', 'tvalue': '15/03/2021 10:44:15'}, {'ap': '3065.12', 'bp': '3093.45', 'bq': '52', 'bs': '76', 'c':
'3057.95', 'cng': '35.50', 'e': 'nse_cm', 'lo': '3041.00', 'ltp': '3093.45', 'ltq': '14', 'ltt': '15/03/2021 10:44:13', 'nam
e': 'sf', 'nc': '01.16', 'sp': '3093.85', 'tbq': '160119', 'tk': '11536', 'to': '3294679097.28', 'tsq': '186969', 'v': '10748
94'}]
```

Figure 15: Real-time data received from SmartAPI format

The data provided by the api contains the following relevant information:

1. **name**: Name of the financial instrument.
2. **tvalue**: Time at which the data has been relayed.
3. **ap**: Ask-price, lowest price that is currently being offered by any seller.
4. **bp**: Bid-price, highest price being offered by the buyer at the time.
5. **bq**: Bid-quantity, quantity being offered at the bid-price.
6. **cng**: Change in ltp from previous close price
7. **e**: Exchange, stock-exchange whose data is being relayed.
8. **lo**: Low, lowest price at which the instrument was traded for on the day.
9. **ltp**: Last traded price, price at which the most recent transaction executed.
10. **ltq**: Last traded quatity, quantity exchanged most recently.

11. **ltt**: Last traded time, time instance when the instrument was exchanged most recently.

12. **tk**: Token number(similar to id number) for financial instruments

13. **v**: Volume, total number of transactions executed till time for the day.

For our purpose of feeding the next time-instant data to the ARIMA model, we have used '**ltp**' data and placed **market orders**, which implies the order should be executed at the best possible offer price(ap).

7 Analysis

We have used time-series based ARIMA model to forecast the value of price data for the next time-instant. This statistical model is very commonly used for time-series forecasting and requires tuning in the parameters, namely p,d & q for improved quality of prediction. The governing equation of the model is:

$$y_t = c + \sum_{i=1}^{p} \phi_i * y_{t-i} + \sum_{j=1}^{q} \theta_j * \epsilon_{t-j} \tag{1}$$

where y_{t-i} is i_{th} lag of y_t, ϵ_{t-j} is j_{th} error term and c is some constant.

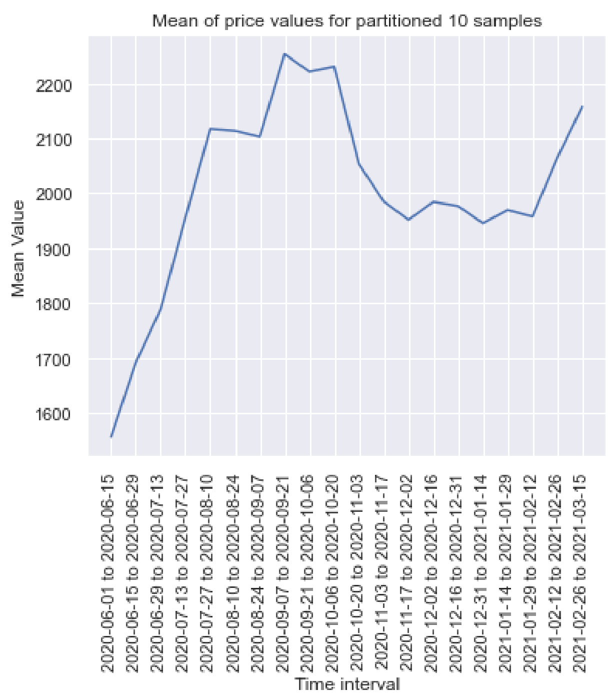
Before fitting ARIMA model, data should be made stationary first by adjusting the level of differencing to be used, d and then decide on the number of lagged observation(p) and the number of error terms(q) to be used by the model for prediction based on the below described methods:

7.1 Estimation of parameter, d

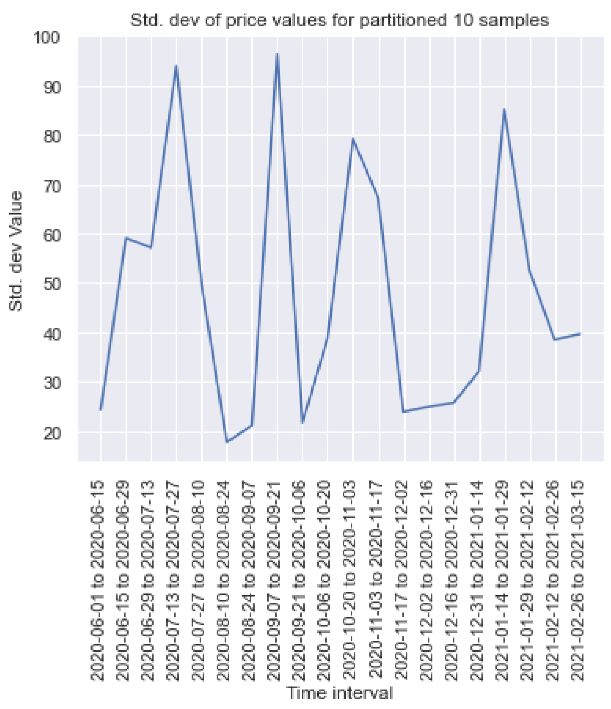
This parameter dictates the order of differencing that the ARIMA model should use to make prediction. Generally 1st order differencing suffice the stationarity requirement of the model, and in very rare cases 2nd, 3rd or higher order of differencing is considered. We first check if the data to be used is already stationary, in which case no differencing(d = 0) needs to be considered, or not. Data being stationary means constant mean and constant standard deviation. Generally two different methods are being followed to check for stationarity:

7.1.1 Visual test of stationarity

In this method, data to be checked for stationarity is being grouped into groups and then mean and standard deviation of each of the groups is being plotted. If these values do change significantly across groups then the data will be classified as a stationary data. This method leaves a lot of responsibility on the user, and he/she need to make a call if the variation is significant or not.
For our purpose, training data has been grouped into 20 groups each with 10 continuous(demand of the test) instances. Their mean and variance plot has been attached below:



(a) Mean across groups



(b) Std. dev across groups

Figure 16: Visual test of stationarity on raw data

Standard deviation values across groups can be assumed to be not varying significantly across groups but mean across groups is definitely far from being constant, shows an increasing trend initially and then kind of becomes constant. So, based on this test, raw-data is non-stationary. We need to check if 1st order of differencing can pass the test.

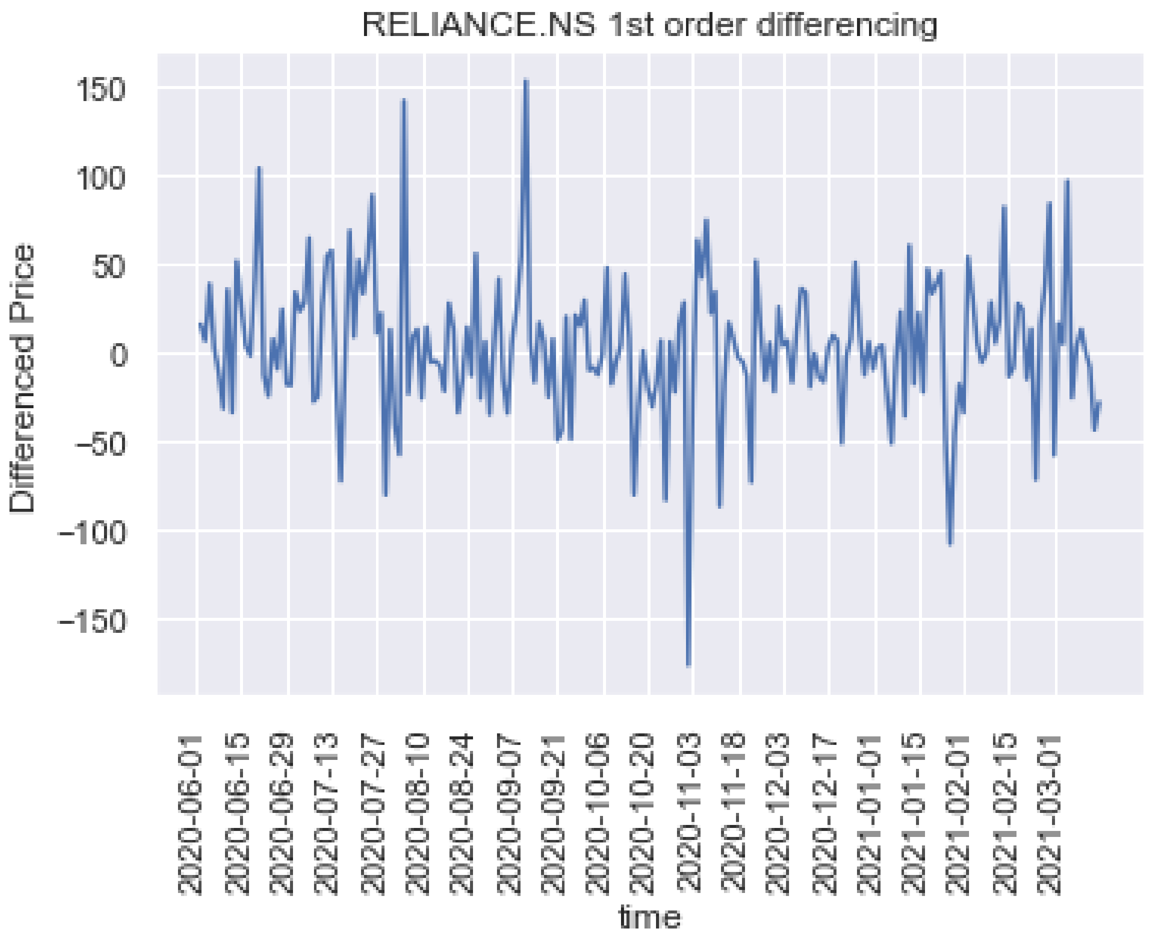
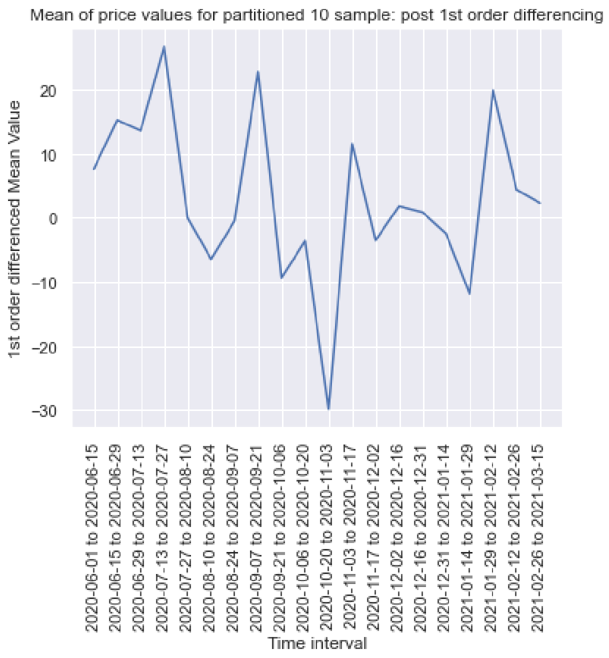
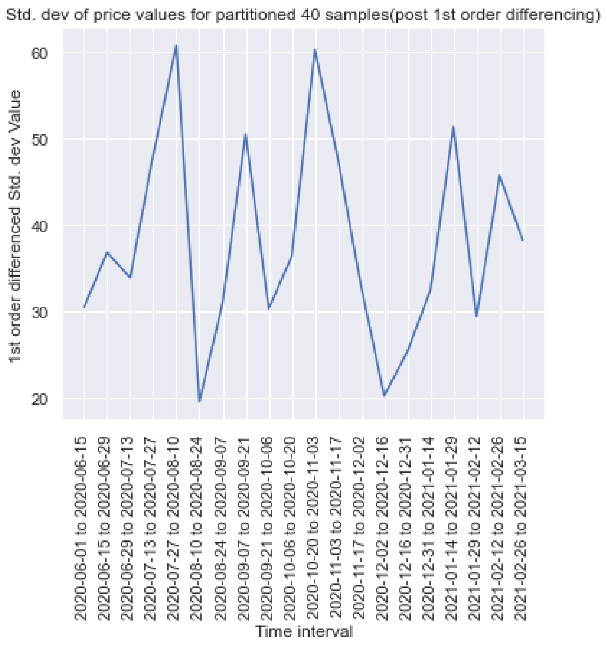


Figure 17: 1st-order differenced price plot



(a) Mean across groups



(b) Std. dev across groups

Figure 18: Visual test of stationarity on 1st order differenced price

Here, we see that the mean and standard deviation values across groups is reverting towards some constant value, close to 0 and 30 respectively. So, 1st order differencing is passing this visual test, let's now check what ADF test has to offer.

