RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi, Accredited By NAAC, Bengaluru And NBA, New Delhi

Go, change the world

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Stock Price Prediction

MINOR PROJECT REPORT

Submitted by

Name 1: Akshat Bansal	**USN1:1RV19CS008**
Name2: Dency Narendra Patel	**USN2:1RV19CS044**
Name3: Khetan Rishabh	**USN3:1RV19CS071**

Under the guidance of

Prof. Sneha M.
Asst. Professor
Dept of CSE
RV College of Engineering

In partial fulfilment for the award of degree

of

Bachelor of Engineering

in

Computer Science and Engineering 2021-2022

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the minor project work titled ***'Stock Price Prediction'*** is carried out by **Akshat Bansal (1RV19CS008), Dency Patel (1RV19CS044), and Khetan Rishabh (1RV19CS071)** who are bonafide students of RV College of Engineering, Bengaluru, in partial fulfilment for the award of degree of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the minor project report deposited in the departmental library. The Minor Project report has been approved as it satisfies the academic requirements in respect of minor project work prescribed by the institution for the said degree.

Signature of Guide	Signature of Head of the Department	Signature of Principal
Prof. Sneha M.	**Dr. Ramakanth Kumar P**	**Dr.K.N.Subramanya**

External Viva

Name of Examiners	Signature with Date
1	
2	

RV COLLEGE OF ENGINEERING®, BENGALURU-59
(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Akshat Bansal, Dency Patel and Rishabh Khetan,** students of sixth semester B.E., department of CSE, RV College of Engineering, Bengaluru, hereby declare that the minor project titled **'Stock Price Prediction'** has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering** during the year 2021-22.

Further we declare that the content of the report has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

Name	**Signature**
1. Akshat Bansal (1RV19CS008)	
2. Dency Patel (1RV19CS044)	
3. Khetan Rishabh (1RV19CS071)	

ACKNOWLEDGEMENT

We are indebted to our guide**, Prof. Senha M,** Asst. Professor**, Dept of CSE** for his/her wholehearted support, suggestions and invaluable advice throughout our project work and also helped in the preparation of this thesis.

We also express gratitude to our Minor Project lab faculty **Anitha Sandeep, Professor,** Department of Computer Science and Engineering for their valuable comments and suggestions.

Our sincere thanks to **Dr. Ramakanth Kumar P.**, Professor and Head, Department of Computer Science and Engineering, RVCE for his support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for his appreciation towards this project work.

We thank all the **teaching staff and technical staff** of Computer Science and Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our **family** members and **friends** who provided all the backup support throughout the project work..

ABSTRACT

The stock market is a dynamic and volatile platform which provides an environment and opportunity for the traders to invest and trade in stocks of particular companies. The price of a stock is dependent on numerous static and dynamic features. Predicting the trend in future price movement of a particular company's stock can be extremely beneficial for investors and traders.

we have decided to use a couple of conventional machine learning algorithms to study the behavior of learning techniques for stock prediction. This paper presents an empirical study to study and analyze the behavior of Decision Tree, Linear Regression, K-Nearest Neighbors, and LSTM learning algorithms to bet on the algorithm that best predicts the stock prices.

After performing a series of experiments, we arrived at the following results.The RMSE values of the proposed algorithms with the bestchosen hyperparameters are computed and are shown in Table II. We have observed that the proposed model is giving the least RMSE as compared to other algorithms with the lowest RMSE of **21.83** at **200 epochs** (iterations) compared to different epochs variations the LSTM model is the best proposed model as it is depicting the near representation of predicted values from the actual stock price value and has the least RMSE value. Thus, for all intent and purposes, we can rely on the prices that LSTM has predicted. Whereas Linear Regression, Decision Tree and KNN cannot predict the prices as accurately as LSTM.

The project has a lot of scope in future and the topic and related research will always be in demand because the application of this has the power to control the flow of money which will keep the interest boosted. Hybrid models could be developed to get better accuracy and eliminate flaws that a single model produced. Apart from that, we can use NLP to view things from a sentimental analysis point of view.

TABLE OF CONTENTS

List of Tables **Page No.**

List of Figures **Page No**

CHAPTER-1

Introduction

1.1 State of the Art work

AI has a huge potential in the prediction of stock prices. Taking the past performance and behavior of any stock and training the data available using neural networks and machine learning models can help in understanding how a stock might behave in the future. Industrially talking, the system would have huge relevance. It can be used by traders to gain an edge over others and can also be used by financial institutions for quant-vol trading.

1.2 Motivation

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. 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. Since most of the dealings in the markets are done by automated systems, it has now been well established that training the past data can help us in finding patterns in the movement of the markets which can be used to predict the future prices. If implemented successfully with a higher accuracy than existing systems, it could turn into a financial support system with minimal amount of risk.

1.3 Problem Statement

With the innovation in technology and their application in the stock market, the system has become increasingly complex and volatile which in turn has made human predictions highly inaccurate, but using Machine Learning to find out the patterns in the system using historical data can help us predict the future prices more accurately. With the introduction of new training models using Machine Learning and Neural networks, it has become increasingly easy to predict patterns in price movements and the accuracy of the predictions has been increasing thereafter day by day and so the competition has significantly increased which has resulted in firms shifting to algorithm based trading even more.

1.4 Objectives

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 a new business or the need for a high salary career. Let's say we want to make money by buying stocks. Since we want to make money, we only want to buy stock on days when the price will go up. We'll create a machine learning algorithm to predict if the stock price will increase tomorrow. If the algorithm says that the price will increase, we'll buy stock. If the algorithm says that the price will go down, we won't do anything. We want to maximize our true positives - days when the algorithm predicts that the price will go up, and it actually goes up. Therefore, we'll be using precision as our error metric for our algorithm, which is true positives / (false positives + true positives). This will ensure that we minimize how much money we lose with false positives (days when we buy the stock, but the price actually goes down). This means that we will have to accept a lot of false negatives - days when we predict that the price will go down, but it actually goes up. This is okay, since we'd rather minimize our potential losses than maximize our potential gains.

1.5 Methodology

The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc.The proposed system that we offer is a hybrid model which provides a combination of more than one existing machine learning model that can be used for increasing the accuracy of the predictions. Bidirectional LSTM (Long Short Term memory) and Sequence to sequence are models that have shown good accuracy in predicting the prices. But to match the competitive environment pertaining to complex algorithms used by financial institutions in today’s world, restricting yourself to only one model can not prove to be that efficient. Integrating the usage of more than one model with the right set of data and parameters can prove to be a more efficient and accurate system to predict the volatile situation in post covid markets.

1.6 Summary

It is now evidently clear that AI and ML can have huge significance in topics of prediction and using these systems in financial markets can be a huge bonus if applied correctly and carefully. AI systems can predict the movements using knowledge of complex mathematical functions on the basis of which the stocks move and by training them could be able to predict how it would move ahead.

