Abstract

The stock market has intrigued everyone since time immemorial. Stock market is an area where extensive research takes place to predict future values of stocks to decide whether to invest in one stock or another to maximise profits. Accurately predicting the stock market is a challenging task and thus it requires extensive study and analysis of the pattern of various data. Nowadays, artificially intelligent machine learning algorithms along with statistical models are used to solve this challenge and predict the market values more accurately than earlier times. Numerous machine learning and deep learning algorithms can make quite an accurate prediction with minimised error possibilities.

We are trying to compare various machine learning and deep learning algorithms to predict accurate future values of stock prices.

In this project we used Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and a combination of RNN and LSTM models on different datasets like that of Facebook and Apple to find out which gives a more accurate prediction. We found that our hybrid combined model of RNN and LSTM provides the most accurate results and has the minimum root mean square error.

Table of Contents

Chapter 1

Introduction

The stock market is the aggregation of buyers and sellers of stocks which are also known as shares, which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately, such as shares of private companies which are sold to investors through equity crowdfunding platforms. Investment in the stock market is most often done through stock brokerages and electronic trading platforms or over-the-counter marketplaces that function under a defined set of regulations. Equities represent fractional ownership in a company, and the stock market is a place where investors can buy and sell ownership of such investable assets.

Stock market prediction is an attempt to predict the future value of an individual stock, a particular sector of the stock market or the market as a whole. Prediction and analysis of stock markets play a very important role in the world economy. An efficiently functioning stock market is considered critical to economic development, as it gives companies the ability to quickly access capital from the public. The stock market is a very dynamic and unsure field, so the stock market's prediction needless to say becomes a significant matter.

Stock market prediction is one of the most difficult tasks in the world. There are various causes for this like the instability, unreliability and volatility of the market and there are many dependent and independent variables which determine the price of any particular stock. Stock values are affected by current news, previous historical data of particular stocks, mood and sentiment of the public, commodity prices and other factors. All these factors make it onerous to anticipate the rise and fall of the market with great precision.

However, with the advancement of computational power in recent times and various machine learning algorithms and the ongoing stock market research, stock market prediction advancements have started and the future values can be forecasted faster with greater accuracy.

The motive of this project is to provide predictions of the stock market price values with good accuracy, for consecutive days, even during extreme market fluctuations.

Chapter 2

Literature Review

2.1 NSE Stock Market Prediction Using Deep-Learning Models

Hiransha Ma, Gopalakrishnan E.Ab, Vijay Krishna Menonab, Soman K.P*

In this piece of work, stock prices of two leading stock markets of the world, NSE and NYSE were predicted using four DL architectures. MLP, RNN, LSTM and CNN networks were used to predict stock price of of MARUTI, HCL and AXIS BANK from NSE stock market and the stock price of BANK OF AMERICA (BAC) and CHESAPEAKE ENERGY (CHK) from NYSE.In the proposed work , CNN has performed better than other three networks as it is capable of capturing the abrupt changes in the system.

EXPERIMENTAL DATASET : The experimental dataset that has been used is the National Stock Exchange (NSE) stock market dataset, specifically the NIFTY price index, ranging in the time frame of April 2008 to April 2018, collected from the National Stock Exchange (NSE) India. The recent dataset of NSE India
from 1 November 2019 to 5 August 2020 is taken into account as a case study.

MODEL STRUCTURE :

1. **Using the ANN with backpropagation algorithm**

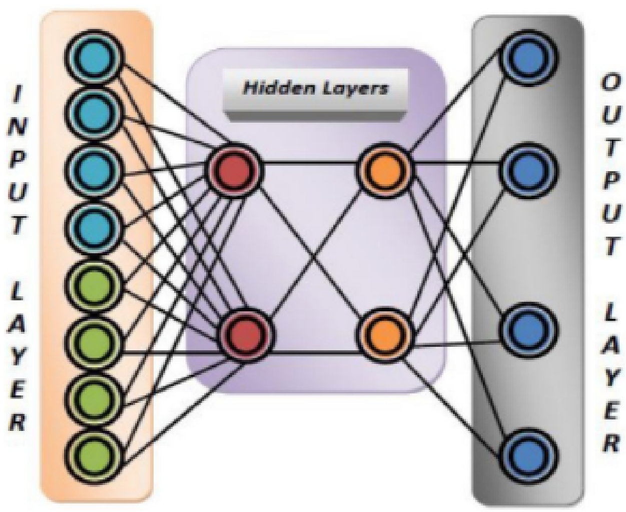


FIGURE 6 ANN model used in this prediction

The model involves predicting the 'Open,' 'Close,' 'High', and 'Low' values of NIFTY of the nth day based on these four features of the (n − 1)th and (n-2)th days. The input layer is followed by two hidden layers consisting of 2 neurons, each, in turn, followed by the output layer with four nodes to predict the following day's values.

$$accuracy_i(in\ \%) = 1 - \left(\frac{|target_i - output_i|}{target_i}\right) * 100$$

2. **Using the convolutional neural network with 2‑D histograms**

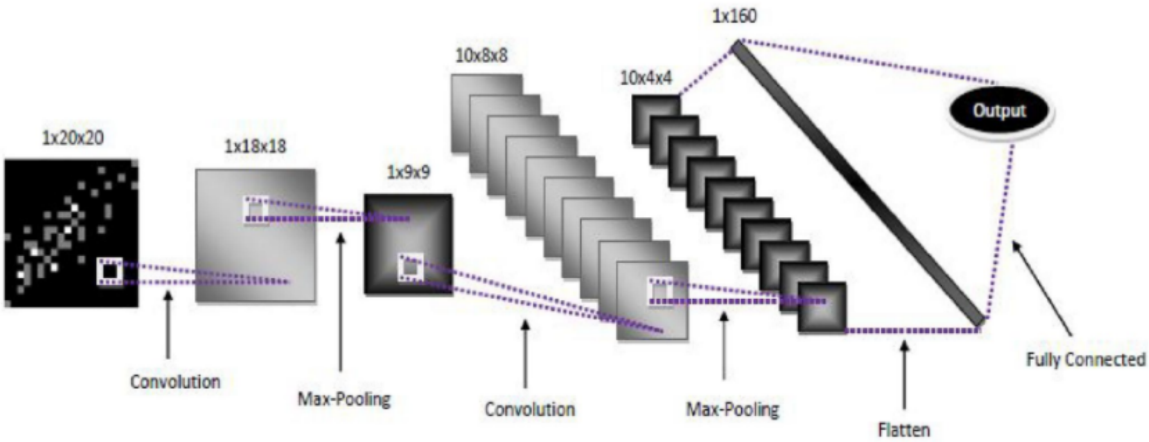


FIGURE 7 CNN model used in this study

This CNN model's training process was much faster and efficient than the ANN model proposed prior. 1000 input combinations and just about 150 epochs were sufficient for the required training. Both MSE and Mean Absolute Error (MAE) were used to judge the model's performance during the training.

$$MSE = \frac{1}{n}\sum (y - \hat{y})^2$$

$$MAE = \frac{1}{n}\sum |y - \hat{y}|$$

THE RESULTS - Artificial Neural Networks which have been used before for stock market prediction gives about 97.66% accuracy, whereas the Convolutional Neural Network model gives 98.92% accuracy on the dataset. The working of CNN model had proved to be better than that of the ANN model. This is because the ANN model loses context from the time series data before the CNN model. The graphs have been presented below.

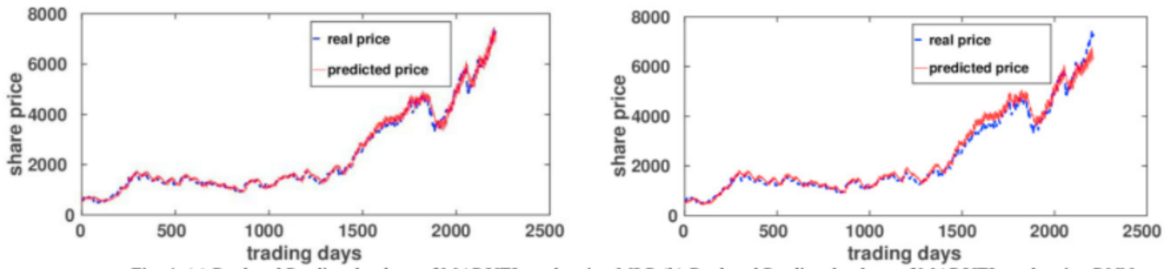


Fig. 4. (a) Real and Predicted values of MARUTI stock using MLP;(b) Real and Predicted values of MARUTI stock using RNN