7.1.2 ADF test of stationarity

Acronym for Augmented Dickey-Fuller test. This is a popular test for stationarity, and it uses the principle of hypothesis testing to let the user judge for data stationarity. Null and alternate hypotheses for the test are as follows:

- H_0: Unit root is present and hence time-series is non-stationary
- H_1: Unit root is not present and hence time-series is stationary

Results of the test performed on raw data(on data with no differencing considered):

- Test statistics: -2.73
- p-value: 0.068
- critical values
  - 1%: -3.46
  - 5%: -2.87

We cannot reject null hypothesis as p-value is greater than significance level(commonly taken as 0.05) and test-statistics is greater than both 1% and 5% critical values. This let's us conclude that the time-series considered is non-stationary.

So, let's analyse the result on 1st order differenced data:

- Test statistics: -13.23
- p-value: 9e-25
- critical values
  - 1%: -3.46
  - 5%: -2.87

As p-value is less than significance level and test-statistics is less than both 1% and 5% critical values, we can reject null hypothesis, concluding that the 1st order differencing should be considered to ensure data stationarity.

Both the test suggests us to use a 1st order differencing to train our ARIMA model, so we finalize the value of d as 1.

7.2 Estimation of parameter, q

This parameter controls the moving average(MA) part of ARIMA model, i.e. number of error terms that the model will essentially use to forecast the time-series model.

$$y_t = c + \sum_{i=1}^{p} \phi_i * y_{t-i} + \sum_{j=1}^{q} \theta_j * \epsilon_{t-j}$$

The parameter controls the third component of the ARIMA's governing equation. To test the optimal number of q terms to be used, ACF(Autocorrelation Function) is utilised, governing equation of ACF is described below:

$$ACF(k) = ACF(y_t, t_{t-k}) = \frac{cov(y_t, y_{t-k})}{\sqrt{var(y_t) * var(y_{t-k}}} \tag{2}$$

where $cov()$ and $var()$ is covariance and variance function respectively.

$$cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) * (y_i - \bar{y})}{n - 1} \tag{3}$$

$$var(x) = cov(x, x) \tag{4}$$

ACF plot on the training set is attached below:

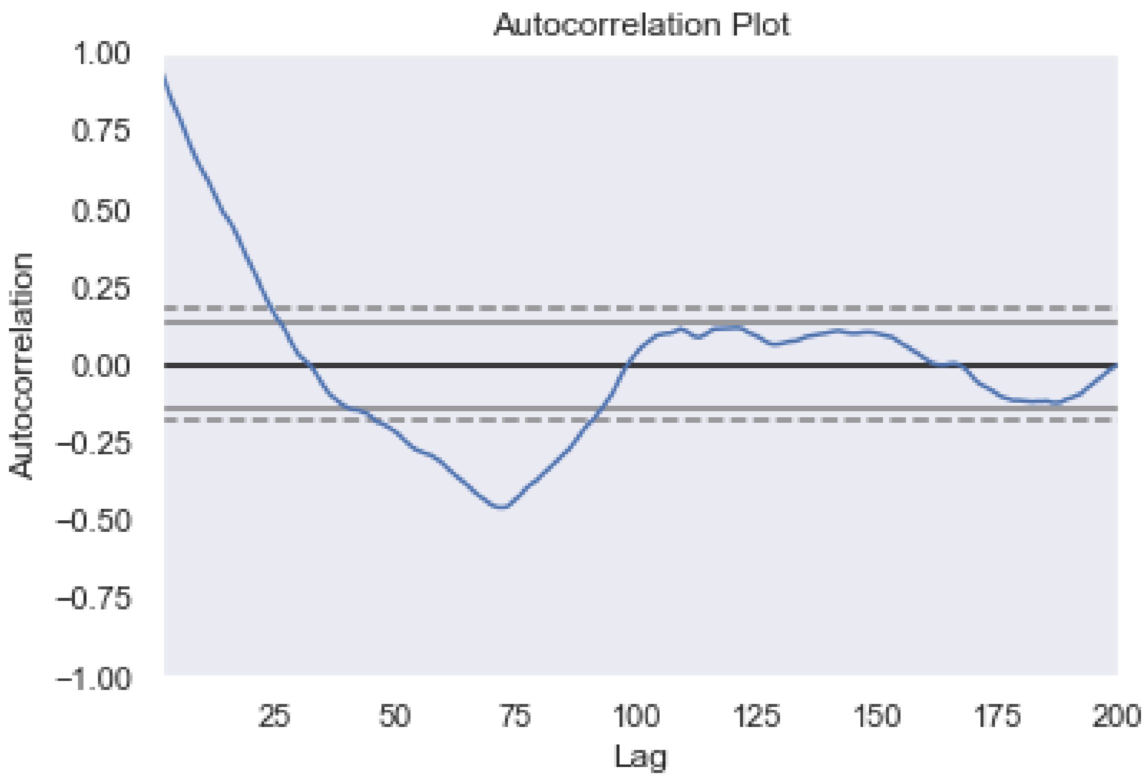


Figure 19: ACF plot

First intersection of the ACF curve with the significance level(SL) if less than 10 is generally suggested to be taken as the value of the parameter, q. Here, we see that the required intersection happens at lag value of 25, which is an indication that user should opt for q value of 0, i.e. MA(0) and look for optimal p parameter for the AR(p) part.

7.3 Estimation of parameter, p

This parameter controls the number of lag terms that ARIMA model should make use of for making forecasting, essentially controls the 2nd component of the ARIMA's equation:

$$y_t = c + \sum_{i=1}^{p} \phi_i * y_{t-i} + \sum_{j=1}^{q} \theta_j * \epsilon_{t-j}$$

To estimate the parameter, p; partial PACF(partial autocorrelation function) is made use of. The governing equation of PACF is outlined below:

$$PACF(1) = ACF(1)$$

$$PACF(2) = \frac{cov(y_t, y_{t-2}|y_{t-1})}{sqrt(var(y_t|y_{t-1}) * var(y_{t-2}|y_{t-1})}$$

$$PACF(3) = \frac{cov(y_t, y_{t-3}|y_{t-1}, y_{t-2})}{sqrt(var(y_t|y_{t-1}, y_{t-2}) * var(y_{t-3}|y_{t-1}, y_{t-2})} \tag{5}$$

and so on ..

PACF plot for the training dataset is attached below:

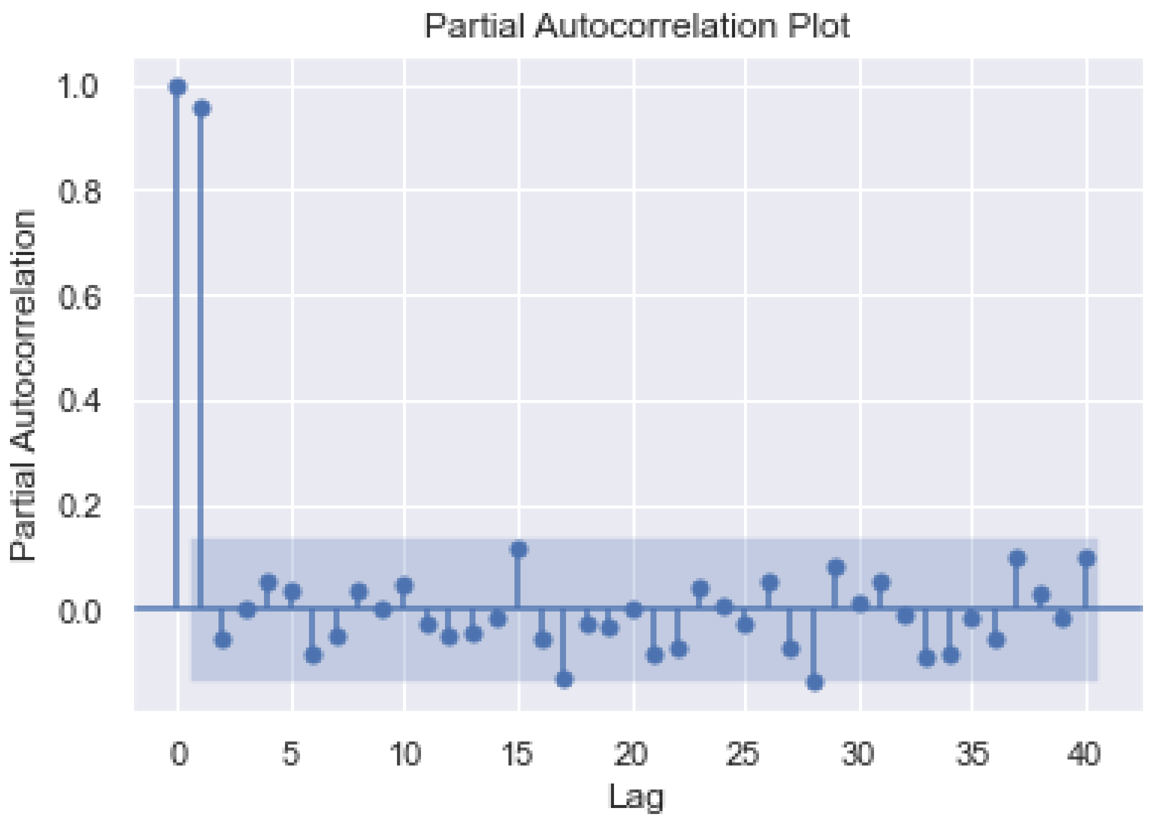


Figure 20: PACF plot

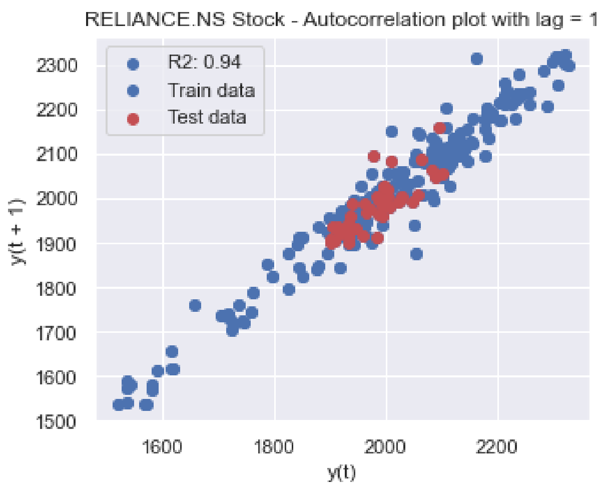
As we can see in the above plot, 5 lags have PACF value greater than the significance level, so this suggests that we should use the p value of 5, i.e. AR(5)

7.4 Lag plot Analysis

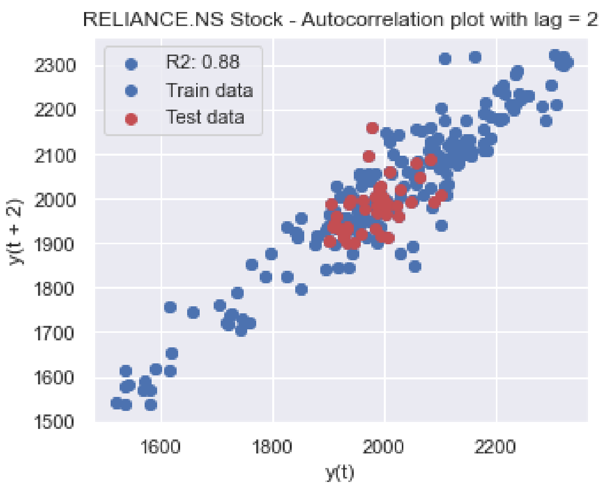
Lag plot is just a scatter plot between a variable and the lags of itself. It provides a good visual representation on how the current instant data is correlated with its previous values.

