CSMSS

CHH. SHAHU COLLEGE OF ENGINEERING

KANCHANWADI, PAITHAN ROAD, AURANGABAD – 431011

A REPORT OF MINI PROJECT ON

"Stock Price Prediction "

SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENT FOR

THIRD YEAR

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By

Mr.Vaibhav Dagwal (Roll No: CS3234)

Miss.Vaishnavi Kharbal (Roll No: CS3238)

Under the guidance of

Prof. V. U. Gaikwad

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Academic Year: 2022-23

CSMSS

CHH. SHAHU COLLEGE OF ENGINEERING

KANCHANWADI, PAITHAN ROAD, AURANGABAD – 431011

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Mini Project work entitled **"Stock Price Prediction "** is bonafide work carried out by **Mr.Vaibhav Dagwal ,** Roll no**:** **CS3234 , Miss.Vaishnavi Kharbal ,** Roll no**:** **CS3238 ,** in partial fulfillment for curriculum of Third Year in **Computer Science & Engineering** of **Dr. Babasaheb Ambedkar Technological University**, Lonere, Raigad, during academic year 2022-23.

Prof. V. U.Gaikwad	**Dr. S. P. Abhang**	**Dr. U. B. Shinde**
(Project Guide)	(Head of the Department)	(Principal)

DECLARATION

We, **Vaibhav Dagwal , Vaishnavi Kharbal & Roll No's CS3234,CS3238** students of **5th** semester in B.Tech in Computer Science & Engineering, CSMSS Chh. Shahu College of Engineering, Aurangabad, hereby declare that the project work entitled **"Stock Price Prediction "** submitted to the Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad during academic year 2022-23, is a record of an original work done by us.

This mini project work is submitted in partial fulfillment for curriculum of Third Year of the requirement for in Computer Science & Engineering.

The results embodied in this report have not been submitted to any University or Institute for the award of any degree.

Date: Vaibhav Dagwal (CS3234)

Place: Aurangabad Vaishnavi Kharbal (CS3238)

ACKNOWLEDGMENT

We take this opportunity to express our profound gratitude and deep regards to our project guide **Prof. V.U.Gaikwad** for her exemplary guidance, monitoring and constant encouragement throughout the course of this report.

We are thankful to **Mr. Arvind S. Sardar** Project Coordinator, **Dr. S. P. Abhang**, Head – Computer Science & Engineering department and **Dr. U. B. Shinde**, Principal- CSMSS Chh. Shahu College of Engineering, Aurangabad for their continuous support and encouragement throughout the academic.

We are also obliged to faculty members of Computer Science & Engineering Department, CSMSS Chh. Shahu College of Engineering, Aurangabad, for their valuable guidance and information provided by them in their respective fields.

Date: Vaibhav Dagwal (CS3234)

Place: Aurangabad Vaishnavi Kharbal (CS3238)

CONTENTS

List of Figures

List of Abbreviations

1. LSTM - Long short-term memory
2. LMS - Least Mean Square
3. SDLC - Software Development life cycle

ABSTRACT

In this project we attempt to implement machine learning approach to predict stock prices. Machine learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices in order to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades.

There are two types of stocks. You may know of intraday trading by the commonly used term "day trading." Intraday traders hold securities positions from at least one day to the next and often for several days to weeks or months. LSTMs are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. While predicting the actual price of a stock is an uphill climb, we can build a model that will predict whether the price will go up or down.

Keywords: LSTM, CNN, ML, DL, Trade Open, Trade Close, Trade Low, Trade High

Chapter 1

INTRODUCTION

1.1 Introduction

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. 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 analysed.

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 of time 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 study on it and proposed various models. In this project, LSTM model is used to predict the stock price.

1.2 Objective of Project

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. main objective of this project is to find the best model to predict the value of the stock market.

During the process of considering various techniques and variables that must be taken into account, we found out that techniques like random forest, support vector machine were not exploited fully.

The successful prediction of the stock will be a great asset for the stock market institutions and will provide real-life.

1.3 Problem Definition

Stock Market Prediction is the act of trying to determine the future values of company stock or other stock market prediction is basically defined as trying to determine the stock value and offer a robust idea for the people to know and predict the market and the stock prices. It is generally presented using the quarterly financial ratio using the dataset.

The use of machine learning and artificial intelligence techniques to predict the prices of the stock is an increasing trend. The main task is we try to predict the future trend by using the historical data of data and we try to train the ml model with available back -testing data of the stock to get future trend approximately. The programming language is used to predict the stock market using machine learning is Python and as there are many ML algorithms like KNN, Recurrent Neural Network, LSTM, Reinforcement learning to predict the stock trend as of now we are using the most basic and widely used machine learning algorithm “linear regression” on dataset.

Chapter 2

LITERATURE SURVEY

2.1 Related Work

The research work done by Manh Ha Duong Boriss Siliverstovs[11]. Investigating the relation between equity prices and aggregate investment in major European countries including France, Germany, Italy, the Netherlands and the United Kingdom. Increasing integration of European financial markets is likely to result in even stronger correlation between equity prices in different European countries. This process can also lead to convergence in economic development across European countries if developments in stock markets influence real economic components, such as investment and consumption. Indeed, our vector autoregressive models suggest that the positive correlation between changes equity prices and investment is, in general, significant. Hence, 6 monetary authorities should monitor reactions of share prices to monetary policy and their effects on the business cycle.

2.1.1 Automated Stock Price Prediction Using Machine Learning

The research work done by Mariam Moukalled Wassim El-Hajj Mohamad Jaber[10] Computer Science Department American University of Beirut. Traditionally and in order to predict market movement, investors used to analyze the stock prices and stock indicators in addition to the news related to these stocks. Hence, the importance of news on the stock price movement. Most of the previous work in this industry focused on either classifying the released market news as (positive, negative, neutral) and demonstrating their effect on the stock price or focused on the historical price movement and predicted their future movement.

In this work, we propose an automated trading system that integrates mathematical functions, machine learning, and other external factors such as news' sentiments for the purpose of achieving better stock prediction accuracy and issuing profitable trades. Particularly, we aim to determine the price or the trend of a certain stock for the coming end-of-day considering the first several trading hours of the day. To achieve this goal, we trained traditional machine learning algorithms and created/trained multiple deep learning models taking into consideration the importance of the relevant news. Various experiments were conducted, the highest accuracy (82.91%) of which was achieved using SVM for Apple Inc. (AAPL) stock.