CHAPTER-2
Literature Survey

2.1 Introduction :

The work on the use of artificial intelligence and especially machine learning to predict the prices of any type of equity and commodity has been going on since a long time. With the increase in the technological developments in the field of Machine learning, it has started becoming clearer that historical patterns can be used in multiple ways to predict what can happen in the future relating to the prices of any type of equity or commodity. With this development, people have started creating more novel models to predict the movements in prices more accurately. Since these markets are a huge arena for making financial profits, all the giant financial institutions started conducting even more research in this field to gain an economic advantage over their competitors and this forced the work on such models to full force.

2.2 Related Work:

SL. NO	Publications	IMPLEMENTATIONS	CONS
1.	Saurav Agrawal, Dev Thakkar, Dhruvil Soni, Krunal Bhimani, Dr. Chirag Patel, “Stock Market Prediction using Machine Learning Techniques”.	Artificial neural network with backpropagation algorithm	Neither growth nor pruning methods were attempted for the selection of network architecture.
2.	K. Hiba Sadia, Aditya Sharma, Adarsh Paul, SarmisthaPadhi, Saurav Sanyal, “Stock Market Prediction Using Machine Learning Algorithms”.	Random forest Algorithms, support vector machine	Previous years dataset is considered. No real-time data are used for predicting stocks.
3.	Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM”. International Journal of Science and Research 2017.	Root Mean Square Error (RMSE), the difference between the target value and the obtained output value is reduced by using RMSE value. Recurrent Neural Network, Long Short-Term Memory	Doesn't focus on events in the environment, like news or social media. It exploits only one data source, thus it is highly biased.

	ISSN: 2319-7064		
4.	S Abdulsalam Sulaiman Olaniyi, Adewole, Kayode S, Jimoh, R. G, “Stock Trend Prediction Using Regression Analysis – A Data Mining Approach”. ARPN Journal of Systems and Software, Volume 1, Issue 4, 2011. ISSN: 2222-9833	Linear regression, moving average	Used for limited company stocks More amount of data is not considered for prediction
5.	Gareja Pradip, Chitrak Bari, J. Shiva Nandhini, “Stock market prediction using machine learning”.	Artificial neural network, multiple linear regression, Bayesian Algorithm	using Bayes theorem bias is found. Predicted price is fluctuating they are not constant
6.	Vivek Kanade, Bhausaheb Devikar, Sayali Phadatare, Pranali Munde, “Stock market prediction: Using historic data analysis”. International journal of advanced research in computer science and software engineering, volume 7, issue 1, 2017. ISSN: 2277 128X. DOI: 10.23956/ijarcsse/V711/0112.	SVM, ANN SVM (Support vector Machine)	Only sentiment data are used from various news and Twitter resources no historical data are considered for predictions.

Table 1 Literature reviews

2.3 Summary

The existing system on stock price prediction consists of basic LSTM models and recurrent neural networks. ANNs use adaptive weights to forecast stock prices. Y. Bing proposed an ANN to predict the index of the Shanghai Stock Exchange. The authors studied the market between March 17, 2010 to April 28, 2010. They considered 5 features of the market, open, high, close, low and volume. The neural network constructed was successful in predicting the daily lowest, highest, and closing value of the Shanghai Stock Exchange. M. Jia proposed a framework which made use of the bidirectional long-short term memory (BLSTM) neural network for predicting the future price of a stock. The authors used the historical data of the GREE stock. They collected data for 568 days from January 1, 2017 to May 14, 2019. The data consisted of 14 features such as open, high, close, volume etc. The data was normalized and pre-processed.The close value was used as the benchmark for the prediction. K. A. Althelaya proposed a Bidirectional LSTM for Short- and Long-Term Stock Market Prediction. The authors had made use of the Standard and Poor 500 Index (S&P500) historical data for their proposed work.

CHAPTER-3
Software Requirements Specifications

3.1 Functional requirements

Functional requirements describe what the software should do (the functions). Think about the core operations.

Because the "functions" are established before development, functional requirements should be written in the future tense. In developing the software for Stock Price Prediction, some of the functional requirements could include:

- The software shall accept the tw_spydata_raw.csv dataset as input.

- The software should shall do pre-processing (like verifying for missing data values) on input for model training.
- The software shall use LSTM ARCHITECTURE as main component of the software.
- It processes the given input data by producing the most possible outcomes of a CLOSING STOCK PRICE.

3.2Non-Functional requirements

Product properties :

- Usability: It defines the user interface of the software in terms of simplicity of understanding the user interface of stock prediction software, for any kind of stock trader and other stakeholders in stock market.
- Efficiency: maintaining the possible highest accuracy in the closing stock prices in shortest time with available data.
- Performance: It is a quality attribute of the stock prediction software that describes the responsiveness to various user interactions with it.

3.3 Hardware Requirements

Hardware requirements define what sort of hardware specifications we will be working with, and what will be needed to replicate in some other scenario.

CPU	2 GHz or faster
RAM	4 GB or higher
Disk Space	500 GB SSD or larger
Architecture	32-bit or 64-bit

The specifications given in table x.x is just an estimation. It can vary based on the kind of model used and the size of the dataset chosen.

3.4 Software Requirements

Software requirements define what software is being used. It includes major stuff like what kind of operating system, what databases are being used. The projects' software requirements are given in table x.x.

Operating System	Windows 10 or newer
Database	Obtained through Yahoo Finance
Programming	Python 3.10.0 (Jupyter notebook)

This project is specifically built in jupyter notebook using python wherein all the dataset collection (imported through csv file), agents, training and testing of models and the results that the prediction produces are all implemented using various python libraries like pandas, numpy, scikit learn etc.

CHAPTER-4
Design

4.1 High Level Design

4.1.1 Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems.
- Goals that your system or application helps those entities (known as actors) achieve.
- The scope of your system.