A higher correlation means greater dependence of current instance data on the previous instance values.

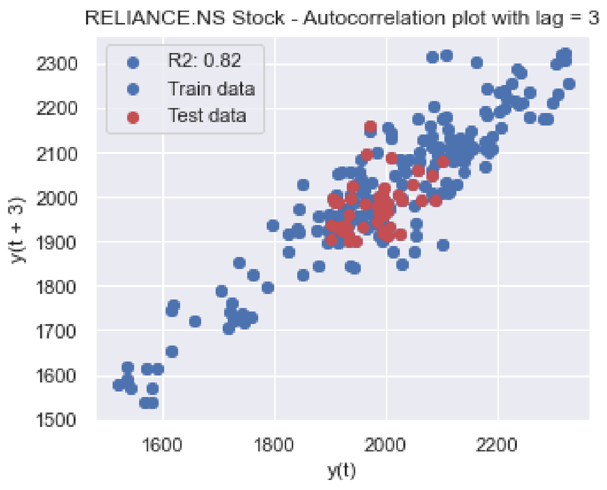
7.4.1 Lag plot of complete dataset



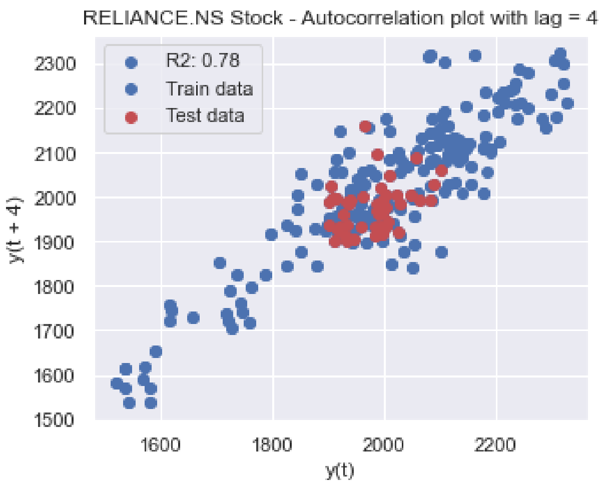
(a) Lag 1



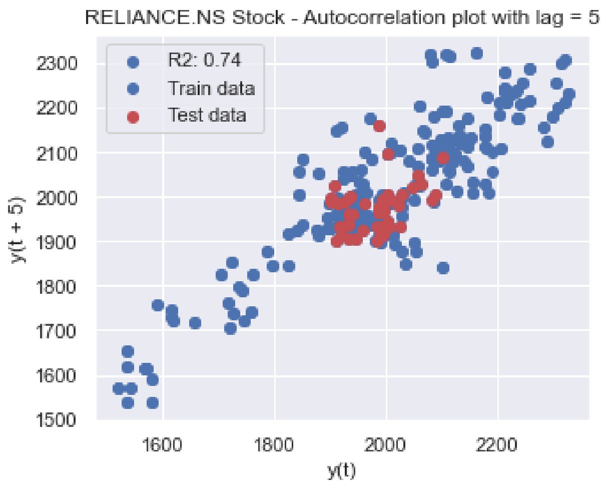
(b) Lag 2



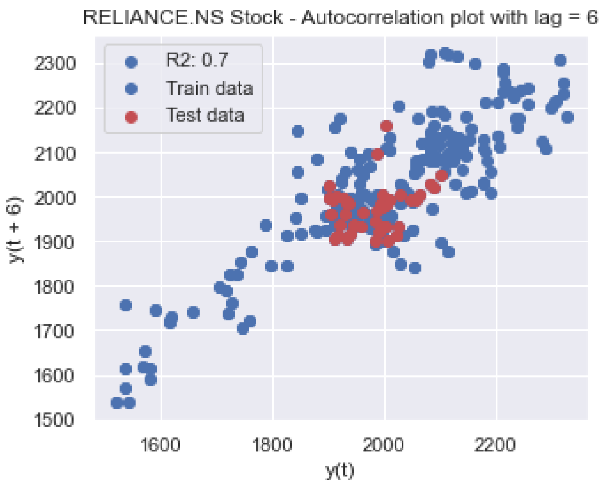
(c) Lag 3



(d) Lag 4



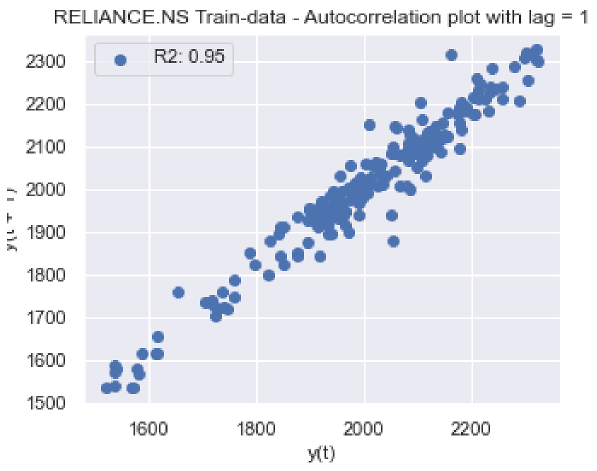
(e) Lag 5



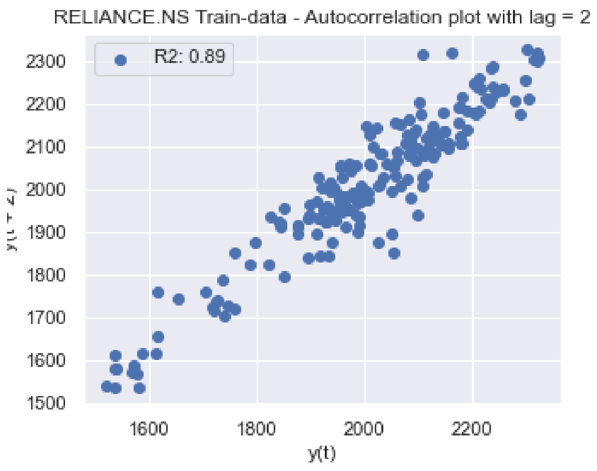
(f) Lag 6

Figure 21: Lag plot of Complete dataset

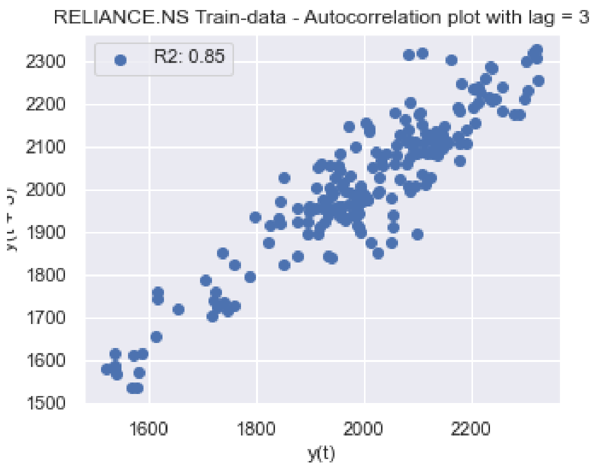
7.4.2 Lag plot of training set



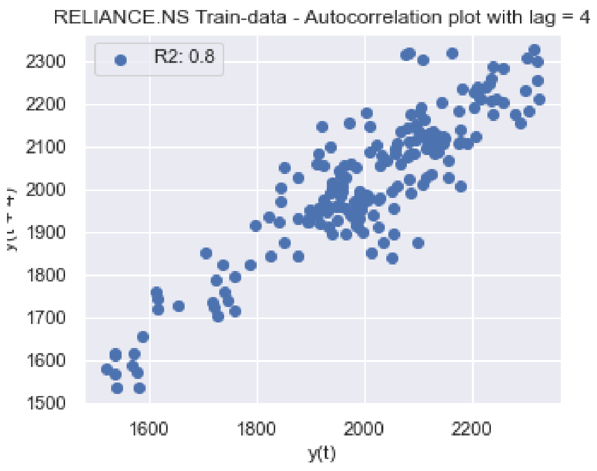
(a) Lag 1



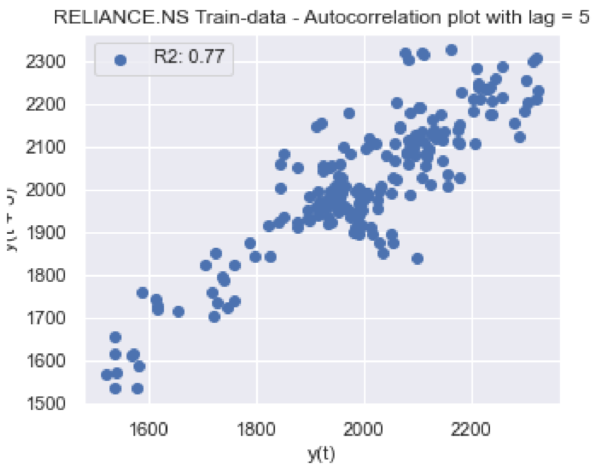
(b) Lag 2



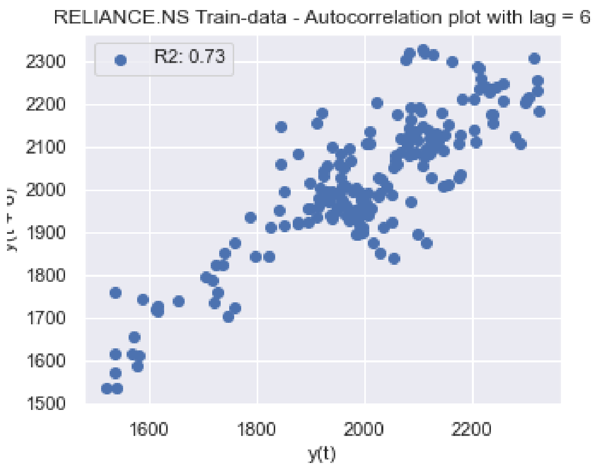
(c) Lag 3



(d) Lag 4



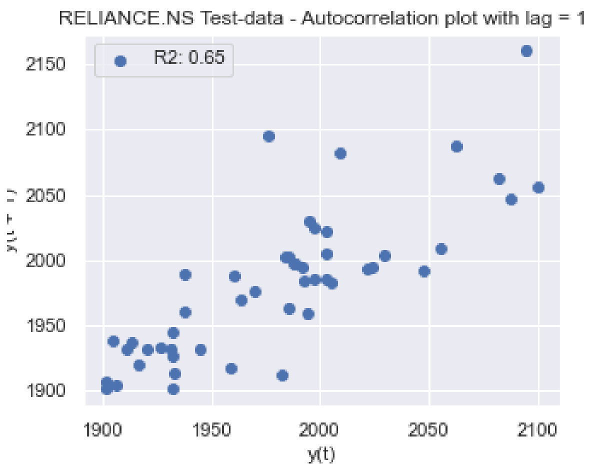
(e) Lag 5



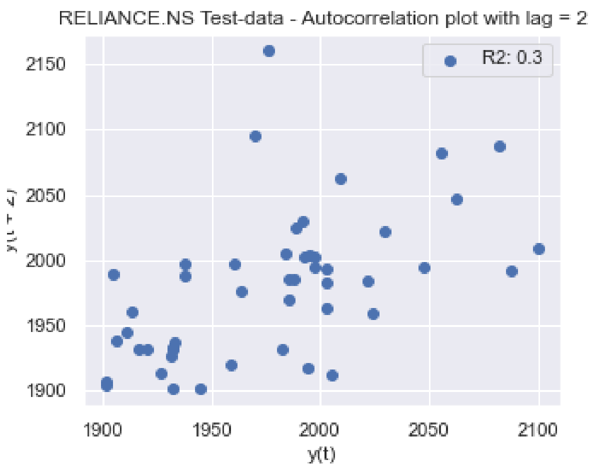
(f) Lag 6

Figure 22: Lag plot of Training Set

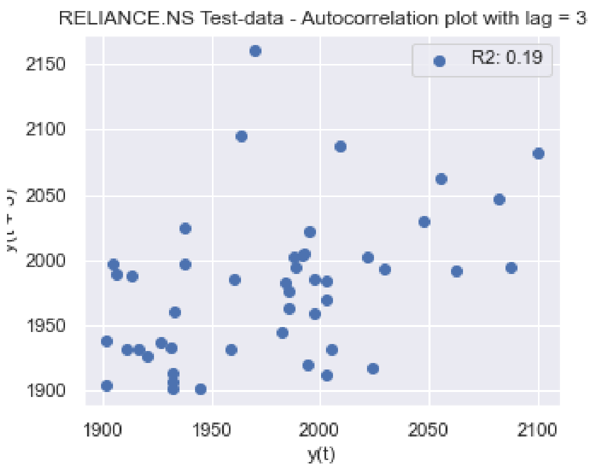
7.4.3 Lag plot of testing set



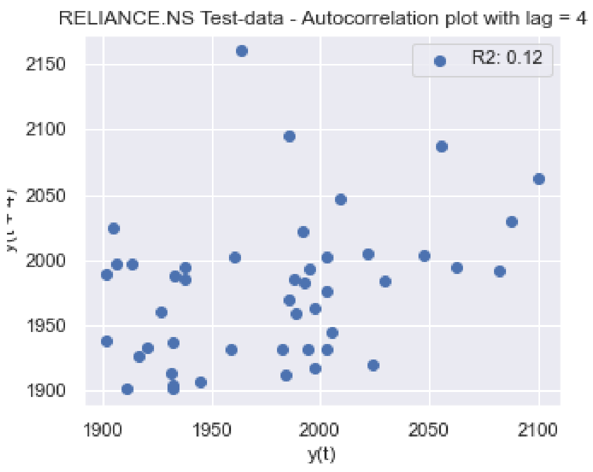
(a) Lag 1



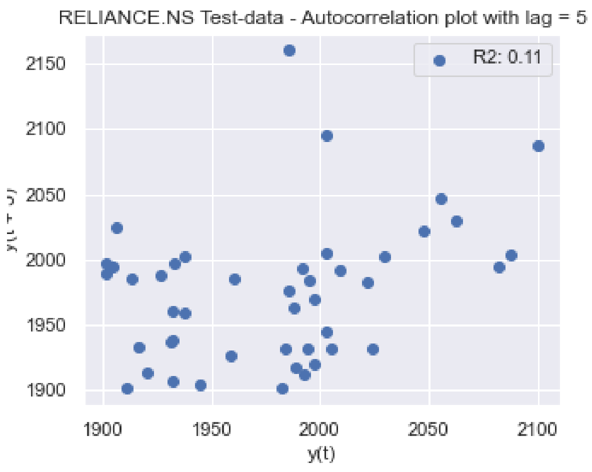
(b) Lag 2



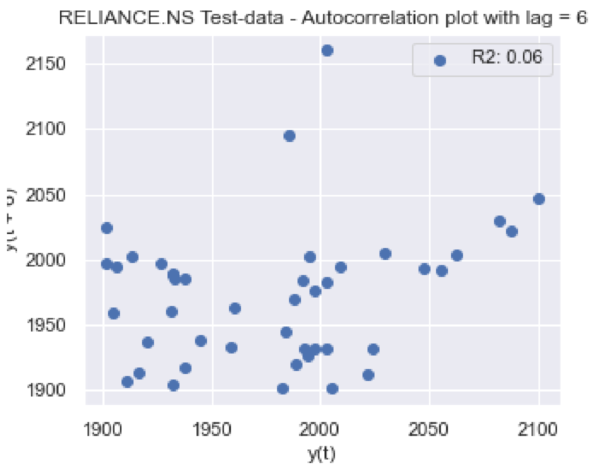
(c) Lag 3



(d) Lag 4



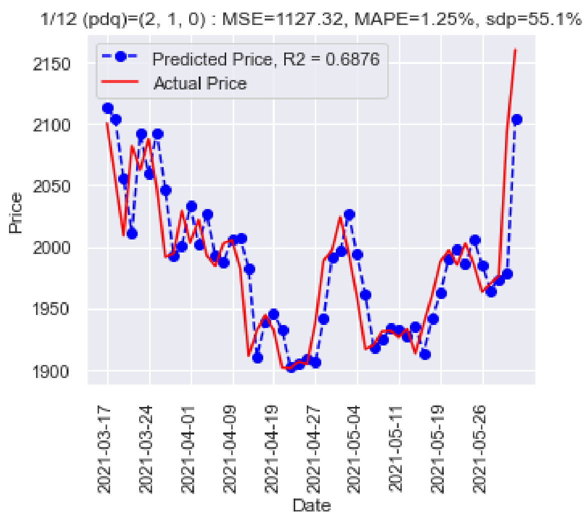
(e) Lag 5



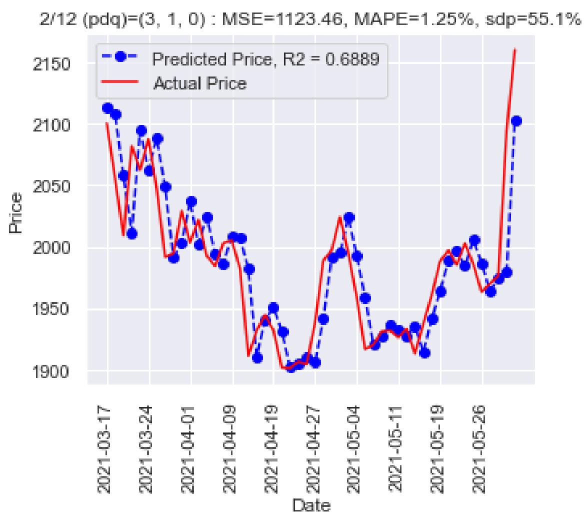
(f) Lag 6

Figure 23: Lag plot of Testing Set

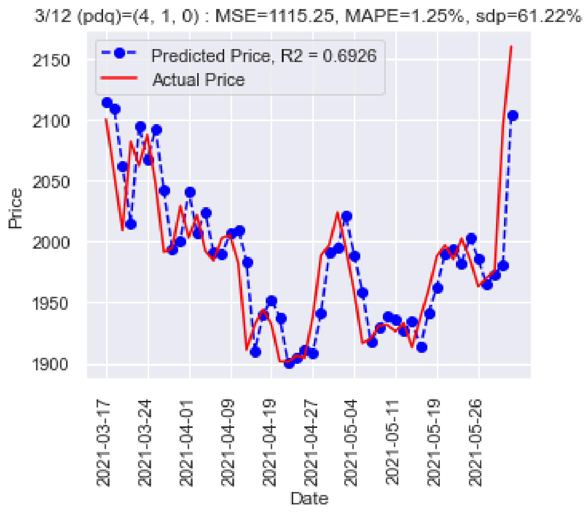
7.5 Cross-Validation of hyper-parameters



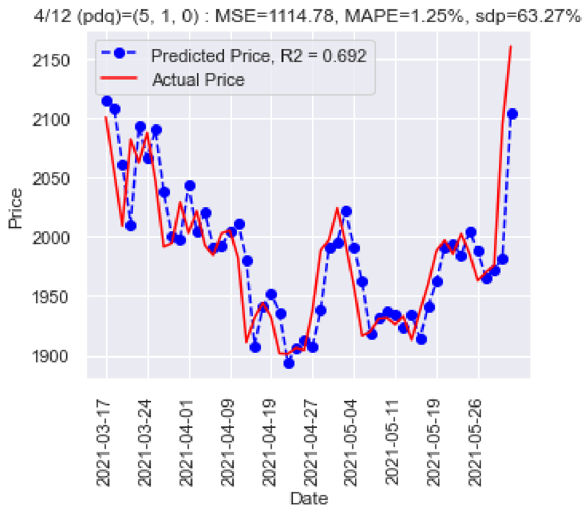