2.1.2 Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model

The research work done by Hyeong Kyu Choi[16], B.A Student Dept. of Business Administration Korea University Seoul, Korea. Predicting the price correlation of two assets for future time periods is important in portfolio optimization. We apply LSTM recurrent neural networks (RNN) in predicting the stock price correlation coefficient of two individual stocks. RNN's are competent in understanding temporal dependencies. The use of LSTM cells further enhances its long-term predictive properties. To encompass both linearity and nonlinearity in the model, we adopt the ARIMA model as well. The ARIMA model filters linear tendencies in the data and passes on the residual value to the LSTM model. The ARIMA-LSTM hybrid model is tested against other traditional predictive financial models such as the full historical model, constant correlation model, single-index model and the multi-group model. In our empirical study, the predictive ability of the ARIMA-LSTM model turned out superior to all other financial models by a significant scale. Our work implies that it is worth considering the ARIMALSTM model to forecast correlation coefficient for portfolio optimization.

2.1.3 Event Representation Learning Enhanced with External Common-sense Knowledge

The research work done by Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li[14], Junwen Duan Research Center for Social Computing and Information Retrieval Harbin Institute of Technology, China. Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event. Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market.

2.1.4 Forecasting directional movements of stock prices for intraday trading using LSTM and random forests

The research work done by Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari SahooDepartment of Computer Science & Information Systems, BITS Pilani K.K.Birla Goa campus, India bDivision of Mathematical Sciences, Nanyang Technological University, Singapore cDepartment of Mathematics, BITS Pilani K.K.Birla Goa campus, India. We employ both random forests and LSTM networks (more precisely CuDNNLSTM) as training methodologies to analyze their effectiveness in forecasting out-of-sample directional movements of constituent stocks of the S&P 500 from January 1993 till December 2018 for intraday trading. We introduce a multi-feature setting consisting not only of the returns with respect to the closing prices, but also with respect to the opening prices and intraday returns. As trading strategy, we use Krauss et al. (2017) and Fischer & Krauss (2018) as benchmark and, on each trading day, buy the 10 stocks with the highest probability and sell short the 10 stocks with the lowest probability to outperform the market in terms of intraday returns – all with equal monetary weight. Our empirical results show that the multi-feature setting provides a daily return, prior to transaction costs, of 0.64% using LSTM networks, and 0.54% using random forests. Hence, we outperform the single feature setting in Fischer & Krauss (2018) and Krauss et al. (2017) consisting only of the daily returns with respect to the closing prices, having corresponding daily returns of 0 .41% and of 0 .39% with respect to LSTM and random forests, respectively. 1 Keywords: Random forest, LSTM, Forecasting, Statistical Arbitrage, Machine learning, Intraday trading.

2.1.5 A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance

The research work done by Xiao-Yang[14] Liu1 Hongyang Yang,Qian Chen4,Runjia ZhangLiuqing Yang Bowen Xiao Christina Dan Wang Electrical Engineering, 2Department of Statistics, 3Computer Science, Columbia University, 3AI4Finance LLC., USA, Ion Media Networks, USA, Department of Computing, Imperial College, 6New York University (Shanghai). As deep reinforcement learning (DRL) has been recognized as an effective approach in quantitative finance, getting hands-on experiences is attractive to beginners. However, to train a practical DRL trading agent that decides where to trade, at what price, and what quantity involves error-prone and arduous development and debugging. In this paper, we introduce a DRL library FinRL that facilitates beginners to expose themselves to quantitative finance and to develop their own stock trading strategies. Along with easily-reproducible tutorials, FinRL library allows users to streamline their own developments and to compare with existing schemes easily. Within FinRL, virtual environments are configured with stock market datasets, trading agents are trained with neural networks, and extensive back testing is analyzed via trading performance. Moreover, it incorporates important trading constraints such as transaction cost, market liquidity and the investor's degree of risk-aversion. FinRL is featured with completeness, hands-on tutorial and reproducibility that favors beginners: (i) at multiple levels of time granularity, FinRL simulates trading environments across various stock markets, including NASDAQ-100, DJIA, S&P 500, HSI, SSE 50, and CSI 300; (ii) organized in a layered architecture with modular structure, FinRL provides fine-tuned state-of-the-art DRL algorithms (DQN, DDPG, PPO, SAC, A2C, TD3, etc.), commonly used reward functions and standard evaluation baselines to alleviate the debugging workloads and promote the reproducibility, and (iii) being highly extendable, FinRL reserves a complete set of user-import interfaces. Furthermore, we incorporated three application demonstrations, namely single stock trading, multiple stock trading, and portfolio allocation. The FinRL library will be available on GitHub at link https://github.com/AI4Finance-LLC/FinRLLibrary.

2.1.6 An innovative neural network approach for stock market prediction

The research work done by Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin[4]. To develop an innovative neural network approach to achieve better stock market predictions. Data were obtained from the live stock market for real-time and off-line analysis and results of visualizations and analytics to demonstrate Internet of Multimedia of Things for stock analysis.

To study the influence of market characteristics on stock prices, traditional neural network algorithms may incorrectly predict the stock market, since the initial weight of the random selection problem can be easily prone to incorrect predictions. Based on the development of word vector in deep learning, we demonstrate the concept of "stock vector." The input is no longer a single index or single stock index, but multi-stock high-dimensional historical data. We propose the deep long short-term memory neural network (LSTM) with embedded layer and the long shortterm memory neural network with automatic encoder to predict the stock market. In these two models, we use the embedded layer and the automatic encoder, respectively, to vectorize the data, in a bid to forecast the stock via long short-term memory neural network. The experimental results show that the deep LSTM with embedded layer is better. Specifically, the accuracy of two models is 57.2 and 56.9%, respectively, for the Shanghai A-shares composite index. Furthermore, they are 52.4 and 52.5%, respectively, for individual stocks. We demonstrate research contributions in IMMT for neural network-based financial analysis.