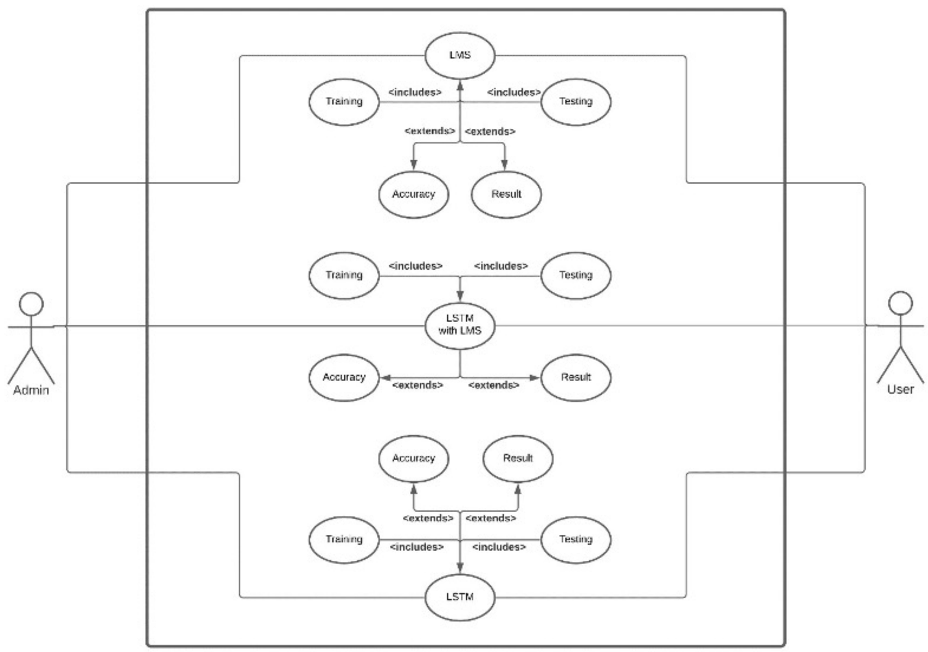


Figure 1- Use Case Diagram

4.1.2Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

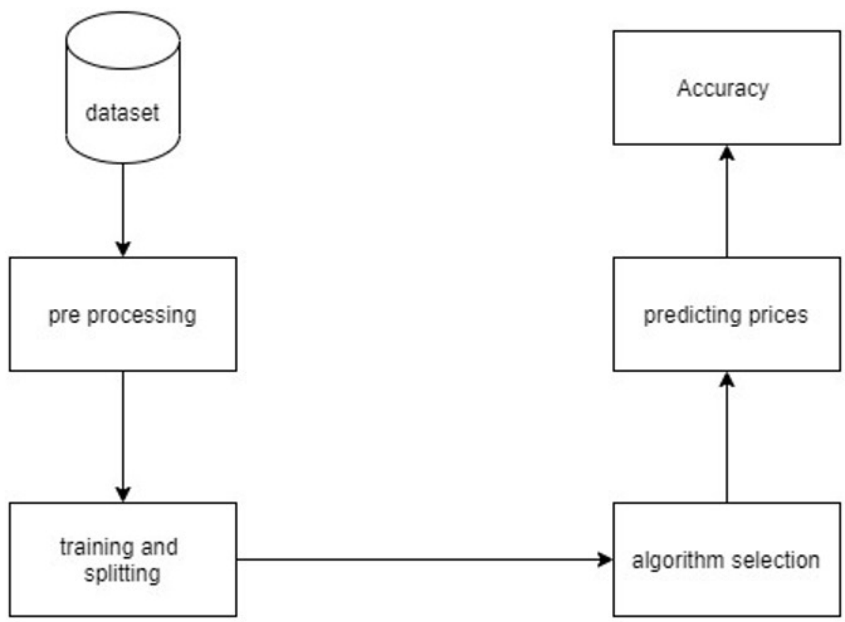


Figure 2: Components present in the system

4.2 System Architecture

1) Preprocessing of data

Fig. 3: Pre-processing of data

2) Overall Architecture

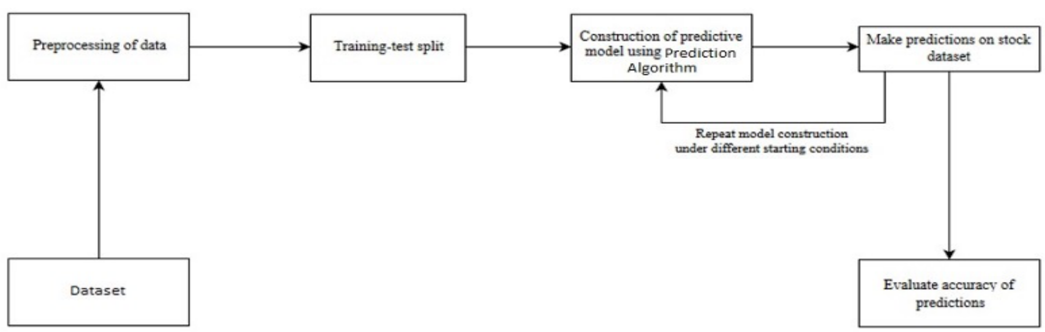


Fig. 4: Overall Architecture

4.3 Detailed Design

Long short-term memory network:

Long short-term memory network (LSTM) is a particular form of recurrent neural network (RNN).

Working of LSTM:

LSTM is a special network structure with three "gate" structures.
Three gates are placed in an LSTM unit, called input gate, forgetting gate and output gate. While information enters the LSTM's network, it can be selected by rules. Only the information conforms to the algorithm will be left, and the information that does not conform will be forgotten through the forgetting gate.

The experimental data in this paper are the actual historical data downloaded from the Internet. Three data sets were used in the experiments. It is needed to find an optimization algorithm that requires less resources and has faster convergence speed.

•Used Long Short-term Memory (LSTM) with embedded layer and the LSTM neural network with automatic encoder.

• LSTM is used instead of RNN to avoid exploding and vanishing gradients.

• In this project python is used to train the model, MATLAB is used to reduce dimensions of the input. MySQL is used as a dataset to store and retrieve data.

• The historical stock data table contains the information of opening price, the highest price, lowest price, closing price, transaction date, volume and so on.

•The accuracy of this LSTM model used in this project is 57%.

LMS filter:

The **LMS filter** is a kind of adaptive filter that is used for solving linear problems. The idea of the filter is to minimize a system (finding the filter coefficients) by minimizing the least mean square of the error signal.

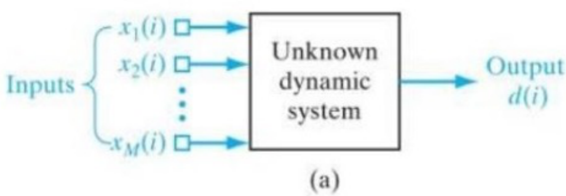


Fig. 5: LMS Inputs and Outputs

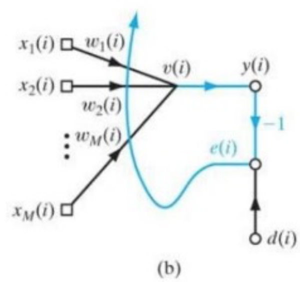


Fig 6: LMS updating weights

Algorithm 1: LMS

Input:

x : input vector

d: desired vector

μ: learning rate

N: filter order

Output:

y: filter response

e: filter error

begin

$M = size(x)$;

$x_n(0) = w_n(0) = [0\ 0\ \ldots\ 0]^T$;

while $n < M$ **do**

$x_{n+1} = [x(n);\ x_n(1:N)]$;

$y(n) = w_n^H * x_n$;

$e(n) = d(n) - y(n)$;

$w_{n+1} = w_n + 2\mu e(n) x_n$;

end

end

In general, we don't know exactly if the problem can be solved very well with linear approach, so we usually test a linear and a non-linear algorithm. Since the internet always shows non-linear approaches, we will use LMS to prove that stock market prediction can be done with linear algorithms with a good precision.

