STOCK MARKET TREND PREDICTIONS AND ANALYSIS USING DEEP LEARNING MODELS AND VISUAL REPRESENTATION

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science Engineering

by

SANCHITA SURYAVANSHI

18BCE0506

KAREN PINTO

18BCE0596

Under the guidance of

Dr. Ramani S

School of Computer Science and Engineering

VIT, Vellore.

June, 2022

DECLARATION

I hereby declare that the thesis entitled "Stock Market Trend Predictions and Analysis Using Deep Learning Models and Visual Representations" submitted by me, for the award of the degree of *Bachelor of Technology in Computer Science and engineering* to VIT is a record of bonafide work carried out by me under the supervision of Dr. Ramani S.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore
Date : 3rd June 2022

Signature of the Candidate
Sanchita Suryavanshi (18BCE0506)
Karen Pinto (18BCE0596)

CERTIFICATE

This is to certify that the thesis entitled "Stock Market Trend Predictions and Analysis Using Deep Learning Models and Visual Representations" submitted by **Sanchita Suryavanshi (18BCE0506) and Karen Pinto (18BCE0596)**, **School of Computer Science and Engineering**, VIT University, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by them under my supervision during the period, 01.01.2022 to 30.05.2022, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date : 3rd June 2022

Signature of the Guide

Internal Examiner

External Examiner

Dr. Vairamuthu S

Head of the Department

School of Computer Science and Engineering

ACKNOWLEDGEMENTS

We are deeply grateful to our course faculty, Dr. Ramani S, without whose constant guidance throughout the semester, it would have been impossible for us to successfully complete this project. His valuable support and supervision are responsible for attaining its present form.

We would like to express our gratitude to Dr. G. Viswanathan, Dr. Ramesh Babu K, Dr. Vairamuthu S for providing us with an environment to work in. In the jubilant mood, we express ingeniously our whole-hearted thanks to School of Computer Science and Engineering (SCOPE), all teaching staff and members working as limbs of our university for their enthusiasm coupled with timely encouragements showered on us with zeal, which prompted the acquirement of the requisite knowledge.

We would like to thank our parents for their support and express our gratitude and appreciation to all those who have helped us directly or indirectly towards the successful completion of this project.

Student Name
Sanchita Suryavanshi (18BCE0506)
Karen Pinto (18BCE0596)

Executive Summary

Nowadays, quite a few stock market investors wind up losing their cash due absence of comprehension of the development of the market and due to lack of exposure. It is impossible to keep track of the performance of stocks and shares every minute making it difficult for new traders to start investing. Hence, this project plans to solve this issue by implementing various deep learning models and comparing them to determine which algorithm would suit perfectly. Algorithms such as LSTM, RNN & GRU along with their bi-directional versions will be used for prediction of stocks, along with Sentimental Analysis to understand the financial status of a company. Some agents which are trading strategies will also be used to help investors on when to buy or sell a unit within a duration of time. It will take input from the predictions made by the deep learning models and for those predictions, investment strategies will be planned. Monte-Carlo and Joint Plot Visualization are used to depict the performances of various stocks.

CONTENTS

List of Figures

List of Tables

List of Abbreviations

LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
MA	Moving Average

1. INTRODUCTION

1.1. OBJECTIVE

Our project is working on the principles of macroeconomics and also on some parameters of Deep Learning. Some commonly used terminology will be used in our project which is used in the stock market.

- The project consists of a mix of three models which are :
  1) RNN
  2) GRU & BI-DIRECTIONAL GRU
  3) LSTM & BI-DIRECTIONAL LSTM
- The comparison of accuracy of these models will tell us which is suitable for the given dataset.
- The aim of this project is to use agents which will help users to predict stock trends and guide users by notifying when to sell and buy a particular stock.
- To make it easier for a layman to understand, this project also has data visualization models like Monte Carlo Simulation and Joint Plot Visualization.

1.2 MOTIVATION

Maintaining stocks in the stock market is equally challenging and it is impossible to have direct and accurate market updates every second as it takes a second for the market to fluctuate. The reason for choosing this topic is the demand as most of the people wish to generate passive income by investing some parts of their active income in share markets. The regular traders who are not very serious, keep losing money on a daily basis in the market which discourages them, eventually quitting the trading. Hence, for building the confidence in people we need to have a certain system which can predict the stock trend so that everyone using the system can have a general idea of the stock market trend and plan their investment strategy accordingly in advance. Therefore, here we are with a solution for regular traders to understand the market movement and predict a certainly good outcome. This project aims to provide such people the platform on how they should approach investing in stocks.

1.3 BACKGROUND

Stock market trend prediction is a very old issue as the dynamics of the stock market keep changing every minute. But now, as many people are showing interest and are willing to invest, they are not able to do so due to their lack of experience.

As we know the stock market has two very important objectives. The first is to provide funding; It gives us an idea about a company's revenue and turnover. Another purpose of investing in stocks is to give investors, those who buy stocks, an opportunity to share the profits of the companies they invested in.

Through this project we are willing to create an exposure for the new investors in this field. Making accurate forecasts for stock prices will not only help the users to understand the stock market trend but will also help the investors to pre-plan investment strategy which will lead to additional profits that investors can make.

2. PROJECT DESCRIPTION AND GOALS

This project uses deep learning models to predict the future outcomes of the stock market, by analyzing and finding the pattern in the data.We will be importing datasets from Yahoo Finance which is one of the most authentic websites with regard to stocks.

The project is composed of four components:

1. Stock Prediction using Deep learning Models
2. Sentiment Analysis on Stock Market News
3. Stock investment Strategy Planning Agents
4. Stock Market Data Visualizations

Here each component will have different models and algorithms which will be evaluating several different tasks and will have separate complexities according to their use cases. We have used 5 deep learning models for training the dataset which are as follows: Basic RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), Bi-directional LSTM and Bi-directional GRU. The second component of the project performs Sentiment Analysis on Stock Market news which will give investors an idea whether a particular company is positively or negatively trending. In the third component of our project we have used popular stock market strategies.

Based on these strategies we developed codes which will help to predict and plan the investment profile for any particular company which we want. Based on the outputs of deep learning models (GRU) we can pre-plan when, where and on which day investors should invest in a particular company.

Apart from training models we also showed the Monte-Carlo simulations of TESLA stock in which we determine the returns and volatility every day. The Monte-Carlo results are explained with the help of graphs. Further, a jointplot simulation of 11 stocks data in reg mode is conducted which draws the line of best fit of linear regression that becomes the defining parameter of whether a given stock data should be bought or sold. With the help of these methods adopted it will be very convenient for people investing in the stock market to carefully take calls during buying and selling of stocks.

3. TECHNICAL SPECIFICATION

3.1 Hardware Requirements

- Working Desktop
- Wifi Routers/Net connectivity for API calls
- Laptop with an Intel Core i7 or i9 processor or an AMD Ryden processor to run complex and time consuming machine learning codes.

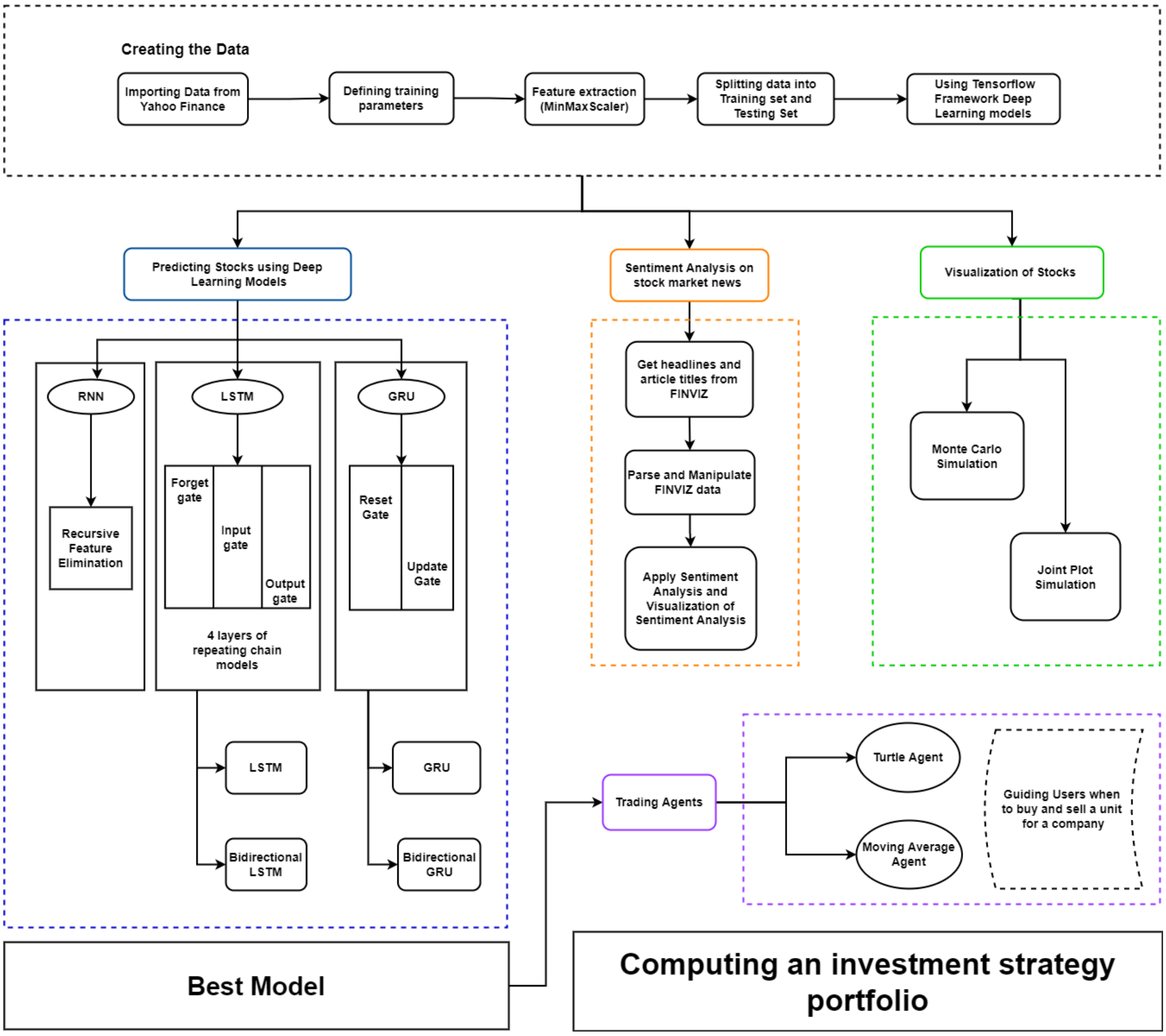
3.2 Software Requirements

- Python/Jupyter Notebook
- Packages such as nltk, tensorflow, pandas, seaborn, matplotlib
- Valid Datasets from Yahoo Finance which includes the date and required trading factors needed for the Deep Learning models.
- Valid API requests are made to finviz.com to acquire company-specific news articles used for Sentimental Analysis.

4. DESIGN APPROACH AND DETAILS

4.1 DESIGN APPROACH

Fig 4.1.1 Architectural Diagram



4.2 CODES AND STANDARDS

4.2.1 Deep Learning Models

- **Datasets**

Google Historical Stock Prices in USD

Time Period: 2years (3years for LSTM)

Imported from yfinance library

Fig 4.2.1.1 Google Historical Stock Records

	Date	Open	High	Low	Close	Adj Close	Volume
753	2022-03-21	2736.949951	2751.649902	2692.229980	2729.570068	2729.570068	1331600
754	2022-03-22	2730.000000	2830.000000	2730.000000	2805.550049	2805.550049	1488800
755	2022-03-23	2782.770020	2800.500000	2763.330078	2770.070068	2770.070068	1265100
756	2022-03-24	2785.449951	2827.929932	2760.788086	2826.239990	2826.239990	1027200
757	2022-03-25	2835.080078	2839.189941	2793.989990	2830.429932	2830.429932	963500

- **Model Creation**

The deep learning models were imported from *keras* and *keras.layers.*
Each Deep Learning Model included an Input Layer and an Output Layer. Additional Hidden layers were used to help model the complex data. We gave different Sequential models a different number of hidden layers ranging from 0 to 2 based on its learning and accurate predictions.

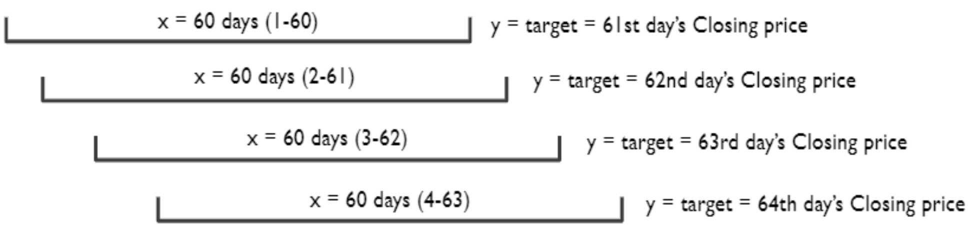
- **Scaling the data**

Some of the values have large numbers and are quite skewed when plotting a regression. Therefore the data is scaled to a specific range suitable to perform computations.
Library used: MinMaxScaler() from sklearn.preprocessing
MinMaxScaler normalizes the data in the range [0,1]

- **Training Dataset**

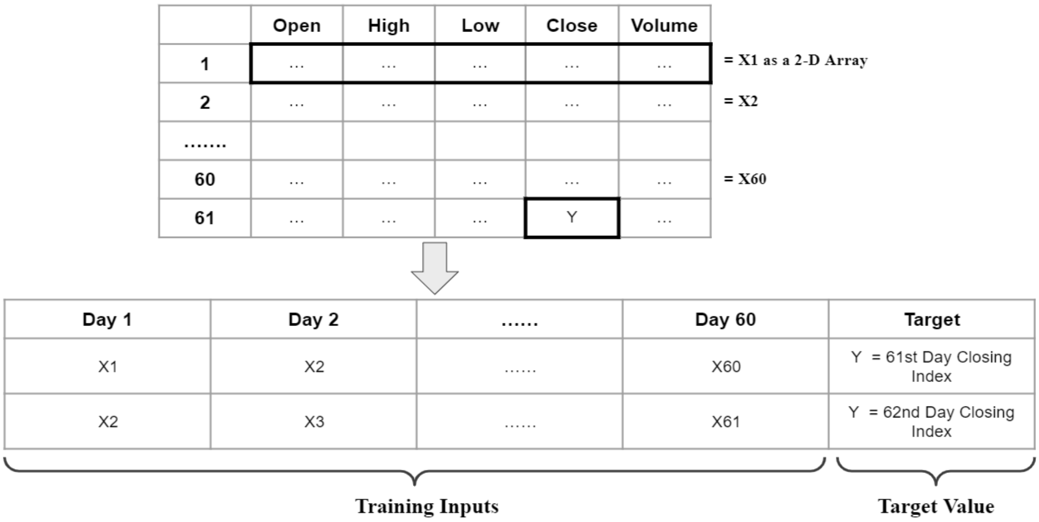
Here, we are dividing and creating a dataset of 60 columns each. Basically our input values will include the Open, High, Low, Close and Volume prices from the dataset of 60 days in order to **predict the target closing index of the 61st day.**

Fig 4.2.1.2 Depicting the creation of the inputs to the model



The dimensions of our inputs will be **[60, 60, 5]** as shown below -

Fig 4.2.1.3 Dimensionality of the training inputs



- **Accuracy**

Two types of accuracy/error computational formulas were used. Both are solely based on the mathematical distance of the predicted value from its real value.

