Project report

on

Algorithmic Trading

A Dissertation submitted in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science & Engineering

Submitted by

L. Bharadwaj
18H51A0517

M.M. Prathyush
18H51A0518

G. Vineetkumar
18H51A0596

Under the esteemed guidance of

DR. Sarat Chandra Nayak
Assoc. Professor, Dept. of CSE

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2018- 2022

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project phase - 1 report entitled " **Algorithmic Trading** " being submitted by **L. Bharadwaj (18H51A0517), M. M. Prathyush (18H51A0518), and G. Vineetkumar (18H51A0596)** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Dr. Sarat Chandra Nayak	**Dr. K. Vijaya Kumar**
Associate Professor	**Professor and HOD**
Dept. of CSE	**Dept. of CSE**

Submitted for viva voice Examination held on ___________

External Examiner

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Algorithmic trading is a method of executing orders using automated pre-programmed trading instructions for the things such as time and price. This type of trading attempts to boost the speed and computational resources of computers relative to human traders. These encompass a variety of trading strategies, some of which are based on formulas and results from mathematical finance, and often rely on specialized software.

Stock market is a probabilistic trading where we can gain and have possibility of loss. Investing in stock market trading might be a risky task. So, we can use algorithmic trading which uses the previous trends of a particular stock and help us predicting investing in the stock. Algorithmic trading using machine learning techniques to increase the probability of profit because it uses the technical analysis of stock, price action strategies, seasonal trends and help us to predict which time is better to invest in stocks. So, we will not fall in debts anymore. We are developing an algorithm using machine learning and deep learning techniques like SMA (Simple Moving Average), Arima, RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory) for predictions.

CHAPTER 1

INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities, and derivatives over virtual platforms. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems (ATS) that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, to evaluate and control the performance of ATSs, the implementation of risk strategies and safety measures applied based on human judgements are required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analyzed.

Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period to predict the result in the next time unit. Many timeseries prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Stock market is a typical area that presents time-series data and many researchers' studies on it and proposed various models. In this project, LSTM model is used to predict the stock price.

1.1 MOTIVATION FOR WORK

Stock price prediction is an attractive area where we gain more profits to develop the business. With a successful model for stock prediction, we can gain insight about market behavior over time. Machine learning technique will be an efficient method to solve the problem of predicting stock movements.

1.2 OBJECTIVE

- To design efficient system for predicting the stock market movement using ML.
- Specifically, use LSTM method to Predict stock price based on the historical data.
- Compare existing methods to show the superiority of LSTM.

1.3 PROBLEM STATEMENT

Time Series forecasting & modelling plays an important role in data analysis. Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. Time Series is being widely used in analytics & data science. Stock prices are volatile in nature and price depends on various factors. The main aim of this project is to predict stock prices using Long short-term memory (LSTM).

P/E * EPS = PRICE

If we can predict stocks future P/E and EPS, we will know accurate PRICE

1.4 SCOPE & LIMITATIONS

1.We predict the stock market up to one month.

2.It is hard to predict in the pandemic situations

CHAPTER 2

LITERATURE SURVEY

2.DOMAIN INTRODUCTION

TRADING: Trade is a basic economic concept involving the buying and selling of goods and services, with compensation paid by a buyer to a seller, or the exchange of goods or services between parties. Trade can take place within an economy between producers and consumers.

• PREDICTION: “Prediction” refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether a customer will churn in 30 days.

• STOCK: Stocks are securities that represent an ownership share in a company. For companies, issuing stock is a way to raise money to grow and invest in their business. For investors, stocks are a way to grow their money and outpace inflation over time.

• STOCK EXCHANGE: The stock exchange is a virtual market, here buyers and sellers trade in existing securities. It is a market hosted by an institute or any such government body where shares, stocks, debentures, bonds, futures, options, etc are traded. A stock exchange is a meeting place for buyers and sellers. Examples of stock exchanges are New York Stock Exchange (NYSE), NASDAQ, Tokyo Stock Exchange (JPX), Bombay Stock Exchange (BSE), National Stock Exchange (NSE).

• BROKER: A stockbroker is a financial professional who executes orders in the market on behalf of clients. A stockbroker may also be known as a registered representative (RR) or an investment advisor. Examples: Zerodha, UpStox etc.