But this filter mimetizes a system, that is, if we apply this filter in our data, we will have thefilter coefficients trained, and when we input a new vector, our filter coefficients will output a response that the original system would (in the best case).

So we just have to do a *tricky* modification for using this filter to predict data.

The system: First, we will delay our input vector by *l* positions, where *l* would be the quantity of days we want to predict, this *l* new positions will be filled by zeros.

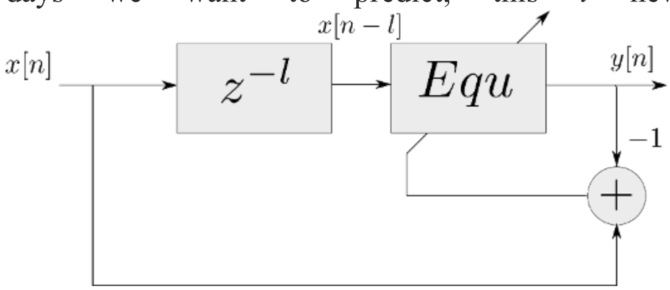
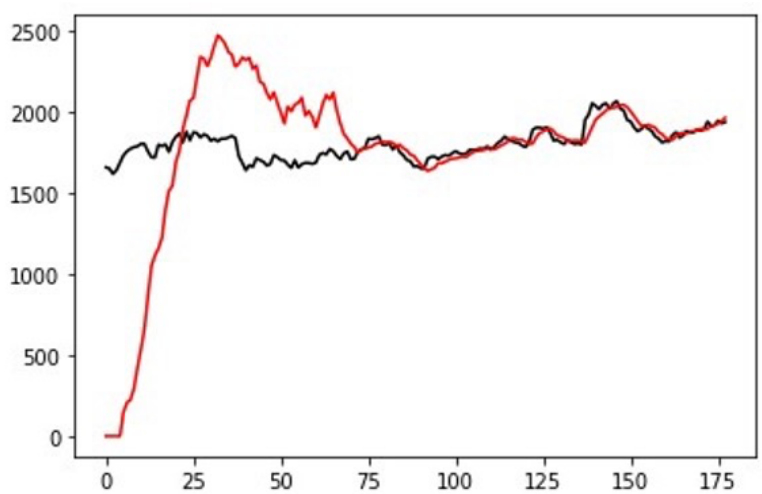


Fig. 7: LMS updating weights

When we apply the LMS filter, we will train the filter to the first 178 data. After that, we will set the error as zero, so the system will start to output the answers as the original system to the last *l* values. We will call the *tricky* modification as the LMSPred algorithm.

Results



One example of stock market prediction result

LSTM Architecture

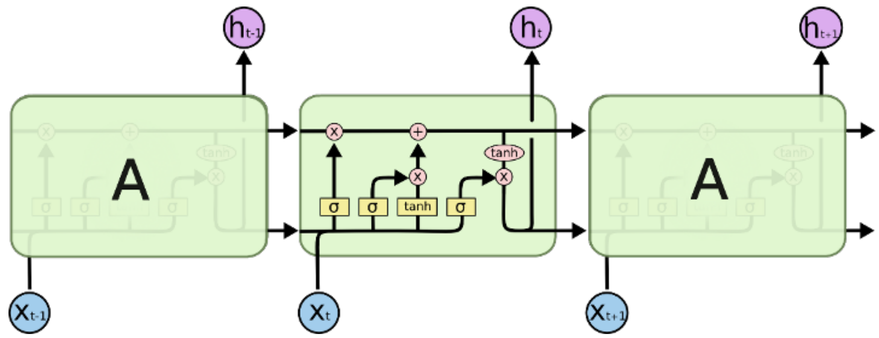


Fig. 8: LSTM Architecture

Forget Gate:

A forget gate is responsible for removing information from the cell state.

- The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter.
- This is required for optimizing the performance of the LSTM network.
- This gate takes in two inputs; h_t-1 and x_t. h_t-1 is the hidden state from the previous cell or the output of the previous cell and x_t is the input at that particular time step.

Input Gate:

1. Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from hi-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

Output Gate:

The functioning of an output gate can again be broken down to three steps:

- Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
- Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.

Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as a output and also to the hidden state of the next cell.

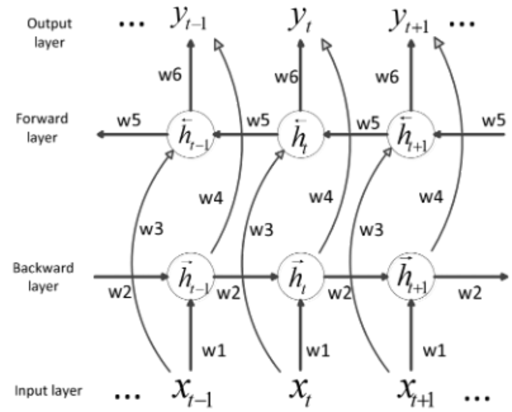
Bidirectional LSTM Principle

In the Forward layer, the forward calculation is performed from 1 moment to t moment, and the output of the forward hidden layer at each time is obtained and saved. In the Backward layer, the calculation is reversed along the time t to the time 1 to obtain and save the output of the backward hidden layer at each time. The six unique weights are repeatedly used in each time step, and the six weights are respectively used. Correspondence: Input to the forward and backward hidden layers (w1, w3), hidden layer to the hidden layer itself (w2, w5), forward and backward hidden layers to the output layer (w4, w6). Finally, at each moment, the final output is obtained by combining the output of the Forward layer and the Backward layer, as shown in Figure 2 is a bidirectional LSTM network diagram [13]. The mathematical expressions are as follows:

$$h_t \; f w x(1 \; t \; w h 2 \; t 1) \qquad (7)$$

$$h_t' \; f w x(3 \; t \; w h 5 \qquad t 1') \qquad (8)$$

$$o_t \; g w h(4 \; 4 \; w h 6 \qquad t') \qquad (9)$$



Bidirectional LSTM Structure

The BLSTM uses the pre-processed data as input, passes through the forward and backward LSTM neural network layer, then goes to the full connection layer, and outputs the prediction result[13], as shown in Figure 3. In Figure 3, the weights and bias initialization of the full connection layer are all based on a random normal distribution.

BLSTM is a variant of recurrent neural network, which solves the long-term dependence of RNN and LSTM. It combines LSTM in two different directions and extracts forward and reverse information data at the same time. Stocks are data with strong time series characteristics.