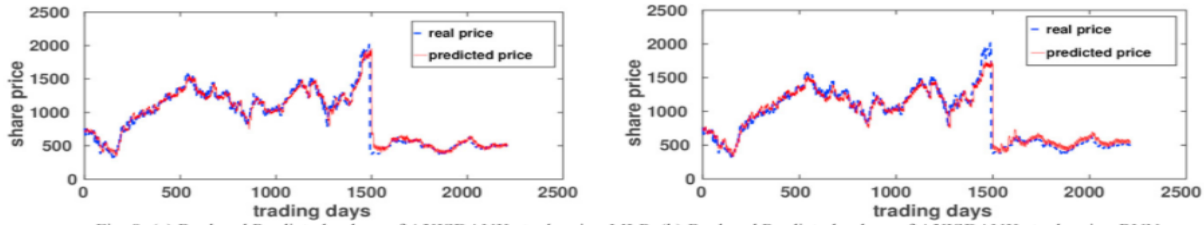


Fig. 8. (a) Real and Predicted values of AXISBANK stock using MLP; (b) Real and Predicted values of AXISBANK stock using RNN

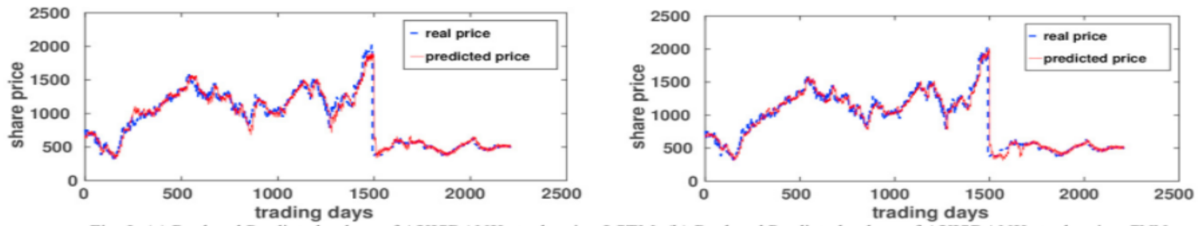


Fig. 9. (a) Real and Predicted values of AXISBANK stock using LSTM; (b) Real and Predicted values of AXISBANK stock using CNN

2.2 An Empirical Research and Comprehensive Analysis of Stock Market Prediction using Machine Learning and Deep Learning techniques Aryendra Singh et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012098

In this work, Linear regression, K-means Clustering, K nearest neighbour, LSTM, etc. has been used in stock prediction. The volatility of a trader's position has been predicted using Bollinger bands. It can be used to evaluate public opinion content and thus establish patterns/connections between public and industry employees.

DATASET : Data consist of returns, i.e. percentage change in close price of different companies divided into small, medium and large capitals with Indian Rupees currency. Below is the diagram of all the stocks taken into consideration.

Table1. Stocks with their respective sector of capitalisation

Small Caps	Mid-Caps	Large Caps
Ashoka	Adani	Adanisport
Bajaj	Ajantfarm	Asian
Bomdyeing	Amarajabat	Axis
Century	Apollo	Bajfinance
Fortis	Bergepaint	BPCL
Gujalkali	Castround	Cipla
IDFC	Cummins	Dreddy
Ircon	DHFL	Eicher
ITDC	Excide	Gail
Jetairways	Gimrin	HDFC

MODEL STRUCTURE:

1. **LSTM (Long-short term memory)** - Today's stock price will depend on; the pattern, the market observed in previous days, which will either be a downtrend or an uptrend, and the stock price the day before. LSTMs are capable of selectively retrieving patterns over long-time spans.

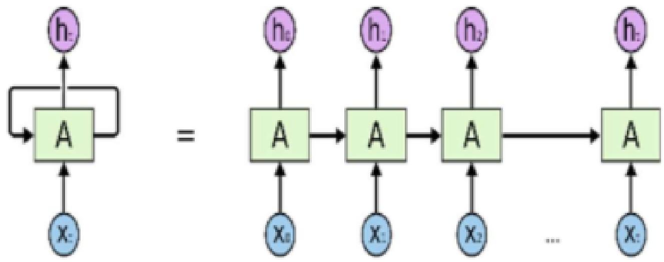


Figure 6. Architecture of LSTM

The work makes use of K-Means Clustering. It was applied on 30 stocks according to their mean annual Volatilities and Returns.This helped to identify the optimum number of clusters using the elbow curve method.

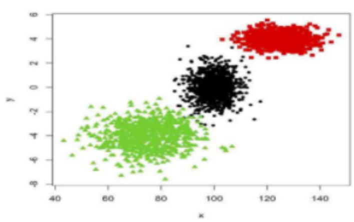


Figure 7. K-means Clustering

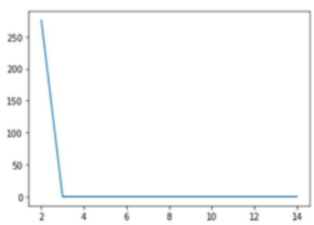


Figure 8. Elbow Curve.

THE RESULTS - The orange line predicted close value for the stock by test data. The green line is the predicted close price for the stock by test data (orange colour). Model efficiency is found to be 58.44 with 1/1 epoch, which is good because it is much less than 180.

Figure 12. LSTM prediction output

2.3 Stock market prediction using deep learning algorithms

Somenath Mukherjee, Bikash Sadhukhan, Nairita Sarkar2, Debajyoti Roy2, Soumil De

This proposed work makes use of two two different approaches, firstly, Feed-forward Neural Network and secondly, Convolution Neural Network. Feed-forward Neural Network provided an average prediction accuracy of 97.66%, but it required much training data and epochs. Whereas when Convolution Neural Network was utilised, an average prediction accuracy of 98.92% was obtained. It utilised grayscale 2-D histograms generated from time-series data for prediction.

THE EXPERIMENTAL DATASET - Dataset is taken from highly traded stocks of three different sectors which are Automobile, Banking and IT sectors
from NSE. The corresponding stocks from these sectors are Maruti, Axis bank, and Hcltech. Information like stock symbol , stock series, stock date and previous closing, opening ,high ,low ,last , closing and average prices, total traded quantity, turnover and number of trades.

MODEL STRUCTURE -

1. **FEED FORWARD NETWORK -** Feed forward network, also known as MLP is an example of a simple neural network. Each input neuron is linked to the succeeding hidden layer neurons through a weighted matrix. A Network has three sections of layers input, hidden and output layers.

Equation for activation function of an i th hidden neuron is given by -

$$h_i = f(u_i) = f\left(\sum_{k=0}^{K} w_{ki} x_k\right)$$

hi : i th hidden neuron , f(ui) : link function which provides non-linearity between input and hidden layer , wki : weight in the (k, i) th entry in a (K X N) weight matrix , xK : Kth input value

$$y_j = f(u'_j) = f\left(\sum_{i=1}^{N} w'_{ij} h_i\right)$$

y_j : j^{th} output value

2. **RECURRENT NEURAL NETWORK -** It takes input from two sources, one is from the present and the other from the past. Information from these two sources are used to decide how they react to the new set of data.
   Each input sequence has plenty of information and this information are stored in the hidden state of recurrent networks. This hidden information is recursively used in the network as it sweeps forward to deal with a new example.

$$h_t = g_n(W_{xh}X_t + W_{hh}h_{t-1} + b_h) \qquad ($$

where as h_t : hidden layer at t^{th} instant , g_n : function , W_{xh} : input to hidden layer weight matrix, X_t : input at t^{th} instant, h_{t-1} :hidden layer at $t - 1^{th}$ instant , b_h :bias or threshold value

hidden to output layer equation is given as :

$$Z_t = g_n(W_{hz}h_t + b_z)$$

whereas Z_t :output vector, W_{hz} :hidden to output layer weight matrix, b_z :bias or threshold

THE RESULTS - The work shows that the models are capable of identifying the patterns existing in both the stock markets. This shows that there exists an underlying dynamics, common to the stock markets.