➔ **Mean Absolute Percentage Error (MAPE)**

accuracy = (1 - np.mean(np.abs((real-predict)/real))) * 100

➔ **Root Mean Squared Error (RMSE)**

accuracy = (1 - np.sqrt(np.mean(np.square((real-predict)/real)))) * 100

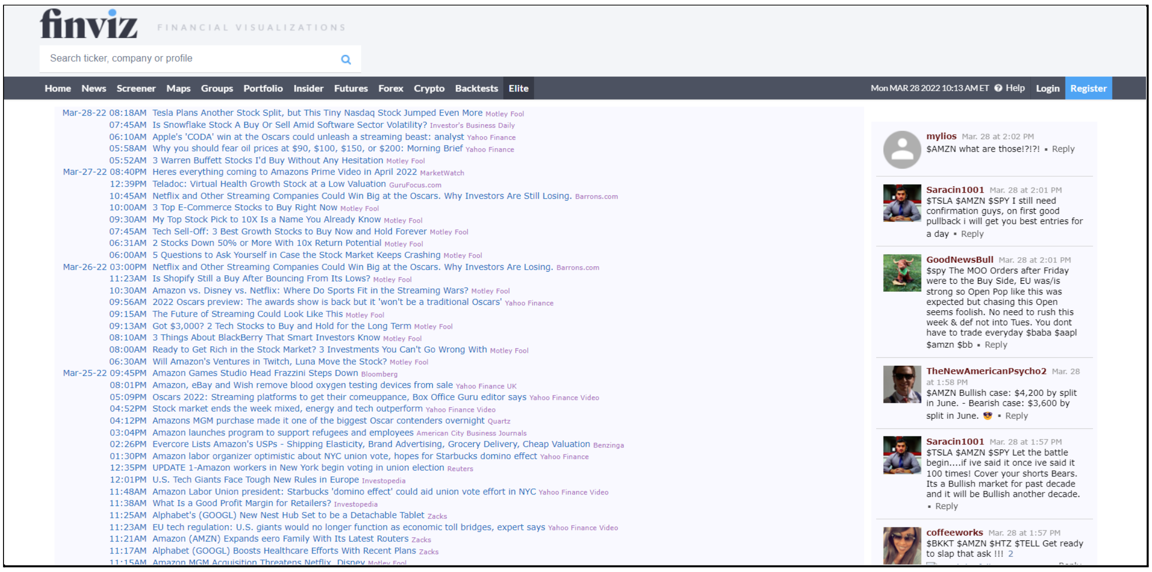
- **Visualization the Prediction Trend**
    - Output: Closing prices of Google for the last 60 days
    - Library used to plot the graph: matplotlib
    - Forecasts: 3 simulations (Blue, Orange and Green)
    - True Trend: Prediction line in black

4.2.2 Sentiment Analysis

We are using Sentiment Analysis to understand financial news and make decisions on stock. In this we used Python to parse through finviz.com, gather all the news article titles and then we do sentiment analysis to understand if everyday averaging news is positive, negative or neutral.

Fig 4.2.2.1 Finviz Website

Fig 4.2.2.2 Finviz News Articles



To perform Sentiment Analysis-

1. We used Beautifulsoup in Python to scrape article headlines from FinViz
2. Then, we'll use Pandas (Python Data Analysis Library) to analyze and run sentiment analysis on the article headlines
3. Finally, we'll use Matplotlib for visualization of our results

We will be doing this in 3 steps -

a. Collecting Data and Parsing FinViz Data

FinViz (finviz.com) is a free financial news website that makes stock data easily accessible to traders and investors. We'll gather the stock data from FinViz for a specific stock ticker.

```
fnvz_url = 'https://finviz.com/quote.ashx?t='
ticker = [GOOG, AMZN, 'FB']
```

Here we are getting the article title/headline from finviz then we are storing the article title along with the timestamp in a dictionary. Then we parse through the table contents which are made from the dictionary and select article titles and we split the timestamp into date and time components. At the end of this step we have scraped the components which are needed for Sentiment Analysis ie. tickers, date, time and article title.

b. Applying Sentiment Analysis

Applying sentiment analysis on the titles is done with the NLTK (Natural Language Toolkit) vader that allows us to pass in a string into its function. We initialize the SentimentIntensityAnalyzer, and then a lambda function is created which takes the “title” as a an input string, then apply the vader.polarity_scores() function on it inorder to get the results and that function gives us the compound score which we use for the sentiment analysis. Using a function, a new 'compound' column is created and added in the data frame wherein all the computed compound scores from each news title are stored along with the title.

c. Visualizing the Results in MatPlotLib

We visualize the data frame in MatPlotLib to see how our Stocks fared every day from public perception in news articles. For that we group our dataset based on the ticker and dates of each row, and then visualize the average compound score of each day.

4.2.3 Agents

These agents will help sketch a stock trend and will also develop an investment plan for a particular company. The two agents as we call them are based on popular techniques/ stock market strategies which are tremendously used. The two agents are Turtle Agent which is based on Turtle Trading Strategy and Moving Average Agent which uses Moving average indicator. The output of the deep learning models (specifically GRU as it is giving the most accurate result) will be used as input for our agents, this will let it predict and plan the investment strategy which the user needs to follow to maximize profit.

a) Moving Average Agent

Moving average (MA) in the stock market is basically a stock indicator, it is a technical tool that helps in analysis which smooths out cost data by making a continually refreshed typical cost. A moving normal (MA) is a generally utilized specialized indicator, here we will be using the moving average to determine points when one should buy / sell stocks.

Key Points -

- Indication to sell or buy a stock/asset is based on the calculated estimation of MA. At the point when resource costs get over their moving midpoints, it might produce an exchange signal for specialized brokers.
- Moving average crossovers are a well known procedure for both entries and exits. MAs can likewise feature areas of possible help or opposition.
- It's assumed that when the price is above a MA, then the trend is considered to be up. If the price is below a MA, the trend is known to be down.
- Whenever the short-term moving avg crosses over the long-term moving avg, it's supposed to be a purchase signal, because the way the pattern moves is considered to be moving up. It is known to be a golden cross.
- In the interim, the time when the short-term moving avg crosses underneath the long-term moving avg, that depicts a sell signal, as that movement of the pattern means the trend is moving down and that is known as a dead/passing cross.

Calculating Total gain and total Investment percentage to determine profit or loss for comparison with the other agent.

```
invst = ((in_money - start_money) / start_money) * 100
tot_gain = in_money - start_money
return buy_state, sell_state, tot_gain, invst
```

b) Turtle Agent

The turtle trading strategy/ methodology is a well-known pattern following the procedure that brokers use to profit from the supported force in the exchanging market. Utilized in a large group of monetary business sectors, dealers utilizing this methodology search for breakouts, to potential gain and disadvantage.

1. The exchanging markets rule
2. The position-measuring rule
3. The passages rule(Entries)
4. The stop-misfortune rule
5. The ways out rule
6. The strategies rule

Key Points -

- The Turtle Trading framework is one of the most popular pattern following systems. Turtle Trading depends on buying a stock or agreement during a breakout and rapidly selling on a retracement or cost fall.
- The thought is just the "pattern is your companion", so you ought to purchase potential gain of exchanging reaches and undercut drawback breakouts.
- It uses the typical genuine reach to work out unpredictability , and utilizes it to change your position size. It Takes bigger situations in less unstable business sectors and reduces your openness to the most unpredictable business sectors.
- One key illustration is that a framework is vital; without an obviously characterized set of boundaries for sections, exits, position-estimating and stopping misfortunes, a dealer is simply utilizing his intuition.
- The turtle exchanging test shows that a bunch of rules is fundamental, yet similarly indispensable is the outlook to observe those guidelines.

Calculating Total gain and total Investment percentage to determine profit or loss for comparison with the other agent.

```
invst = ((in_money - start_money) / start_money) * 100
tot_gain = in_money - start_money
return buy_state, sell_state, tot_gain, invst
```

4.2.4 Data Visualization Models

In this project we have used Monte Carlo Simulations and Jointplot regression mode visualization to depict the performances of various stocks. In both these visualizations we need to compute returns and volatilities. Returns are determined by the percentage of closing values of a particular day with respect to the previous day whereas Volatility is calculated by taking the standard deviation of all the closing values of the dataset.

- **Monte Carlo Simulation**

In Monte Carlo Simulations, we took the Tesla dataset and performed the simulation analysis. Monte Carlo Simulation is based on probability and its typical normal distribution. Here the idea is to choose a random number from a random number of intervals that are generated from probability which is determined by the likelihood of a particular category of event to occur and its cumulative probability distribution. This random number helps to draw logical conclusions thereby helping in decision making with a high amount of uncertainty. Since, Stock market is a game of uncertainty we feel choosing Monte Carlo simulations as the best possible chance to help predict the status of closing values in future.

For this we took the last closing value and tried to generate 30 random values based on this price along with the influence of volatility which is calculated with the help of standard deviation calculation as mentioned above. We generate 10 simulations and then store them in a multi-dimensional list. Then this multi-dimensional list is converted to a single dimensional list with the help of ravel function. We then compare this single predicted value of the list generated from epochs with the actual closing values and plot the curve.

- **Joint Plot Visualization**

For jointplot simulations we took 11 datasets of the companies namely, AMD (Advanced Micro Devices, Inc.), FB(Facebook), FSV (First Service Corporation), INFY(Infosys), JINDALSTEL.NS(Jindal Steel Pvt. Ltd.), KNX (Knight-Swift Transportation Holdings Inc.), MONDY (Mondi plc), MTDR (Matador Resources Company), TMUS (T-Mobile US, Inc.), TSLA(Tesla), TWTR(Twitter).

This reg mode draws the line of best fit which demarcates and compares the buying and selling of all the stocks feeded and helps us to take calls in future.

Fig 4.2.4.2 Computing Return and Volatile

```
from functools import reduce
data = reduce(lambda left,right: pd.merge(left,right,on='Date'), dfs).iloc[:, 1:]
data.head()
```

	Close_x	Close_y	Close_x	Close_y	Close_x	Close_y	Close_x	Close_y	Close_x	Close_y	Close
0	83.910004	329.510010	165.660004	18.459999	47.439999	56.680000	27.270000	454.250000	134.009995	677.000000	65.089996
1	81.620003	325.079987	162.250000	18.080000	47.119999	55.367001	26.309999	441.500000	132.130005	709.440002	55.220001
2	78.550003	322.579987	162.250000	18.200001	47.580002	55.674999	26.170000	445.299988	131.449997	684.900024	54.580002
3	78.610001	318.359985	159.910004	18.110001	48.500000	54.980000	27.020000	435.750000	128.479996	673.599976	54.400002
4	77.830002	315.019989	163.169998	18.290001	48.619999	56.165001	28.709999	439.399994	134.130005	670.940002	53.560001

```
returns = data.pct_change()
mean_daily_returns = returns.mean()
volatilities = returns.std()
```

```
mean_daily_returns * 252
```

We then computed returns and volatilities as mentioned above and then used jointplot function available in seaborn library of python for which we used regression mode-

```
g = sns.regplot("volatility", "returns", data=combine)
```

4.3 CONSTRAINTS, ALTERNATIVES AND TRADEOFFS

Constraints:

- To train the deep learning model, it necessarily requires at least 2 years of historical records of the company stocks.

- Although the models have shown to have great accuracy, it is limited to an unstable certainty, especially companies that often fluctuateThe trading strategy developed using the agents is dependent on the accuracy of the prediction trend from the deep learning model.
- The news articles retrieved from the Finviz website for conducting Sentiment Analysis are limited to the 100 most recent articles pertaining to a given company which could result in any duration of the past few days.

Alternatives:

- Due to the complexity of codes, the sentiment analyzer could be incorporated with the training model's code that would increase the accuracy and would also consider various other outliers.
- The comparative analysis done on deep learning models and agents were tested on a lot of companies from various levels, right from companies whose stocks are stable to companies with unstable stocks, for instance, the new companies or startups as well as the companies which have recently gained or lost ie. recent stock gainers and stock losers.
- We have proven that irrespective of different situations we get a common answer and have thereby showcased which algorithm/model and agent works best. For an even detailed overview, we could increase the number of companies to confirm the outcome of the acquired study.

Trade Offs:

- Model can be made more accurate with datasets of larger timestamps and trained with a higher epoch value. Unfortunately this increases the time taken while building the model.