(a) Combination 1



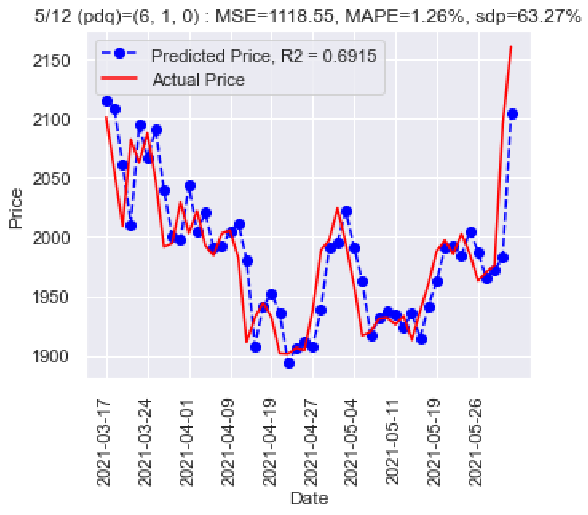
(b) Combination 2



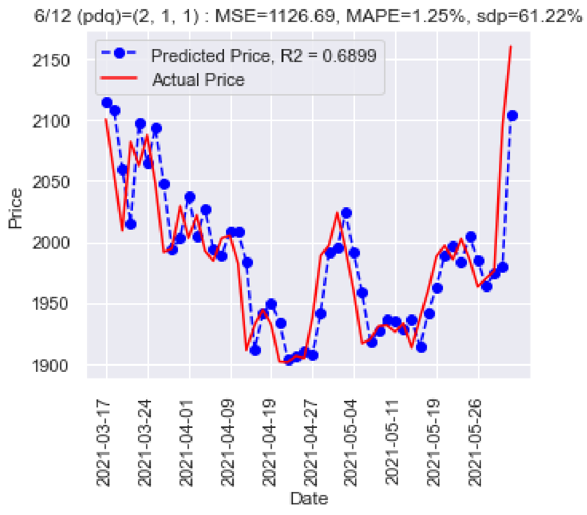
(c) Combination 3



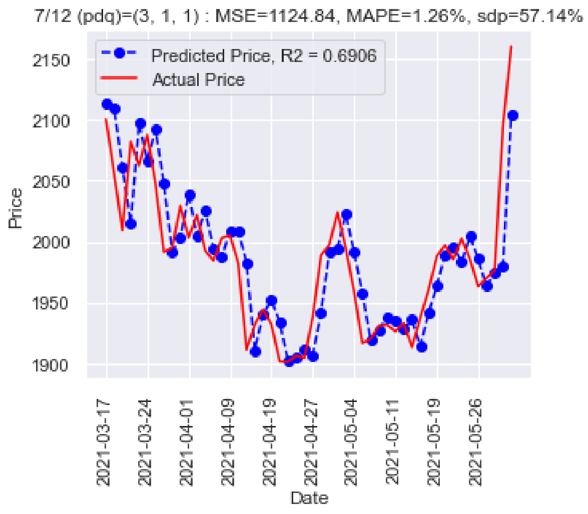
(d) Combination 4



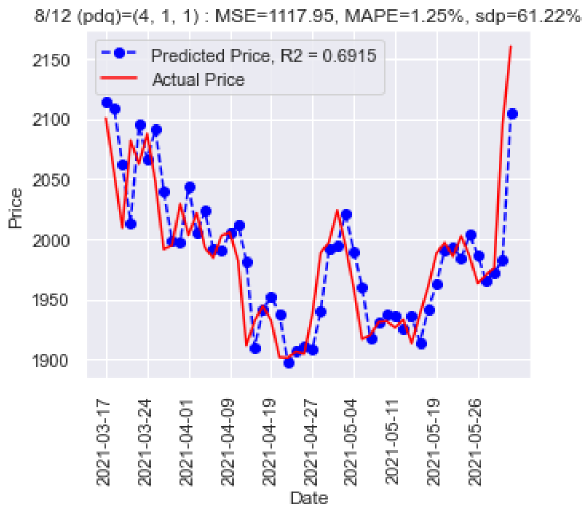
(e) Combination 5



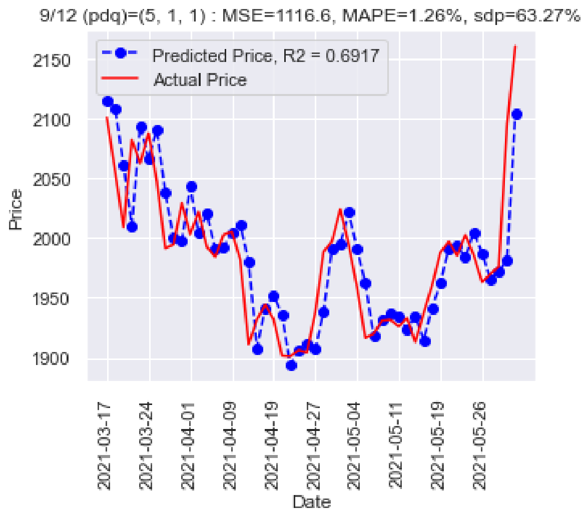
(f) Combination 6



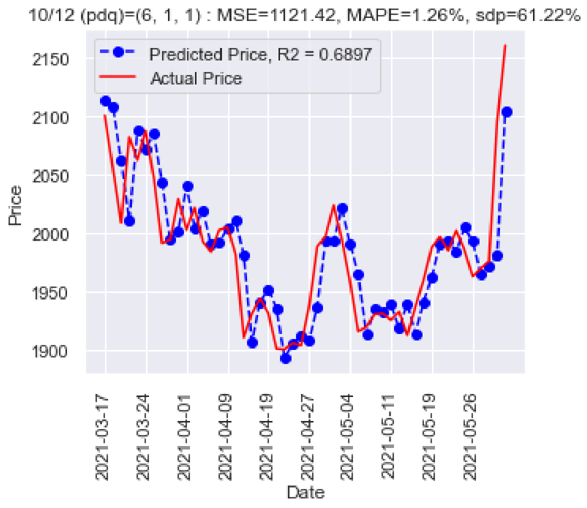
(g) Combination 7



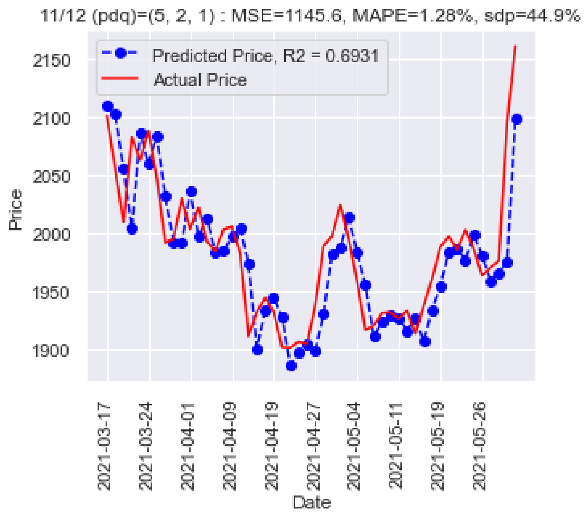
(h) Combination 8



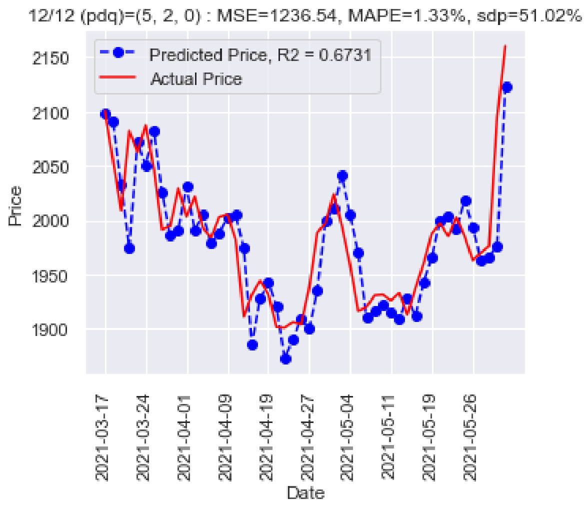
(i) Combination 9



(j) Combination 10



(k) Combination 11



(l) Combination 12

Figure 24: Hyper-parameter cross-validation

7.6 Error plots

Error plots help user select the hyper-parameters based on aspects that are of greater importance. Popular measures of error are MSE(mean-sqaured error) and MAPE(mean absolute percentage error). In the domain of forecasting AIC(Akaike's Information Criterion) and BIC(Bayesian Information Criterion) are also popularly used. Also, we have used this section to talk about R2 variaton across the hyper-parameters. Apart from these, as we are interested in trading aspects of forecasting, CDP(Correct Direction Percentage) has also been analyzed and it help us quantify the percentage of instances for which direction of price movement, either increase or decrease from previous day price value is rightly captured. Mathematical expressions of these terms are described below:

1. **MSE**