2.1.7 An Intelligent Technique for Stock Market Prediction

The research work done by M. Mekayel Anik · M. Shamsul Arefin (B)[19] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh. A stock market is a loose network of economic transactions between buyers and sellers based on stocks also known as shares. In stock markets, stocks represent the ownership claims on businesses. These may include securities listed on a stock exchange as well as those only traded privately. A stock exchange is a place where brokers can buy and/or sell stocks, bonds, and other securities. Stock market is a very vulnerable place for investment due to its volatile nature. In the near past, we faced huge financial problems due to huge drop in price of shares in stock markets worldwide. This phenomenon brought a heavy toll on the international as well as on our national financial structure. Many people lost their last savings of money on the stock market. In 2010–2011 financial year, Bangladeshi stock market faced massive collapse [1]. This phenomenon can be brought under control especially by strict monitoring and instance stock market analysis. If we can analyse stock market correctly in time, it can become a field of large profit and may become comparatively less vulnerable for the investors. Stock market is all about prediction and rapid decision making about investment, which cannot be done without thorough analysis of the market. If we can predict the stock market by analysing historical data properly, we can avoid the consequences of serious market collapse and to be able to take necessary steps to make market immune to such situations.

2.2 Requirement Analysis

System configuration :

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

2.2.1 Hardware Requirements:

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

2.2.2 Software requirements

- Python 3.5 in Google Colab is used for data pre-processing, model training and prediction.
- Operating System: windows 7 and above or Linux based OS or MAC OS.

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.

Notice that each requirement is directly related to what we expect the software to do. They represent some of the core functions.

Non-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.

Chapter 3

SYSTEM ANALYSIS & DESIGN

In software design, as in mathematics, the representation schemes used are of fundamental importance. At least three levels of design notations exist: external design specifications, which describe the external characteristics of a software system; architectural design specifications, which describe the structure of the system; and detailed design specifications, which describe control flow, data representation, and other algorithmic details within the modules.

3.1 Type of SDLC Model Used:

Software Development life cycle (SDLC) is a spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.

There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "Software Development Process Models." Each process model follows a series of phase unique to its type to ensure success in the step of software development.

3.2 UML Diagrams

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as templated texts.

The kind of the diagram is defined by the primary graphical symbols shown on the diagram. For example, a diagram where the primary symbols in the contents area are classes is class diagram. A diagram which shows use cases and actors is use case diagram. A sequence diagram shows sequence of message exchanges between lifelines.

UML specification does not preclude mixing of different kinds of diagrams, e.g. to combine structural and behavioral elements to show a state machine nested inside a use case. Consequently, the boundaries between the various kinds of diagrams are not strictly enforced. At the same time, some UML Tools do restrict set of available graphical elements which could be used when working on specific type of diagram.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams.

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

3.2.1 Use Case Diagram

In the Unified Modelling 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.