5. SCHEDULE, TASKS AND MILESTONES

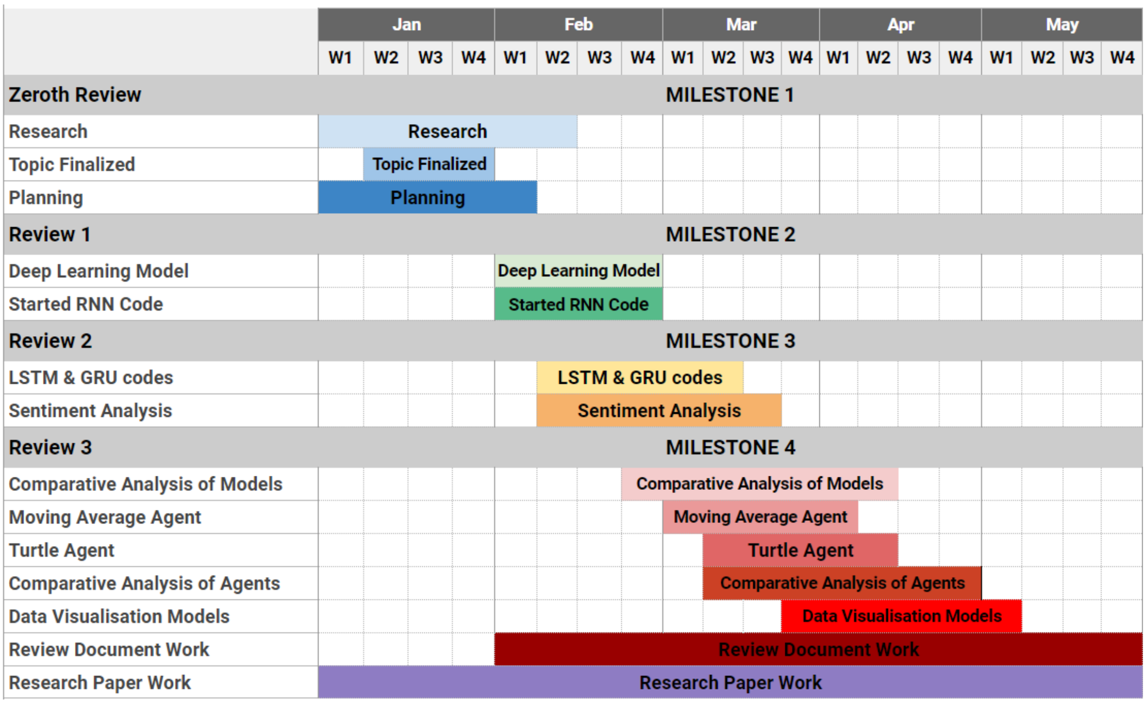


Table 5.1 Gantt chart displaying Tasks, Schedule and Milestones

6. PROJECT DEMONSTRATION

a) Dataset screenshots

Fig 6.1 Importing Datasets from yfinance

```
import yfinance as yf
stock_symbol = 'TSLA'
data = yf.download(tickers=stock_symbol,period='4y',interval='1d')
data = data.reset_index()
data.tail()
[7]  ✓  0.2s
```

```
[*********************100%***********************]  1 of 1 completed
```

	Date	Open	High	Low	Close	Adj Close	Volume
1002	2022-04-21	1074.729980	1092.219971	996.419983	1008.780029	1008.780029	35138800
1003	2022-04-22	1014.909973	1034.849976	994.000000	1005.049988	1005.049988	23181600
1004	2022-04-25	978.969971	1008.619995	975.299988	998.020020	998.020020	22780400
1005	2022-04-26	995.429993	1000.000000	875.000000	876.419983	876.419983	45377900
1006	2022-04-27	898.580017	918.000000	877.359985	881.510010	881.510010	25585300

b) Code simulation screenshots

Deep Learning Models

- Each Model has an Input and an Output layer along with 0-2 Hidden layers.
- Every layer is accompanied with a dropout layer to prevent overfitting.
- Models are built with ‘relu’ activation and are compiled using the ‘adam’ optimizer.
- Suitable epoch of 300 and a batch size of either 32 or 64 is used.

Fig 6.2 Simulation

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/300
16/16 [==============================] - 2s 56ms/step - loss: 0.1801 - accuracy: 0.0000e+00
Epoch 2/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0519 - accuracy: 0.0000e+00
Epoch 3/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0394 - accuracy: 0.0020
Epoch 4/300
16/16 [==============================] - 1s 56ms/step - loss: 0.0260 - accuracy: 0.0020
Epoch 5/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0248 - accuracy: 0.0020
Epoch 6/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0174 - accuracy: 0.0020
Epoch 7/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0208 - accuracy: 0.0020
Epoch 8/300
16/16 [==============================] - 1s 60ms/step - loss: 0.0177 - accuracy: 0.0000e+00
Epoch 9/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0180 - accuracy: 0.0020
Epoch 10/300
16/16 [==============================] - 1s 59ms/step - loss: 0.0186 - accuracy: 0.0020
Epoch 11/300
16/16 [==============================] - 1s 57ms/step - loss: 0.0173 - accuracy: 0.0020
```

c) Predictions and Analysis of the Deep Learning Models

- **Simple RNN:**

Here, the model used 3 layers, with a batch size of 64 at 300 epochs. In terms of error functions, MAPE gave a mean accuracy of 91.91% and RMSE gave an accuracy of 86.01%.

- **LSTM:**

Here, the model used 4 layers, with a batch size of 32 at 300 epochs. In terms of error functions, MAPE gave a mean accuracy of 92.90% and RMSE gave an accuracy of 89.73%.

- **GRU:**

Here, the model used 2 layers, with a batch size of 32 at 300 epochs. In terms of error functions, MAPE gave a mean accuracy of 93.09% and RMSE gave an accuracy of 91.39%.

- **Bidirectional LSTM:**

Here, the model used 4 layers, with a batch size of 32 at 300 epochs. In terms of error functions, MAPE gave a mean accuracy of 94.53% and RMSE gave an accuracy of 94.06%. Here, we used a timestamp of 30 days, as it gave a higher accurate prediction trend.

- **Bidirectional GRU:**

Here, the model used 3 layers, with a batch size of 32 at 300 epochs. In terms of error functions, MAPE gave a mean accuracy of 92.98% and RMSE gave an accuracy of 90.95%.

d) Sentimental Analysis

For Sentiment Analysis we have used python for scraping the finviz.com to gather the article titles then we used pandas to analyze and run nltk module for sentiment analysis and finally in the end we use matplotlib for visualization. In the output we get graphs where we have specified 3 companies - Amazon, Google and Facebook. These three companies are our tickers, we have gathered news articles of these companies and performed sentiment analysis on them. We later on visualized the output and we can see the correlation and average daily score (compound score) of how that stock of a company has performed based on the news.

We get the data frames as such which will have tickers, date, time, article titles related to stocks, compound scores, positive, negative and neutral scores.

Fig 6.3 Dataframe prepared for conducting Sentiment Analysis

```
Open in Notebook Editor
     ticker        date     time  \
0      AMZN  2022-03-28  08:18AM
1      AMZN  2022-03-28  07:45AM
2      AMZN  2022-03-28  06:10AM
3      AMZN  2022-03-28  05:58AM
4      AMZN  2022-03-28  05:52AM
..      ...         ...      ...
295      FB  2022-03-18  04:40PM
296      FB  2022-03-18  03:18PM
297      FB  2022-03-18  01:45PM
298      FB  2022-03-18  12:00PM
299      FB  2022-03-18  11:54AM
```

```
                                                 title  compound    pos  \
0    Tesla Plans Another Stock Split, but This Tiny...    0.0000  0.000
1    Is Snowflake Stock A Buy Or Sell Amid Software...    0.0000  0.000
2    Apple's 'CODA' win at the Oscars could unleash...    0.5859  0.275
3    Why you should fear oil prices at $90, $100, $...   -0.4939  0.000
4    3 Warren Buffett Stocks I'd Buy Without Any He...    0.2057  0.206
..                                                 ...       ...    ...
295  Russian copycats are stealing intellectual pro...   -0.1027  0.236
296  Stocks turn positive to end the week, led by m...    0.5574  0.265
297  McDonald's, Starbucks, and others have no reco...   -0.6597  0.000
298  Instagram users targeted for get-rich-quick in...   -0.5859  0.000
299  Instagram releases new parental supervisory to...    0.0000  0.000
```

```
       neg    neu  compund
0    0.000  0.000   0.0000
1    0.000  0.000   0.0000
2    0.275  0.275   0.5859
3    0.000  0.000  -0.4939
4    0.206  0.206   0.2057
..     ...    ...      ...
295  0.236  0.236  -0.1027
296  0.265  0.265   0.5574
297  0.000  0.000  -0.6597
298  0.000  0.000  -0.5859
299  0.000  0.000   0.0000
```

e) Moving Average Agent

This agent will sketch an investment plan which will have a day to day task of whether one should invest in a particular company's stock or no. (eg. Microsoft)

```
Buy_state, sell_state, tot_gain, invst = stock_buy(df.Close,
                                   signal['position'])
```

Figure 6.4 Strategic Portfolio of Buying and Selling condition in MA

```
day 24: buy 1 units at price 251.720001, total balance 9748.279999
day 32, sell 1 units at price 253.809998, investment 0.830286 %, total balance 10002.089997,
day 34: buy 1 units at price 253.589996, total balance 9748.500001
day 75, sell 1 units at price 289.459991, investment 14.144878 %, total balance 10037.959992,
day 76: buy 1 units at price 288.329987, total balance 9749.630005
day 95, sell 1 units at price 301.140015, investment 4.442836 %, total balance 10050.770020,
day 104: buy 1 units at price 299.869995, total balance 9750.900025
day 108, sell 1 units at price 299.559998, investment -0.103377 %, total balance 10050.460023,
day 119: buy 1 units at price 294.850006, total balance 9755.610017
day 154, sell 1 units at price 336.630005, investment 14.169916 %, total balance 10092.240022,
day 163: buy 1 units at price 342.540009, total balance 9749.700013
day 169, sell 1 units at price 319.910004, investment -6.606529 %, total balance 10069.610017,
day 174: buy 1 units at price 341.250000, total balance 9728.360017
day 180, sell 1 units at price 316.380005, investment -7.287911 %, total balance 10044.740022,
day 198: buy 1 units at price 308.760010, total balance 9735.980012
day 205, sell 1 units at price 302.380005, investment -2.066331 %, total balance 10038.360017,
day 218: buy 1 units at price 300.190002, total balance 9738.170015
day 222, sell 1 units at price 275.850006, investment -8.108197 %, total balance 10014.020021,
day 230: buy 1 units at price 300.429993, total balance 9713.590028
```

The investment plan will be on the basis that if we assume we have Rs 10,000 then on which day should we invest how much money in a particular company's stock and in the end how much profit/gain will we have. This plan also gives us an estimation of the investment strategy which we must follow to maximize our profit. It gives details like total net balance, total gain, total investment percentage etc. The user can compare the investment strategy for various companies and check out which company will give a maximum profit.

f) Turtle Agent

Similarly, the Turtle agent also gives us an investment plan.

Figure 6.5 Strategic Portfolio of Buying and Selling condition in Turtle

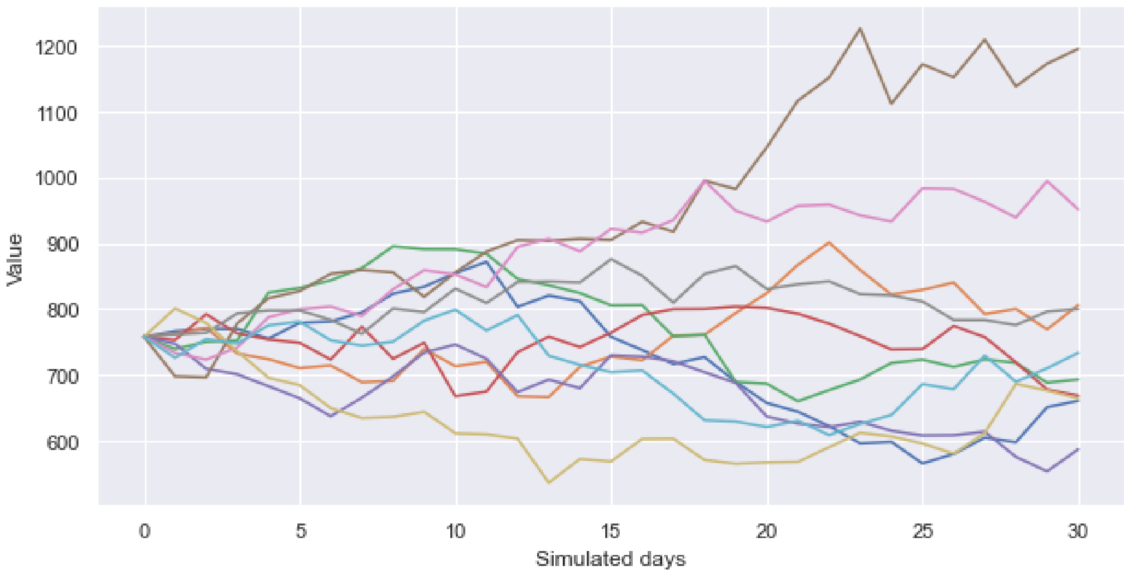
```
day 53: cannot sell anything, inventory 0
day 54: cannot sell anything, inventory 0
day 70: buy 1 units at price 3327.590088, total balance 6672.409912
day 77: buy 1 units at price 3320.679932, total balance 3351.729980
day 78: buy 1 units at price 3292.110107, total balance 59.619873
day 82: total balances 59.619873, not enough money to buy a unit price 3241.959961
day 83: total balances 59.619873, not enough money to buy a unit price 3201.219971
day 84: total balances 59.619873, not enough money to buy a unit price 3187.750000
day 96, sell 1 units at price 3509.290039, investment 10.086740 %, total balance 3568.909912,
day 97, sell 1 units at price 3525.500000, investment 10.595247 %, total balance 7094.409912,
day 113: buy 1 units at price 3285.040039, total balance 3809.369873
day 114: buy 1 units at price 3283.260010, total balance 526.109863
```

This agent uses the turtle strategy to draft the investment plan. It will give us an idea in advance about how one should plan the investment.

g) Monte Carlo Visualization

As we hear about the growth of Tesla Motors and how Elon Musk boosted the company but there may be loopholes so we thought of using this data to use for MonteCarlo Simulations which is useful in case of uncertainties as Tesla Motors. For this we are using the latest dataset ie. from April 2021 to April 2022 it has seen a rising boom as evident from results head and tail of the dataset despite pandemic being the biggest challenge for stock markets. This factor further motivated us to analyze using Monte-Carlo simulations.

Fig 6.6 10-simulations of Monte Carlo 30 day prediction



h) Joint Plot Visualization

We took a dataset of 11 companies and then computed returns and volatilities and then used a joint plot function available in seaborn library of python for which we used regression mode.

Fig 6.7 Computing average daily returns and volatiles

```
mean_daily_returns * 252

Close_x    -0.427263
Close_y     0.735194
Close_x     0.509847
Close_y    -0.063756
Close_x     0.324793
Close_y     0.280127
Close_x     2.122266
Close_y     1.539646
Close_x    -0.036006
Close_y    -0.524399
Close       0.426694
dtype: float64
```