2.1 MACHINE LEARNING ALGORITHMS

Machine learning is a concept that provides systems the ability to learn automatically and improve from experience without being explicitly programmed. Machine learning concepts focuses on the development of computer programs that can access data and use it learn for themselves. The learning process begins with observing the data, such as examples, direct experience, or instruction, to look for patterns in data and make better decisions in the future based on the sample data that we provide. The primary goal is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as unsupervised and supervised.

2.2 Unsupervised learning algorithm

In unsupervised learning, the training of machine is done using information which is neither classified nor labelled and allowing the algorithm to work on that information without guidance. The goal of machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data. Different from supervised learning, no teacher is provided, that means no training will be given to the machine. That's why machine is restricted to find the hidden structure in unlabelled data by our-self. Unsupervised learning classified into two categories of algorithms:

1. Clustering: A clustering problem is where we want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.

2. Association: An association learning problem is where we want to discover rules that describe huge portions of your data, such as people that buy X also tend to buy Y.

2.3 Supervised learning algorithm

In Supervised learning, as the name indicates there is a supervisor as teacher. Learning in which we train the machine using data which is well labeled that means some data is already tagged with correct answer is known as Supervised learning. Next, machine is provided with new set of examples (data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data. Supervised learning algorithms increases consistently with the data. It is a type of inductive learning. Supervised learning classified into two categories of algorithms:

1. Classification: A classification problem is where the output variable is a category, such as "Red" or "blue" or "disease" and "no disease".

2. Regression: A regression problem is where the output variable is a real value, such as "dollars" or "weight".

CHAPTER 3

EXISTING SYSTEMS

3.1 Simple Moving Average [2]

A simple moving average (SMA) is an arithmetic moving average calculated by adding recent prices and then dividing that figure by the number of time periods in the calculation average. For example, one could add the closing price of a security for several time periods and then divide this total by that same number of periods. Short-term averages respond quickly to changes in the price of the underlying security, while long-term averages are slower to react. There are other types of moving averages, including the exponential moving average (EMA) and the weighted moving average (WMA).

FORMULA:

$$SMA = \frac{A_1 + A_2 + \cdots + A_n}{n} \qquad \text{eq(1)}$$

Where:

An= The price of an asset

n = The number of total periods

For example, this is how you would calculate the simple moving average of a security with the following closing prices over a 15-day period.

Week One (5 days): 20, 22, 24, 25, 23

Week Two (5 days): 26, 28, 26, 29, 27

Week Three (5 days): 28, 30, 27, 29, 28

A 10-day moving average would average out the closing prices for the first 10 days as the first data point. The next data point would drop the earliest price, add the price on day 11, and then take the average, and so on. Likewise, a 50-day moving average would accumulate enough data to average 50 consecutive days of data on a rolling basis.

A simple moving average smooths out volatility and makes it easier to view the price trend of a security. If the simple moving average points up, this means that the security's price is increasing. If it is pointing down, it means that the security's price is decreasing. The longer the time frame for the moving average, the smoother the simple moving average.

A shorter-term moving average is more volatile, but its reading is closer to the source data. One of the most popular simple moving averages is the 200-day SMA. However, there is a danger to following the crowd. As the Wall Street Journal explains, since thousands of traders base their strategies around the 200-day SMA, there is a chance that these predictions could become self-fulfilling and limit price growth.

Special Considerations

Analytical Significance Moving averages are an important analytical tool used to identify current price trends and the potential for a change in an established trend. The simplest use of an SMA in technical analysis is using it to quickly determine if an asset is in an uptrend or downtrend. Another popular, albeit slightly more complex, analytical use is to compare a pair of simple moving averages with each covering different time frames. If a shorter-term simple moving average is above a longer-term average, an uptrend is expected. On the other hand, if the long-term average is above a shorter-term average, then a downtrend might be the expected outcome.

Popular Trading Patterns

Two popular trading patterns that use simple moving averages include the death cross and a golden cross. A death cross occurs when the 50-day SMA crosses below the 200-day SMA. This is considered a bearish signal, indicating that further losses are in store. The golden cross occurs when a short-term SMA breaks above a long-term SMA. Reinforced by high trading volumes, this can signal further gains are in store.