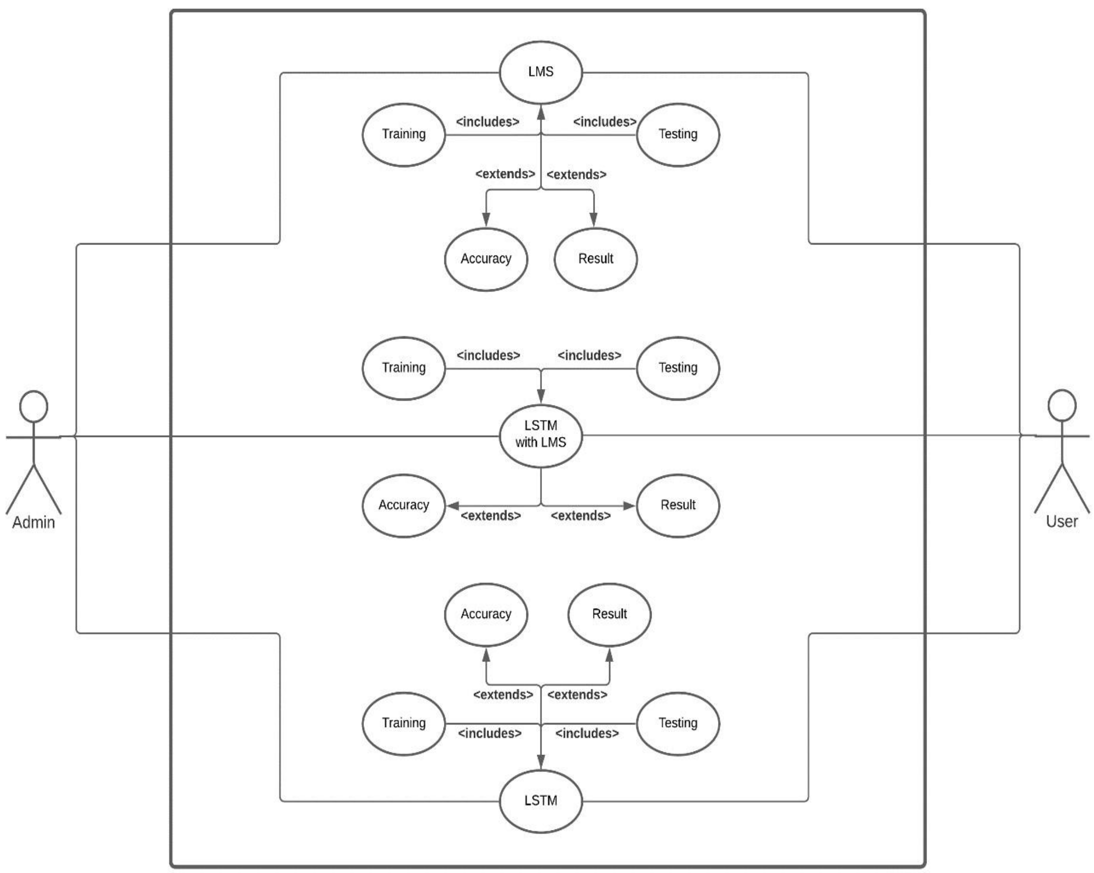


Fig. 1: Using LMS, LSTM and LSTM with LMS in the system

3.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

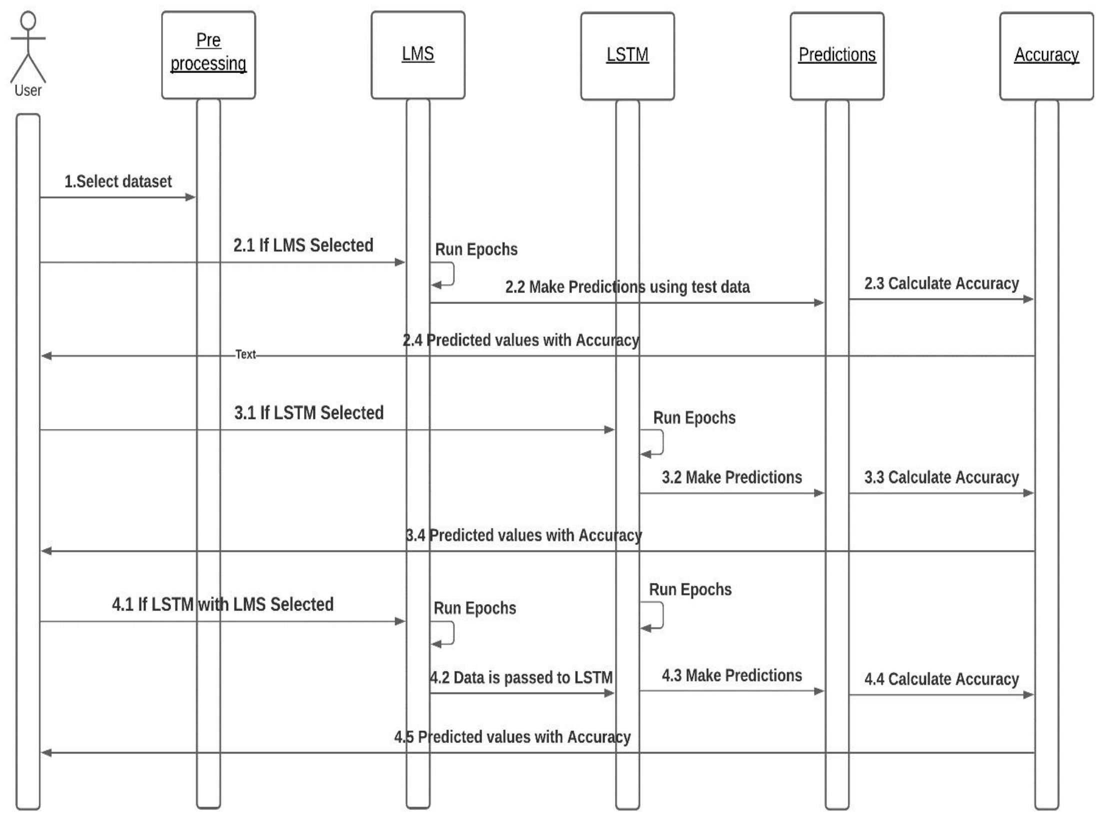


Fig. 2: Execution based on model selection

3.2.3 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

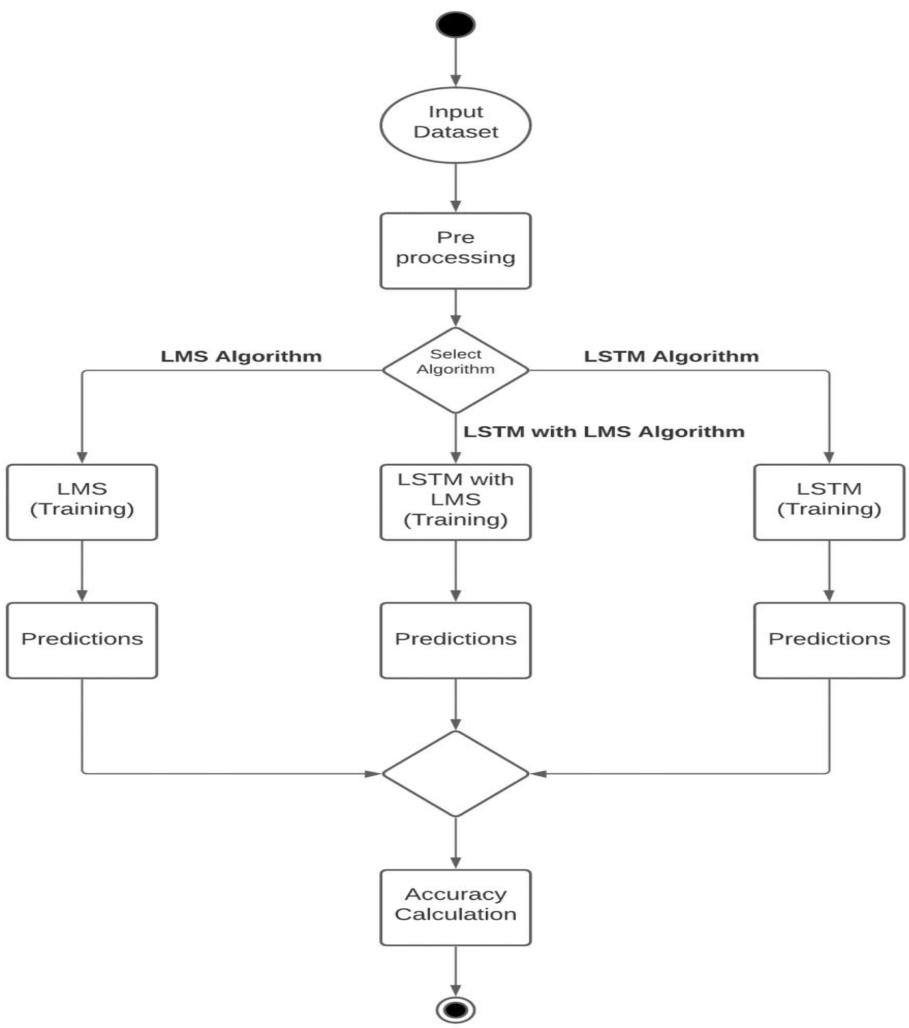


Fig. 3: Execution based on algorithm selection

3.2.4 Collaboration Diagram

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case.