```
volatilities * 252

Close_x     6.690635
Close_y     5.215759
Close_x     4.579849
Close_y     4.135928
Close_x     4.630070
Close_y     4.849706
Close_x    11.648769
Close_y     7.533855
Close_x     4.555902
Close_y    10.963129
Close       9.564248
dtype: float64
```

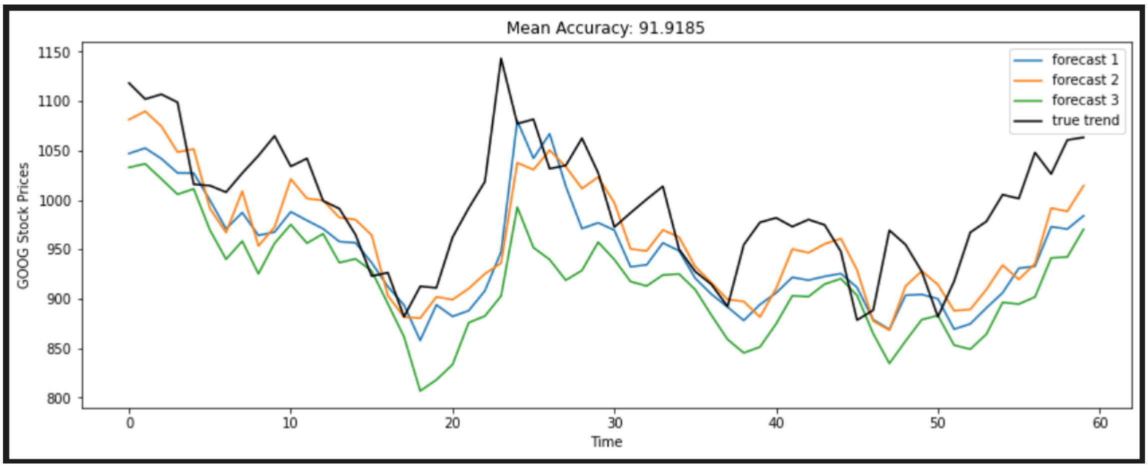
7. RESULT & DISCUSSION

a) Model Outputs

- **Simple RNN**

For testing the accuracy of the RNN model we took google's dataset and in the output below we ran 3 simulations and got 91.3% MAPE mean accuracy.

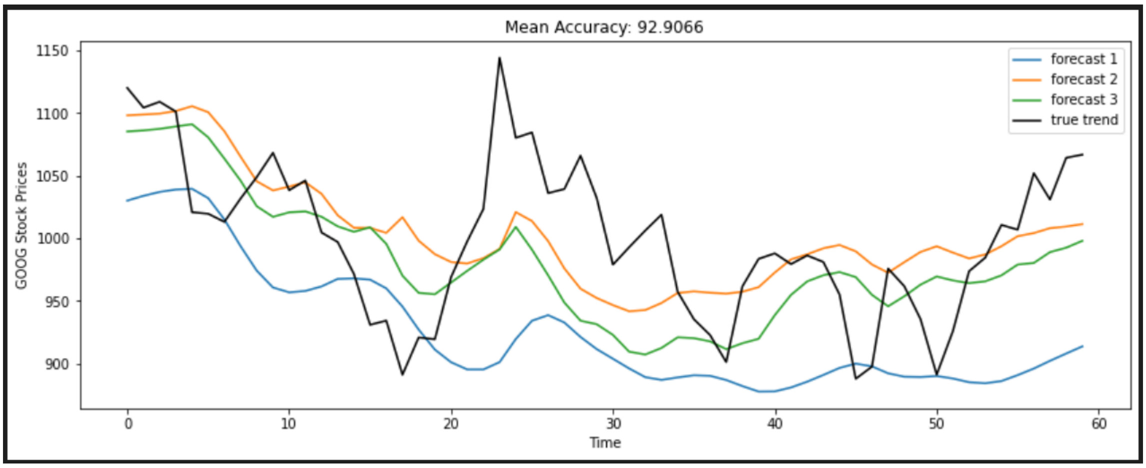
Fig 7.1. Simple RNN Model Predictions



- **LSTM**

Similarly in LSTM we ran 3 simulations and predicted the output for the stocks of google and got a MAPE mean accuracy of 92.9%.

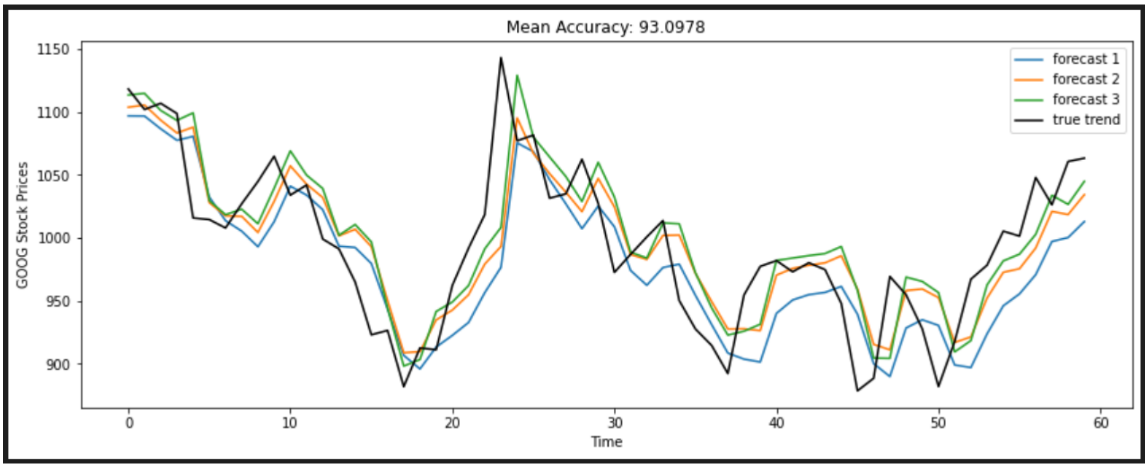
Fig 7.2. LSTM Model Predictions



- **GRU**

This model gives the best results for most of the companies. Here is a simulation for google and we got a MAPE mean accuracy of 93%.

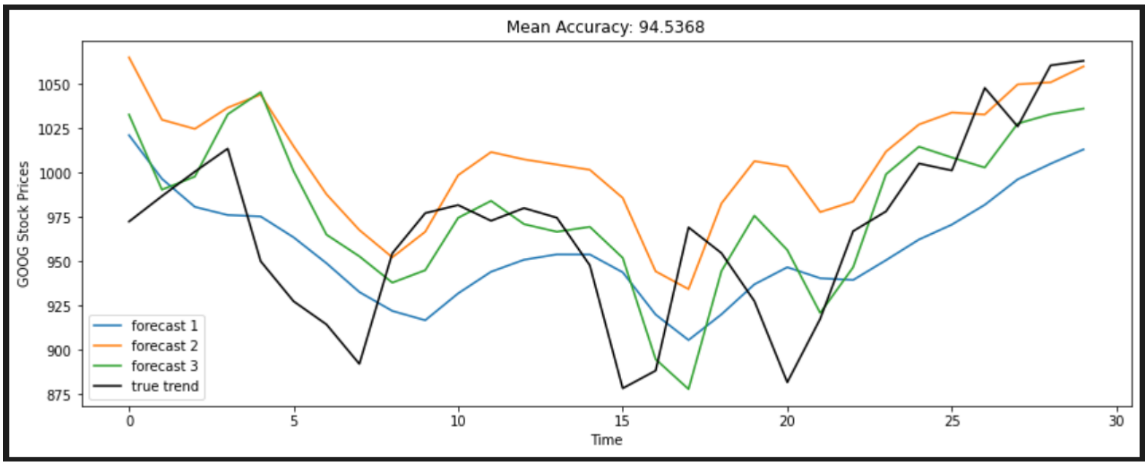
Fig 7.3. GRU Model Predictions



- **Bidirectional LSTM**

The bidirectional version of models are better than normal deep learning models , the Bidirectional LSTM is giving better results for google when predicted for 30 days.

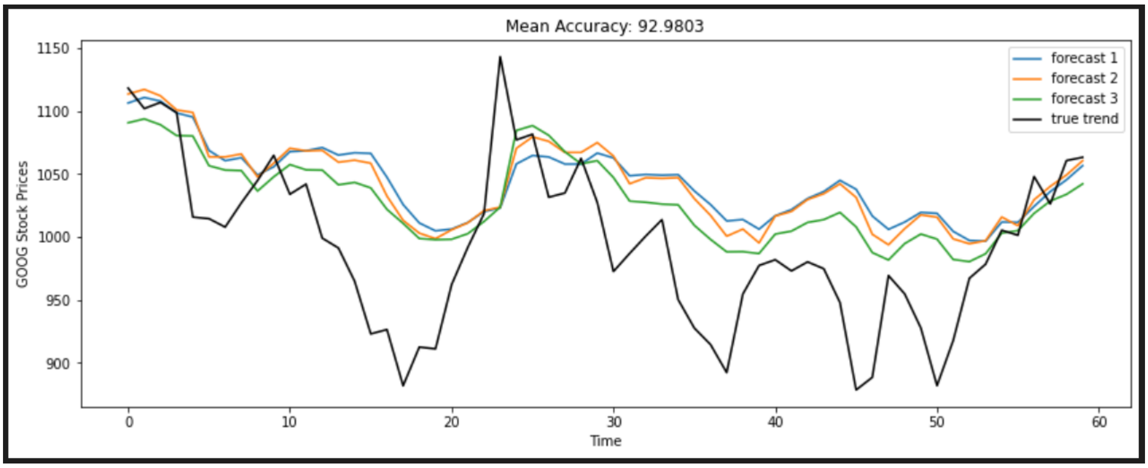
Fig 7.4. Bidirectional-LSTM Model Predictions



- **Bidirectional GRU**

Bidirectional GRU is working well and gives us 92.9% accuracy. We tested it on google.

Fig 7.5. Bidirectional-GRU Model Predictions



b) Sentiment Analysis

In the first graph we calculated the average compound scores of all articles to see whether it's positive, negative or neutral. We have visualized it in bar chart format -

Fig 7.6 Compound Scores

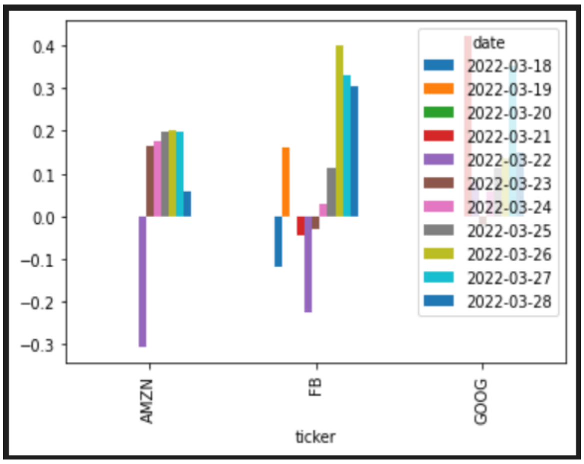


Fig 7.7 Positive Scores

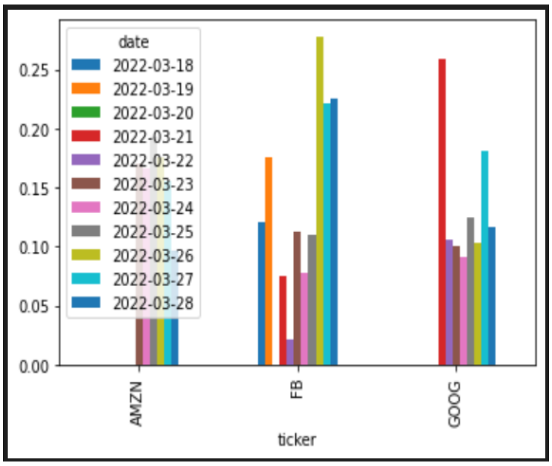


Fig 7.8 Negative Scores

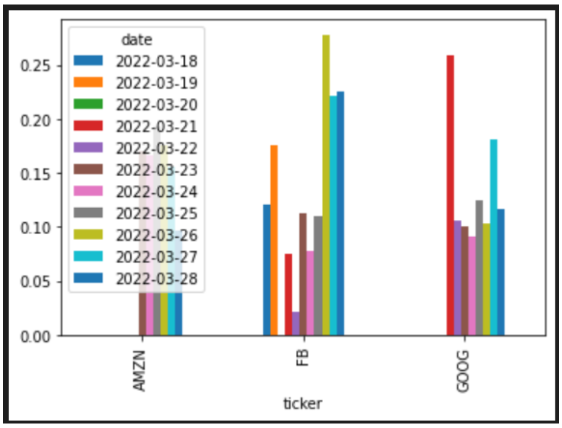
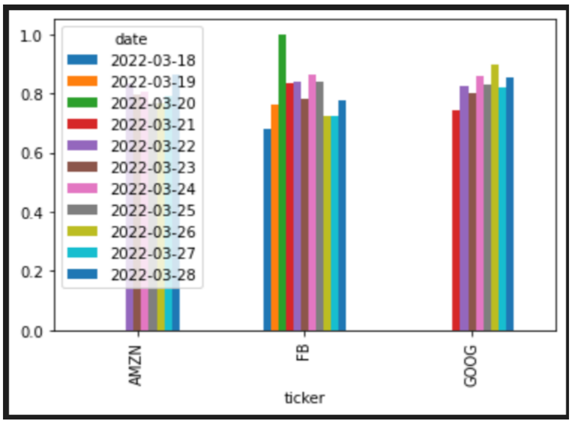


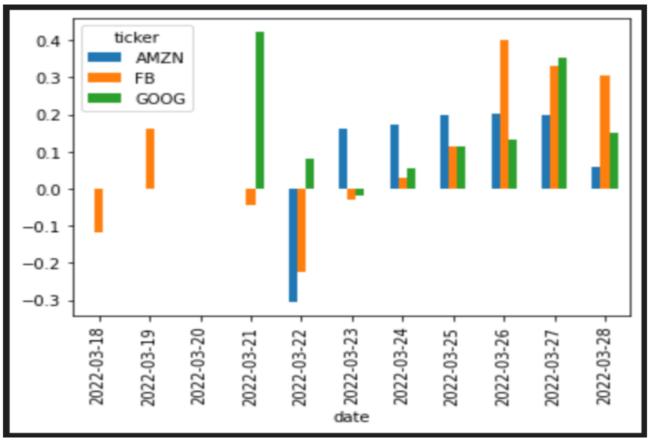
Fig 7.9 Neural Scores



These graphs give us an idea about whether the daily news of a particular company is positive or negative and whether one should invest in it or not. Each graph has bars of different days and we have done that for three companies.

This bar chart tells us what the average score of each company is for the past 9 days. This will make it easier to compare between different companies. Here we took the cross section of the compound score row, flipped the data frame so that we have the dates as the x-axis, and then plotted it as a Bar chart -

Fig 7.10. Average compound company scores of 9 days



Various companies on different levels were tested, and accordingly the accuracy of the models were compared to determine which among them stayed true to its prediction.

c) Moving Average Agent

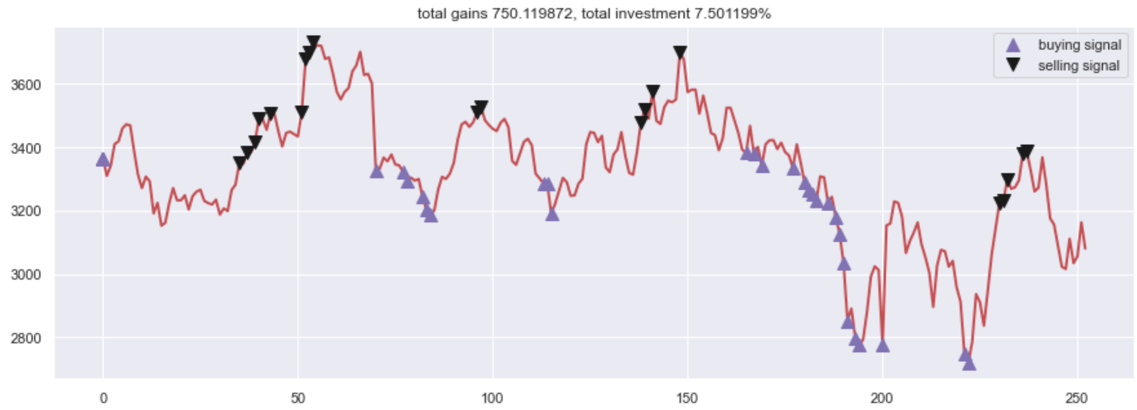
Along with the textual plan, the project also has a graph depicting points (days) when to sell/ buy certain stocks of a particular company , which will make it easier to understand on which day one should buy/sell a unit of stock.

Fig 7.11 Line graph depicting buying and selling pins for MSFT

d) Turtle Agent

This agent uses the turtle strategy to draft the investment plan. It will give us an idea in advance about how one should plan the investment. . This graph depicts (days) when to sell/ buy certain stocks of a particular company, it will make it easier to understand on which day one should buy/sell a unit of stock.

Fig 7.12 Line graph depicting buying and selling pins for AMZN