Selecting a cycling neural network can make better use of historical information.

Compared with LSTM, BLSTM can simultaneously make use of temporal information in both directions, so it is easier to mine potential unused data [14].

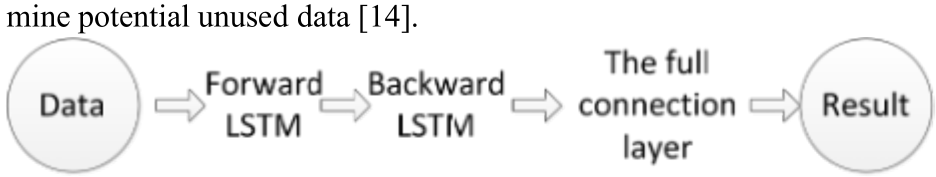


Fig 9 :*Seq2Seq long–short-term memory layer*

Inspired by the success of machine translation (Cho *et al.*, 2014), we have recognized the power of the Seq2Seq model in NLP. More specifically, two crucial components make up the standard Seq2Seq model, one is an encoder and the other is a decoder. The former maps the source input *x* to a vector representation, while the latter produces an output series based on the source. Both the encoder and decoder are LSTMs. By transmitting the last memory condition of the encoder to the decoder as the original memory condition, the encoder is capable of accessing information from the encoder. Input and output generally apply various LSTMs that possess their own compositional parameters to capture various compositional patterns. We apply a Seq2Seq LSTMs model to address the non-linear time-series forecasting issue as the third layer. In the encoder part, the input LSTM mechanism is used for inputting into series data. In the decoder part, an output LSTM mechanism is employed to decode the hidden states of encoder across all time steps before.

4.3.1 Structure Chart:

A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.

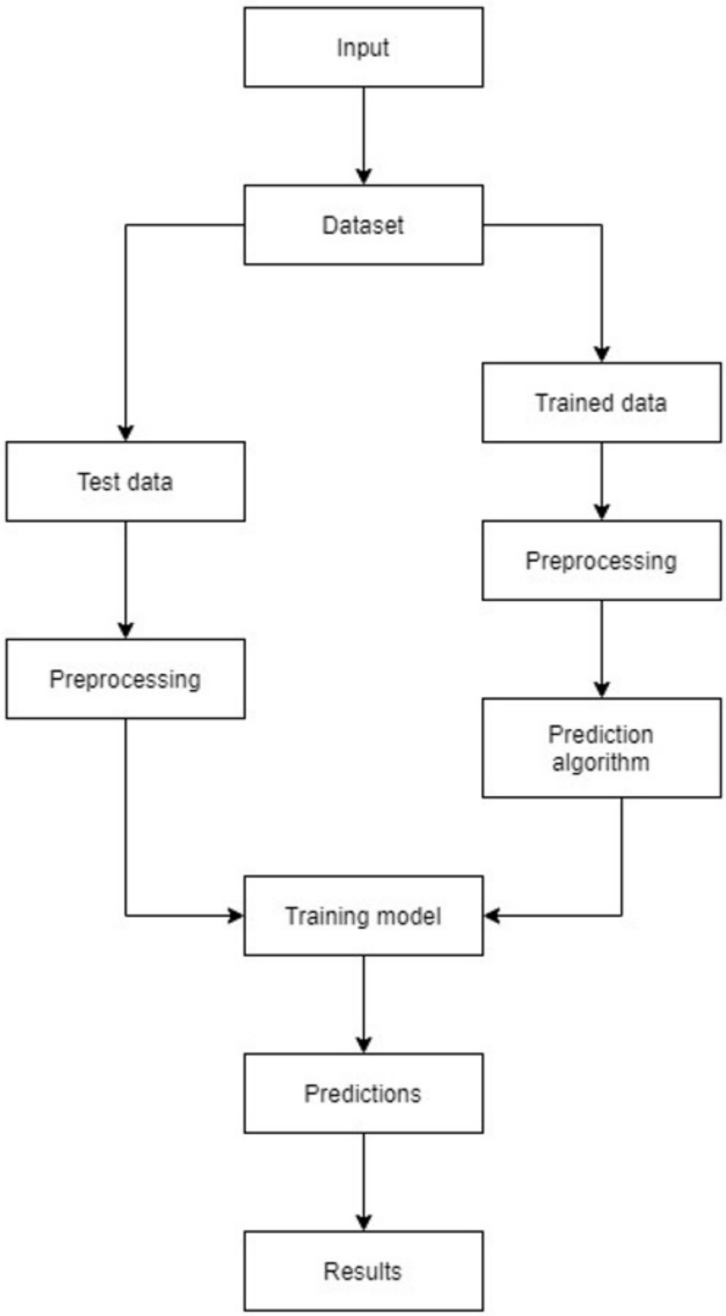


Fig 10: Structure Chart

CHAPTER-5
Implementation

DATASET :

In this project we have mainly used data consisting of stock prices for the well-known company Google from Yahoo! Finance from the year 2004 to May 2020. This includes Date, Open, High, Low, Close, Adj Close and Volume for a given day.

Date	Open	High	Low	Close	Adj Close	Volume
2004-08-19	49.813286	51.835709	47.800831	49.982655	49.982655	44871300
2004-08-20	50.316402	54.336334	50.062355	53.952770	53.952770	22942800
2004-08-23	55.168217	56.528118	54.321388	54.495735	54.495735	18342800
2004-08-24	55.412300	55.591629	51.591621	52.239193	52.239193	15319700
2004-08-25	52.284027	53.798351	51.746044	52.802086	52.802086	9232100

Step 1: Data Visualization

We have plotted a box plot as shown in Figure 1. That shows the mean of each attribute and the highest and lowest value they take. We also plotted a histogram as shown in Figure 2 for every attribute of the data to observe the dependency of stocks on the given attributes.