They are the primary source of information used to determining class responsibilities and interfaces. The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

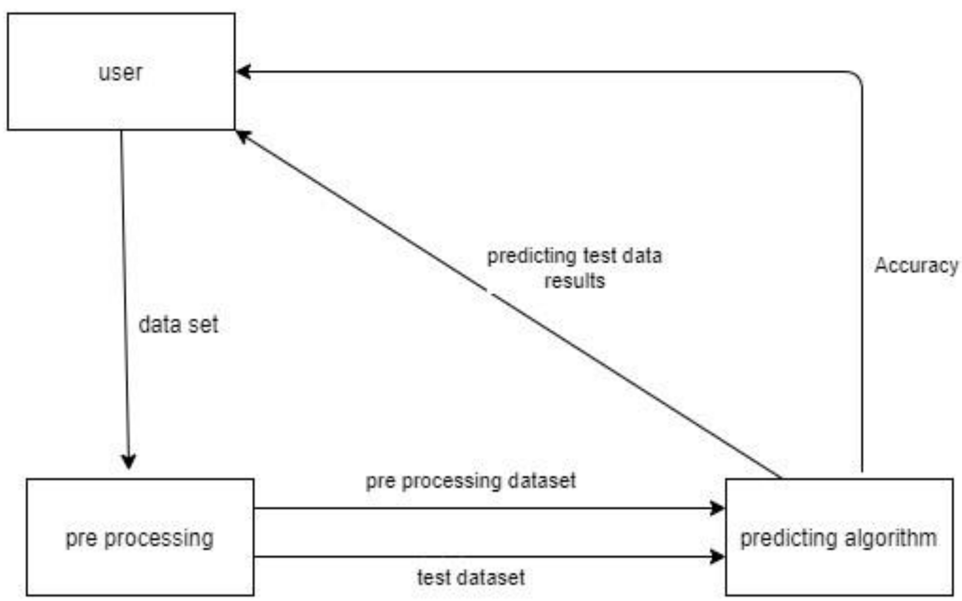


Fig. 4: Data transfer between modules

3.2.5 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

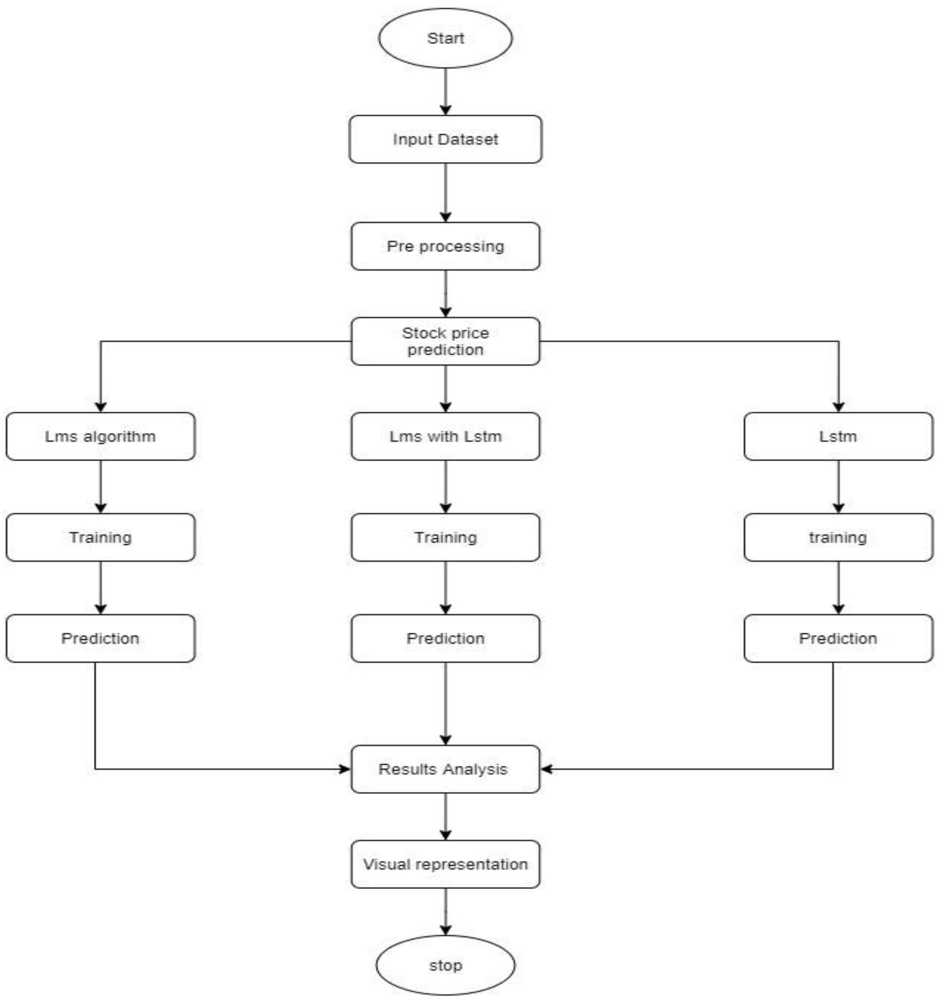


Fig. 5: Flow of execution

Chapter 4

SYSTEM IMPLEMENTATION

4.1 Implementation Details

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. They used Long short-term memory network (LSTM).

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.

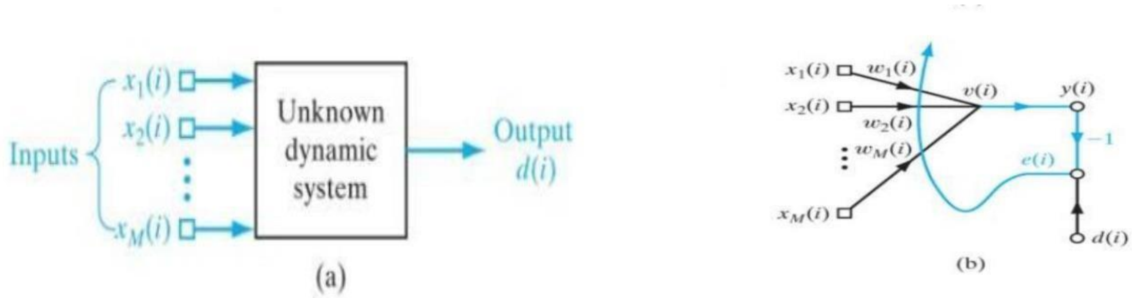


Fig. 6: LMS Inputs and Outputs

Algorithm 1: LMS

Input:
x : input vector
d: desired vector
μ: learning rate
N: filter order
Output:
y: filter response
e: filter error
begin
  $M = size(x)$;
  $x_n(0) = w_n(0) = [0\ 0\ \dots\ 0]^T$;
  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 the filter 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.