e) Comparative Analysis of Deep Learning Models

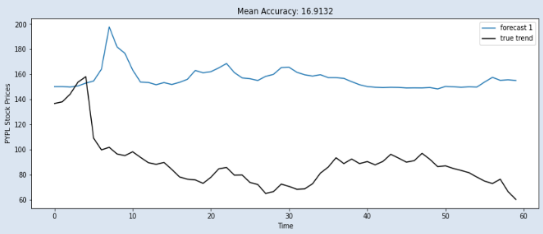
UNSTABLE STOCKS
STOCK LOSERS
STOCK GAINERS
STABLE STOCKS

We compared the MAPE accuracy of 3 deep learning models (RNN , GRU & LSTM) and the results are as follows.

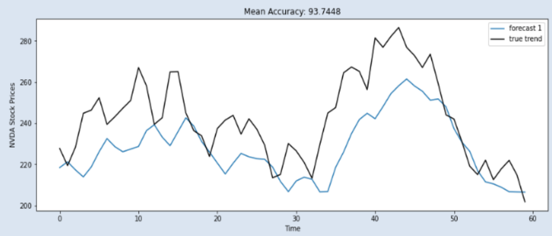
RNN

In this table we have added outputs of 22 companies from the above categories and along with the graphs we have mean accuracy of the RNN model. Later on, we have compared the accuracies of all models.

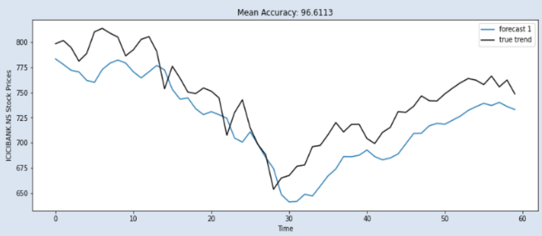
PYPL



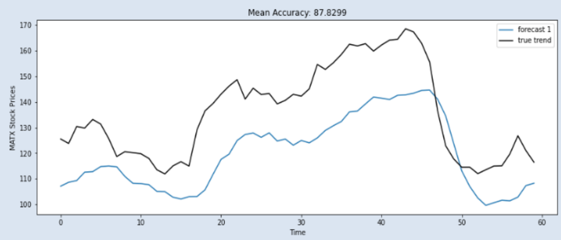
NVDA



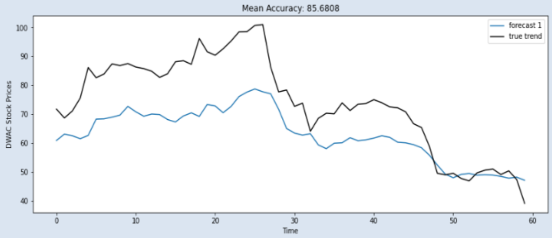
ICICIBANK.NS



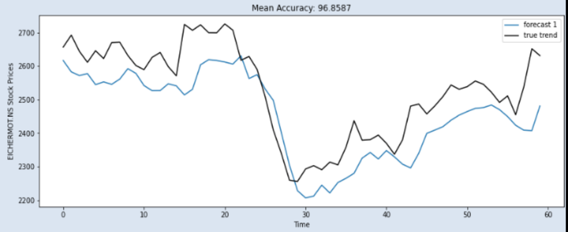
MATX



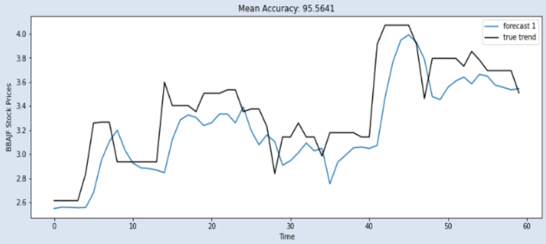
DWAC



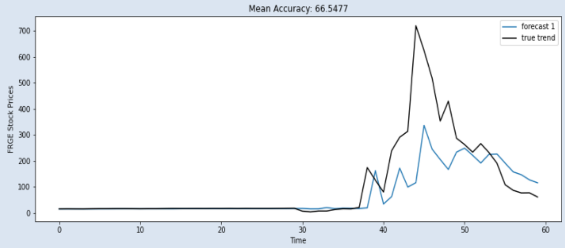
EICHERMOT.NS



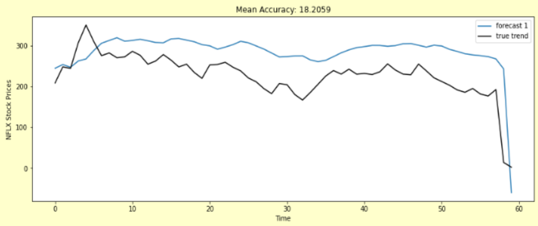
BBAJF



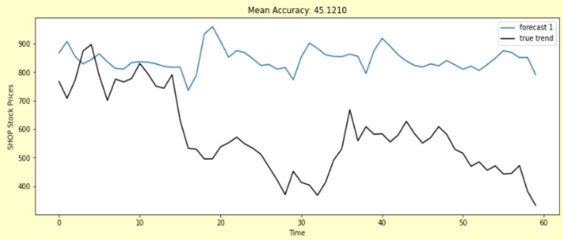
FRGE



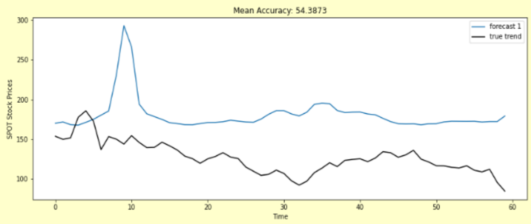
NFLX



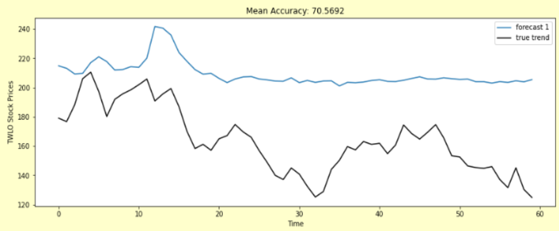
SHOP



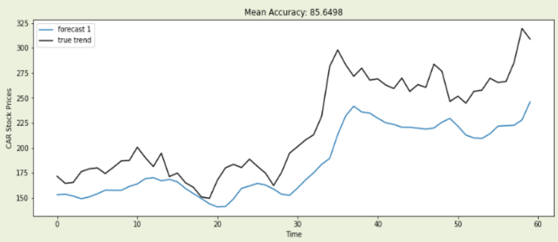
SPOT



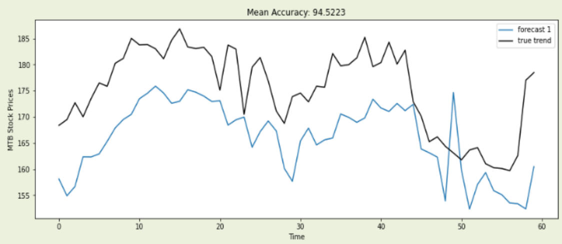
TWLO



CAR – AVIS BUDGET



MTB



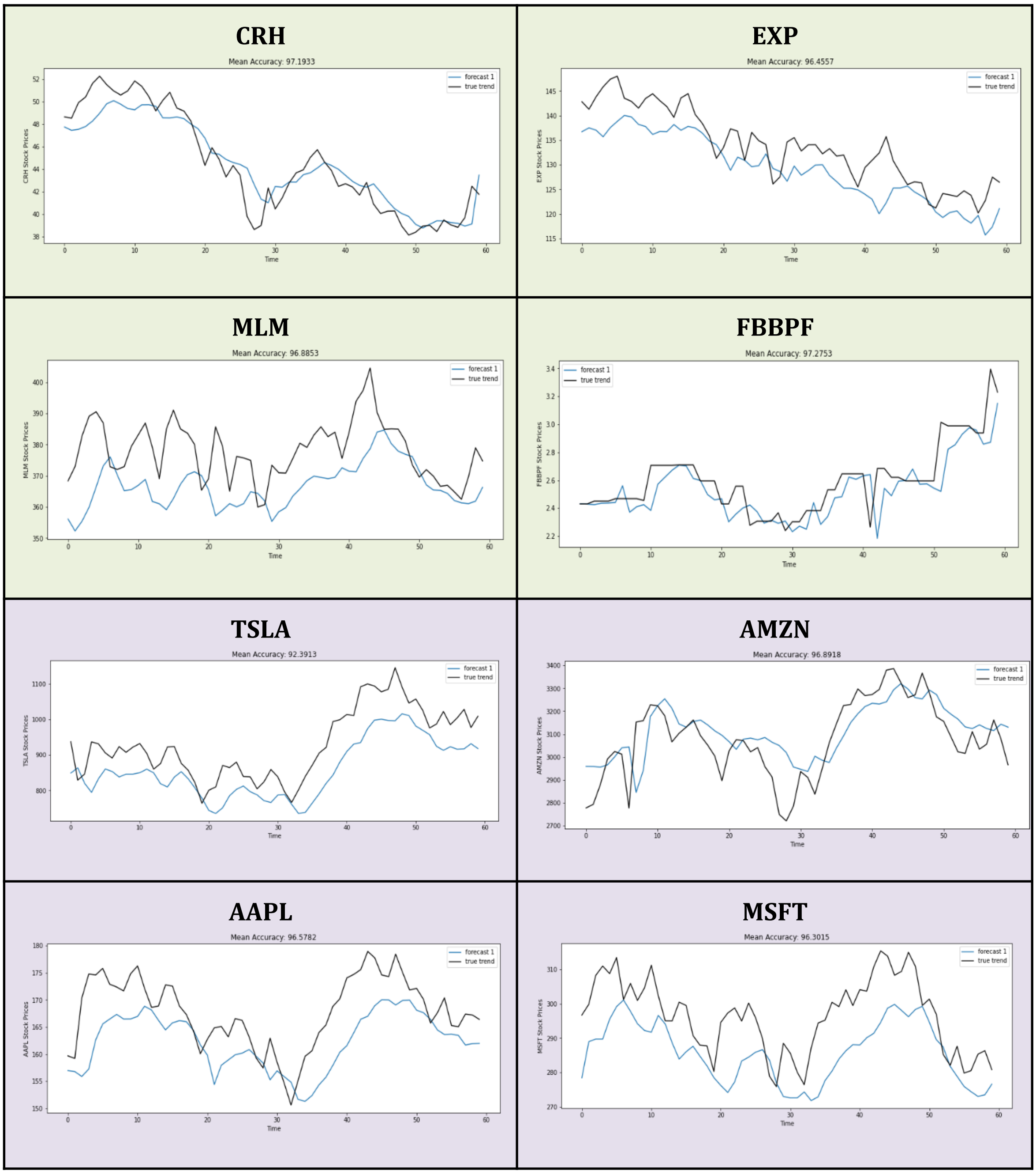
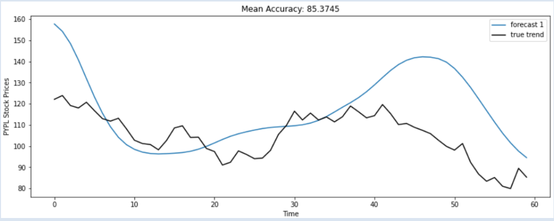


Table 8.1 RNN Prediction Trends of different companies

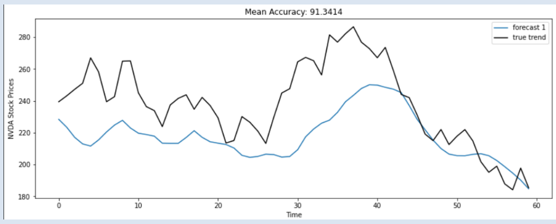
LSTM

In this table we have added outputs of 22 companies from the above categories and along with the graphs we have mean accuracy of the LSTM model. And we have also compared these outputs to the outputs of other two models.

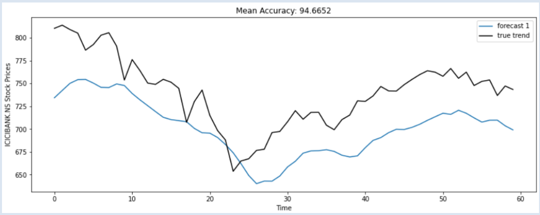
PYPL



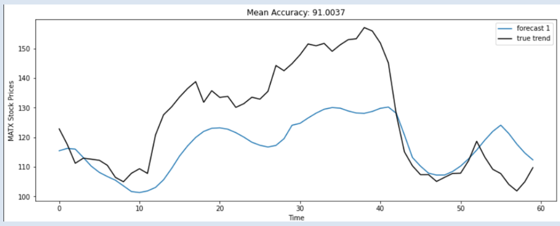
NVDA



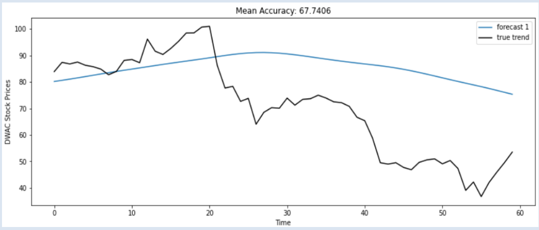
ICICIBANK.NS



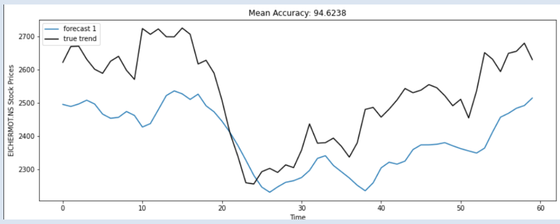
MATX



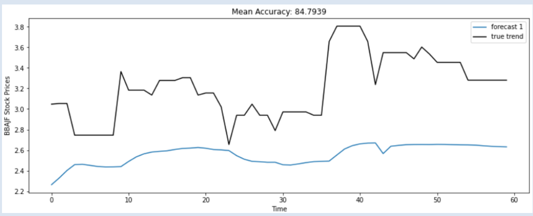
DWAC



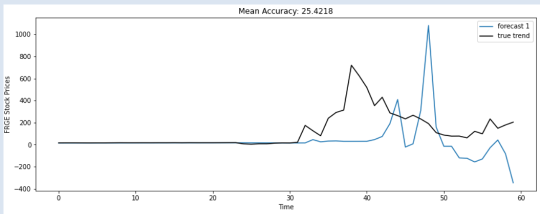
EICHERMOT.NS



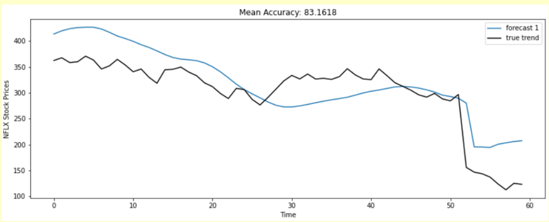
BBAJF



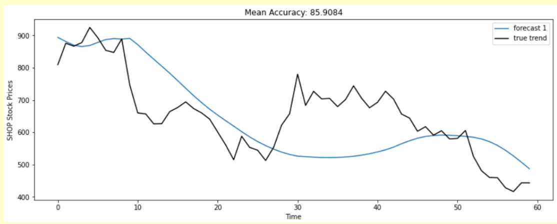
FRGE



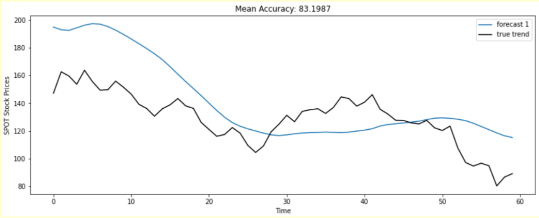
NFLX



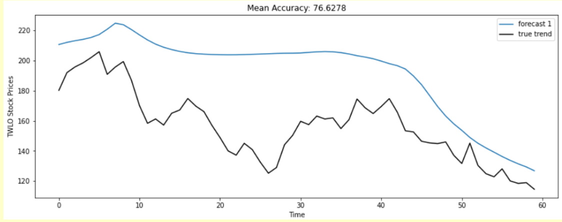
SHOP



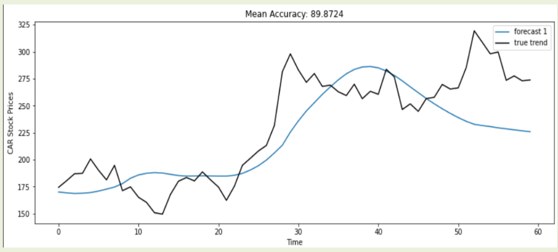
SPOT



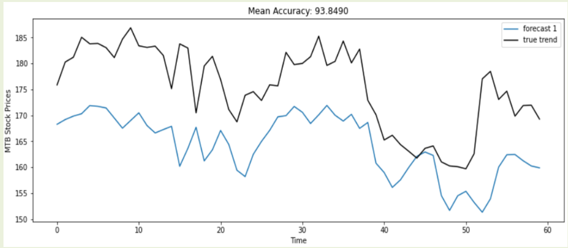
TWLO



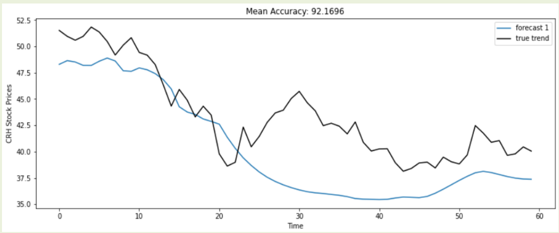
CAR – AVIS BUDGET



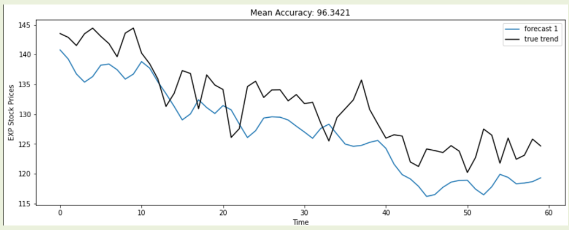
MTB



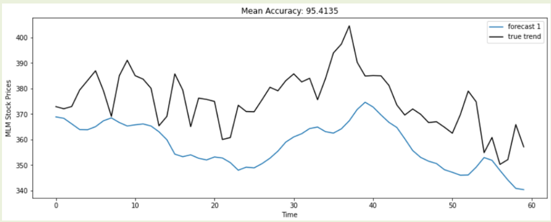
CRH



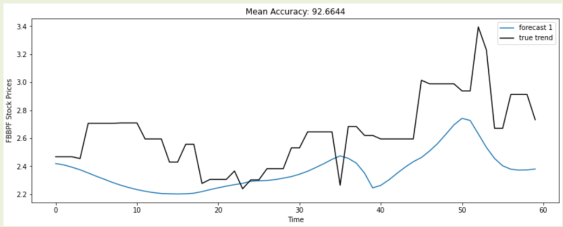
EXP



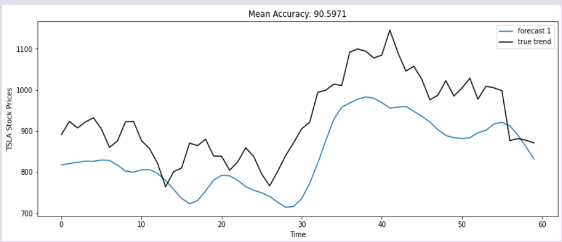
MLM



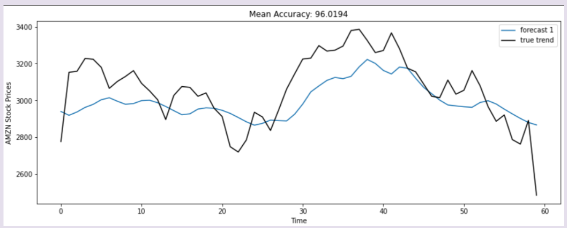
FBBPF



TSLA



AMZN



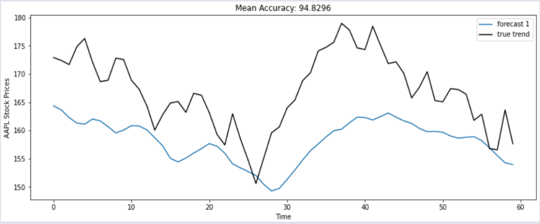
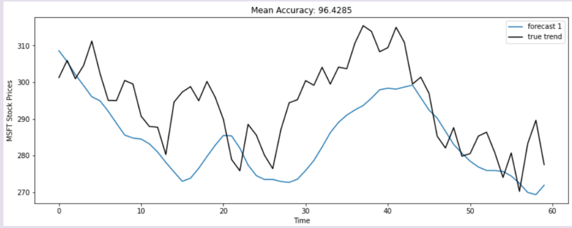
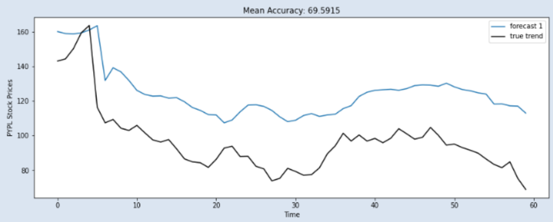
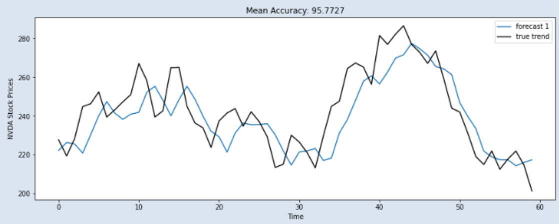
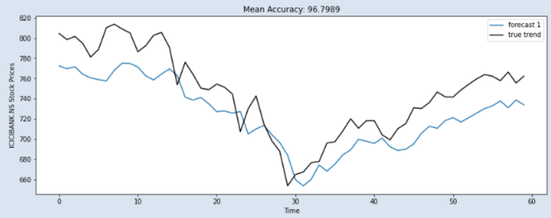
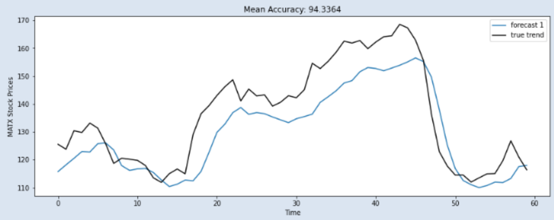
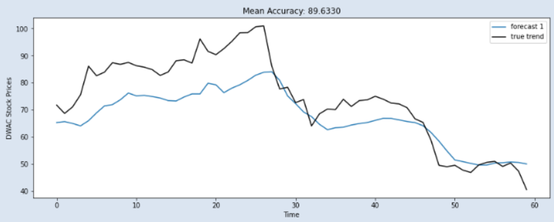
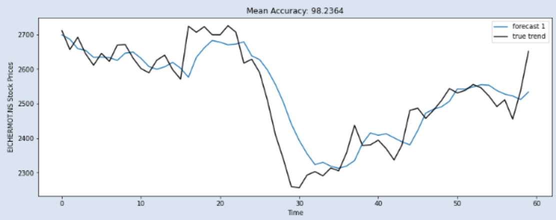
AAPL	MSFT
	

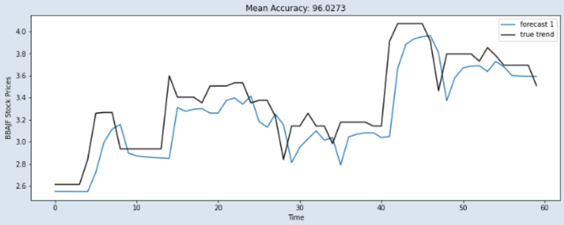
Table 8.2 LSTM Prediction Trends of different companies

GRU

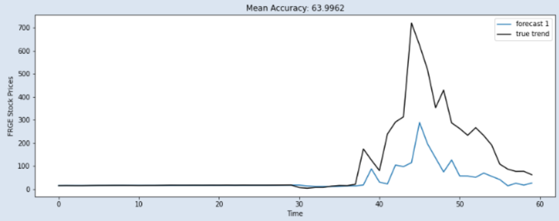
In this table we have added outputs of 22 companies from the above categories and along with the graphs we have mean accuracy of the GRU model. As we can see GRU model is giving us the best results and in all cases the accuracy is best.

PYPL	NVDA
	
ICICIBANK.NS	**MATX**
	
DWAC	**EICHERMOT.NS**
	

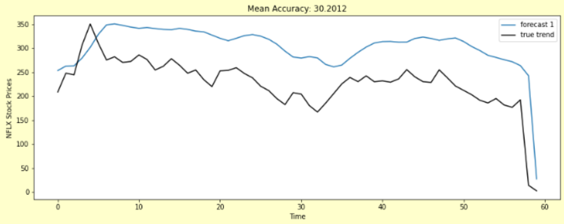
BBAJF



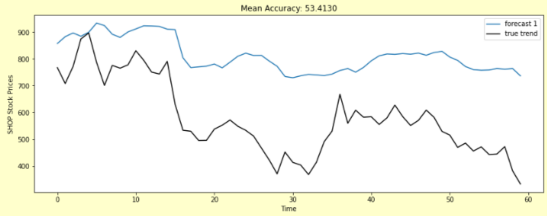
FRGE



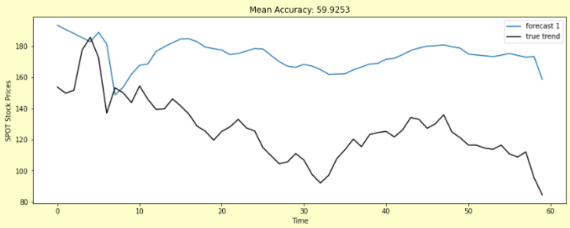
NFLX



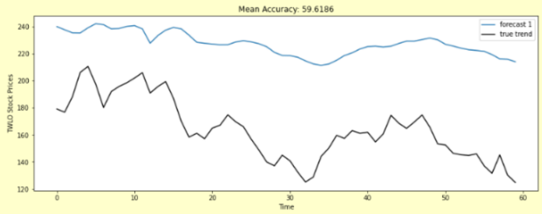
SHOP



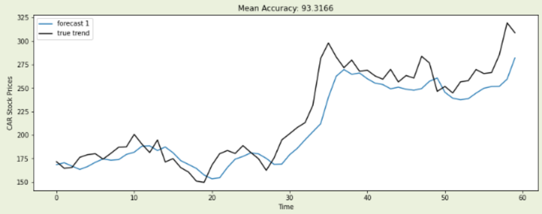
SPOT



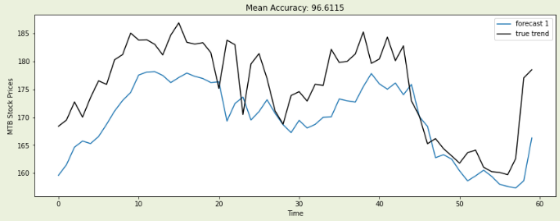
TWLO



