Real time Stock Prediction

Raktim Bijoypuri (IIT2018125)
Project Supervisor - Dr Manish Kumar
Department of Information Technology
Indian Institute of Information Technology, Allahabad, Prayagraj

Abstract

The aim of this study is to develop a real-time stock forecaster system using not only historical stock price data but also incorporating some indicators used by intraday traders . 2 technical indicators i.e ema and vwap have been used in this study in addition to the stock price for making predictions. We had taken some stock prices of 15 mins frequency for training the model and predicting the stock price, i.e we predict the stock price after 15 minutes on the basis of current,previous few values of closing price,ema and vwap.

Keywords

Real time, Stock Prediction, CNN, LSTM, ARIMA, EMA, VWAP

1. INTRODUCTION

Stock market is a very uncertain market whose trend is based on several micro, macro factors such as the global economy, healthcare situation, oil prices, interest rates, news articles, public sentiment etc.
Because of the presence of so many factors stock price prediction has been an extremely challenging problem which has given rise to several papers and studies trying to make the best predictions and models which they can, because of the tremendous business potential such models pose.
Infact presently there are many companies involved in high frequency trading whose sole purpose is to buy and sell stocks in a way which maximises their profits.
Since it is impossible to analyse so many factors as discussed earlier we have tried to focus on only the historical stock price to predict the price, also since we were dealing with such short intervals it is expected that data patterns will be much more helpful in most of the cases rather than trying to analyse sentiments or news articles.

In this paper, we present the results of a few models such as LSTM, CNN,ARIMA etc trained on historical data for stock price prediction. This paper is organised as follows: section 2 describes the relevant study done on existing state of the art methods of stock prediction. Section 3 describes the dataset we have used. Section 4 is experimental setup. Section 5 states the methodology we adopted while section 6 states the experimental results we obtained from this research. Section 7 includes conclusion and discusses future works.

2. LITERATURE REVIEW

Dharmaraja Selvamuthu et al. proposed using ANN with several different learning

algorithms for predicting stock prices[1] they have used **3 different learning algorithms. Bayesian Regularization, Levenberg - Marquardt, Scaled Conjugate Gradient** on a common ANN architecture of 20 hidden layers, they have used a paid dataset of Reliance Private Ltd, from Thomson Reuter over a period of 1 year with 15,000 data points per day. From this data they also made a dataset of 15-minutes time intervals to evaluate their model on 2 datasets. A lag of 50 data points was used so the prices of previous 50 data points were used as features in training. They were able to obtain an **MAPE** of 99.9% using tick data and 98.9% over 15 minutes dataset.Nayak et al. have made 2 models, one which predicts prices on the next day using historical data and the other which predicts prices for the next month[2]. This paper doesn't predict the prices, instead it just predicts the trend of the price i..e whether it will be UP or DOWN, For the next day prediction historical data as well as sentiment analysis of twitter data is down, but for 1 month prediction only historical data is used. The historical data has been taken from yahoo so only opening,closing price etc. per day are taken into account. They have used Logistic regression, SVM, Boosted Decision tree in order to predict the trend on the basis of features such as volume variation, sentiment, continuous up/down. They were able to achieve an accuracy upto 70%.

Shakya et al. have also used artificial neural networks in order to predict stock prices of Nepal stock exchange [3]. They have tried to predict the percentage increase/decrease in stock prices for every second minute, they have used 3 features based on the data of previous 30 minutes i.e Number of transaction in 30 minutes, Traded share volume in 30 minutes, Standard deviation of Last Traded Price (LTP) in 30 minutes. Their output label is then one of 10 classes, where the classes are defined on the basis of percentage increase/decrease in stock price for example label "1" corresponds to (1-2) percent price increase, they were able to obtain best accuracy of 86% for one of the stocks. They had used approximately 6 months of data.Jingpei Dan et al. have showcased in this paper, forecasting performances of **deterministic ESNs** which are investigated in stock price prediction applications.[4] The experiment was performed on two benchmark datasets (Shanghai Composite Index and S&P500) of which the experimental results on Shanghai Composite Index dataset show that deterministic ESNs outperform standard ESN in accuracy by about 20% and stability by 52% averagely. Experiments with S&P 500 dataset shows that the deterministic ESNs have improvement in efficiency by about 23% while having insignificant improvement in forecasting accuracy. Since deterministic ESNs have advantages of higher forecasting accuracy and efficiency as indicated by the experimental results, they have great prospects in applications of stock price prediction and even other time series forecasting applications.