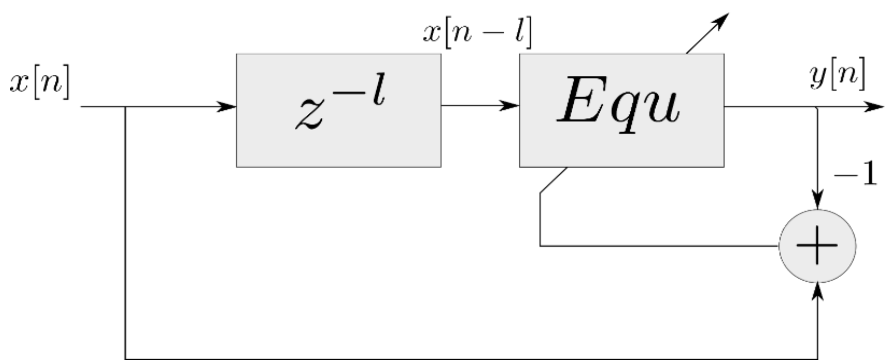


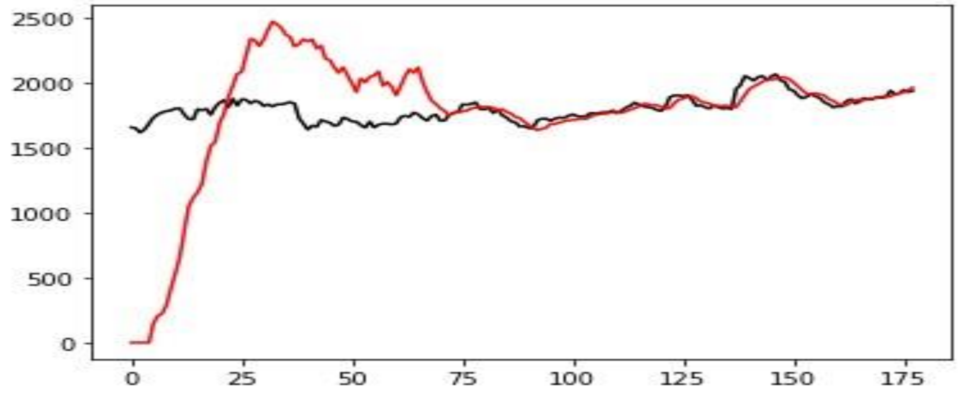
Fig. 7: LMS updating weights

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.

Algorithm 2: LMSPred

Input:
x: input vector
l: quantity of days to predict
μ: learning rate
N: filter order
Output:
y: filter response
begin
  $M = size(x_d)$;
  $x_n(0) = w_n(0) = [0\ 0\ \dots\ 0]$;
  $x_d = [0\ 0\ ..\ 0\ x]$;
  while $n < M$ **do**
    $x_{n+1} = [x_d(n);\ x_n(1:N)]$;
    $y(n) = w_n^H * x_n$;
    if $n > M - l$ **then**
      $e = 0$;
    else
      $e(n) = d(n) - y(n)$;
    end
    $w_{n+1} = w_n + 2\mu e(n) x_n$;
  end
end

Results:



One example of stock market prediction result

LSTM Architecture

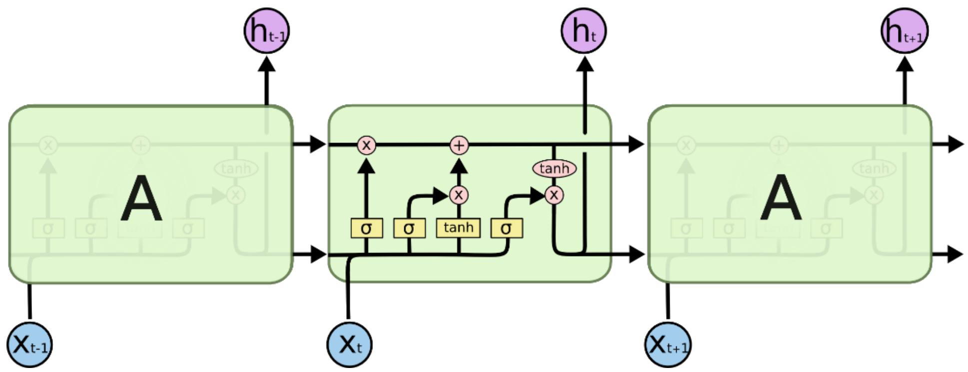


Fig. 8: LSTM Architecture

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_t1 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.

```
• # LSTM
• Inputs: dataset
• Outputs: RMSE of the forecasted data
•
• # Split dataset into 75% training and 25% testing data
• size = length(dataset) * 0.75
• train = dataset [0 to size]
• test = dataset [size to length(dataset)]
•
• # Procedure to fit the LSTM model
• Procedure LSTMAlgorithm (train, test, train_size, epochs)
• X = train
• y = test
```

- model = Sequential ()
- model.add(LSTM(50), stateful=True)
- model.compile(optimizer='adam', loss='mse')
-
- epochs = 100

- # Procedure to make predictions
- Procedure getPredictonsFromModel (model, X)
- predictions = model.predict(X)
- return predictions
- neurons = 50
- predictions = empty 16
- # Fit the LSTM model
- model = LSTMAlgorithm (train, epoch, neurons)
- # Make predictions
- pred = model.predict(train)
- # Validate the model
- n = len(dataset)
- error = 0
- for i in range(n): error += (abs(real[i] - pred[i])/real[i]) * 100
- accuracy = 100 - error/n