3.1.1 IMPLEMENTATION RESULTS

We have implemented SMA for the dates between 08-2018 and 07-2019

We have tested SMA for different stocks datasets like google, amazon, apple, tesla

The platform we used is Google Colab

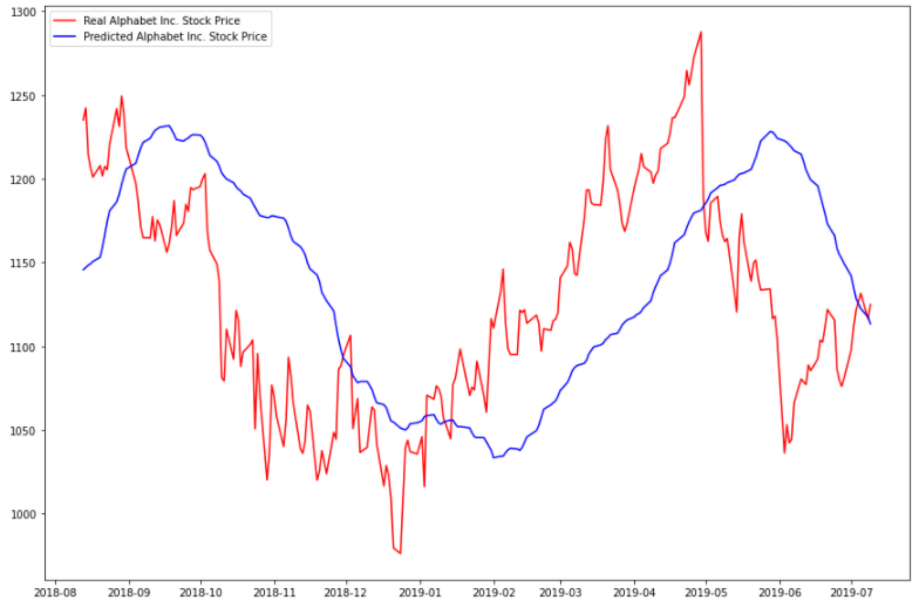

Fig 1: SMA Google stock

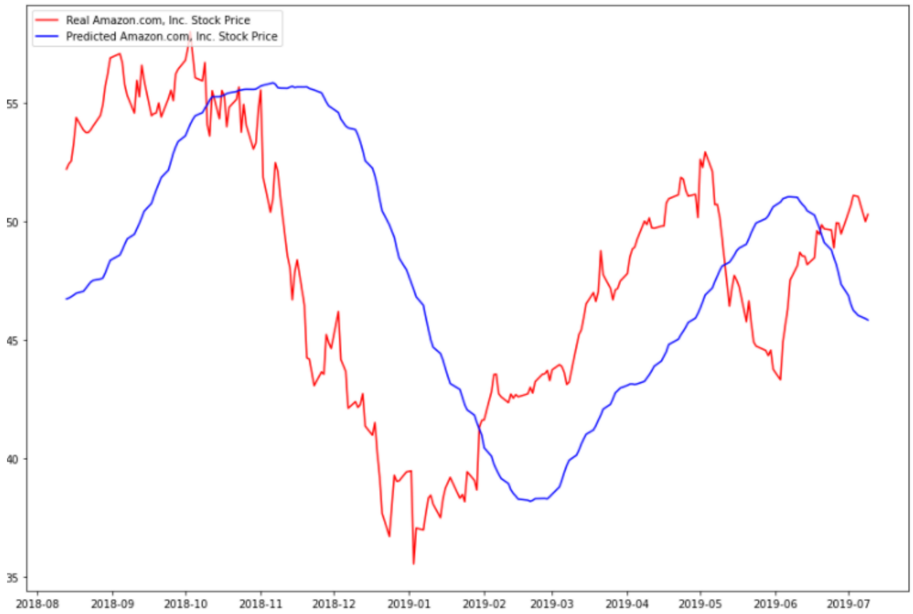

Fig 2: SMA Amazon stock

Fig 3: SMA Apple stock

Fig 4: SMA Tesla stock

PERFORMANCE ANALYSIS:

From the above graphs we have noticed the difference between the predicted values and the actual values is very high. So, this model not preferable for stock prediction

3.1.2 PERFORMANCE METRICS

We calculated the model performance using the formulas given below:

MSE = $\frac{1}{n}\sum_{i=1}^{n}(Y_i - P_i)^2$

MSE = mean squared error

n = number of data points

Yi = observed values

Pi = predicted values

RMSE = $\sqrt{(\frac{1}{n}\sum_{i=1}^{n}(Y_i - P_i)^2)}$

RMSD = root mean square deviation

N = no. of missing data points

Yi = observed values

Pi = predicted values

R^2 = 1-RSS/TSS

R^2 = coefficient of determination

RSS = sum of squares of residuals

TSS = total sum of squares

Calculated values of SMA:

From the below table, we noticed that the MSE and RMSE values are very high and R^2 is too low, it denotes that this model is not preferable for good results

MODEL	MSE	RMSE	R^2
SMA	**5668.585465423127**	**75.2900090677583**	**-0.31223237668214**

3.2 Auto Regressive Integrated Moving Average (ARIMA) [4]

This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

• AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
• I: Integrated. The use of differencing of raw observations (e.g., subtracting an observation from an observation at the previous time step) to make the time series stationary.
• MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA (p, d, q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used. The parameters of the ARIMA model are defined as follows:

• p: The number of lag observations included in the model, also called the lag order.
• d: The number of times that the raw observations are differenced, also called the degree of differencing.
• q: The size of the moving average window, also called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing to make it stationary, i.e., to remove trend and seasonal structures that negatively affect the regression model.

3.2.1 ARIMA IMPLEMENTED RESULTS

We have implemented ARIMA for the dates between 08-2018 and 07-2019

We have tested ARIMA for different stocks datasets like google, amazon, apple, tesla

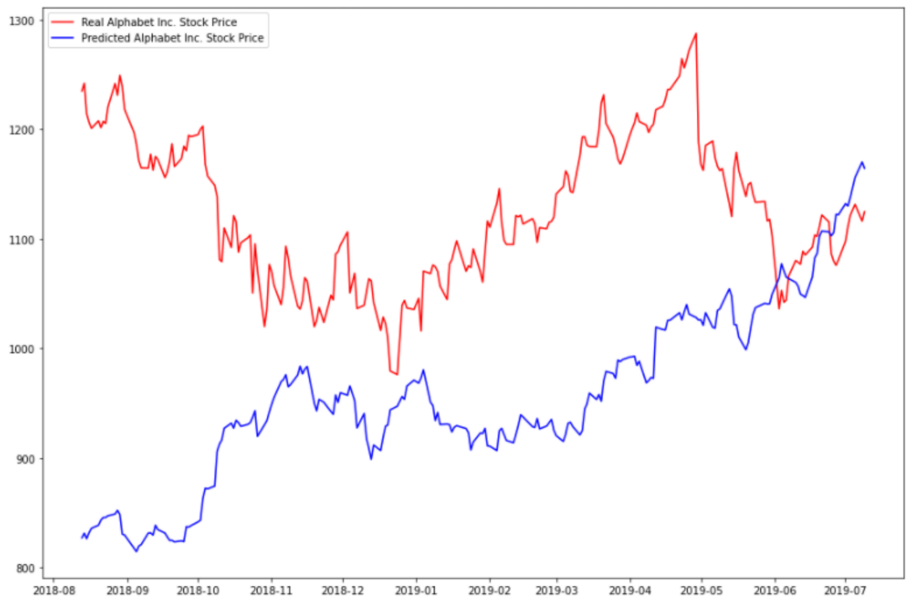

Fig 5: ARIMA Google stock

Fig 5: ARIMA Amazon stock

Fig 5: ARIMA Apple stock

Fig 5: ARIMA Tesla stock

PERFORMANCE ANALYSIS:

From the above graphs we have noticed the difference between the predicted values and the actual values is very high. So, this model not preferable for stock prediction

3.2.2 PERFORMANCE METRICS

From the below table, we noticed that the MSE and RMSE values are very high and R^2 is too low, it denotes that this model is not preferable for good results

Calculated values of ARIMA:

MODEL	MSE	RMSE	R^2
ARIMA	**42071.70516123586**	**205.113883394877**	**-8.73926458226190**

3.3 Recurrent Neural Networks

Recurrent neural networks (RNN) are a class of neural networks that are helpful in modeling sequence data. Derived from feedforward networks, exhibit similar behavior to how human brains function. Simply put recurrent neural networks produce predictive results in sequential data that other algorithms can’t. RNNs are a powerful and robust type of neural network and belong to the most promising algorithms in use because it is the only one with an internal memory.