   MSE stands for Mean Squared Error, and is defined as below:

$$MSE = \sum_{t=1}^{n} (\hat{y_t} - y_t)^2 * \frac{1}{n} \tag{6}$$

2. **MAPE**

   MAPE stands for Mean Absolute Percentage Error, and is defined as below:

$$MAPE = \sum_{t=1}^{n} |\frac{\hat{y_t} - y_t}{y_t}| * \frac{100}{n} \tag{7}$$

3. **CDP**

   CDP stands for Correct Direction Percentage, and is defined as below:

$$CDP = [\sum_{t=1}^{n} 1\, if\ (\hat{y_t} - y_{t-1}) * (y_t - y_{t-1}) > 0\ else\ 0] * \frac{100}{n} \tag{8}$$

4. **R2**

   R2 again a very popular metric termed as "Coefficient of Determination" and is defined as square of correlation coefficient, which in turn is defined as covariance of two vectors divided by product of their standard deviation as below:

$$Corr(\hat{y}, y) = \frac{Cov(\hat{y}, y)}{\sqrt{Var(\hat{y}) * Var(y)}} \tag{9}$$

$$Cov(\hat{y}, y) = \sum_{t=1}^{n} (\hat{y}_t - \bar{\hat{y}}) * (y_t - \bar{y}) * \frac{1}{n-1} \tag{10}$$

$$R2(\hat{y}, y) = Corr(\hat{y}, y)^2 \tag{11}$$

5. **AIC**

   Acronym for Akaike's Information Criterion. AIC & BIC defined next are popular metric used in time-series forecasting. AIC in particular, is useful in selecting predictors for regression; lower the value of AIC, better the model in general is considered. AIC is defined as below:

$$AIC = -2 * log(L) + 2 * (p + q + k + 1) \tag{12}$$

   where L is the likelihood of the data, k = 1 if c $\neq$ 0 and k = 0 if c = 0.