Wenjie Lu et al. proposed a **CNN-LSTM** based stock forecasting method.[5] They have used MLP, CNN, RNN, LSTM, CNN-RNN, and other forecasting models to predict the stock price one by one. The forecasting results of these models were then analyzed and compared. The dataset in this research were from the daily stock prices of a company listed on the **Shanghai Stock Exchange** from July 1, 1991, to August 31, 2020, including **7127 trading days**. In terms of historical data, they choose eight features, including opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change. At first, they adopted CNN to efficiently extract features from the data, which were the items of the previous 10 days. And then, they adopted LSTM to predict the stock price with the extracted feature data. According to the experimental results, the CNN-LSTM can provide a

reliable stock price forecasting with the highest prediction accuracy having MAE of 27.564 and RMSE of 39.688.In the paper, **Stock Closing Price Prediction using Machine Learning Techniques** they propose a model which uses the ANN and Random Forest[6]**,** they have used 6 features to predict the closing price of next day, their features were high-low of previous day, close-open of previous day, previous 7 days MA etc. They had used only 3 layers i.e the Input layer, Hidden layer(with 3 units), Output layer in their ANN part and were able to obtain best MAPE of 1.14% with random forest on one of the stocks. They had used Yahoo Stock market daily price data of 10 years and used their model on stocks of 5 companies.

Majumder et.al[7] tried predicting stock prices for 35 bangladesh companies they tried running feed forward neural networks and ARIMA and Holt-winters approaches on these stocks they found ARIMA(1,0,0) to perform the best for them with an MAPE of 82% they had used approximately 1 year of training data and 6 months of testing data moreover the average accuracy of all algorithms was greater than 75% for them. Zhang et al[8] had used advanced neural networks such as BP, Elman network and PSO-elman network they had used MSE, MAPE for model evaluation and they found that PSO-elman network was working the best for them with good tolerance to fluctuations they were predicting opening price of stock data with a model trained on historical data.

3. DATASET DESCRIPTION

US stock markets historical data of intraday prices is available for free on many websites.

For US stocks we will use AlphaVantage API which gives real-time stock prices as JSON objects and also historical data.

We can get historical stock prices over a period of 5,10,15 minutes.

A sample time-series is shown in the below image.


```
"2022-02-28 14:40:00": {
    "1. open": "121.1050",
    "2. high": "121.3800",
    "3. low": "121.1050",
    "4. close": "121.2450",
    "5. volume": "38987"
},
"2022-02-28 14:35:00": {
    "1. open": "121.0998",
    "2. high": "121.2000",
    "3. low": "121.0388",
    "4. close": "121.1050",
    "5. volume": "27560"
},
```


5. PROPOSED METHODOLOGY

The proposed methodology contains three sections that have been discussed below:

5.1 Technical Indicators

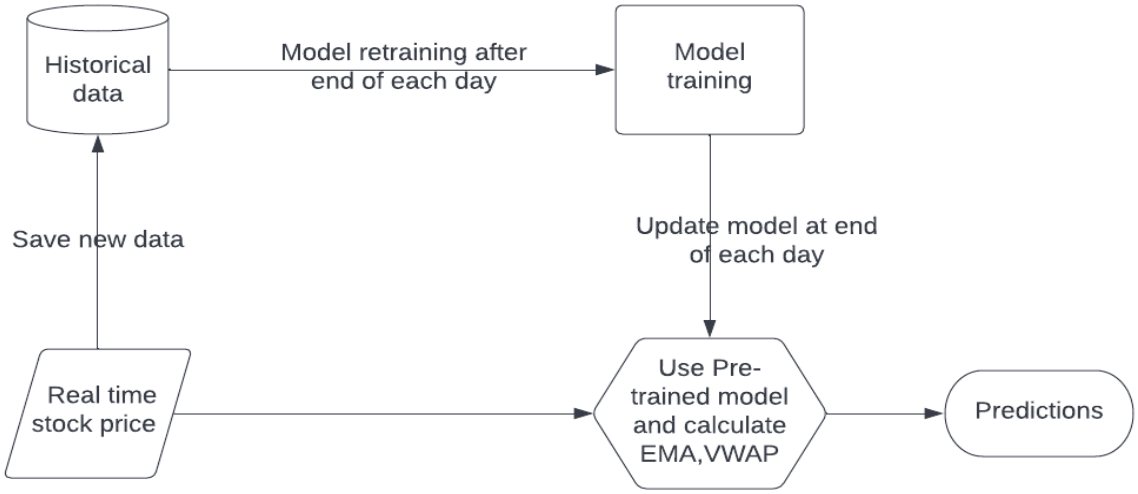
Intra-day traders often use many technical; indicators on when to buy/sell a particular stock, so we will try to incorporate some of the common technical indicators used in our model as features. Some of them are -

- EMA(d) [Exponential moving average] - The average price of the stock in the previous “d” days weighted exponentially(this way the prices of recent days are given more weight)[9]
- Williams %R - It tells us about the momentum of the stock, i.e whether the stock is overbought or oversold according to certain thresholds.
- VWAP [Volume weighted average price] - It gives us the average price the stock is being traded in the day based on both price and volume[10].

- RSI - This is a momentum Indicator which can help investors decide whether the stock is following a bullish or bearish trend.

Other technical indicators are subject to experimentation.

Flowchart-



5.2 LSTM

Neural networks have been proven to give good results when the feature space is non linear. Recurrent Neural Networks (RNN) have been historically used for stock market prediction. However there are several issues that come up while using RNN for stock prediction, one of the most noticeable ones is the vanishing gradient problem which makes RNNs unable to handle long term memory. There are mainly 2 ways to get rid of this problem - 1. Gradient Clipping, 2. Use RNN with leaky units like - LSTM

Gradient Clipping: It involves regulating the gradient and to prevent it from getting too big or too small. But for this, we are losing information. So this is not an ideal approach.

LSTM: It is a variety of RNN which are explicitly designed to overcome the long term dependency problem. As other standard RNN networks LSTM also has the form of a chain of repetitive modules of neural network. But unlike other RNNs the repeating unit has a different structure.

The goal is to create a forecasting model that predicts the price as closely as possible while prioritizing more permanent price movements (e.g., weekly trends) over smaller, more variable intra-week movements.

LSTM Architecture:

Here we will be explaining the architecture of repetitive modules of LSTM in brief. The main concept of LSTM lies in its cell state and various gates. The cell state acts as a transport highway that transfers relative information all the way down to the sequence chain. This acts as the memory of the network. It is able to carry information from the earlier time steps up to the last time step.

While going through various units information gets added or removed via gates. Gates are neural networks that decide which

information should be in the cell state. They learn to keep or forget the relevant information during the training of the model. Gates have sigmoid activation. The Sigmoid activation function gives an output in the range of 0 to 1. 0 means forget the data and 1 means keep the data. There are a total 3 different gates in a unit of lstm. Those 3 are -

Forget Gate - It decides which information to store and which to throw away. It takes input from previous hidden layers and current information, passes it through a sigmoid layer and reflects the result in cell state.