Like many other deep learning algorithms, recurrent neural networks are relatively old. They were initially created in the 1980’s, but only in recent years have we seen their true potential. An increase in computational power along with the massive amounts of data that we now have to work with, and the invention of long short-term memory (LSTM) in the 1990s, has really brought RNNs to the foreground.

Because of their internal memory, RNN’s can remember important things about the input they received, which allows them to be very precise in predicting what’s coming next. Therefore, they're the preferred algorithm for sequential data like time series, speech, text, financial data, audio, video, weather and much more. Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms.

To understand RNNs properly, you'll need a working knowledge of "normal” feed-forward neural networks and sequential data.

In a feed-forward neural network, the information only moves in one direction — from the input layer, through the hidden layers, to the output layer. The information moves straight through the network and never touches a node twice.

Feed-forward neural networks have no memory of the input they receive and are bad at predicting what’s coming next. Because a feed-forward network only considers the current input, it has no notion of order in time. It simply can’t remember anything about what happened in the past except its training.

In a RNN the information cycles through a loop. When it decides, it considers the current input and what it has learned from the inputs it received previously

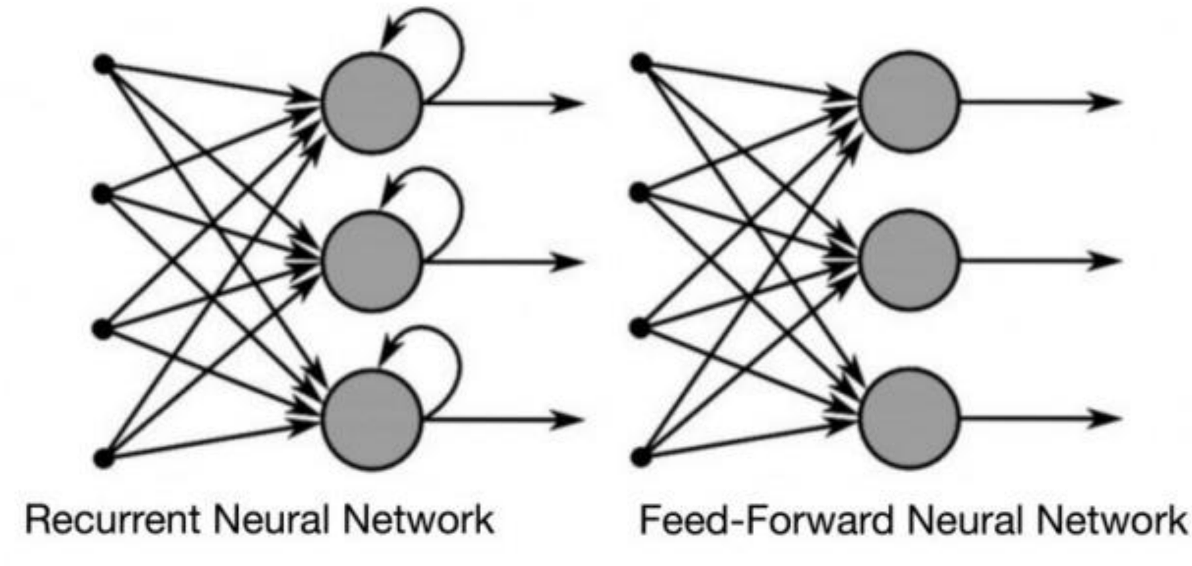

Fig 6: Difference between Feed Forward and Recurrent Neural Network

Also note that while feed-forward neural networks map one input to one output, RNNs can map one to many, many to many (translation) and many to one (classifying a voice)