SYSTEM ARCHITECTURE

1) Preprocessing of data

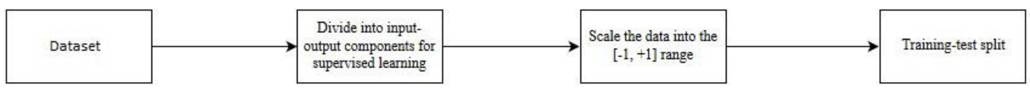


Fig. 9: Pre-processing of data

2) Overall Architecture

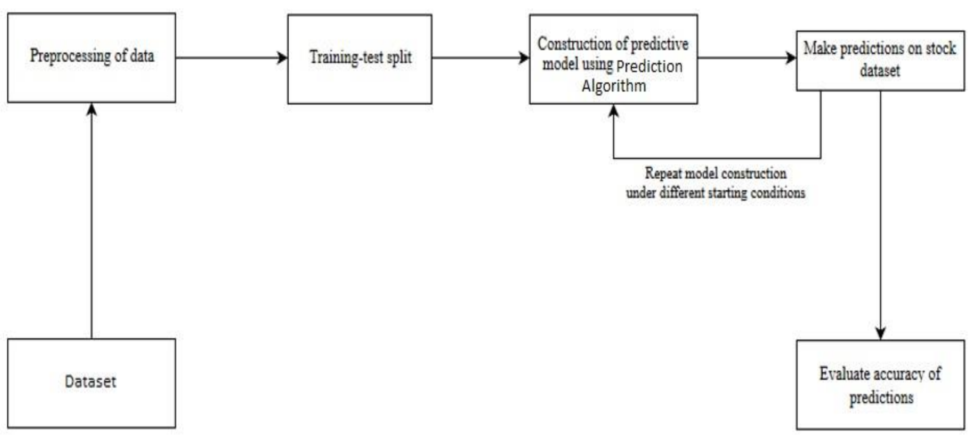
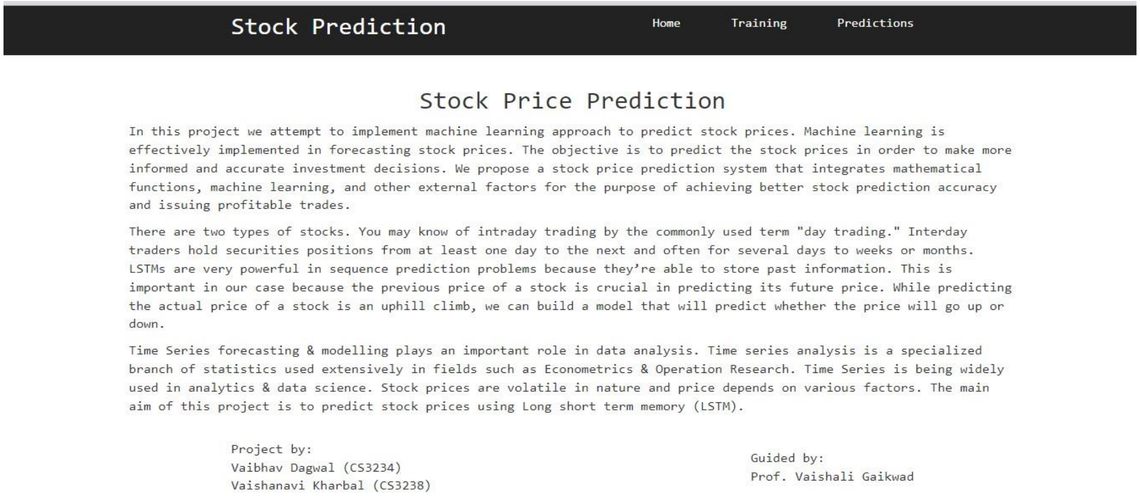
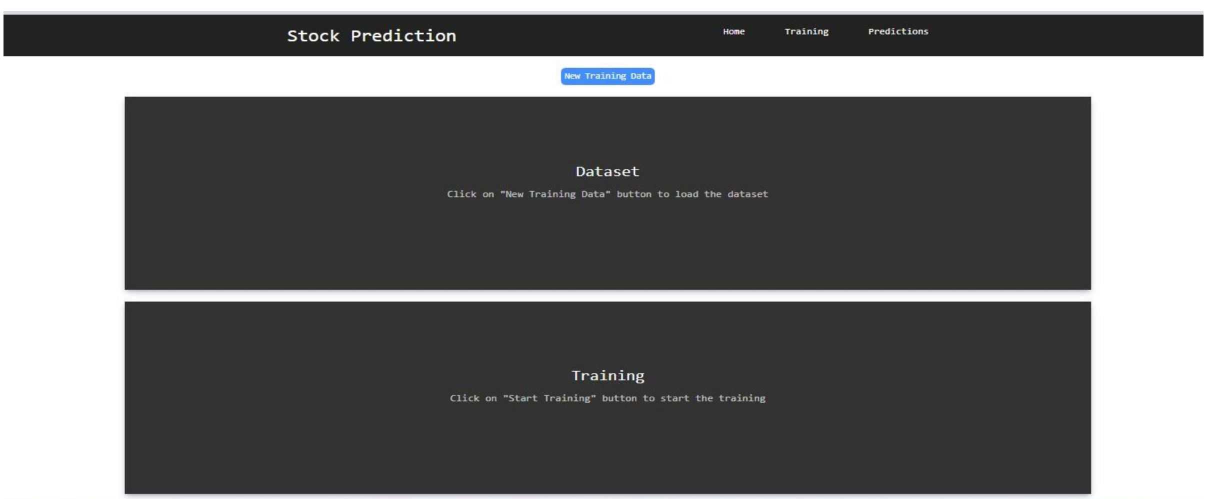