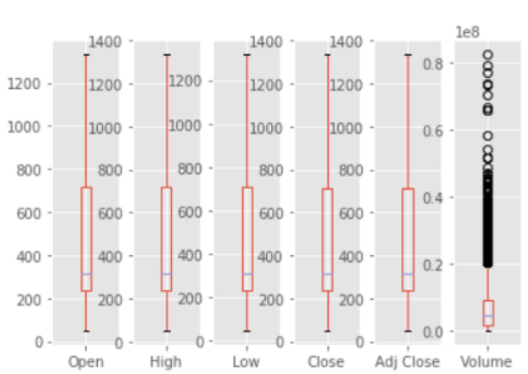


Figure 11: Box plot of all the attributes

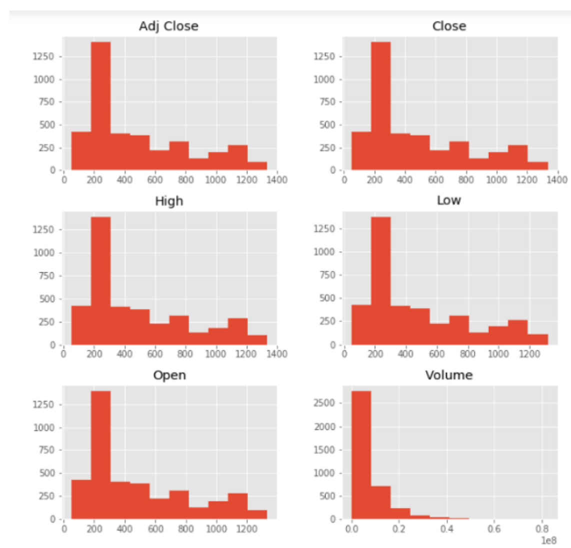


Figure 12: Histogram of all the attributes

Step 2: Data Preprocessing:

In data preprocessing we dropped the unnecessary attributes from the given set.

Date	Open	High	Low	Close	Volume
2004-08-19	49.813286	51.835709	47.800831	49.982655	44871300
2004-08-20	50.316402	54.336334	50.062355	53.952770	22942800
2004-08-23	55.168217	56.528118	54.321388	54.495735	18342800
2004-08-24	55.412300	55.591629	51.591621	52.239193	15319700
2004-08-25	52.284027	53.798351	51.746044	52.802086	9232100

Figure 13: Adjusted Close attribute dropped from the given attribute

Step 3: The LSTM model's architecture with the layers that we applied and the parameters that the model gets at the training phase can be seen in Table I.

Layer(type)	Output Shape	Parameters
lstm (LSTM)	(None, 60, 50)	10400
lstm_1(LSTM)	(None, 50)	20200
dense (Dense)	(None, 1)	1275
dense_1(Dense)	(None, 1)	26
Total parameters: 31,901 Trainable parameters: 31,901 Non-trainable parameters: 0		

TABLE 2. LSTM ARCHITECTURE

Moreover, it might be also be taken into consideration that optimization is one of the core aspects of training a machine learning model, thus on an intuitive level, the essence of most of the machine learning models is to define an optimization algorithm that can reduce the cost function used to quantify the error between the predicted value and the expected value, more so we can also point that depending on the context of the problem, a cost function can either converge at its local minimum or a local maximum. And since the situation at hand can be considered as a regression problem, it suits us quite well to use MSE (Mean Square Error) as defined in (2) as our loss function because, as evident from before, we have transformed our positively skewed data into a normal distribution.

As MSE is more flexible in penalizing the outlier than the absolute mean error, it often ensures our dataset is not robust in considering outliers while making predictions

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

As for choosing our optimization algorithm, we decide to subside with Adam as the optimizer as unlike most of the stochastic optimization methods that maintain a single learning rate throughout the training, the optimization algorithms involving Adam as an optimizer calculates adaptive learning rates while updating the parameters as estimated from the first and second moment of gradients. Thus, we can say that an optimization method like Adam maintains the adaptive learning rate based on both the first and second moments of the gradients and keeps track of the exponentially decaying average of past gradients that are much more reliable, especially when dealing with sparse gradients.

CHAPTER-6

Experimental results and testing

After performing a series of experiments, we arrived at the following results.

The RMSE values of the proposed algorithms with the best chosen hyperparameters are computed and are shown in Table II. We have observed that LSTM is giving the least RMSE as compared to other algorithms.

Algorithm	*Parameters*	*RMSE*
LSTM	Optimizer = Adam, Loss Function = Mean Squared Error	21.83
Linear Regression	α = 0.1	54
Decision Tree	Max Depth = 5	59
KNN	K=13	149

TABLE 3 COMPARATIVE RMSE VALUE OF DIFFERENT ALGORITHMS

After performing the series of observations, we have concluded that the LSTM model has been able to predict prices with the lowest RMSE of 21.83 at 200 epochs (iterations) compared to different epochs variations, as shown in Table III.

No. of Epochs	***RMSE***
50	26.94
100	23.86
200	**21.83**
300	31.37
400	49.21
500	25.21

TABLE 4 LSTM MODEL'S RMSE VALUE WITH A DIFFERENT VARIATION OF EPOCHS

As evident from Fig. 3, the LSTM model is the best proposed model as it depicts the near representation of predicted values from the actual stock price value and has the least RMSE value. Thus, for all intent and purposes, we can rely on the prices that LSTM has predicted. Whereas Linear Regression, Decision Tree and KNN cannot predict the prices as accurately as LSTM.

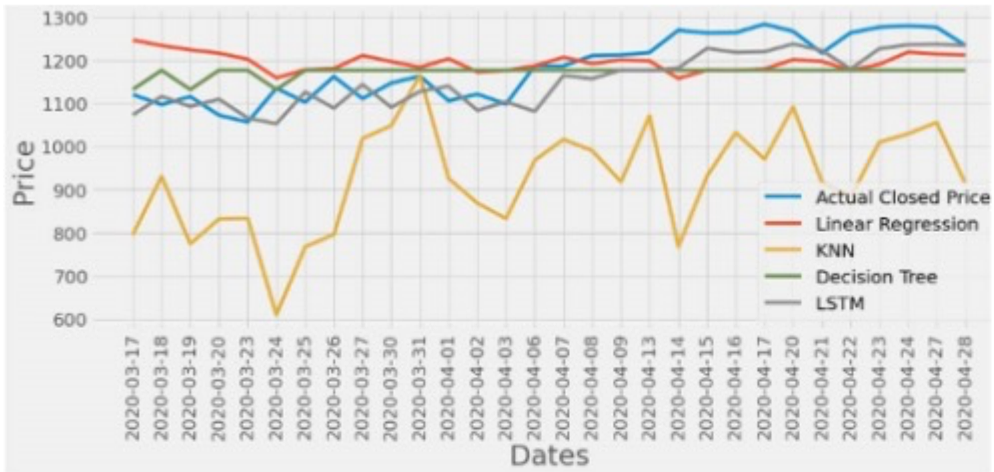


Fig. 13. Graphical Comparison of Actual Close Price and Predicted Closed Price of proposed algorithms.

The LSTM model that we worked on can thus have predicted the stock prices most accurately. A detailed comparison between the price predicted by all the models with the actual price can be seen in Fig. 4.