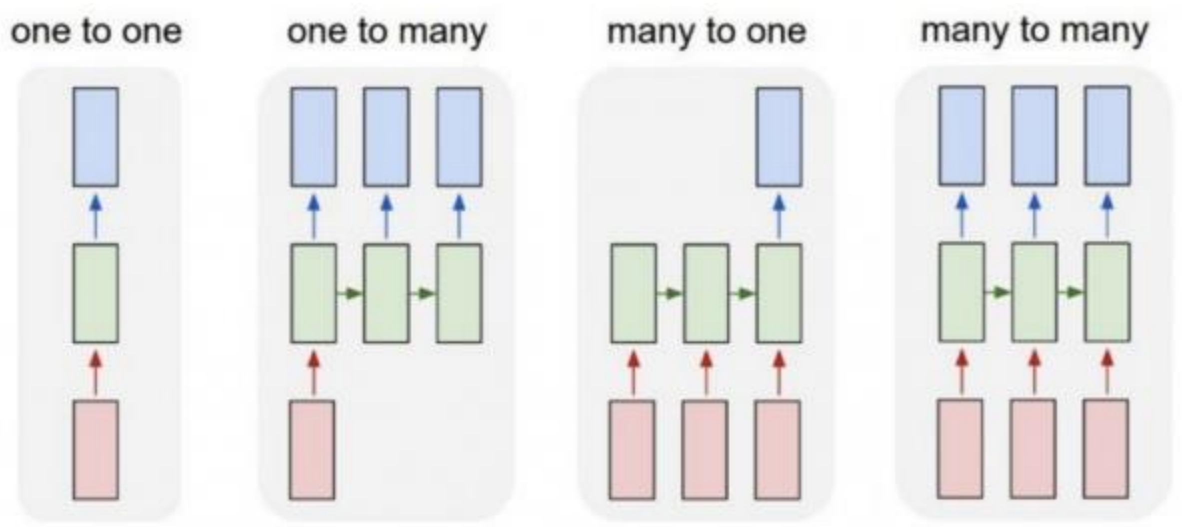

Fig: RNN mapping different type of inputs to outputs [1]

3.3.1 RESULTS

We have implemented RNN for the dates between 08-2018 and 07-2019

We have tested RNN for different stocks datasets like google, amazon, apple, tesla

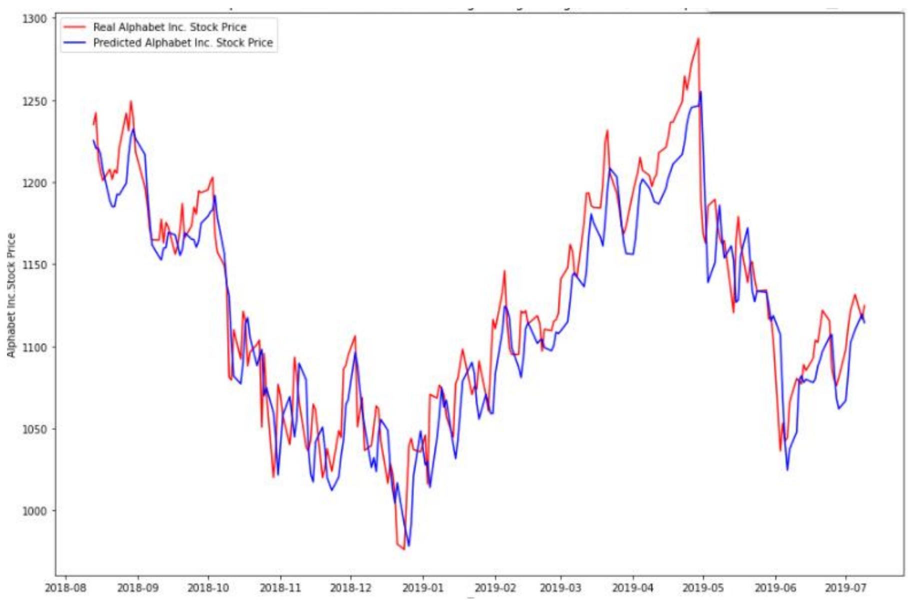

Fig 7: RNN Google stock

Fig 8: RNN Amazon stock

Fig 9: RNN Apple stock

Fig 10: RNN Tesla stock

PERFORMANCE ANALYSIS:

From the above graphs we have noticed the difference between the predicted values and the actual values is very low. So, this model preferable for stock prediction

3.3.2 PERFORMANCE METRICS

From the below table, we noticed that the MSE and RMSE values are slightly low and R^2 is near to 1, it denotes that this model is preferable for good results

Calculated values of RNN:

MODEL	MSE	RMSE	R^2
RNN	**608.6947111664791**	**24.671739225687**	**0.8590920235074586**

CHAPTER 4

PROPOSED SYSTEM

LSTM Long-Short-Term Memory

- ✓ LSTM is an advanced type of RNN
- ✓ LSTM is proficient in learning about long-term dependencies.
- ✓ LSTM is popularly used on time-series data for classification, processing, and making predictions.