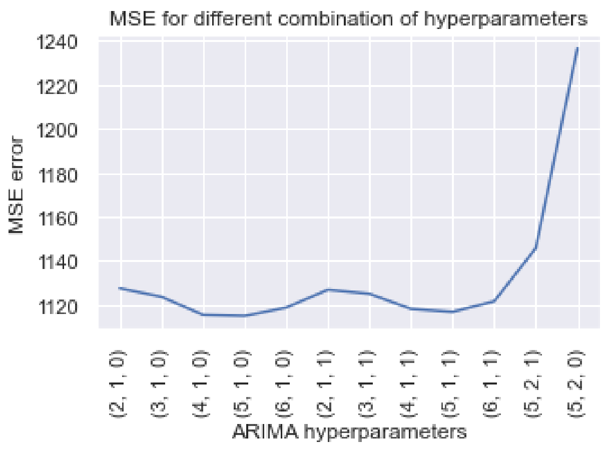
   For ARIMA models, a slighltly modified version of AIC is commonly used, let's say AICc defined as below:

$$AICc = AIC + \frac{(2 * (p + q + k + 1) * (p + q + k + 2)}{T - p - q - k - 2} \tag{13}$$

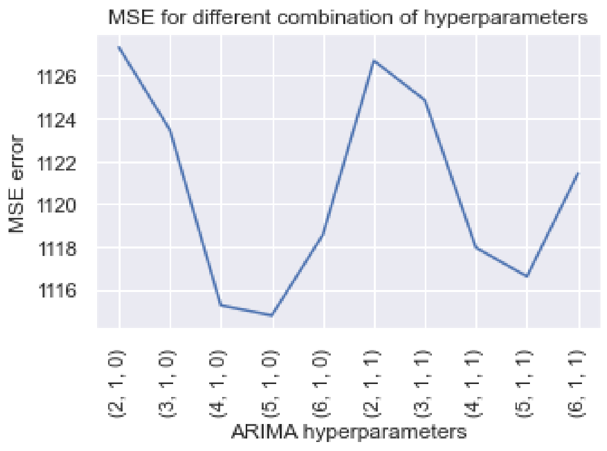
6. **BIC**

   BIC stands for Bayesian Information Criterion and this is another metric used popularly in forecasting based models. Mathematical expression of the quantity is described below:

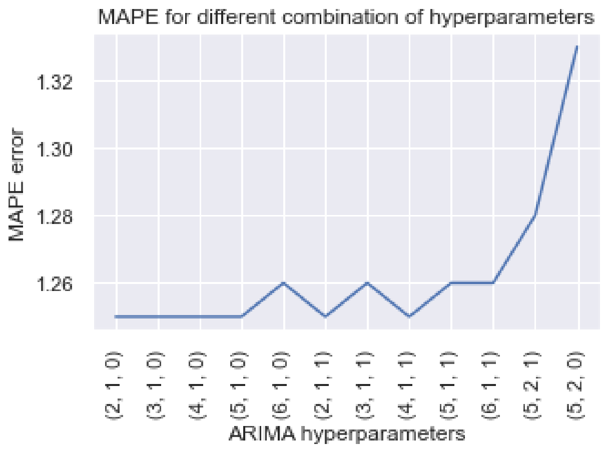
$$BIC = AIC + (log(T) - 2) * (p + q + k + 1) \tag{14}$$



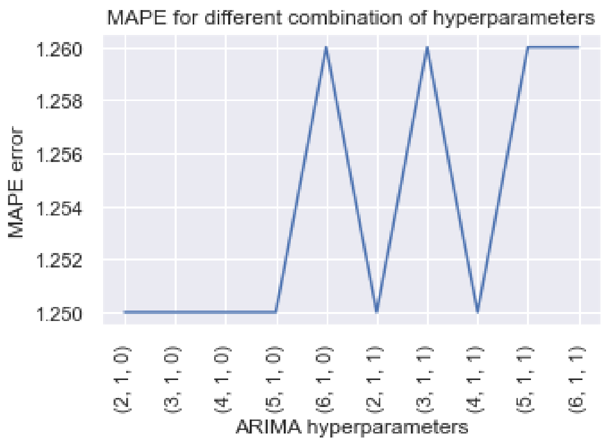
(a) MSE



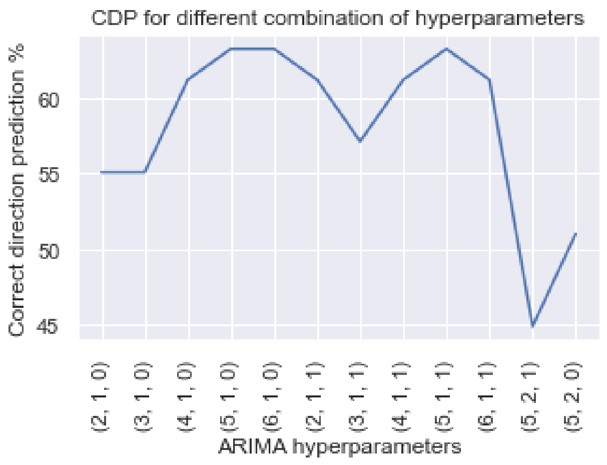
(b) MSE trimmed



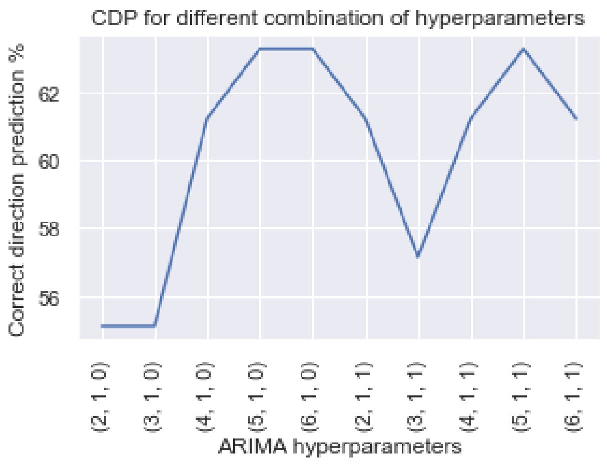
(c) MAPE



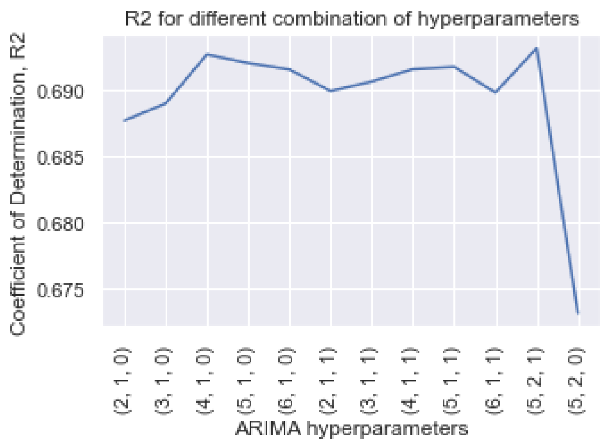
(d) MAPE trimmed



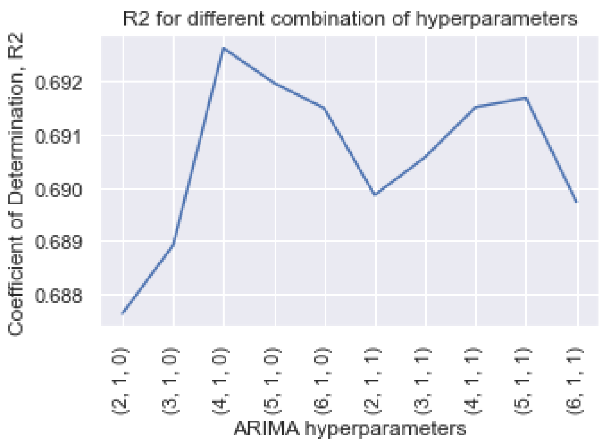
(e) CDP



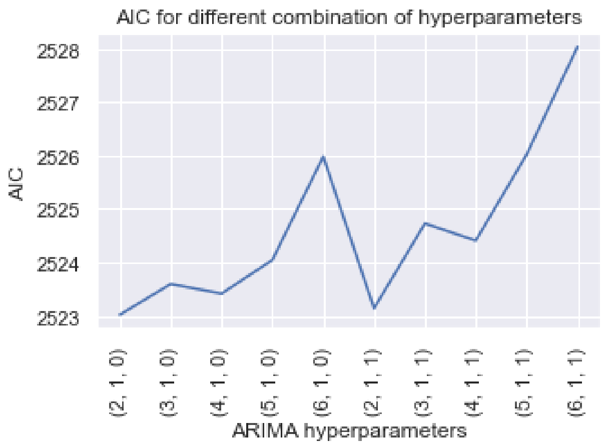
(f) CDP trimmed



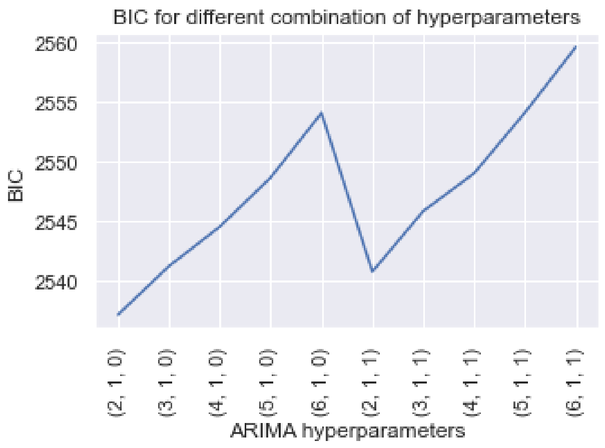
(a) R2



(b) R2 trimmed



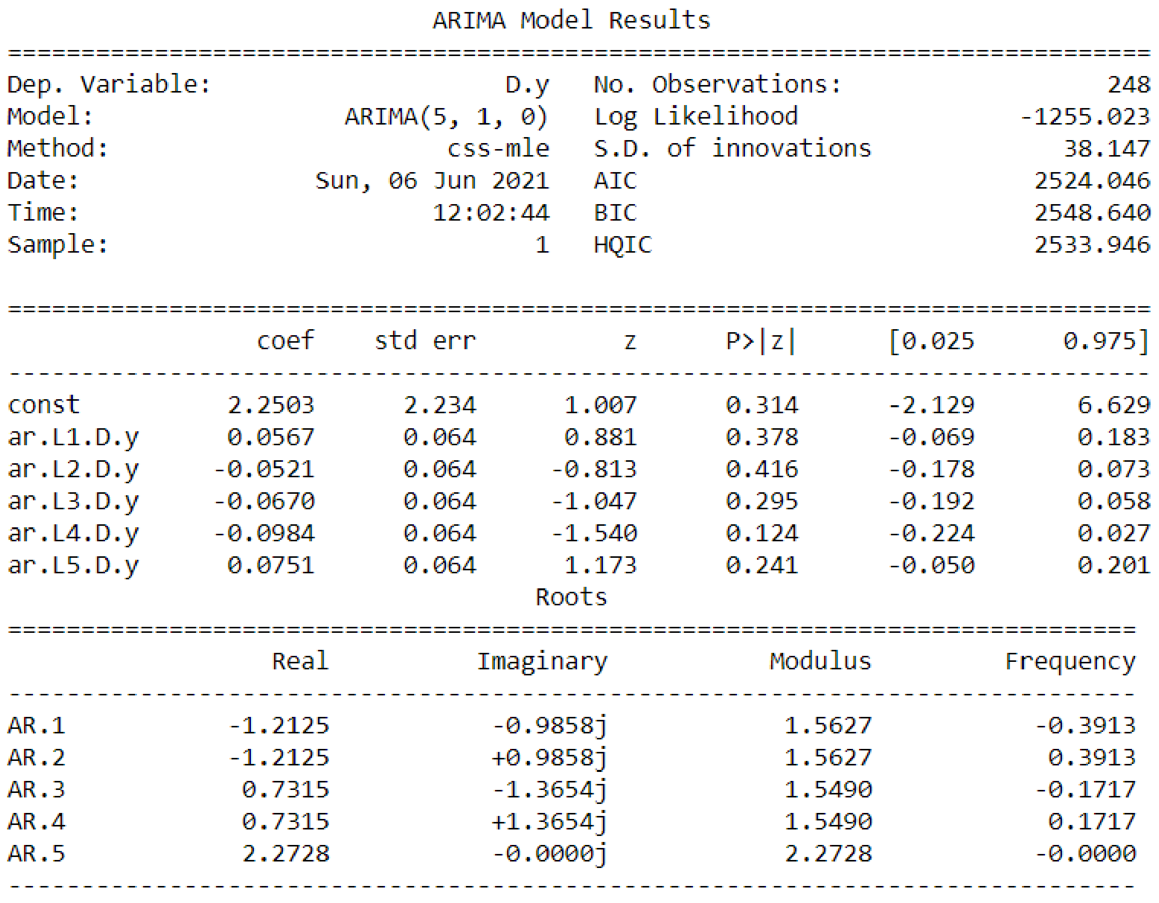
(c) AIC



(d) BIC

Figure 26: Error plots

7.7 Model Summary



```
                              ARIMA Model Results
==============================================================================
Dep. Variable:                    D.y   No. Observations:                  248
Model:                 ARIMA(5, 1, 0)   Log Likelihood               -1255.023
Method:                       css-mle   S.D. of innovations             38.147
Date:                Sun, 06 Jun 2021   AIC                           2524.046
Time:                        12:02:44   BIC                           2548.640
Sample:                             1   HQIC                          2533.946

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.2503      2.234      1.007      0.314      -2.129       6.629
ar.L1.D.y      0.0567      0.064      0.881      0.378      -0.069       0.183
ar.L2.D.y     -0.0521      0.064     -0.813      0.416      -0.178       0.073