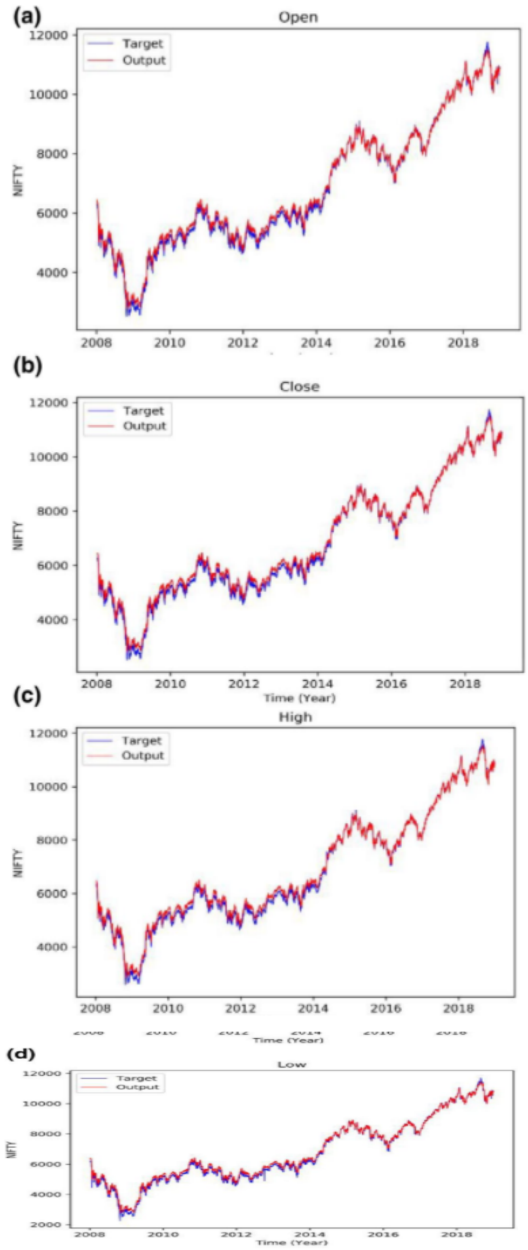


FIGURE 9 (a–d). The prediction of all the features based on the ANN model

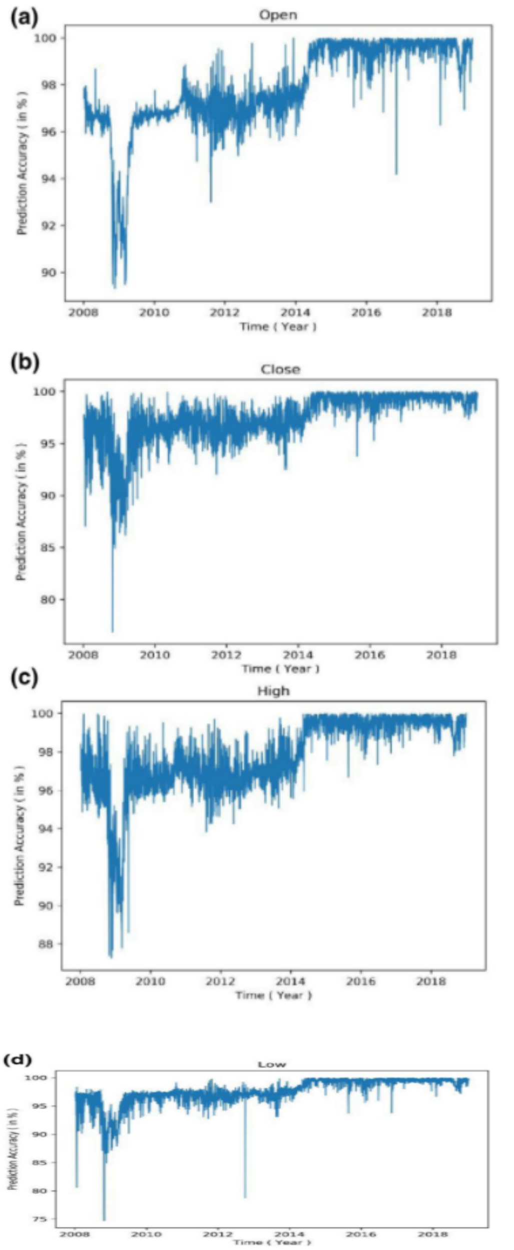


FIGURE 10 (a–d). The accuracy layup of all the features based on the ANN model

Chapter 3

Proposed Work

3.1 Dataset

Our approach to for this project consist of these major steps:

1. Dataset Creation
2. Implementing algorithms
3. Comparison of result and analysis

In this project, we used the following datasets:

1. Apple (27th May 2015 to 22nd May 2020)
2. Facebook (18th May 2012 to 24th March 2022)

The variables in our datasets are:

Close Open High Low Volume Adj Close Date

A brief description of the variables in our dataset is below:
Close: The closing price of stock on a particular day.
Open: The opening price of stock on a particular day.
High: The highest price that the stock rose to on a particular day.
Low: The lowest price that the stock fell to on a particular day.
Volume: The number of shares traded on a particular day.
Adj Close: The closing price on a particular day is adjusted for other corporate actions such as dividends.
Date: The specific date that corresponds to the data entry.

The Apple Dataset:

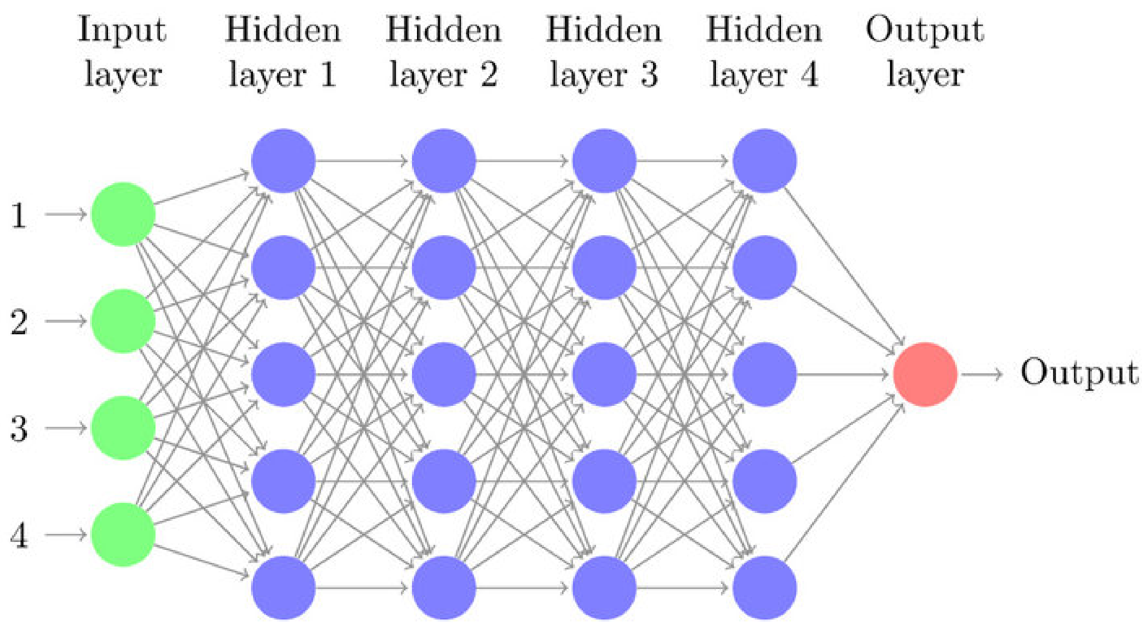
	Close	Open	High	Low	Volume	Adj Close	Date
0	132.045	130.34	132.260	130.05	45833246	121.682558	2015-05-27 00:00:00+00:00
1	131.780	131.86	131.950	131.10	30733309	121.438354	2015-05-28 00:00:00+00:00
2	130.280	131.23	131.450	129.90	50884452	120.056069	2015-05-29 00:00:00+00:00
3	130.535	131.20	131.390	130.05	32112797	120.291057	2015-06-01 00:00:00+00:00
4	129.960	129.86	130.655	129.32	33667627	119.761181	2015-06-02 00:00:00+00:00

The Facebook Dataset:

	Close	Open	High	Low	Volume	Adj Close	Date
0	38.230000	42.049999	45.000000	38.000000	573576400	38.230000	2012-05-18
1	34.029999	36.529999	36.660000	33.000000	168192700	34.029999	2012-05-21
2	31.000000	32.610001	33.590000	30.940001	101786600	31.000000	2012-05-22
3	32.000000	31.370001	32.500000	31.360001	73600000	32.000000	2012-05-23
4	33.029999	32.950001	33.209999	31.770000	50237200	33.029999	2012-05-24

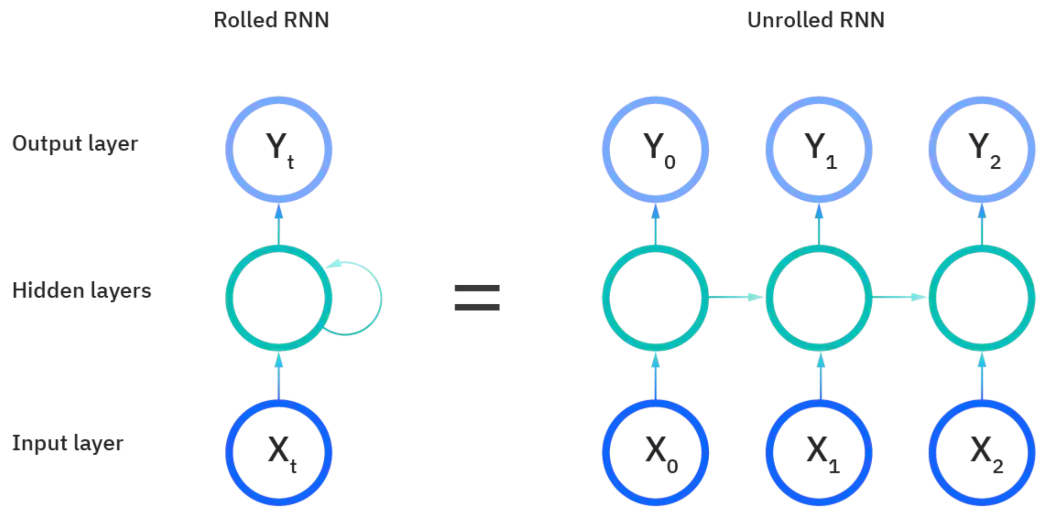
3.2 Algorithms Used

3.2.1 MLP[3]: MLP stands for Multilayer Perceptron. Multi layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers—the input layer, output layer and hidden layer. The input layer receives the input signal to be processed. The required task such as prediction and classification is performed by the output layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the true computational engine of the MLP. Similar to a feed forward network in a MLP the data flows in the forward direction from input to output layer. The neurons in the MLP are trained with the back propagation learning algorithm. MLPs are designed to approximate any continuous function and can solve problems which are not linearly separable. The major use cases of MLP are pattern classification, recognition, prediction and approximation.