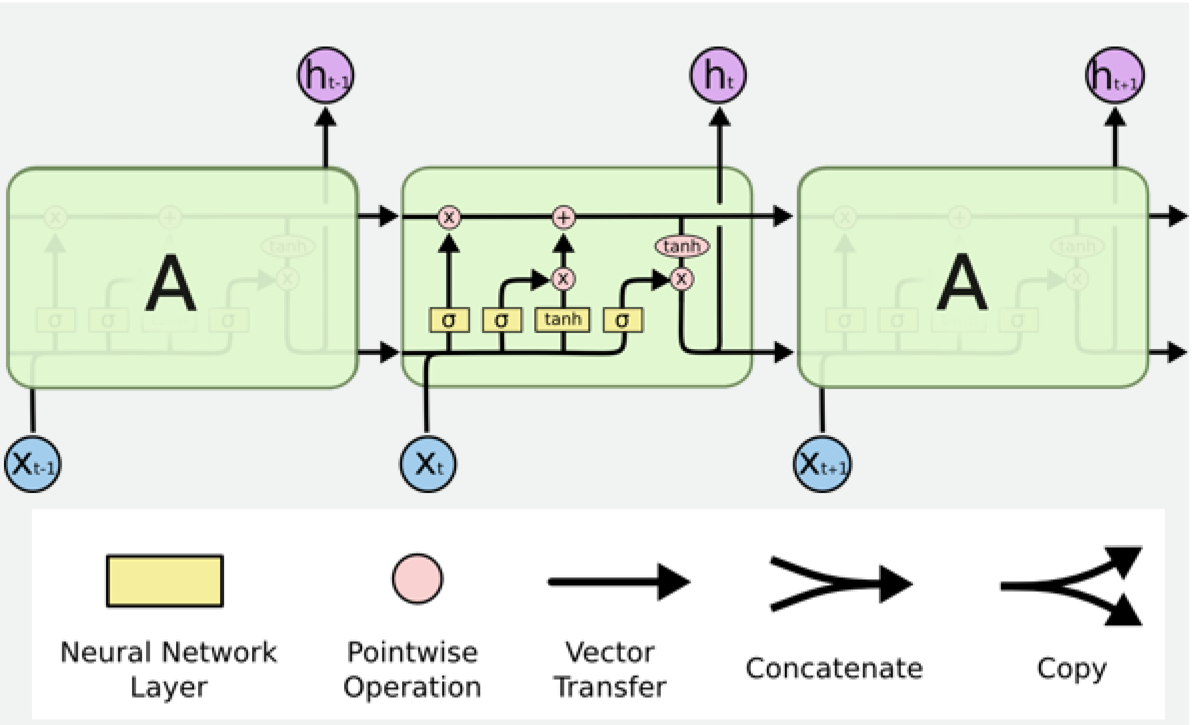

Fig: Structure of LSTM

- Long Short-Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It can handle the vanishing gradient problem faced by RNN. A recurrent neural network is also known as RNN is used for persistent memory.
- The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp. These three parts of an LSTM cell are known as gates. The first part is called Forget gate, the second part is known as the Input gate and the last one is the Output gate.

CHAPTER 5
CONCLUSION & FUTURE WORK

5.1 Conclusion

In this project, we are predicting closing stock price of any given organization. We have applied datasets belonging to Google, Amazon, Apple, and Tesla Stocks and achieved above 85% accuracy for these datasets.

5.2 Future work

- We want to extend this project for predicting cryptocurrency trading.
- We want to further add features of GRU model.

CHAPTER 6

REFERENCES

[1] https://builtin.com/data-science/recurrent-neural-networks-and-lstm

[For RNN details and images]

[2] https://www.investopedia.com/terms/s/sma.asp

[For SMA description and formula]

[3] https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[For ARIMA terms and details]

[4] https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-an-arima-model-2e3b3080bd70

[For ARIMA implementation]

[5] https://github.com/0xpranjal/Stock-Prediction-using-different models/blob/main/Notebooks/4.%20RNN_seqtovec_seqtoseq.ipynb

[For RNN implementation]

[6] https://github.com/0xpranjal/Stock-Prediction-using-different-models/blob/main/Notebooks/1.%20Time%20Series%20Forecasting%20with%20Naive%2C%20Moving%20Averages%2C%20and%20ARIMA.ipynb

[For SMA implementation]