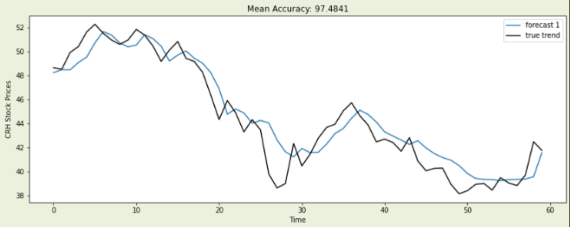
CAR – AVIS BUDGET



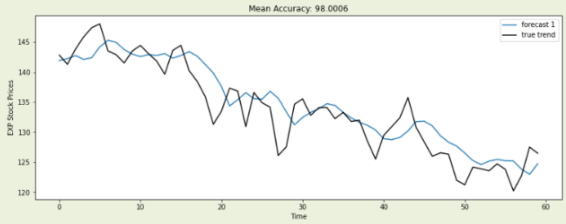
MTB



CRH



EXP



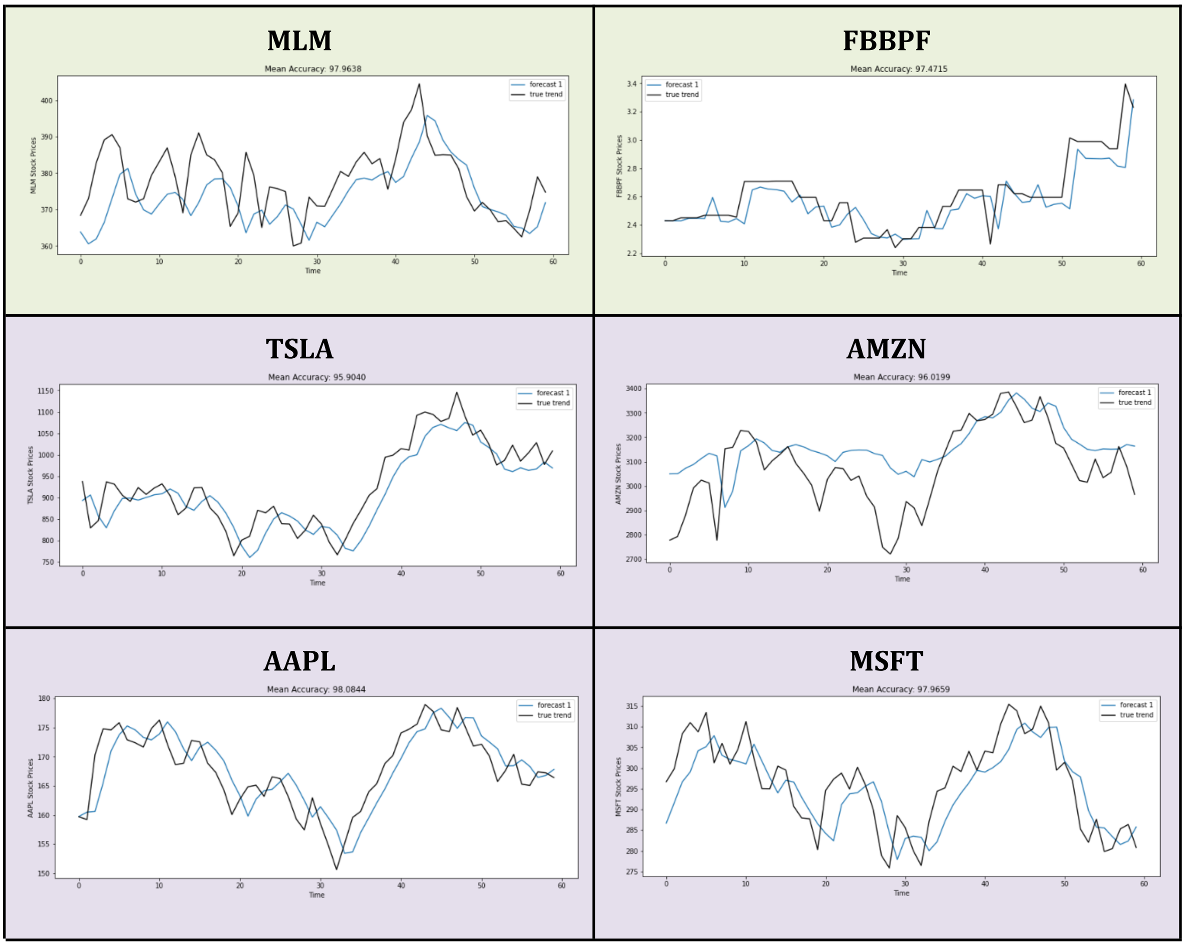


Table 8.3 GRU Prediction Trends of different companies

f) Comparative Analysis of Agents

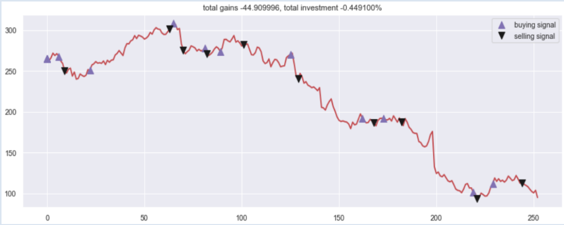
From the models, GRU turns out to be the most precise model, hence it is further used to prepare trading strategies using turtle and moving average. The comparison in terms of the agents is as shown below.

UNSTABLE STOCKS
STOCK LOSERS
STOCK GAINERS
STABLE STOCKS

Moving Average Agent

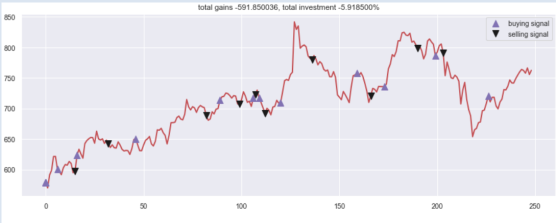
Here we are comparing the agents and this table has outputs of 22 companies and moving average agent is giving the best outcome total gains and investment wise.

PYPL

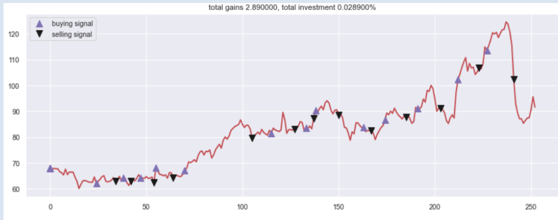


NVDA

ICICIBANK.NS



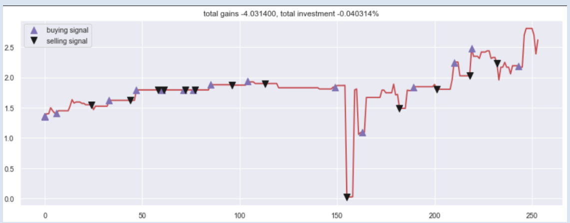
MATX



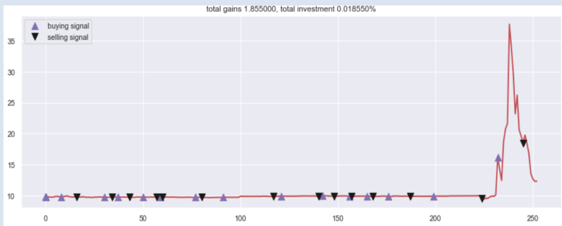
DWAC

EICHERMOT.NS

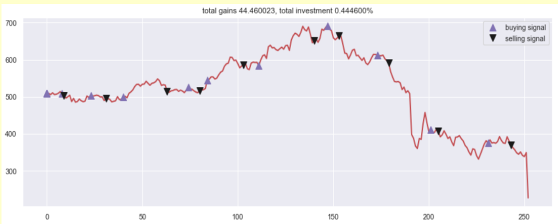
BBAJF



FRGE

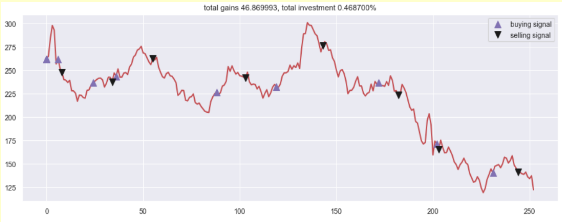


NFLX



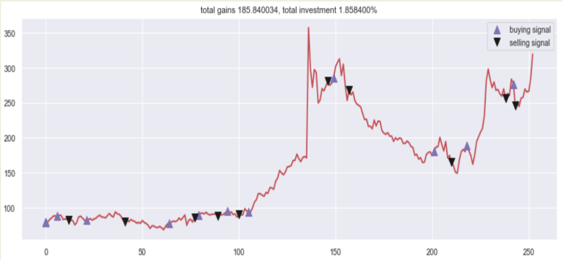
SHOP

SPOT



TWLO

CAR – AVIS BUDGET



MTB

CRH

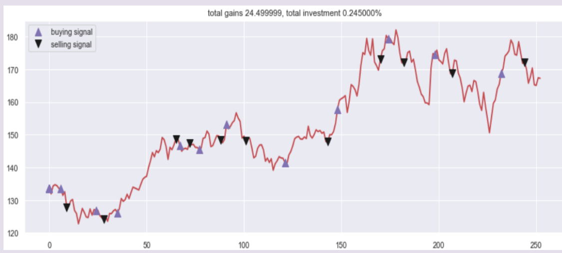
EXP

MLM

FBBPF

TSLA

AAPL

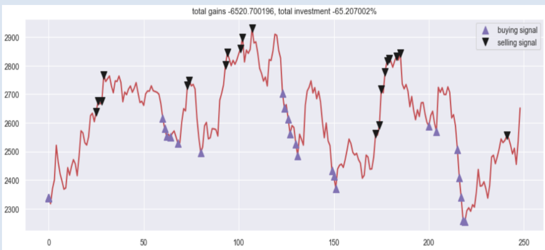


MSFT	AMZN
	

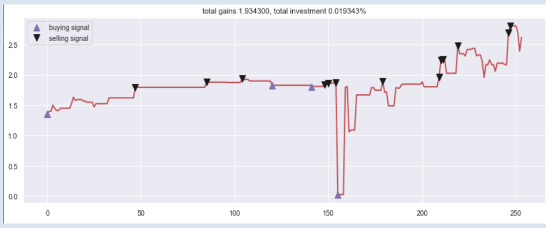
Table 8.4 Moving Average Visual Trading Strategy using buying and selling pins.

Turtle Agent

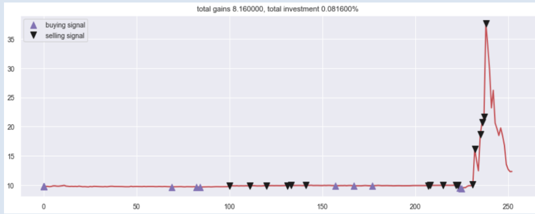
Here we are showing the outputs of turtle agents. This table has outputs of 22 companies.It not only shows us the graph but also tells us the outcome, total gains and investment wise.

PYPL	NVDA
	
ICICI BANK.NS	**MATX**
	
DWAC	**EICHERMOT.NS**
	

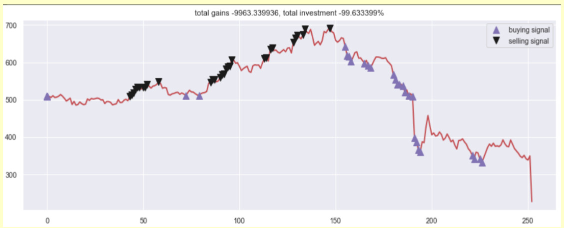
BBAJF



FRGE

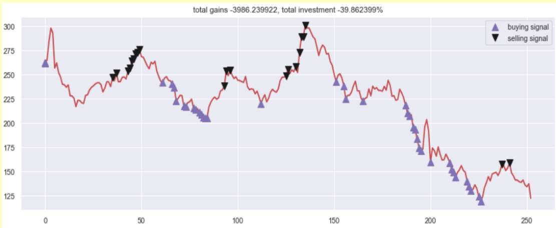


NFLX



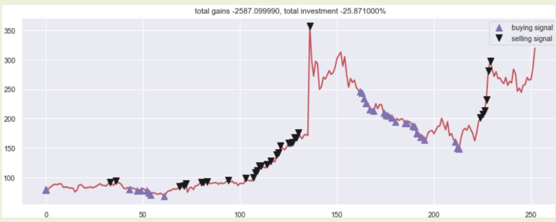
SHOP

SPOT



TWLO

CAR – AVIS BUDGET



MTB

CRH

EXP

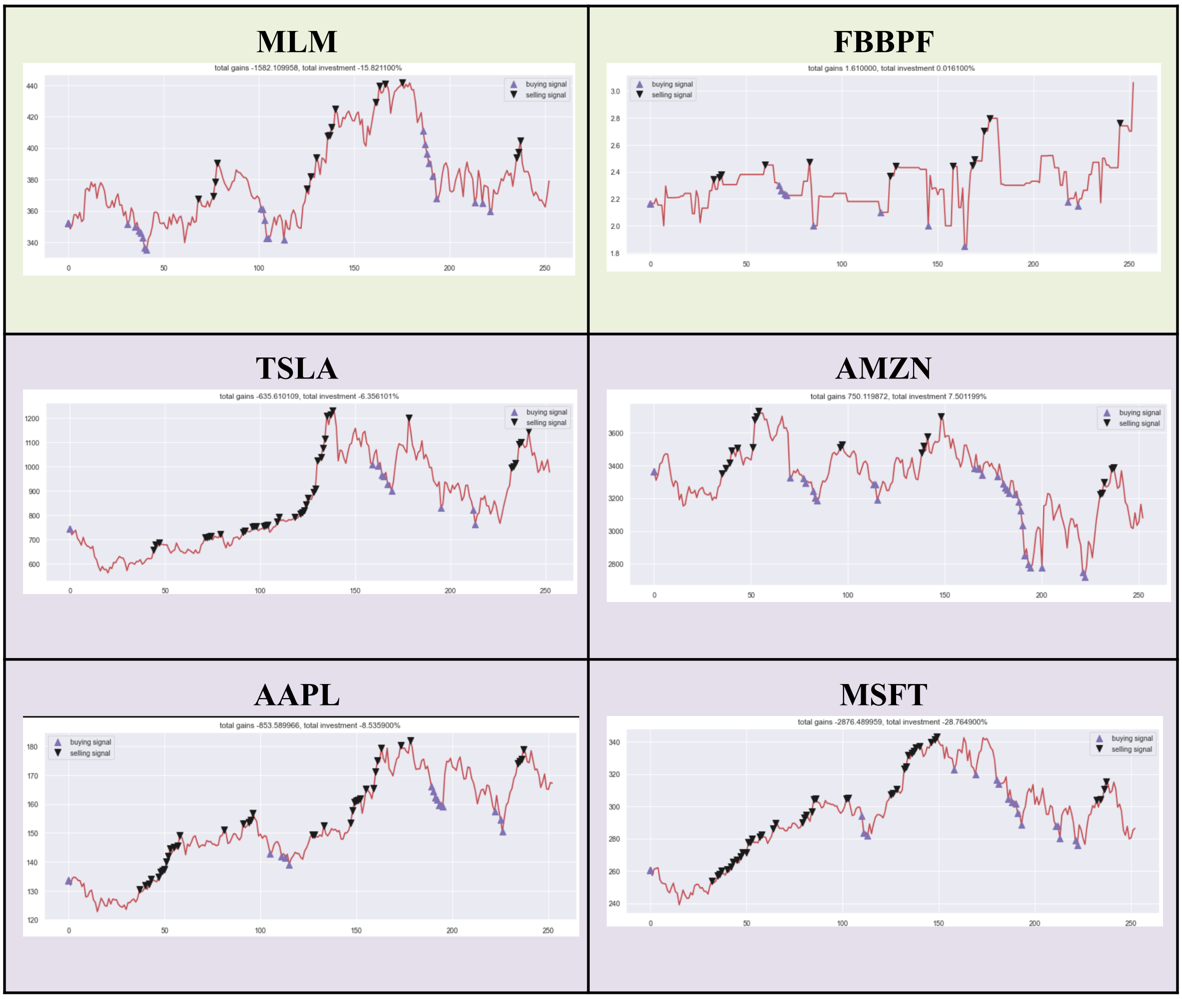


Table 8.5 Turtle Agent Visual Trading Strategy using buying and selling pins.

From the above comparison, GRU evidently stayed true to its certainty for companies of different fluctuation rates. Hence, using the predictions given by GRU, trading strategies are developed and accordingly a comparison between the agents is visualized as given below.

Sr No	Company	Accuracy %			Total Gain	
		RNN	GRU	LSTM	Turtle	Moving Avg
1	PYPL	16.9	69.5	65.3	-8123	-44
2	NVDA	93.7	95.7	91.3	-2545	109
3	ICICI BANK.NS	96.6	96.7	94.6	-2730	-591
4	MATX	87.8	94.3	91.0	-418	2

5	**DWAC**	85.6	89.6	67.7	-382	11
6	**EICHERMOT.NS**	96.8	98.2	94.6	-6520	-2665
7	**BBAJF**	95.5	96.7	84.7	1	-4
8	**FRGE**	66.5	63.9	25.4	8	1
9	**NFLX**	18.2	30.2	83.1	-9963	44
10	**SHOP**	45.1	73.4	65.9	-9677	-217
11	**SPOT**	54.3	59.9	83.1	-3986	46.8
12	**TWLO**	70.5	56.6	76.6	-5909	-129
13	**CAR**	85.6	93.3	89.8	-2587	185
14	**MTB**	94.5	96.6	93.8	-555	-22
15	**CRH**	97.1	97.484	92.1	-308	-1
16	**EXP**	96.4	98.0	96.3	-1729	-2
17	**MLM**	96.8	97.96	95.4	-1582	38
18	**FBBPF**	97.2	97.7	92.6	1	-3
19	**TSLA**	92.3	95.9	90.5	-635	234
20	**AMZN**	96.8	96.1	96.0	750	-22
21	**APPL**	96.5	98.0	94.8	-853	24
22	**MSFT**	96.3	97.9	96.4	-2876	14

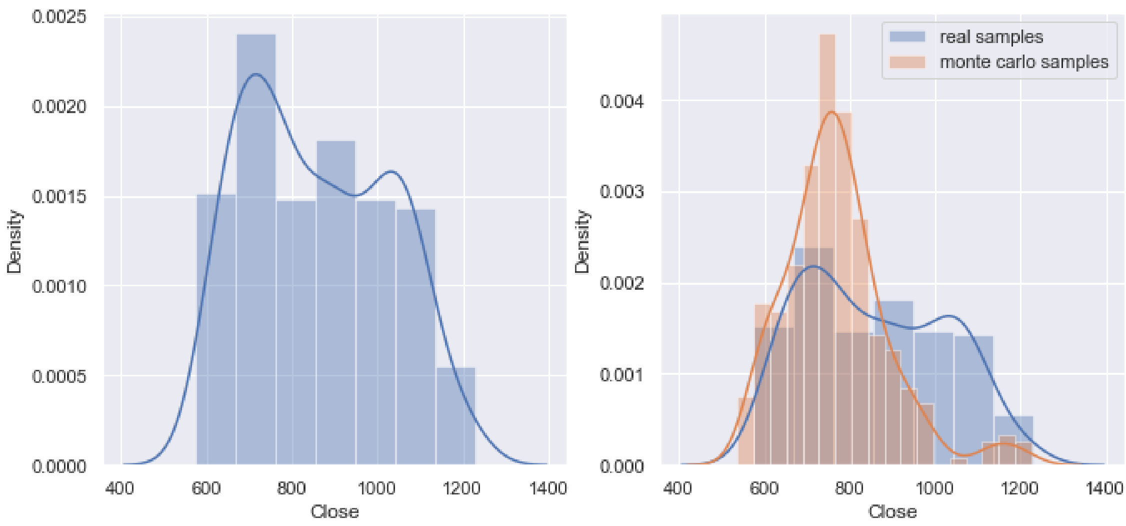
The above tables and their respective outcomes are summarized in the following table. We compared the deep learning models and agents among themselves and the result is clear that GRU and Moving Average agent are giving the best results.

g) Visualization

Monte- Carlo Simulation:

We have used 2 visual simulations one is Monte Carlo simulations extremely useful in case of uncertainties and especially useful in uncertain stocks like Tesla for which the market flourished in no time as evident from the dataset. It is useful in predicting uncertain values as random numbers are chosen based on likely outcomes of whether the closing values are high or low and how it will help take calls for future making the simulation extremely useful.

Fig 7.13 Comparison between Monte Carlo and the Real Closing Index

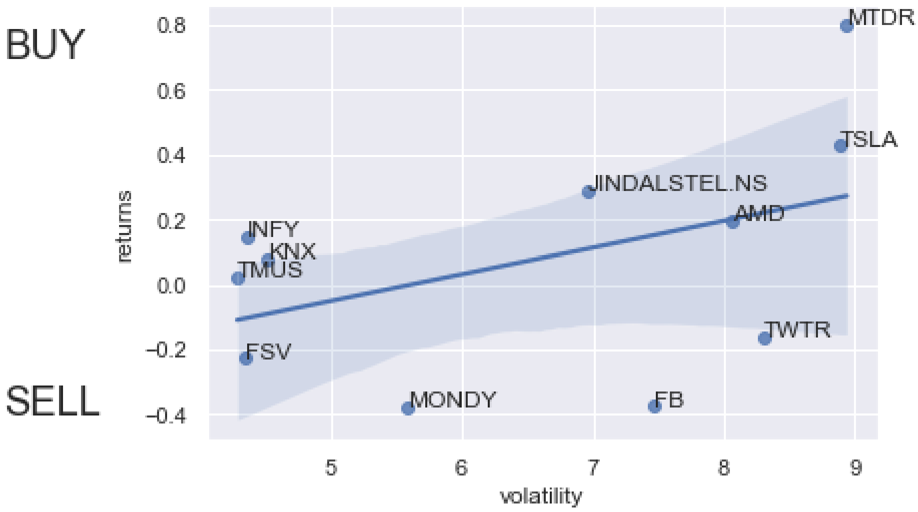


We hope with these analysis and visualization, we were able to influence, encourage and generate self-belief in people to trade with our system.

Joint Plot Visualization:

- From the jointplot graph it is clearly evident that Jindal steel, Matador industries and Tesla will give high returns despite high volatilities implying the stocks are worth investing for a decent amount of period not for long as prices fluctuate quite often.
- TMUS, INFY and KNX yield returns despite low volatilities implying stocks are worth buying for the long term as prices don’t fluctuate much.
- Twitter and FB as per jointplot are not worth buying as their prices fluctuated a lot and did not yield significant return in comparison to other stocks.
- AMD is on edge implies hold if bought else no need to buy.

Fig 7.14 Stocks comparison using linear regression line of best fit from joint plot visualization



8. SUMMARY

The main idea of this project is to make a prediction model which could help people trade in the stock market despite being inexperienced in the ups and downs of it. A visual comparison of various stock models on the basis of which a person will be able to take calls regarding buying and selling of a particular stock is shown.

A detailed comparison of various companies on all levels of stability, loss and gain are depicted. Out of all the models, GRU displayed results with the highest accuracy. The predicted trend from GRU was further experimented on by using trading agents to make a strategic model to help the investors when and how much to buy or sell during a certain time-stamp. To add to the decision making process in stock investments, Sentimental Analysis is used to compound the positive or negative reviews from the financial news segments.

At last, to identify the patterns and get a deeper understanding of the stocks trends, visualization techniques from Markov Chain Models, particularly Monte Carlo Simulation, is used which searches for a pattern in the data set and makes predictions on the basis of it. This is done along with Joint Plot Visualization to understand the relationship between different companies and get a thorough idea of which company it is safer to invest in.

9. REFERENCES

[1] Ghosh, A., Bose, S., Maji, G., Debnath, N., & Sen, S. (2019, September). Stock price prediction using LSTM on the Indian share market. In Proceedings of 32nd international conference on (Vol. 63, pp. 101-110).

[2] Lee, M. C., Chang, J. W., Hung, J. C., & Chen, B. L. (2021). Exploring the effectiveness of deep neural networks with technical analysis applied to stock market prediction. ComputerScience and Information Systems, (00), 2-2.

[3] Parmar, I., Agarwal, N., Saxena, S., Arora, R., Gupta, S., Dhiman, H., & Chouhan, L. (2018, December). Stock market prediction using Machine Learning. In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) (pp. 574-576). IEEE.

[4] Saud, A. S., & Shakya, S. (2020). Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. Procedia Computer Science, 167, 788-798.