Date	Actual Price	LSTM	Linear Reg.	Decision Tree	K-NN
2020-04-15	1262.469971	1249.831055	1178.453431	1144.758281	889.541354
2020-04-16	1263.469971	1234.953613	1178.701617	1198.114363	999.325331
2020-04-17	1283.250000	1235.863892	1180.725238	1198.114363	993.635690
2020-04-20	1266.609985	1256.876099	1202.507832	1198.114363	1081.082257
2020-04-21	1216.339966	1236.862427	1199.284338	1198.114363	906.231156
2020-04-22	1263.209961	1186.866699	1178.426143	1144.758281	841.254103
2020-04-23	1276.310059	1244.269531	1191.902488	1198.114363	1002.173795
2020-04-24	1279.310059	1252.695312	1219.459388	1144.758281	971.730004
2020-04-27	1275.880005	1252.447876	1216.085958	1198.114363	1063.193852
2020-04-28	1233.670044	1248.345337	1213.198386	1144.758281	893.822223

Fig. 14. Actual Price of stocks and its comparison with the predicted values

CHAPTER-7
Conclusion

7.1 Limitations of the Project

There are limitations when this technique is applied to solving business problems because the problem complexity makes it difficult to completely explain the results provided by ML-driven classifiers.

7.2 Conclusions and Future Enhancements

The project has a lot of scope in future and the topic and related research will always be in demand because the application of this has the power to control the flow of money which will keep the interest boosted. Hybrid models could be developed to get better accuracy and eliminate flaws that a single model produced. Apart from that, we can use NLP to view things from a sentimental analysis point of view.

7.3. Summary

Authors opine that application of machine learning techniques in stock price forecasting needs to be a well thought process and demands painstakingly detailed execution. The proposed approach is a paradigm shift in this class of problems by reformulating a traditional forecasting model as a classification problem. Moreover, knowledge discovery from the analysis should create new frontiers or applications such as a trading strategy based on the strengths of the classification accuracy, investigating the behavior of certain classes of stocks.

References

[1] B. Chhimwal and V. Bapat, “Impact of foreign and domestic investment in stock market volatility: Empirical evidence from India,” Cogent Economics & Finance, vol. 8, no. 1, Apr. 2020.

[2] S. Sridhar, S. Mootha, and S. Subramanian, “Decentralized Stock Exchange Implementation using Ethereum,” in 2020 International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 234- 241, 2020.

[3] C. Pop et al., “Decentralizing the Stock Exchange using Blockchain And Ethereum-based implementation of the Bucharest Stock Exchange,” in 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 459-466, 2018.

[4] N. Sakthivel and A. Saravanakumar, “Investors’ Satisfaction on Online Share Trading and Technical Problems Faced by the Investors: A Study in Coimbatore District of Tamilnadu,” International Journal of Management Studies, vol. V, no. 3(9), p. 71, Jul. 2018.

[5] D. Shah, H. Isah, and F. Zulkernine, “Stock Market Analysis: A Review and Taxonomy of Prediction Techniques,” International Journal of Financial Studies, vol. 7, no. 2, p. 26, May 2019.

[6] U. Hathi, “Indian Companies Act 2013 Highlights and Review,” SSRN Electronic Journal, 2014.

[7] S. V. Shenoy and K. Srinivasan, “Relationship of IPO Issue Price and Listing Day Returns with IPO Pricing Parameters,” International Journal of Management Studies, vol. V, no. 4(1), p. 11, Oct. 2018.

[8] G. Tanty and P. K. Patjoshi, “A Study on Stock Market Volatility Pattern of BSE and NSE in India,” Asian Journal of Management, vol. 7, no. 3, p. 193, 2016.

[9] S. Sridhar, S. Mootha, and S. Subramanian, “Detection of Market Manipulation using Ensemble Neural Networks,” in 2020 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1–8, 2020

[10] S. Zavadzki, M. Kleina, F. Drozda, and M. Marques, “Computational Intelligence Techniques Used for Stock Market Prediction: A Systematic Review,” IEEE Latin America Transactions, vol. 18, no. 04, pp. 744–755, Apr. 2020.

[11] M. C. Joshi, “Factors Affecting Indian Stock Market,” SSRN Electronic Journal, 2013.

[12] S. Alhazbi, A. B. Said and A. Al-Maadid, "Using Deep Learning to Predict Stock Movements Direction in Emerging Markets: The Case of Qatar Stock Exchange," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, pp. 440-444, 2020.

[13] D. Wei, “Prediction of Stock Price Based on LSTM Neural Network,” in 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), pp. 544-547, 2019.

[14] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, “Deep Reinforcement Learning for Sequence-to-Sequence Models,” IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 7, pp. 2469–2489, 2019.

[15] K. Palasundram, N. Mohd Sharef, N. A. Nasharuddin, K. A. Kasmiran, and A. Azman, “Sequence to Sequence Model Performance for Education Chatbot,” International Journal of Emerging Technologies in Learning (iJET), vol. 14, no. 24, p. 56, Dec. 2019.

[16] H. Jain and G. Harit, "An Unsupervised Sequence-to-Sequence Autoencoder Based Human Action Scoring Model," 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, pp. 1-5, 2019.

[17] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, “Evaluation of bidirectional LSTM for short-and long-term stock market prediction,” in 2018 9th International Conference on Information and Communication Systems (ICICS), pp. 151-156, 2018.

[18] J. Chou and T. Nguyen, "Forward Forecast of Stock Price Using SlidingWindo Metaheuristic-Optimized Machine-Learning Regression," in IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3132- 3142, July 2018.

[19] Y. Zhang and Q. Yang, “An overview of multi-task learning,” National Science Review, vol. 5, no. 1, pp. 30–43, 2018.

Appendices

Appendix 1: Screenshots

Loading Dataset

In [53]:
```
df = pd.read_csv('GOOG.csv',index_col='Date',parse_dates=True)
```

In [54]:
```
print(df.head())
```

```
                 Open       High        Low      Close  Adj Close    Volume
Date
2004-08-19  49.813286  51.835709  47.800831  49.982655  49.982655  44871300
2004-08-20  50.316402  54.336334  50.062355  53.952770  53.952770  22942800
2004-08-23  55.168217  56.528118  54.321388  54.495735  54.495735  18342800
2004-08-24  55.412300  55.591629  51.591621  52.239193  52.239193  15319700
2004-08-25  52.284027  53.798351  51.746044  52.802086  52.802086   9232100
```

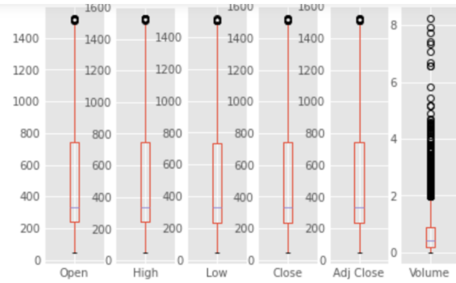
In [55]:
```
df.drop(['Adj Close'], axis=1)
```

Out[55]:

Date	Open	High	Low	Close	Volume
2004-08-19	49.813286	51.835709	47.800831	49.982655	44871300
2004-08-20	50.316402	54.336334	50.062355	53.952770	22942800
2004-08-23	55.168217	56.528118	54.321388	54.495735	18342800
2004-08-24	55.412300	55.591629	51.591621	52.239193	15319700
2004-08-25	52.284027	53.798351	51.746044	52.802086	9232100
...	...	...	...	...	...
2020-04-29	1341.459961	1359.989990	1325.339966	1341.479980	3793600
2020-04-30	1324.880005	1352.819946	1322.489990	1348.660034	2668900
2020-05-01	1328.500000	1352.069946	1311.000000	1320.609985	2072500
2020-05-04	1308.229980	1327.660034	1299.000000	1326.800049	1504000
2020-05-05	1337.920044	1373.939941	1337.459961	1351.109985	1650700

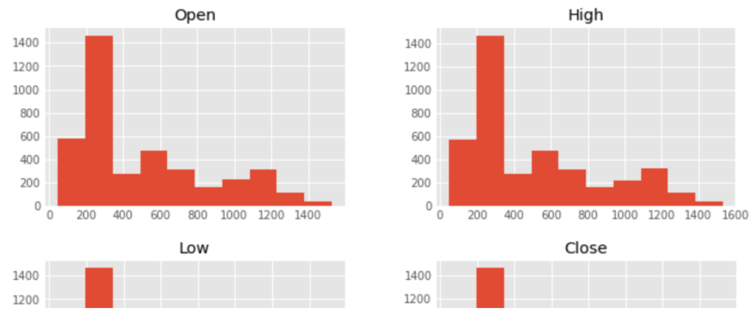
3955 rows × 5 columns

A1.1 Dataset Details




```
In [57]: df.hist(figsize = (10,10))
```

```
Out[57]: array([[<AxesSubplot:title={'center':'Open'}>,
                 <AxesSubplot:title={'center':'High'}>],
                [<AxesSubplot:title={'center':'Low'}>,
                 <AxesSubplot:title={'center':'Close'}>],
                [<AxesSubplot:title={'center':'Adj Close'}>,
                 <AxesSubplot:title={'center':'Volume'}>]], dtype=object)
```



A1.2 Dataset representation

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  iloc._setitem_with_indexer(indexer, value)
```

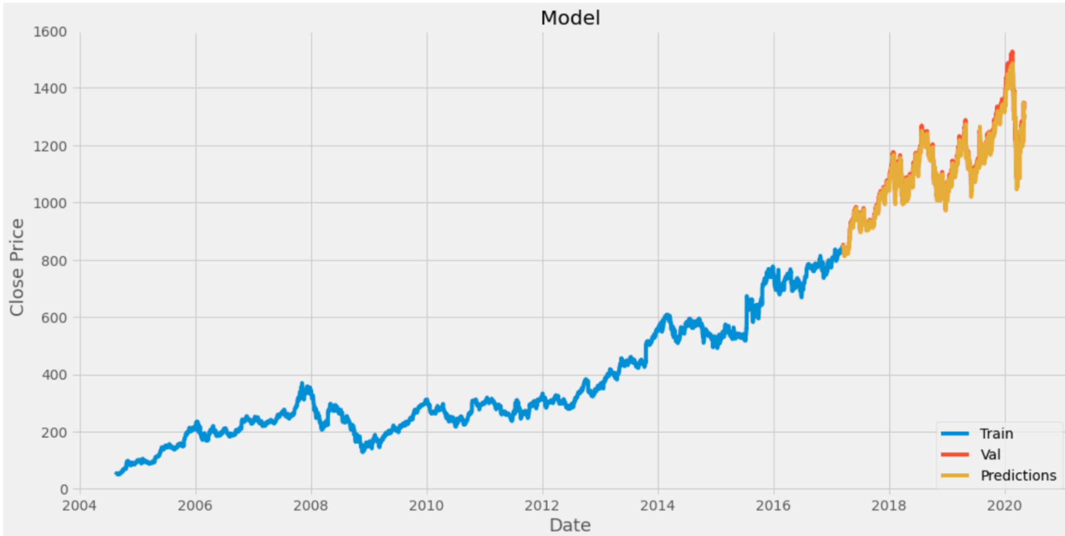
Decision Tree

A1.3 Historical Data

```
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
plt.show()
```

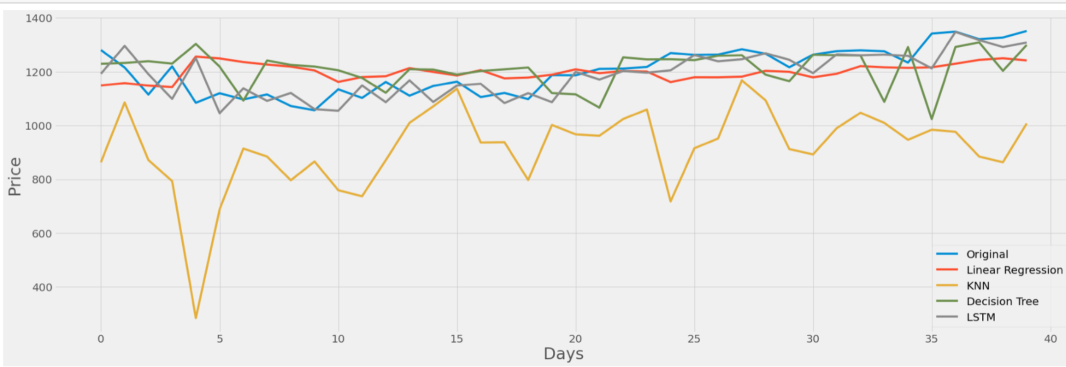
```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/ipykernel_launcher.py:4: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy
  after removing the cwd from sys.path.
```



A1.4 Predicted Graph

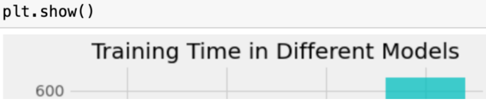
```
predictions['Decision Tree'].plot(figsize = (30,10), fontsize = 20)
predictions['LSTM'].plot(figsize = (30,10), fontsize = 20)
plt.legend(['Original','Linear Regression', 'KNN', 'Decision Tree', 'LSTM'], fontsize=20)
plt.xlabel('Days', fontsize = 30)
plt.ylabel('Price', fontsize = 30)
plt.show()
```



In [102]:
```
objects = ('LR', 'DT', 'KNN', 'LSTM')
y_pos = np.arange(len(objects))
performance = [time_lr, time_dt, time_knn, total_lstm]

plt.bar(y_pos, performance, align='center', color='c', alpha = 0.75)
plt.xticks(y_pos, objects)
plt.ylabel('Training Time (s)')
plt.title('Training Time in Different Models')

plt.show()
```



A1.5 Comparison with other models