Multi-Layer Perceptron (MLP) diagram with four hidden layers and n inputs

3.2.2 RNN[3]: **Recurrent Neural Network(RNN)** is a type of Neural Network where the **output from the previous step is fed as input to the current step**. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is the Hidden **state**, which remembers some information about a sequence.

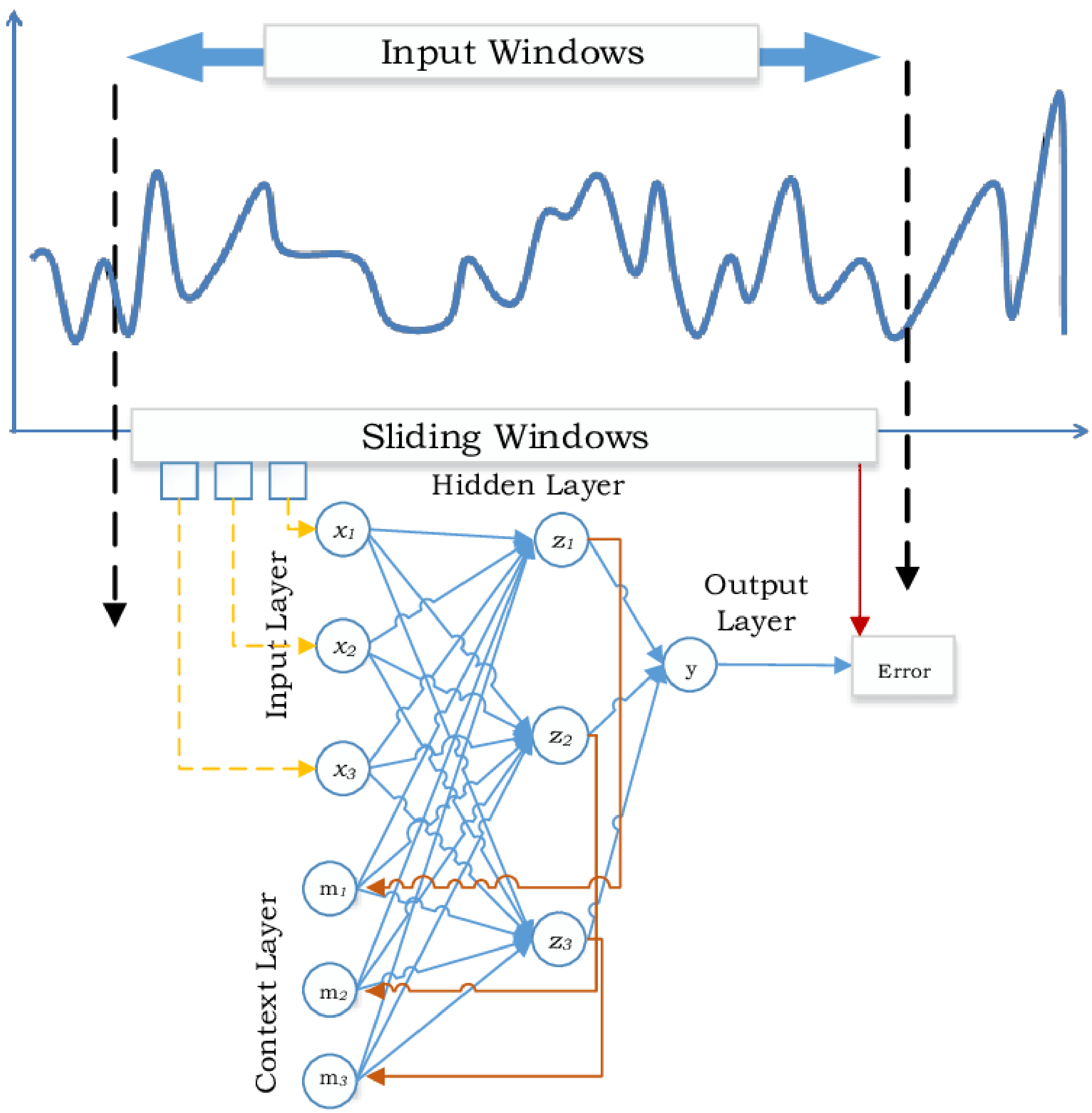


RNN has a **"memory"** which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

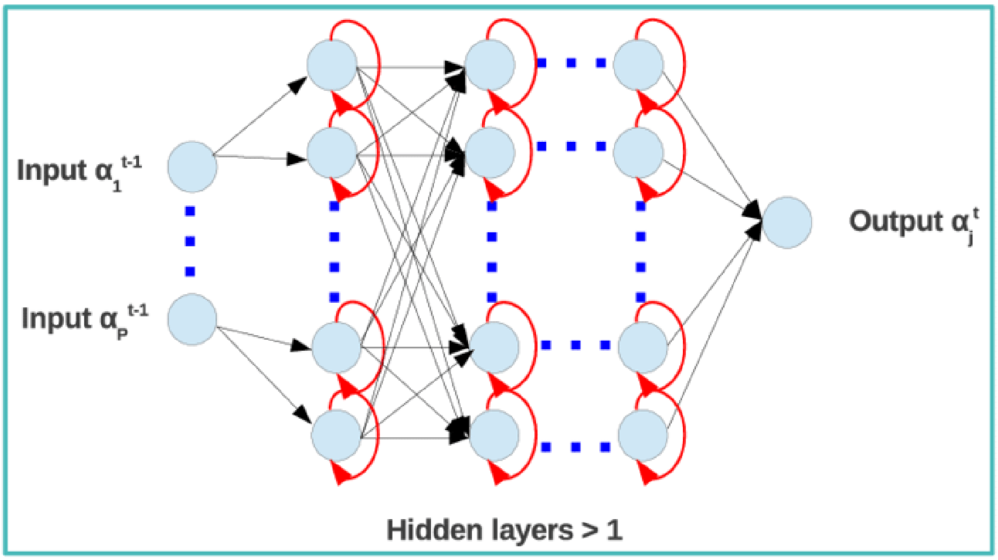
3.2.3 Sliding Windows Technique:

Sliding windows technique is a kind of processing method of concept drift in data streams, which has many applications in intrusion detection. The essence of this technique is the data update mechanism. The data stream (x) is divided into several parts of data blocks. When the sliding window moves to the next block, a new block is added to the window at intervals, and the oldest block is deleted. Through this dynamic sample selection method, the sample for modelling is updated [23]. The technique of sliding windows comes with a particular size of window; and this impacts the size of sample data.

Suppose that hx0, x1, x2, . . . , xn−1, xn, xn+1, . . .i is a series of time-series data. When the window size is fixed at k, the data interval will be changed to hxi−k, xi−k+1, . . . , xi , xi+1i and has different data streams from older data streams.



3.2.4 LSTM[2]: LSTM stands for Long Short Term memory. It is the building block of a neural network (like a perceptron). LSTM blocks are used to build a recurrent neural network. An RNN is a type of neural network where the output of a block is fed as input to the next iteration. An LSTM block is composed of four main components: a cell, an input gate, an output gate and a forget gate. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Each of the three gates can be thought of as a "conventional" artificial neuron, as in a multi-layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation "gate". There are connections between these gates and the cell. Some of the connections are recurrent, some of them are not.
As mentioned earlier stock prediction is a time series problem. LSTM can be used for time series predictions. LSTM doesn't have the vanishing gradient problem which a traditional RNN has.



3.3 RNN & LSTM Combined Model:

We take the predicted values of the individual days that have been calculated by running RNN and LSTM models on our datasets and find the average of individual days.

The new predicted values obtained from the combined RNN and LSTM model give a lower Root Mean Square Error (RMSE) than the values we got from MLP, RNN and LSTM models.

Thus, our proposed combined model provides a better result than the already existing models.

Algorithm for the combined RNN & LSTM model :

1. Download the dataset.
2. Read the dataset/dataframe.
3. Do data cleaning i.e remove the null values and outliers from the dataset.
4. Take our target variable i.e closing price from the dataset.
5. Create some independent variables using that dataset using sliding window technique.(Our window size is 15)
   For example -
   Suppose our last 5 days closing price are 128 140 120 155 117.
   Then the dataframe should look like this →

X1	X2	X3	y
128	140	120	155
140	120	155	117

6. Scale the whole dataframe so that all the values are between 0 and 1.
7. Split the dataframe Train and Test set. (We used 80% for the training set and 20% for the testing set.)
8. Take two models i.e one RNN and one LSTM and train those models with the Training set.(For better results take at least 100 epochs)
9. Predict the values with the Test set for both the models.
10. Use inverse scaling on the predicted values to get the predicted stock prices.
11. Take the average of RNN and LSTM predicted values to get a good accuracy.
12. Plot the necessary graphs and also print the original vs predicted values side by side to check the difference.