Input Gate - It updates the cell state. Input from hidden layers and current information is passed into this. It decides which value should be updated. Hidden state and current input is also passed into the tanh function which keeps output values between -1 and 1 which helps in regulating the network.

Output Gate - It decides what the next hidden state should be.

Workflow in LSTM unit -

1. First previous hidden state and current input get concatenated. We call it combined input This is fed to 4 places
2. Forget layer takes this combined input and removes nonrelevant data.
3. Combined input is fed to tanh layer to create new candidate values.
4. Input layer takes this combined input and decides what information should be added in new cell state.
5. New candidate values and output of the sigmoid layer(input layer) is multiplied to determine which information is to keep from the tanh output.
6. Output from the forget layer and input layer is polarized which updates the cell state to new values. This is the new cell state. This is passed through a tanh layer.
7. Result from the output layer and tanh layer is then multiplied to form a new hidden state.

Finally a new hidden state and cell state is passed to the next unit.

The equations for the gates in LSTM are -

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

where -

$i_t \rightarrow represents\ input\ gate.$
$f_t \rightarrow represents\ forget\ gate.$
$o_t \rightarrow represents\ output\ gate.$
$\sigma \rightarrow represents\ sigmoid\ function.$
$w_x \rightarrow weight\ for\ the\ respective\ gate(x)\ neurons.$
$h_{t-1} \rightarrow output\ of\ the\ previous\ lstm\ block(at\ timestamp\ t-1).$
$x_t \rightarrow input\ at\ current\ timestamp.$
$b_x \rightarrow biases\ for\ the\ respective\ gates(x).$

The equation for cell state -

$$\tilde{c_t} = tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$$
$$h_t = o_t * tanh(c^t)$$

where -

$c_t \rightarrow cell\ state(memory)\ at\ timestamp(t).$
$\tilde{c_t} \rightarrow represents\ candidate\ for\ cell\ state\ at\ timestamp(t).$
$note^*\ others\ are\ same\ as\ above.$

Data preparation - To train LSTM model for a time series we need to convert the time series into a supervised learning problem, Because while training LSTM the model actually learns a function that maps a sequence of past observations as input to an output observation. That's why dataset has to be prepared in form of some samples. Each sample will take current timestep observation as target value with n number of previous data points as features where n is called lag observations, a hyperparameter. Apart from this we may need to normalize and scale data for better results.

There are mainly 2 methodologies based on the number of features of time series we take into consideration.

1. **Univariate -** In this case for each time step we only consider observation for one property. In this case the above mentioned method for data preprocessing is enough.
2. **Multivariate -** Here we have to consider values for multiple properties for a single moment in time. This type of problem can be handled in 2 ways -

    a. **Multiple Input Series -** In this case problem may have two or more parallel input time series and an output time series that is dependent on the input time series. If we take lag observation = 3, then values of input series for last 3 timesteps are considered as input and 3rd step value of the output series is considered as output.

    b. **Multiple Parallel Series -** In this case there are multiple parallel time series and a value must be predicted for each. Here for lag observations = 3 we consider previous 3 values for each series as input and current value of each series as output.

In our case we can observe the problem in 2 different ways -

1. Univariate time series of stock price
2. Multivariste time serie of multiple technical indicators.

For both these cases apart from vanilla LSTM we can also use

1. Stacked LSTM
2. Bidirectional LSTM

Stacked LSTM - A Stacked LSTM architecture can be defined as an LSTM model comprised of multiple LSTM layers. An LSTM layer above provides a sequence output rather than a single value output to the LSTM layer below. Specifically, one output per input time step, rather than one output time step for all input time steps.

Bidirectional LSTM - Bidirectional long-short term memory(Bidirectional LSTM) is the process of making any neural network o have the sequence information in both directions backwards (future to past) or forward(past to future)

5.3 ARIMAX

ARIMAX is an extension of the ARIMA model which is a statistical model of time series forecasting. First we will take a look at the ARIMA model.