[5] Awoke, T., Rout, M., Mohanty, L., & Satapathy, S. C. (2021). Bitcoin price prediction and analysis using deep learning models. In Communication Software and Networks (pp. 631-640). Springer, Singapore.

[6] Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. International Journal of Science and Research (IJSR), 6(4), 1754-1756.

[7] Moghar, A., & Hamiche, M. (2020). Stock market prediction using lstm recurrent neural network. Procedia Computer Science, 170, 1168-1173.

[8] Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in the Indian stock market. Procedia Computer Science, 167, 2091-2100.

[9] Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high‐frequency stock market classification. Journal of Forecasting, 38(6), 600-619.

[10] Li, H., Shen, Y., & Zhu, Y. (2018, November). Stock price prediction using attention-based multi input LSTM. In Asian Conference on Machine Learning (pp. 454-469). PMLR.

[11] Wu, J. L., Yang, C. S., Liu, K. H., & Huang, M. T. (2019, October). A Deep Learning Model for Dimensional ValenceArousal Intensity Prediction in Stock Market. In 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), IEEE.

[12] Al-Thelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2019, April). Forecasting of bahrain stock market with deep learning: Methodology and case study. In 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO) (pp. 1-5). IEEE.

[13] Vargas, M. R., dos Anjos, C. E., Bichara, G. L., & Evsukoff, A. G. (2018, July). Deep learning for stock market prediction using technical indicators and financial news articles. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[14] Jia, W. U., Chen, W. A. N. G., Xiong, L., & Hongyong, S. U. N. (2019, July). Quantitative trading on stock market based on deep reinforcement learning. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[15] Ojo, S. O., Owolawi, P. A., Mphahlele, M., & Adisa, J. A. (2019, November). Stock market behaviour prediction using stacked LSTM networks. In 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), IEEE.

[16] Aishwarya Singh(2018), Stock Prices Prediction Using Machine Learning and Deep Learning Techniques, Analytics Vidhya.

[17] Chouhan, Lokesh & Agarwal, Navanshu & Parmar, Ishita & Saxena, Sheirsh & Arora, Ridam & Gupta, Shikhin & Dhiman, Himanshu. (2018). Stock Market Prediction Using Machine Learning. 10.1109/ICSCCC.2018.8703332.

[18] Sotirios P. Chatzis, Vassilis Siakoulis, Anastasios Petropoulos, Evangelos Stavroulakis, Nikos Vlachogiannakis, Forecasting stock market crisis events using deep and statistical machine learning techniques, Expert Systems with Applications, Volume 112, 2018, Pages 353-371, ISSN 0957-4174.

[19] Wenpeng Yin, Katharina Kann, Mo Yu and Hinrich Schutze(2017), Comparative Study of CNN and RNN for Natural Language Processing, Arxiv:1702.01923v1.

[20] Will Koehrsen, Recurrent Neural Networks by Example in Python, 2018, towardsdatascience.com.

[21] Madhav Kumar Bhusal, Application of Markov Chain Model in the Stock Market Trend Analysis of Nepal, 2017, International Journal of Scientific & Engineering Research, Volume 8, ISSN 2229-5518.

[22] Derrick Mwiti(2018), Using Keras Long Short-Term Memory (LSTM) Model to predict Stock Prices, KDnuggets, 18:n45.

[23] Nagesh Singh Chauhan(2020), Stock Market Forecasting Using Time Series Analysis, 20:n20.

[24] Boldt, M., Borg, A., Ickin, S. et al. Anomaly detection of event sequences using multiple temporal resolutions and Markov chains. Knowl Inf Syst 62, 669–686 (2020).