Chapter 4

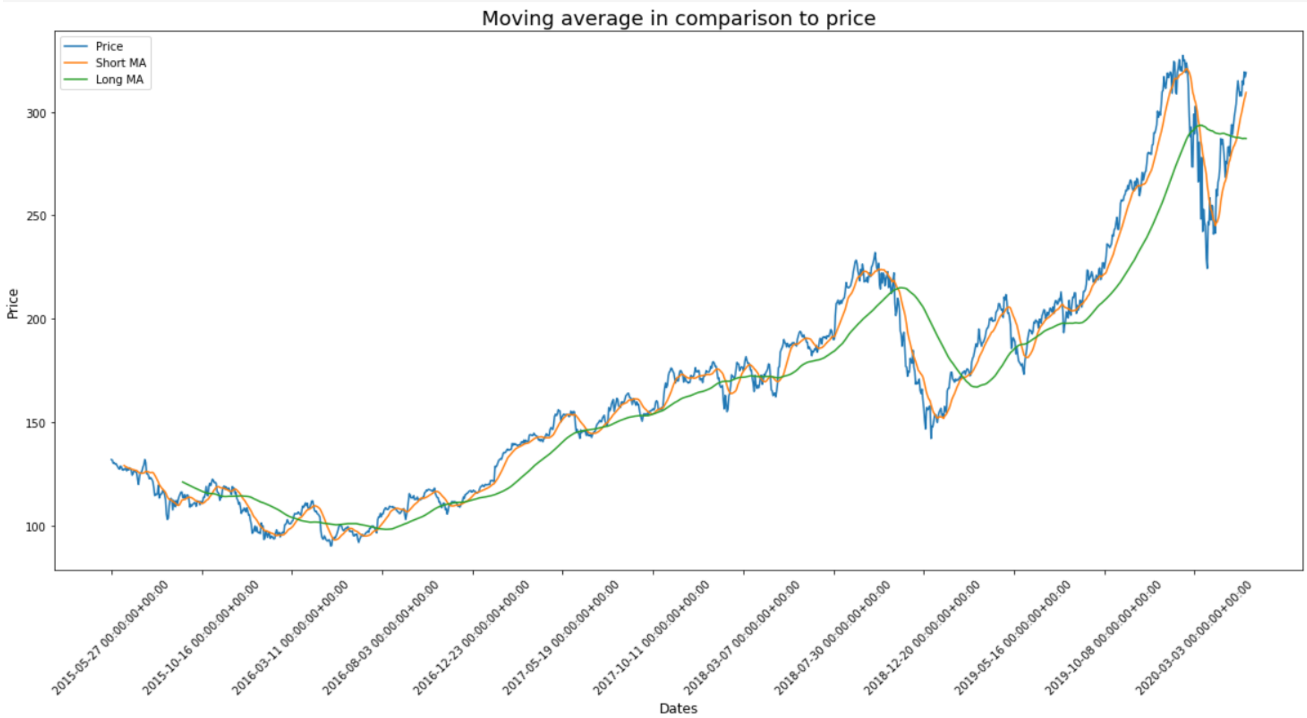
Result and Discussions

4.1 Overview

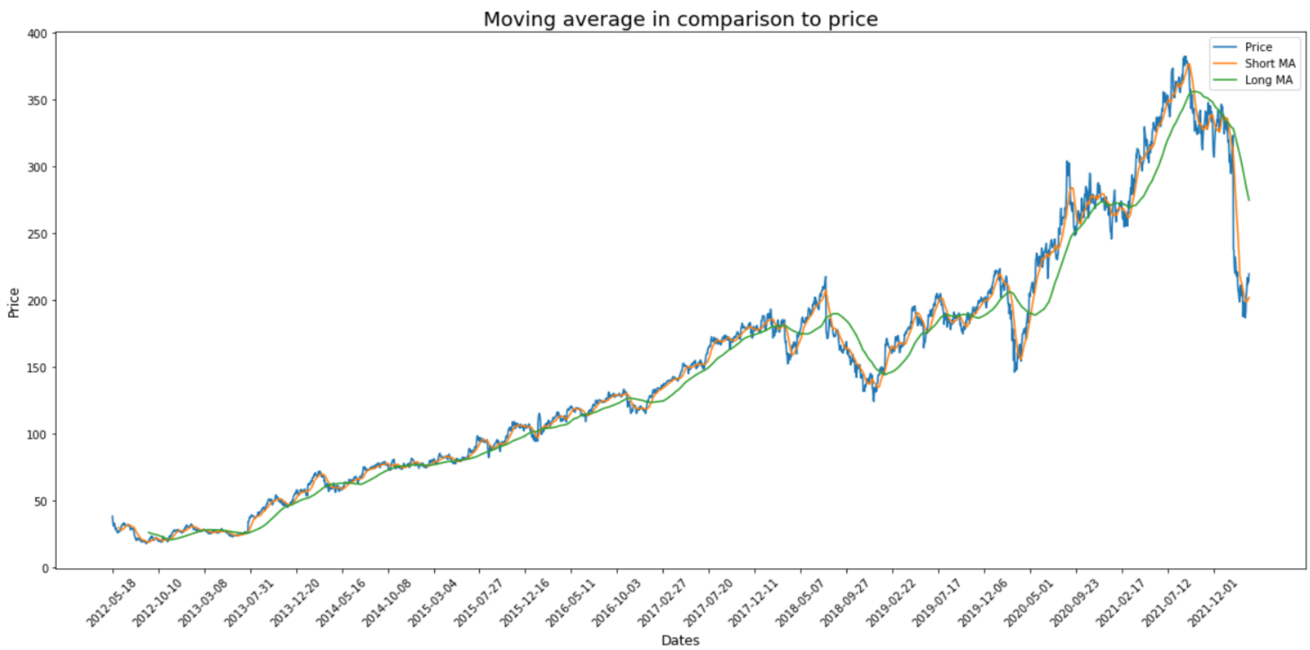
In this project, we worked on Apple dataset and Facebook Dataset and applied MLP model, RNN model, LSTM model and a hybrid combined model of RNN and LSTM.
We trained our models using 80% of the dataset and used the remaining 20% to test and predict the stock market prices.
It is worth noting that the models we built are more relevant to short-term stock predictions. We designed the MLP, RNN, and LSTM models to predict today's price based on the last t-1 days of prices.
The MLP model had the highest RMSE and so had the least accurate prediction and the combined RNN and LSTM model had the lowest RMSE and so had highest prediction accuracy. If we compare the RNN and LSTM models, both performed effectively.
Thus, by using RMSE to judge the accuracy, we concluded that our combined RNN and LSTM model outperformed all the other three models.

4.2 Moving Average to prices graphs

The **moving average to prices graph of the Apple Stock** data is as shown below:

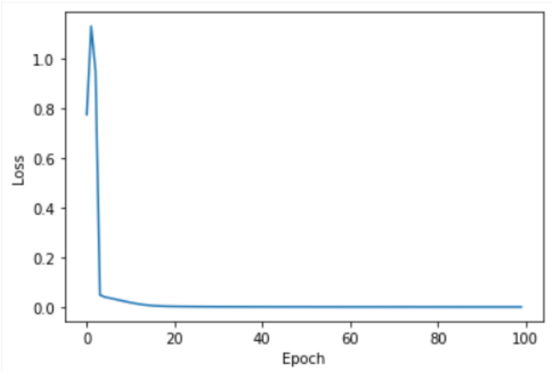


The **moving average to prices graph of the Facebook Stock** data is as shown below:



4.3 The results obtained from MLP model on the Apple Dataset:

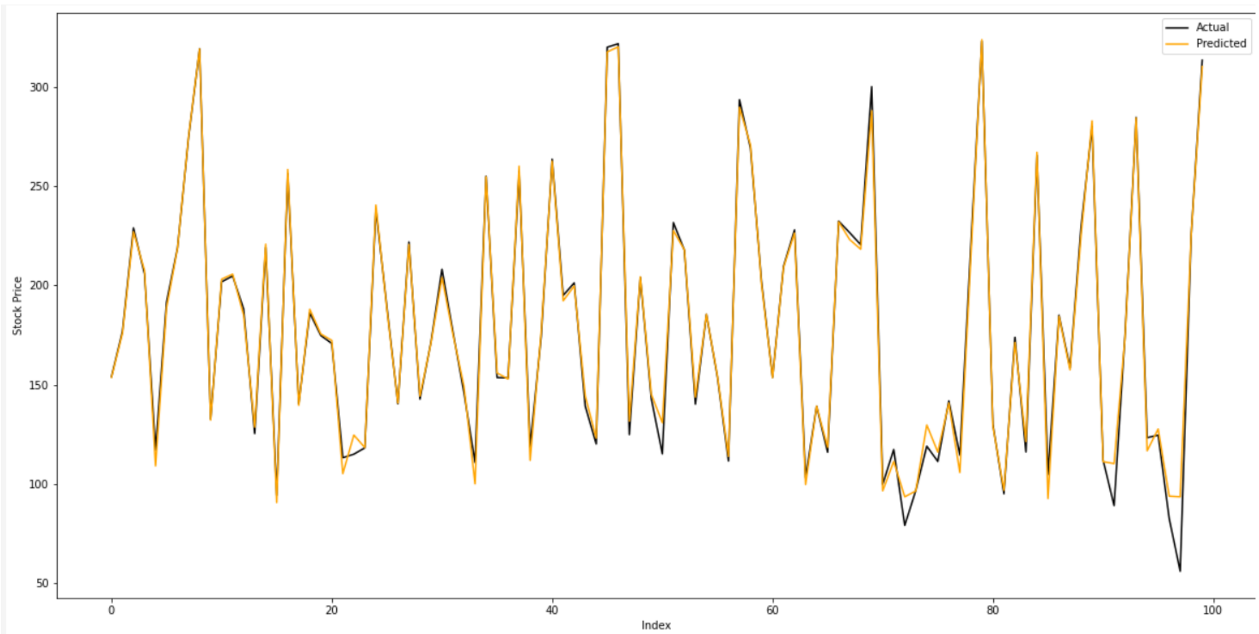
The epoch loss while training the MLP model:



Some of the values of the actual prices and predicted prices using MLP model are:

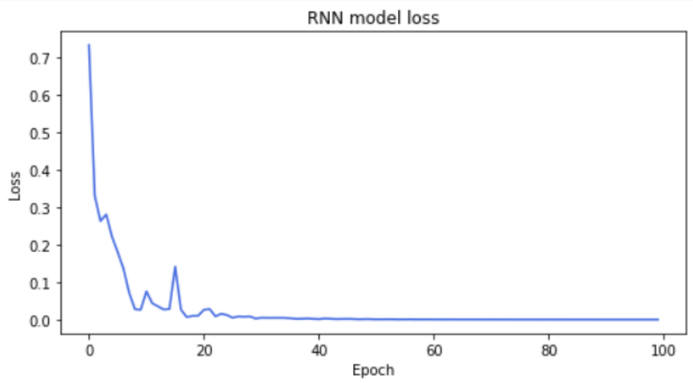
	Actual_Data	Predicted_Data
0	153.898483	153.610001
1	176.804367	175.820007
2	228.969345	227.059998
3	205.843826	207.479996
4	117.018768	108.990005
5	191.516998	188.740005
6	219.016861	218.719986
7	274.625671	275.149994
8	319.260101	319.230011
9	132.662247	132.040009
10	201.802917	203.130005
11	204.727066	205.660004
12	188.077530	184.919998
13	125.299454	128.529999
14	219.125183	220.790009

The graph obtained for the testing on the **Apple Dataset using MLP model**:



4.4 The results obtained from RNN model on the Apple Dataset:

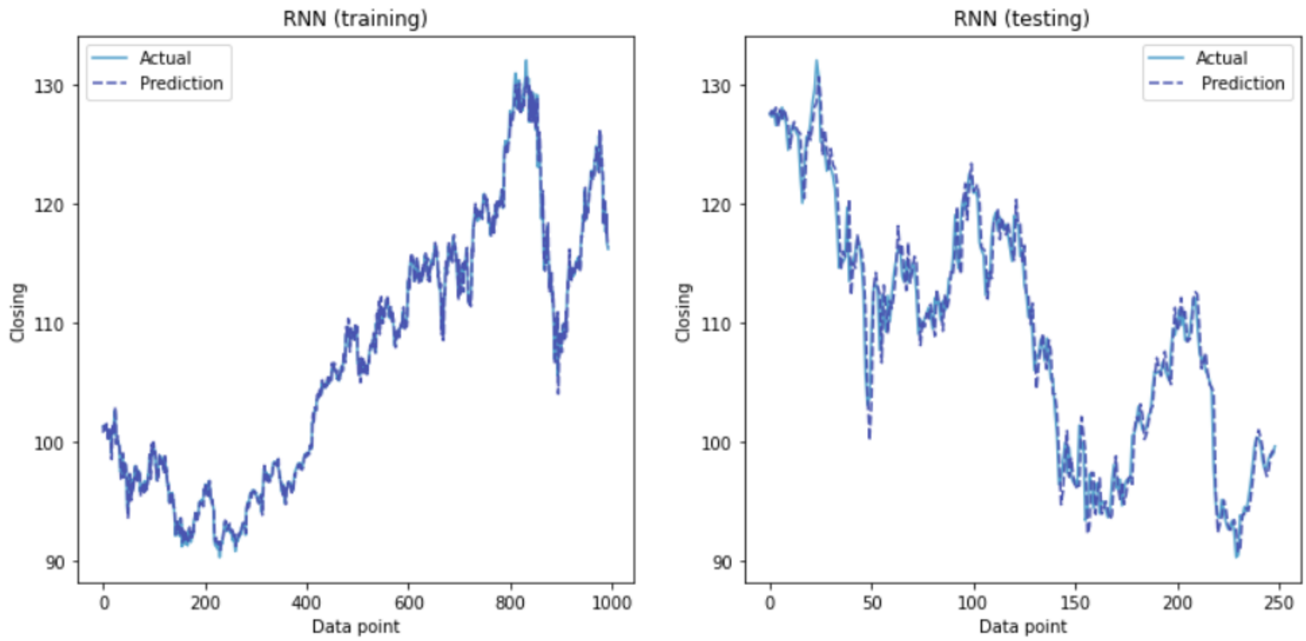
The training epoch loss while using RNN model:



Some of the values of the actual prices and predicted prices using RNN model are:

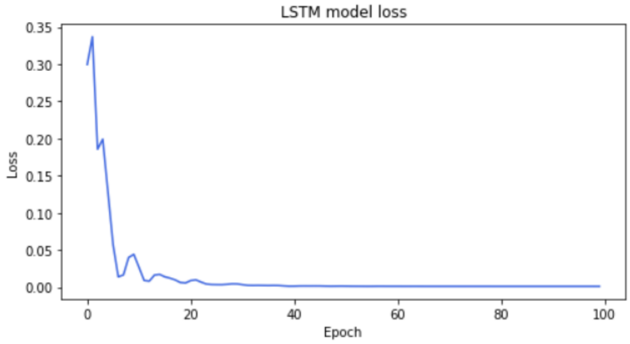
	Original_Test_Data	Predicted_Test_Data
0	127.599998	127.434067
1	127.299995	127.843063
2	127.879997	127.305939
3	126.599998	128.089493
4	127.610008	126.479012
5	127.029999	127.909187
6	128.110001	127.036613
7	127.500000	127.998718
8	126.749992	127.400414
9	124.529999	126.709038

The graph obtained for the testing on the **Apple Dataset using RNN model**



4.5 The results obtained from LSTM model on the Apple Dataset:

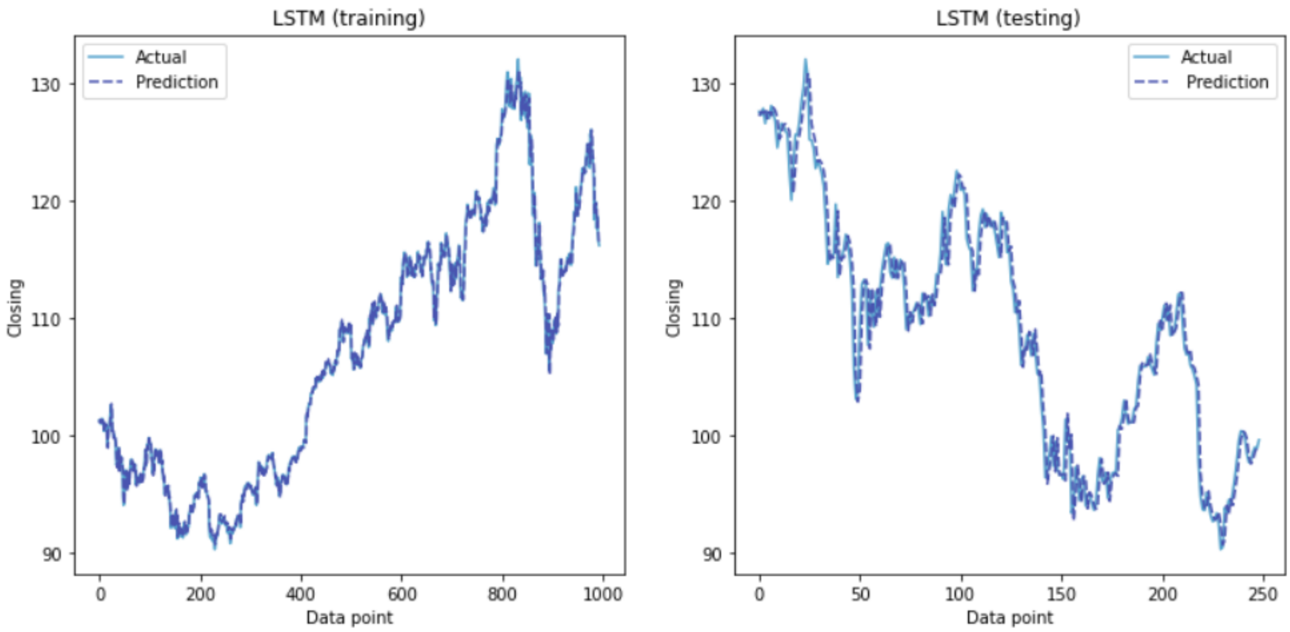
The epoch loss while training the LSTM model:



Some of the values of the actual prices and predicted prices using LSTM model are:

	Actual_Data	Predicted_Data
0	127.599998	127.331680
1	127.299995	127.726257
2	127.879997	127.389969
3	126.599998	127.970398
4	127.610008	126.751694
5	127.029999	127.737366
6	128.110001	127.152069
7	127.500000	127.979294
8	126.749992	127.544968
9	124.529999	126.906479
10	125.425003	124.981705
11	126.599998	125.825287
12	126.439995	126.686615
13	126.000008	126.474190
14	125.689995	126.171860

The graph obtained for the testing on the **Apple Dataset using LSTM model**:



4.6 Combined RNN and LSTM model on Apple Dataset:

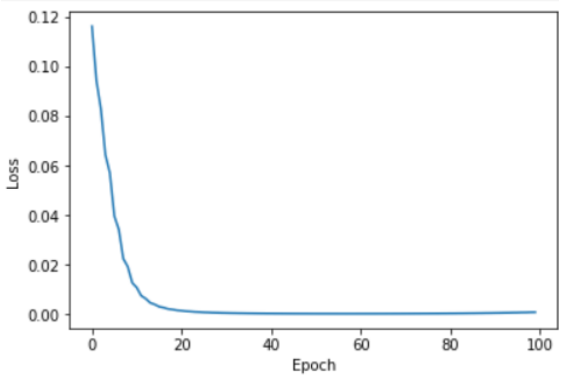
	Actual_Data	Predicted_Data
0	127.599998	127.331680
1	127.299995	127.726257
2	127.879997	127.389969
3	126.599998	127.970398
4	127.610008	126.751694
5	127.029999	127.737366
6	128.110001	127.152069
7	127.500000	127.979294
8	126.749992	127.544968
9	124.529999	126.906479
10	125.425003	124.981705
11	126.599998	125.825287
12	126.439995	126.686615
13	126.000008	126.474190
14	125.689995	126.171860

The RMSE of the combined RNN and LSTM Model on the APPLE Dataset is:

```
Combined RNN and LSTM Test RMSE: 1.93
```

4.7 <u>The results obtained from MLP model on the Facebook Dataset</u>:

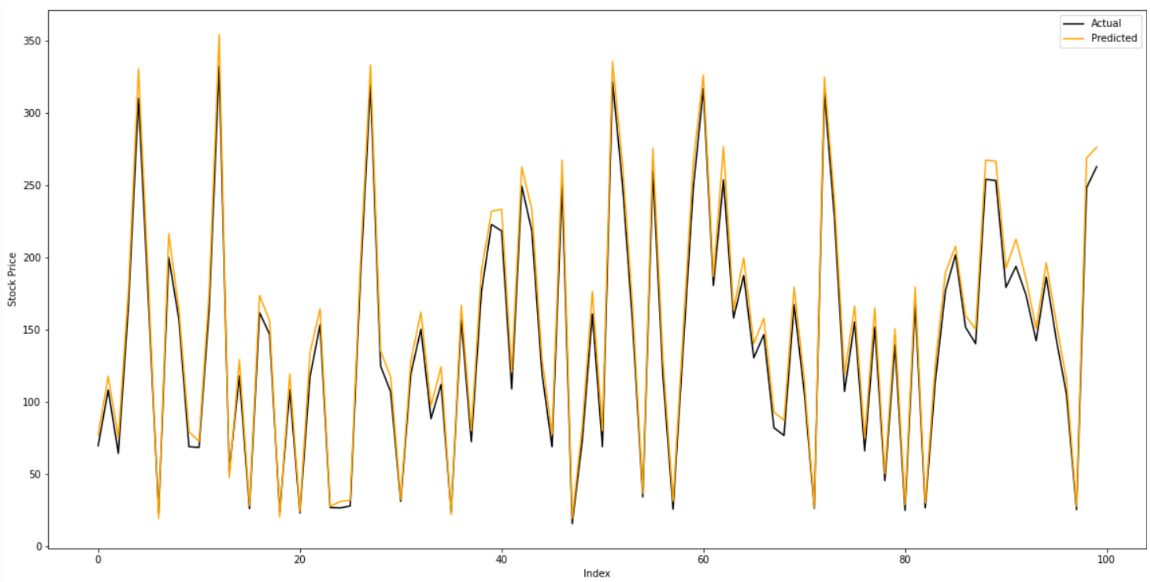
The epoch loss while training the MLP model on Facebook dataset:



Some of the values of the actual prices and predicted prices using MLP model are:

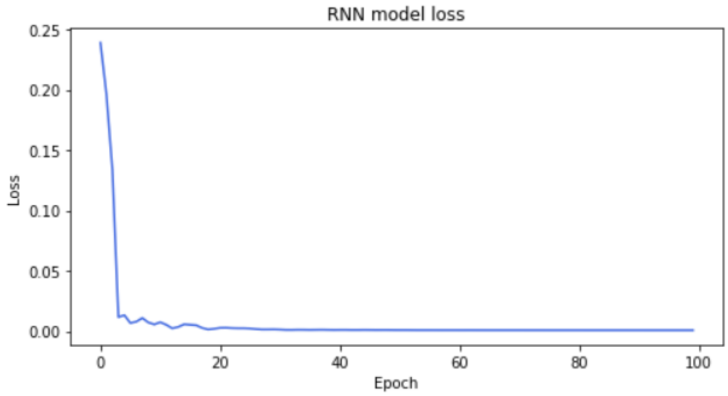
	Actual_Data	Predicted_data
0	69.636139	77.080002
1	108.235771	117.769997
2	64.309975	75.190002
3	166.283218	177.100006
4	310.426422	330.559998
5	168.209122	180.250000
6	22.988525	19.090000
7	200.085159	216.649994
8	157.412033	164.460007
9	68.963860	79.360008
10	68.436302	72.650002
11	163.575027	174.979996
12	332.656891	354.390015
13	51.624161	47.560001
14	118.023109	129.369995

The graph obtained for the testing on the **Facebook Dataset using MLP model**:



4.8 <u>The results obtained from RNN model on the Facebook Dataset</u>:

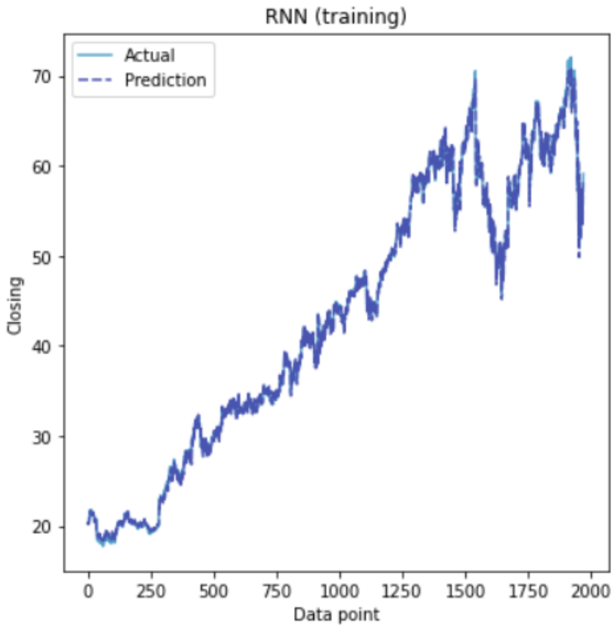
The epoch loss while training the RNN model on Facebook dataset:

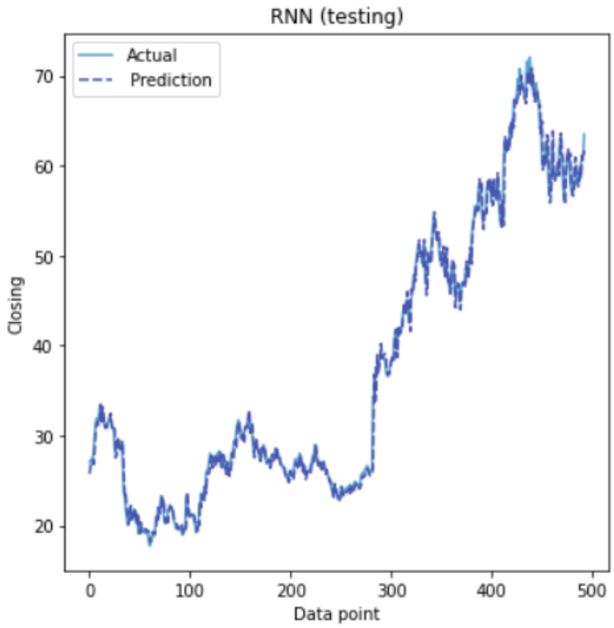


Some of the values of the actual prices and predicted prices using RNN model are:

	Original_Test_Data	Predicted_Test_Data
0	27.100000	25.734173
1	27.010000	26.723417
2	27.400002	27.163511
3	27.270000	27.194288
4	28.290003	26.786028
5	30.010000	28.266325
6	31.410000	30.407404
7	31.910000	31.360336
8	31.600002	31.511850
9	31.840000	31.152422

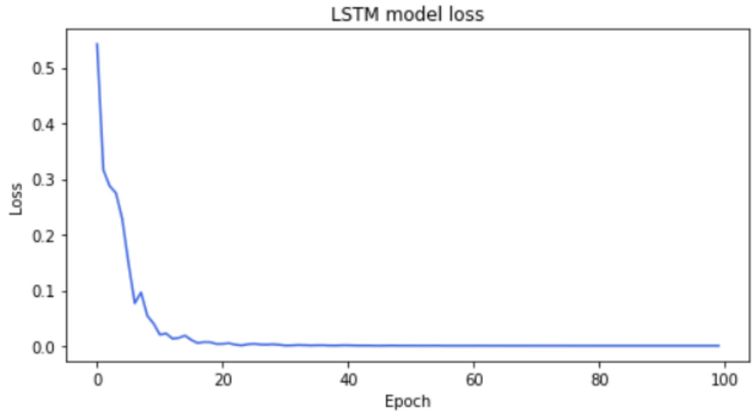
The graph obtained for the testing on the **Facebook Dataset using RNN model**:





4.9 The results obtained from LSTM model on the Facebook Dataset:

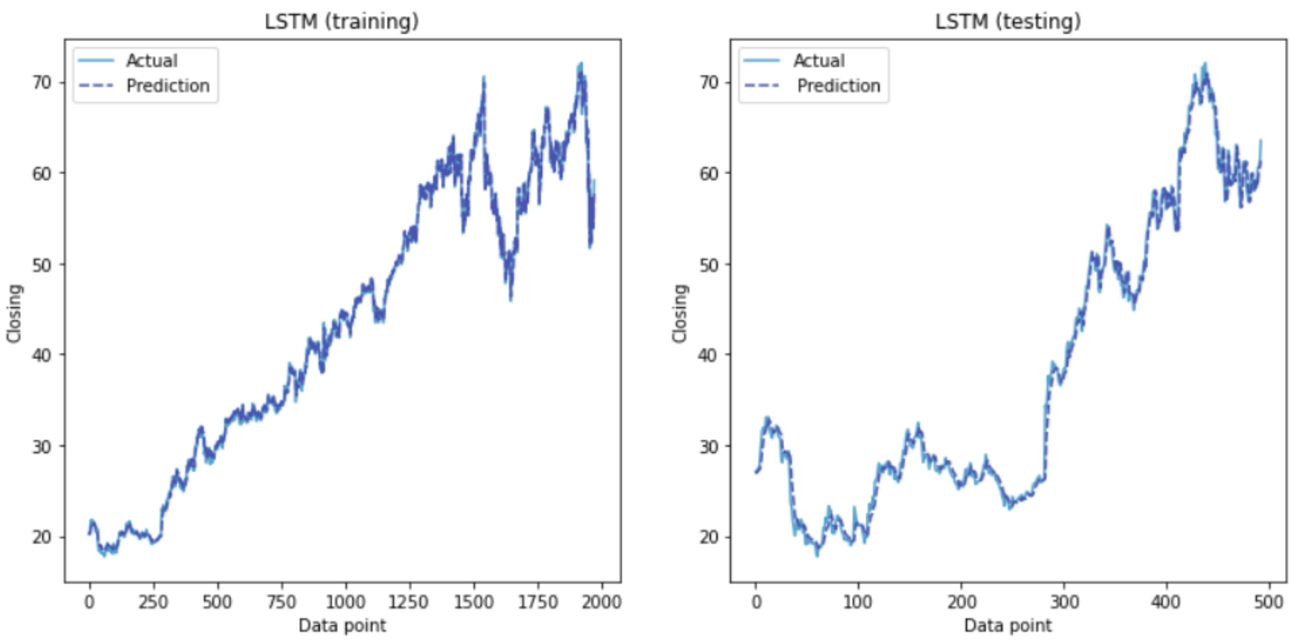
The epoch loss while training the LSTM model on Facebook dataset:



Some of the values of the actual prices and predicted prices using LSTM model are:

	Original_Test_Data	Predicted_Test_Data
0	27.100000	26.831661
1	27.010000	27.049294
2	27.400002	27.087410
3	27.270000	27.294392
4	28.290003	27.327789
5	30.010000	27.863710
6	31.410000	29.004183
7	31.910000	30.267710
8	31.600002	31.115133
9	31.840000	31.369257

The graph obtained for the testing on the **Facebook Dataset using LSTM model**:



4.10 <u>Combined RNN and LSTM model on Facebook Dataset:</u>

	Actual_Data	Predicted_Data
0	27.100000	26.282917
1	27.010000	26.886356
2	27.400002	27.125462
3	27.270000	27.244339
4	28.290003	27.056908
5	30.010000	28.065018
6	31.410000	29.705793
7	31.910000	30.814022
8	31.600002	31.313492
9	31.840000	31.260839
10	33.049999	31.761137
11	32.060001	32.961205
12	33.099998	31.921715
13	32.230000	32.989517
14	31.360003	32.406174

The RMSE of the combined RNN and LSTM Model on the APPLE Dataset is:

```
Combined RNN and LSTM Test RMSE: 1.21
```

4.11 RMSE Comparison

The metric for comparison used is Root Mean Square Error(RMSE). Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Stock prediction is a regression problem rather than a classification problem, therefore, RMSE is the most suitable metric compared to confusion matrices. The RMSE is a measure of standard deviation of the prediction errors, therefore, a smaller RMSE indicates better performance.

The formula of RMSE can be written as:

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Where:

- Σ = summation
- (zfi – Zoi)2 = differences, squared
- N = sample size

RMSE Comparison Table for different Models on Apple Dataset

Model	Root Mean Square Error(RMSE)
MLP	6.39
RNN	2.00
LSTM	1.94
RNN and LSTM Combined	1.93

RMSE Comparison Table for different Models on Facebook Dataset

Model	Root Mean Square Error(RMSE)
MLP	10.64
RNN	1.25
LSTM	1.29
RNN and LSTM Combined	1.21

Chapter 5

Conclusion

Our project work was carried out to test the time series parameters and to analyse how stock market prediction actually works. As a whole project, it is all about predicting the closing prices of the top MNCs around the world (we used Apple and Facebook stocks here). Our main focus with this project was to lower the Root Mean Square Error to get the closing stock prices values closer to the original values while predicting the closing prices of the stocks. After applying deep learning models like MLP, RNN and LSTM we got some good scores. We tried to combine LSTM and RNN to average the error and we can see that the combination of LSTM and RNN performs better than all 3 models we tested. So we can conclude that LSTM & RNN combined is better than all three algorithms we took for testing. The takeaway from our project is - these approaches would be a great help for brokers and investors to invest money in the stock market because they are based on an immense range of historical data after being checked on sample data. The analytics can also be used to evaluate public opinion content and thus establish patterns/connections between public and industry employees. Hence, evaluating the outcomes through this project, we can see that technology also has a fair distance to go before it is completely capable of resolving the stock markets' mystery.

Chapter 6

References

1. Stock Market Prediction using Machine Learning Techniques by Mehak Usmani,Syed Hasan Adil,Kamran Raza,Syed Saad Azhar Ali
2. Mukherjee, S., et al.: Stock market prediction using deep learning algorithms. CAAI Trans. Intell. Technol. 1–13 (2021)
3. NSE Stock Market Prediction Using Deep-Learning Models Hiransha Ma, Gopalakrishnan E.Ab , Vijay Krishna Menonab, Soman K.P.
4. An Empirical Research and Comprehensive Analysis of Stock Market Prediction using Machine Learning and Deep Learning techniques Aryendra Singh et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012098.
5. Shi Yan, Understanding LSTM and its diagrams[Online] - Available at: https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714
6. Erdin¸c Altay, M. Hakan Satman, Stock Market Forecasting: Artificial Neural Network and Linear Regression Comparison in An Emerging Market
7. Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall, Activation Functions: Comparison of Trends in Practice and Research for Deep Learning
8. Christopher Olah, Understanding LSTM Networks - Available at:https://colah.github.io/posts/2015-08-Understanding-LSTMs
9. Gaurav Singhal, Introduction to LSTM Units in RNN - Available at: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn
10. H Tawfeig; Vijanth S. Asirvadam; Nordin Saad, Sliding-window learning using MLP networks with data store management.