ARIMA stands for Auto Regressive Integrated Moving Average where

- Auto - The price of stock will be predicted on its own historical prices.

- Regressive - because it is a regression model
- Integrated - Differencing required to make the time series stationary
- Moving Average - The price of stock will be determined by error in predictions in previous time periods.

For this we have to determine the values of (p,q,d) :

1. **p - Auto regressive lag**

A simple AR model of order "p" looks like

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

Here we can observe that Yt depends on $Y_{t-1}, Y_{t-2} \ldots Y_{t-p}$, so in the AR model we use the value of the time-series in the previous "p" points to predict the current value .

The value of "p" is calculated using the PACF plot by observing the lag after which the correlation drops below a threshold value to be statistically significant.

2. **q - Moving average lag**

A simple MA model of order "q" looks like

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

Here we can observe that Y_t depends on the errors in prediction in previous time steps, so if we use MA(q) then we are just using the error made in predictions in previous "q" time steps with respect to the AR model predictions.
The value of "q" is calculated similar to "p" in AR model but here we use the ACF plot instead.

3. **d - Differencing**

The number of times we want to difference the series shifted by 1 in order to make it a stationary time series.

Finally the equation for ARIMA model is a combination of the AR,MA model as

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

We will be using an extended version of ARIMA for multivariate time series in order to incorporate other technical indicators, in terms of the model equation we just add the corresponding exogenous variables(i.e other indicators) in the RHS multiplied by its coefficient.

An important factor to look at while applying ARIMA or ARIMAX models is ensuring that the data is stationary(almost).
A stationary time series means that mean,variance of the data doesn't change with time and there is no seasonality.
So for determining the value of "d" in ARIMA model we have to ensure this either by visual checks or by using more robust methods such as Augmented Dickey Fuller Test (ADF Test) which provides us with the p-value and a p-value < 0.05 indicates the data is stationary(the closer to zero the better).

5. Model Evaluation

1. **RMSE** - Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values
$y_1, y_2, \ldots, y_n$ are observed values
n is the number of observations

2. **MAPE** - The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics. It usually expresses the accuracy as a ratio defined by the formula:

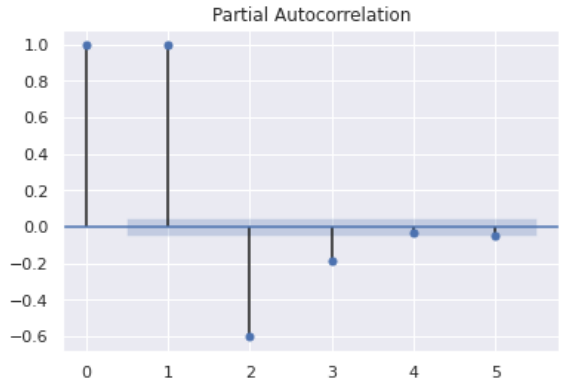
$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

6. RESULTS

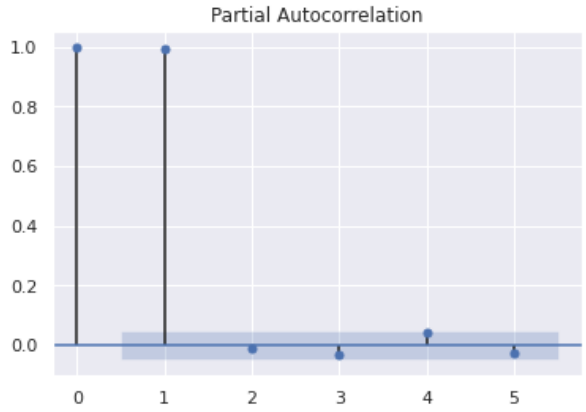
Lags

- For time series problems we use the value of the same stock price in previous few time steps as features, since we have used 2 extra technical indicators - EMA, VWAP so we have also taken their lagged values.
- First we tried manually some lag values such as 3,9,27 in multiple models and we found lag=3 a good enough value to be used in most of the models.
- We then also plotted PACF plots for price,EMA,VWAP which also validate lag=3 as after lag=3 the dependence is negligible(i.e PACF plot shuts off after lag=3).