ar.L3.D.y     -0.0670      0.064     -1.047      0.295      -0.192       0.058
ar.L4.D.y     -0.0984      0.064     -1.540      0.124      -0.224       0.027
ar.L5.D.y      0.0751      0.064      1.173      0.241      -0.050       0.201
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -1.2125           -0.9858j            1.5627           -0.3913
AR.2           -1.2125           +0.9858j            1.5627            0.3913
AR.3            0.7315           -1.3654j            1.5490           -0.1717
AR.4            0.7315           +1.3654j            1.5490            0.1717
AR.5            2.2728           -0.0000j            2.2728           -0.0000
-----------------------------------------------------------------------------
```

Figure 27: Model Result for (p,d,q) = (5,1,0)

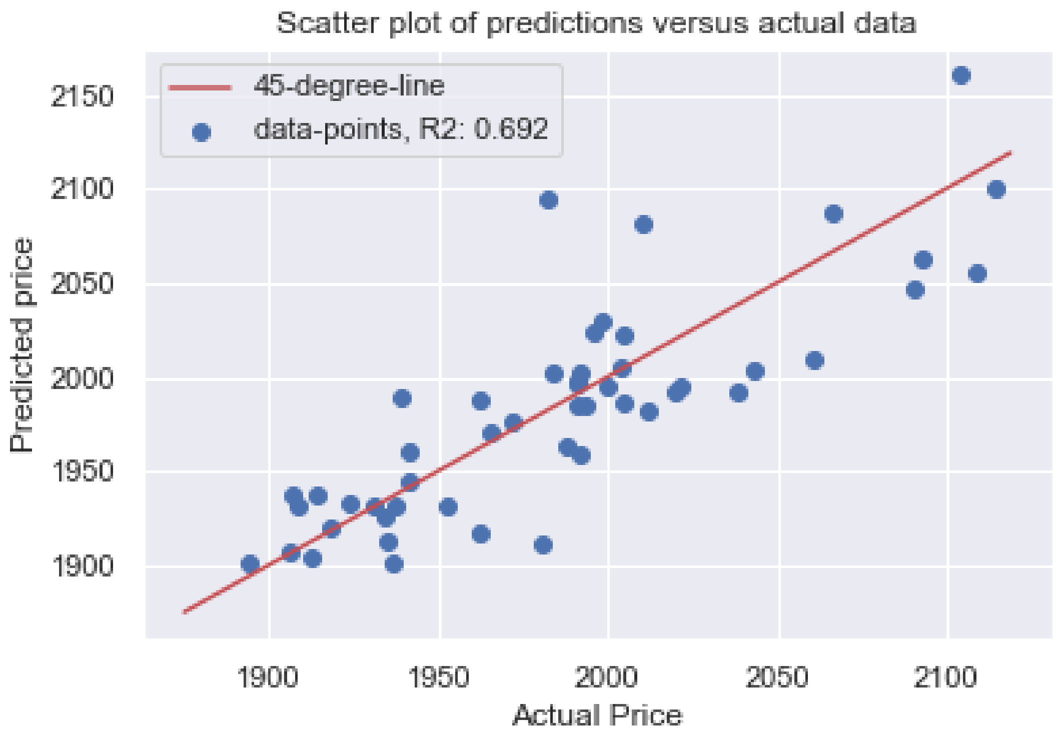


Figure 28: Scatter-plot of predictions versus actual data

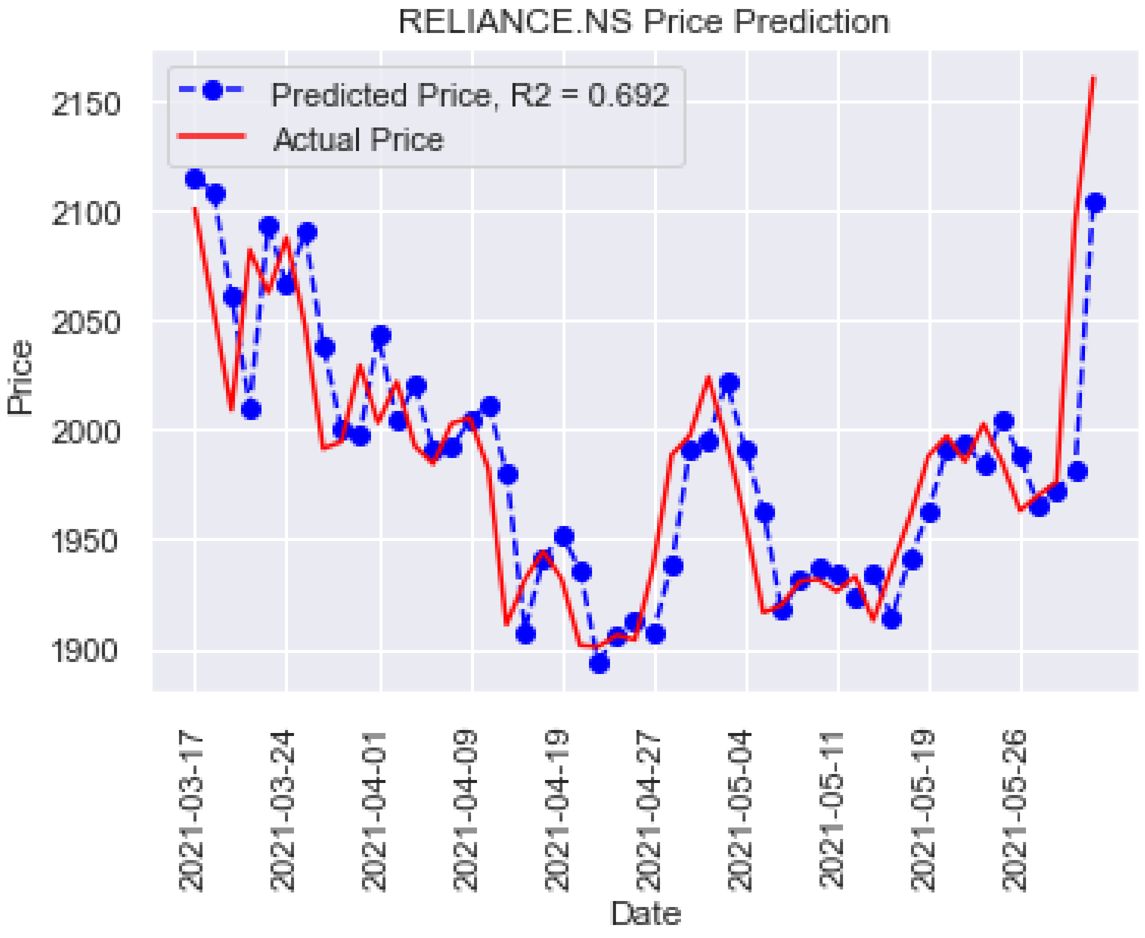


Figure 29: Forecasting using Optimal Parameter

Table 2: **Error, R2 & IC values**

MSE	1114.78
MAPE	1.25
CDP	63.27
R2	0.692
AIC	2524.04
BIC	2548.64

7.8 Trading Strategy

A simple yet effective strategy has been implemented and tested on Reliance Industries daily "Close" price data. Conditions for buy and sell, along with quantities has been listed below:

- Buy when next instance predicted price is at least 2% greater than today's closing price using one-fifth of capital.
- Sell one-third of stocks when the next instance predicted price is at least 2% lower than today's closing price.
- By next instance, next trading day closing price is implied.

Buy and sell signals received from the above strategy are plotted and attached below:

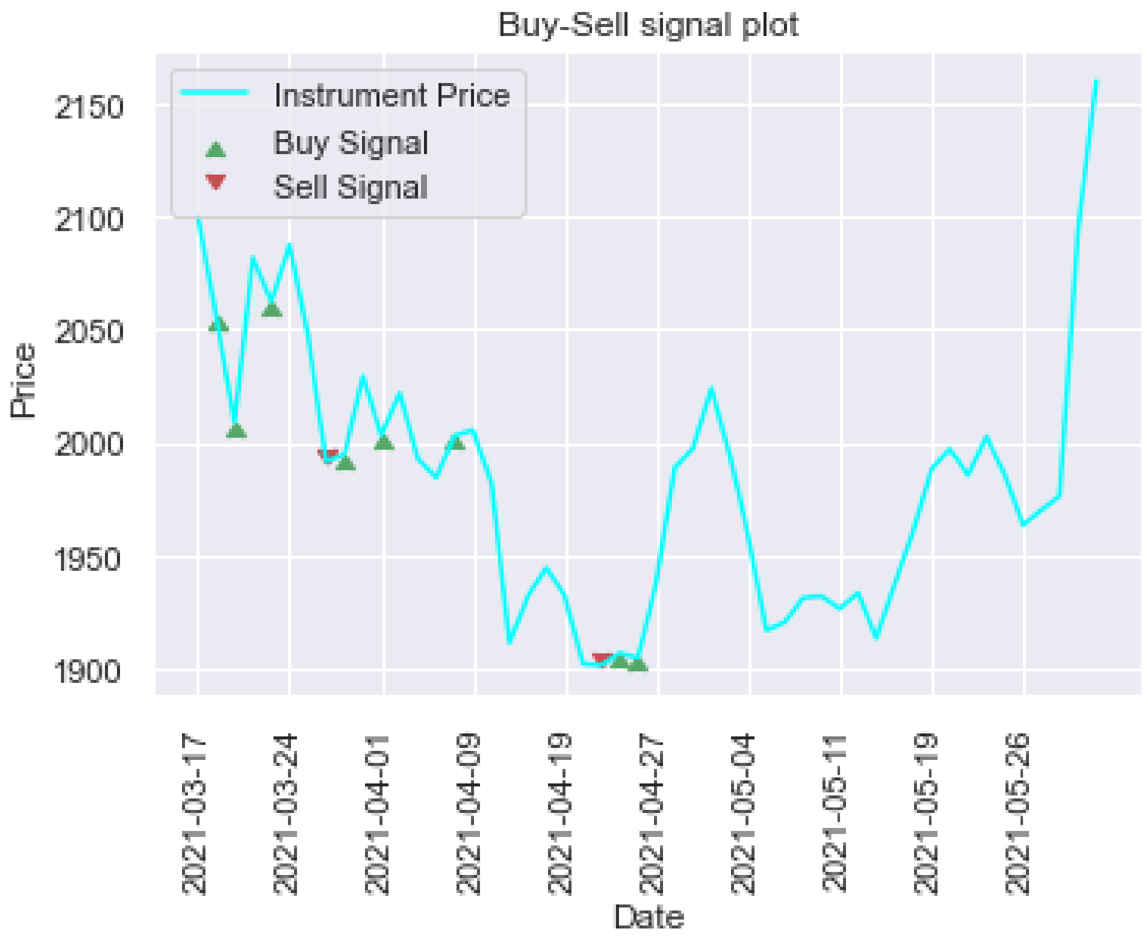


Figure 30: Buy-Sell Signals

Green upward arrow represents buy signal and red downward arrow represents sell signal. Total number of trades executed during the interval = 10.