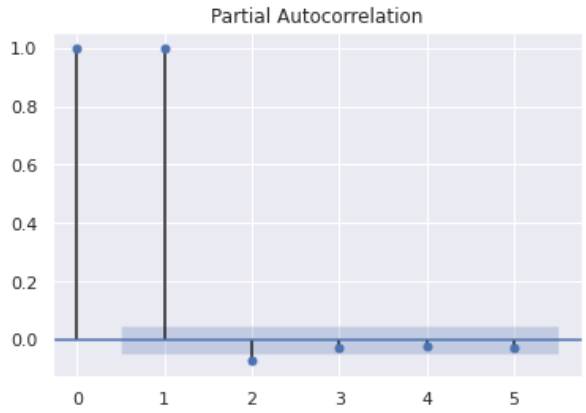
PACF Plot for EMA



PACF Plot for Price



PACF Plot for VWAP



Univariate vs Multivariate comparison

- We tried simple univariate models first by using only lagged price values as features and after few tries we found the multivariate models to perform better as a whole in most of the cases

so we have calculated most of the results using multivariate models only.

- The following tables are the results for AMAZON stock closing prices.

Univariate

Model	Train RMSE	Test RMSE
CNN	20.39	19.55
MLP	24.35	17.92

Multivariate

Model	Train RMSE	Test RMSE
CNN	5.38	10.50
MLP	8.70	15.61

ARIMA results

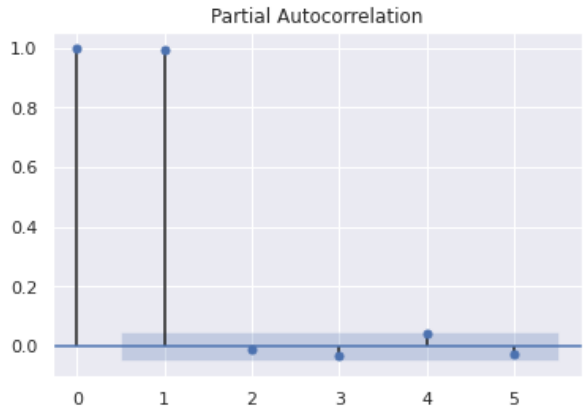
Again from the PACF, ACF plots the plots were shutting off after lag =1. So we have used p=1, q=1.

For the differencing part we first tried simple differencing and ran ADfuller test since the differenced time series was stationary according to AD test so we finalised using ARIMA model of order (p=1,d=1,q=1).

AD Fuller results

```
ADF Statistic: -63.78557921665654
p-value: 0.0
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
```

PACF plot



ARIMA was giving poorer results as compared to other models so decided to focus more on other models, also since ARIMA is a linear model so it is expected that more finely tuned non-linear models such as MLP will work better than ARIMA.

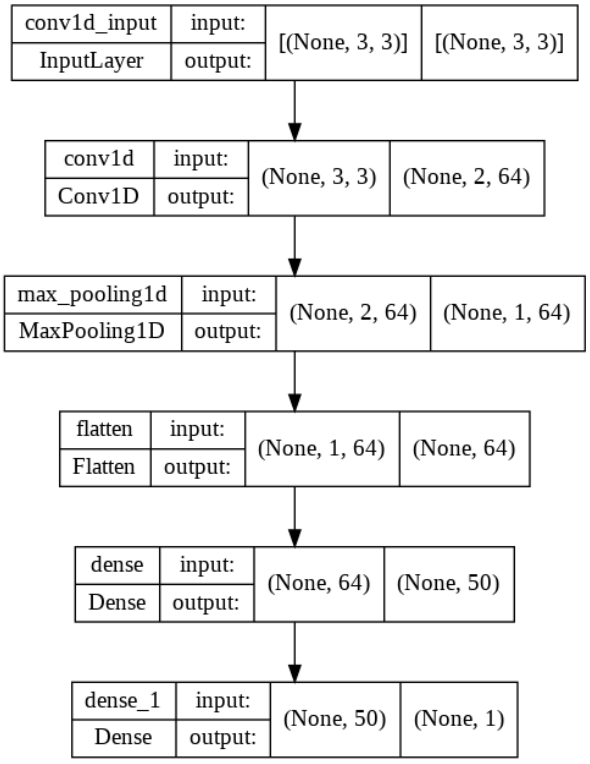
stock_symbol	Test RMSE
AAPL	0.188
MSFT	0.808
AMZN	191.628
TSLA	23.341
GOOGL	109.764
NVDA	1.507
BRK-B	0.561
FB	1.553

Sample plot with EMA(10), VWAP, Price

Architectures of models

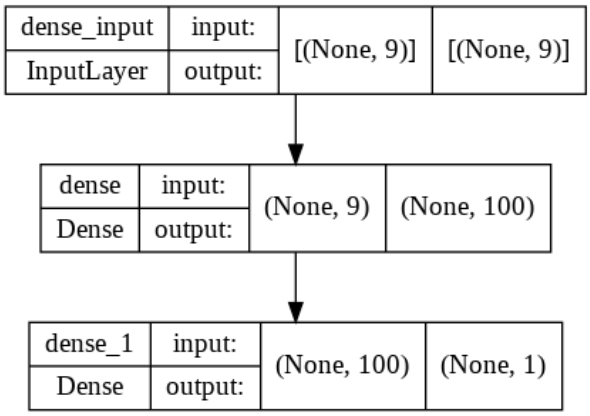
CNN

Using the lag values mentioned earlier we ran the CNN model on stock prices of 10 stocks with the following architecture

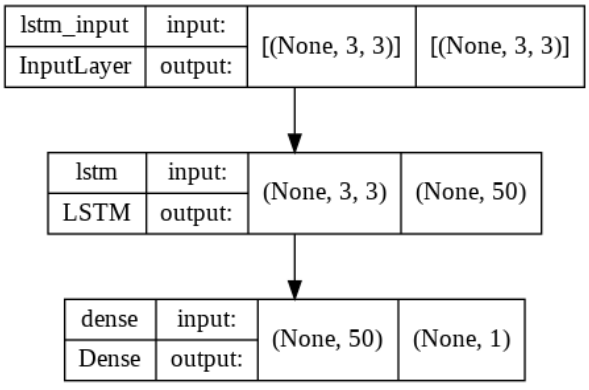


MLP

Again since we are using the same lag value for all the models, we also ran the stocks on MLP with 1 hidden layer of 100 neurons.



LSTM



We have used 2-3 variations of LSTM, above is the architecture of simple LSTM, we found bidirectional LSTM to work the best for us.

Given below is a sample image of the prices predicted by our model and the actual prices for CNN. Red legend is the actual price and blue legend is our predicted price

We can certainly say that our model was able to predict the trend of data for most of the cases if not the actual price value

CNN

stock_symbol	train_rmse	test_rmse	train_mape	test_mape
AAPL	0.18966	0.28142	0.07414	0.10955
MSFT	0.43442	0.76339	0.09734	0.16676
AMZN	5.38355	10.50368	0.09741	0.18189
TSLA	3.20554	3.85980	0.32571	0.33540
GOOGL	6.74588	9.73710	0.20111	0.26676
NVDA	0.78250	1.05153	0.18070	0.29252
BRK-B	0.39107	0.63105	0.09007	0.13602
FB	1.20862	1.34771	0.31825	0.37715
Average	2.29265	3.52196	0.17309	0.23325