7.9 Return Analysis

$$initial\ capital = 1000000 \tag{15}$$

At the end of trading interval considered:

$$Portfolio\,Value = cash\ +\ value\,of\ stocks = 1059600 \tag{16}$$

$$Portfolio\,Return = \frac{(1059600 - 1000000)}{1000000} * 100 = 5.96\% \tag{17}$$

$$\begin{aligned} Idle\ Return &= \frac{cash\ + value\,of\ stocks\ held\,passively)}{initial\ capital} * 100 \\ &= \frac{cash\ + value\,of\ stocks\ held\,passively)}{initial\ capital} * 100 = 2.84\% \end{aligned} \tag{18}$$

Portfolio return is defined as:

$$Portfolio\,Return(t) = \frac{cash\ +\ current\,value\,of\ stocks\ held}{Capital\,invested} * 100 \tag{19}$$

Idle return is defined as:

$$Idle\ Return(t) = \frac{cash\ +\ current\,value\,of\ stocks\,purchased\,initially}{Capital\,invested} * 100 \tag{20}$$

As can be seen from the above calculation and graph, we have portfolio return value: 5.96 % & idle return of 2.84 % in 50 trading sessions/days, this translates to (5.96*5) = 29.8 % p.a. portfolio return(considering 250 trading trading as markets generally do not open on Saturdays and Sundays) & 14.2 % p.a. idle return annually.

The above described return figures are the values that would have been realised had there not been any transaction charges for the exchange of these financial instruments. Accounting for the transaction charges and other duties that are generally being levied on such transactions(generally a flat fee of Rs. 20 for each transaction), the adjusted return figures would come out to be something like:

$$\begin{aligned} adjusted\,portfolio\,value &= 1059600 - (No.\,of\,trades + 1) * 20 \\ &= 1059600 - 11 * 20 \\ &= 1059380 \end{aligned} \tag{21}$$

+1 because trader will be supposed to sell the holdings at the end of analysis period to receive cash.

So, after accounting for transaction and brokerage charges, the return figures will look like

$$Adjusted\,Portfolio\,Return = \frac{(1059380 - 1000000)}{1000000} * 100 = 5.938\% \tag{22}$$

$$\begin{aligned} Adjusted\,Idle\,Return &= \frac{\frac{102.84}{100} * 1000000 - (1+1) * 20}{1000000} * 100 \\ &= 2.836\% \end{aligned} \tag{23}$$

The above figures, when translated to annually:

- Annual adjusted portfolio return = 5.938 * 5 = 29.69 % p.a.
- Annual adjusted idle return = 2.836 * 5 = 14.18 % p.a.

Ratio of the portfolio and idle return:

$$r = \frac{Portfolio\,Return}{Idle\,Return} = \frac{29.8}{14.2} = 2.1 \tag{24}$$

$$r_c = \frac{Adjusted\,Portfolio\,Return}{Adjusted\,Idle\,Return} = \frac{29.69}{14.18} = 2.09 \tag{25}$$

So, we can say that the strategy implemented is realising more than twice return than would have been realised with idle/passive investing. However, this in no way means that such high returns would always be realised if someone invests their capital on this strategy, but on an average this strategy is expected to beat passive investing returns more often than not as the strategy seems to execute transactions when the price point actually seem reversing.

8 Conclusion

Detailed analysis of Reliance Stock "Close" price data has been performed to forecast future values and build a trading strategy around it. To optimally fit the hyperparameters of the ARIMA model; namely p, d q values various plots and frameworks have been used. For stationarity test, visual and ADF test has been performed, which led us to use optimal d value of 1, i.e. 1st order of differencing. To estimate the moving average parameter, q autocorrelation plot has been used where which hints is that the value of q to be used for optimal forecasting if used will be quite large and hence we should use q value of 0, i.e. MA(0) and love ahead to find the optimal value of parameter p. To get this optimal value of autoregressive terms that should be used, partial autocorrelation plot has been exploited where we seen that excatly 5 of the lags had partial autocorrelation value greater than significance level(SL), and hence directing to use p value of 5, i.e. AR(5). So, these tests and plots concluded that ARIMA(5,1,0) should be used to optimally forecast the price value of the stock price.

We then cross-validated the performance of (p,d,q) value of (5,1,0) against other possible local minima and found out that this particular combination indeed minimises MSE & MAPE and maximizes CDP(Correct Direction Percentage), capturing the trend 63.27 % time correctly. However, AIC & BIC plots recommend us to use lower number of regressed terms but that comes at a cost of significant decrease in CDP value and hence We finalised (p,d,q) value of (5,1,0).

The prediction made with (p,d,q) value of (5,1,0) helped us achieve MSE of 1114.78, MAPE of 1.25, CDP of 63.27 and R2 value of 0.692. At first glance this R2 value of 0.692 seems too less compared to R2 that we had seen in correlation plot of train data with its lags, in excess of 0.95 but the test data shows best R2 value of 0.65 from its closest lag, so the model does improve R2 value from 0.65 to 0.692. We could have expected higher R2 value on predicted data had the data been not that random(R2 value of 0.65 initially), should have been tough for the model to actually study the trend and seasonal component, as the price value has been quite volatile during the period(test period) based on the 2nd wave of covid dominating the market. We can conclude that the model actually does a good job on forecasting price on solely analyzing price aspect of the stock, which in general is affected by a variety of numerous reasons ranging from mere market sentiments to news(earnings

announcement, lawsuits, M&As, potential deals, R&D outbreaks), GDP growth, Bond yields, FIIs investments, government policy, market analysts prediction going off the mark, etc.
In the final phase of the project, we built a simple yet efficient trading strategy utilising the trained ARIMA(5,1,0) model where returns in excess of 29 % p.a. could have been expected and this value of return would have been more than double compared to passive investing.
As part of further improvement to this project:

1. Concepts of batch training can be incorporated. Data can be divided into smaller sets say of 50 each(instead of 200 used in training set in one go) and then an individual ARIMA model can be fitted for each of those intervals and then some ways of combining these individual parameters to optimally predict the values on testing set could be tried and tested out.

2. Trading strategy can be played around with, the 2% cut-off value selected for buy or sell decision can be adjusted based on different trends and patterns.

3. The quantity to buy and sell on occurence of each favourable instances can be played around with as well. Say, buying with one one-tenth of capital and selling one forth of stocks, and many more combinations can be incorporated to test how they affect the overall bottomline, return figure.

Code files, papers and ppt directory

References

[1] R. K. Si, S. K. Padhan, and D. B. Bishi, “Application of Box – Jenkins ARIMA (p, d, q) Model for Stock Price Forecasting and Detect Trend of SP BSE Stock Index: An Evidence from Bombay Stock Exchange,” *Scholars Journal of Physics, Mathematics and Statistics*, vol. 7, no. 7, pp. 110–125, 2020.

[2] M. Deb and P. Saha, “Arima model in forecasting share prices,” vol. 5, no. 4, pp. 147–151, 2020.

[3] B. K. Meher, I. T. Hawaldar, C. Spulbar, and R. Birau, “Forecasting stock market prices using mixed ARIMA model: A case study of Indian pharmaceutical companies,” *Investment Management and Financial Innovations*, vol. 18, no. 1, pp. 42–54, 2021.

[4] J. Christy Jackson, J. Prassanna, M. Abdul Quadir, and V. Sivakumar, “Stock market analysis and prediction using time series analysis,” *Materials Today: Proceedings*, no. xxxx, 2021. [Online]. Available: https://doi.org/10.1016/j.matpr.2020.11.364

[5] B. Dhyani, M. Kumar, P. Verma, and A. Jain, “Stock Market Forecasting Technique using Arima Model,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 6, pp. 2694–2697, 2020.

[6] E. Ponomarev, I. Oseledets, and A. Cichocki, “Using reinforcement learning in the algorithmic trading problem,” *Journal of Communications Technology and Electronics*, vol. 64, no. 12, pp. 1450–1457, 2019.

[7] T. Theate and D. Ernst, “An aplication of deep reinforcement learning to algorithmic trading,” 2020.

[8] C. Y. Huang, “Financial trading as a game: A deep reinforcement learning approach,” *arXiv*, pp. 1–15, 2018.

[9] Z. Tan, C. Quek, and P. Y. Cheng, “Stock trading with cycles: A financial application of ANFIS and reinforcement learning,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 4741–4755, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.09.001

[10] C. Evans, K. Pappas, and F. Xhafa, “Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation,” *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1249–1266, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.mcm.2013.02.002

[11] B. Hirchoua, B. Ouhbi, and B. Frikh, “Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy,” *Expert Systems with Applications*, vol. 170, no. August 2020, p. 114553, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.114553

[12] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019,” *Applied Soft Computing Journal*, vol. 90, p. 106181, 2020. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106181

[13] Z. Zhang, S. Zohren, and S. Roberts, “Deep reinforcement learning for trading,” *arXiv*, vol. i, no. Chan 2009, 2019.

[14] S. Colianni, S. Rosales, and M. Signorotti, “Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis,” *CS229 Project*, pp. 1–5, 2015. [Online]. Available: http://cs229.stanford.edu/proj2015/029_report.pdf

[15] A. Nan, A. Perumal, and O. R. Zaiane, “Sentiment and Knowledge Based Algorithmic Trading with Deep Reinforcement Learning,” *arXiv*, pp. 1–13, 2020.

[16] O. B. Sezer, A. M. Ozbayoglu, and E. Dogdu, “An artificial neural network-based stock trading system using technical analysis and big data framework,” *Proceedings of the SouthEast Conference, ACMSE 2017*, no. 2, pp. 223–226, 2017.

[17] F. Giacomel, R. Galante, and A. Pereira, “An algorithmic trading agent based on a neural network ensemble: A case of study in North American and Brazilian stock markets,” *Proceedings - 2015 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015*, vol. 2, pp. 230–233, 2016.

[18] Y. Li, W. Zheng, and Z. Zheng, “Deep Robust Reinforcement Learning for Practical Algorithmic Trading,” *IEEE Access*, vol. 7, pp. 108 014–108 021, 2019.

[19] F. Rundo, F. Trenta, A. L. Di Stallo, and S. Battiato, “Advanced Markov-based machine learning framework for making adaptive trading system,” *Computation*, vol. 7, no. 1, 2019.

[20] Y. Y. Chen, C. T. Chen, C. Y. Sang, Y. C. Yang, and S. H. Huang, “Adversarial attacks against reinforcement learning-based portfolio management strategy,” *IEEE Access*, vol. 9, pp. 50 667–50 685, 2021.