MLP

stock_symbol	train_RMSE	test_rmse	train_mape	test_mape
AAPL	0.1856	0.2797	0.0805	0.1187
MSFT	0.4576	0.8287	0.1028	0.1851
AMZN	8.7015	15.6119	0.1691	0.3114
TSLA	2.9172	3.6003	0.2671	0.2964
GOOGL	4.3833	7.6320	0.1179	0.2004
NVDA	0.5508	0.6999	0.1591	0.2086
BRK-B	0.7627	1.1929	0.1989	0.2755
FB	0.8991	1.3197	0.1987	0.3030
Average	2.3572	3.8956	0.1618	0.2374

LSTM Bidirectional

stock_symbol	train_RMSE	test_rmse	train_mape	test_mape
AAPL	0.056	0.086	0.000	0.000
MSFT	0.085	0.169	0.000	0.000
AMZN	12.766	21.225	0.003	0.004
TSLA	2.823	3.742	0.002	0.003
GOOGL	7.357	12.794	0.002	0.003
NVDA	0.354	0.367	0.001	0.001
BRK-B	0.247	0.368	0.001	0.001
FB	0.150	0.860	0.000	0.002
Average	2.980	4.952	0.001	0.002

7. CONCLUSION

Based on average RMSE across all the stocks we can say that CNN performs the best for us. Multivariate was found to better than univariate even though we used only 2 extra features EMA, VWAP.

There are many such technical indicators commonly used by day traders so in future works one can try to include even more no. of features.

CNN,LSTM other such non-linear models were found to be better than linear model of ARIMA.

8. TIMELINE

Component	Plan
C1	Problem identification & Definition, Literature Survey, Proposed methodology formation
C2	Conduct of the experiment / Software designing, developing and implementing.
C3	Final Presentation, Demonstration, Parameters tuning.

8. REFERENCES

[1] Selvamuthu, D., Kumar, V. & Mishra, A. Indian stock market prediction using artificial neural networks on tick data. Financ Innov 5, 16(2019). https://doi.org/10.1186/s40854-019-0131-7

[2] Nayak, A., Pai, M. M. M., & Pai, R. M. (2016). Prediction Models for Indian Stock Market. Procedia Computer Science, 89, 441-449. https://doi.org/10.1016/j.procs.2016.06.096

[3]A. Shakva, A. Pokhrel, A. Bhattarai, P. Sitikhu and S. Shakva, "Real-Time Stock Prediction Using Neural Network," 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018, pp. 1-4, doi: 10.1109/CONFLUENCE.2018.8443057.

[4] J. Dan, W. Guo, W. Shi, B. Fang and T. Zhang, "Deterministic echo state networks based stock price forecasting", Abstract Appl. Anal., vol. 2014, pp. 1-6, Jun. 2014.

[5] Lu W., Li J., Li Y., Sun A., Wang J. "A CNN-LSTM-based model to forecast stock prices". Complexity, 2020 (2020), 10.1155/2020/6622927

[6] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, Stock Closing Price Prediction using Machine Learning Techniques, Procedia Computer Science, Volume 167, 2020, Pages, 599-606, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.326.

[7] M. M. R. Majumder, M. I. Hossain and M. K. Hasan, "Indices prediction of Bangladeshi stock by using time series forecasting and performance analysis," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019, pp. 1-5, doi: 10.1109/ECACE.2019.8679480.

[8] Z. Zhang, Y. Shen, G. Zhang, Y. Song and Y. Zhu, "Short-term prediction for opening price of stock market based on self-adapting variant PSO-Elman neural network," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017, pp. 225-228, doi: 10.1109/ICSESS.2017.8342901.

[9] Chen.J (2022, April 25) Exponential Moving Average (EMA). Investopedia. Retrieved May 15,2022 from https://www.investopedia.com/terms/e/ema.asp

[10] Fernando.J (2022, April 25) Volume-Weighted Average Price. Investopedia. Retrieved May 15,2022 from https://www.investopedia.com/terms/v/vwap.asp