Fig. 10: Overall Architecture

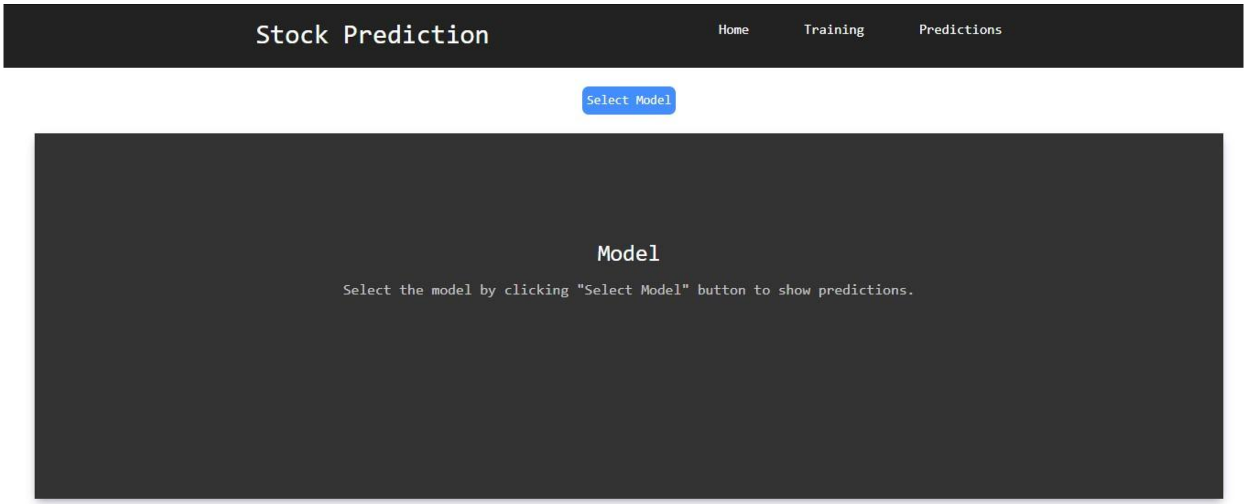
4.2 Snapshots



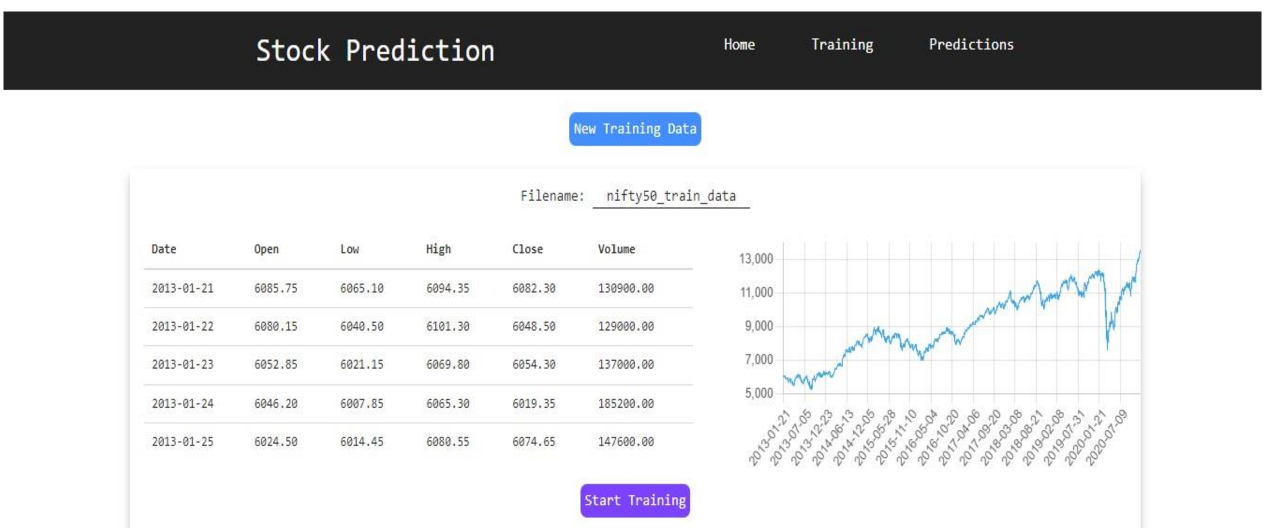
Snapshot 4.1



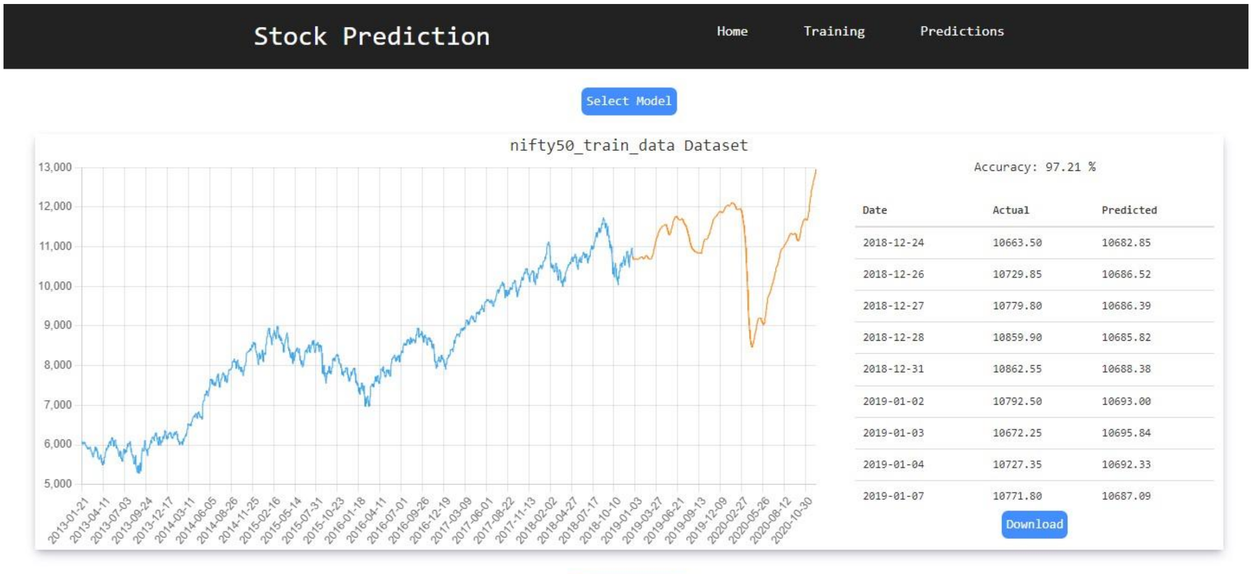
Snapshot 4.2



Snapshot 4.3



Snapshot 4.4


Stock Prediction
Home
Training
Predictions
Select Model
nifty50_train_data Dataset
Accuracy: 97.21 %
Date Actual Predicted
2018-12-24 10663.50 10682.85
2018-12-26 10729.85 10686.52
2018-12-27 10779.80 10686.39
2018-12-28 10859.90 10685.82
2018-12-31 10862.55 10688.38
2019-01-02 10792.50 10693.00
2019-01-03 10672.25 10695.84
2019-01-04 10727.35 10692.33
2019-01-07 10771.80 10687.09
Download

Snaphot 4.5

4.3 Coding:

LSTM Algorithm

```
def LSTMAlgorithm(fileName, train_size, epochs, updateEpochs):
    df = pd.read_csv('./datasets/' + fileName + '.csv')
    cols, dateColName, trade_close_col = getRequiredColumns(df)


    scaling_data_frame = df.filter(cols)

    scaler = MinMaxScaler(feature_range=(0,1))
    scaled_Data = scaler.fit_transform(scaling_data_frame)
    scaled_data_frame = pd.DataFrame(data=scaled_Data, index=[df[trade_close_col]], columns=co

    stock_close_data = df.filter([trade_close_col])
    stock_close_dataset = stock_close_data.values
```

```
    trainingDataLength = math.ceil( len(stock_close_dataset) * train_size )

    scaler = MinMaxScaler(feature_range=(0,1))
    scaledData = scaler.fit_transform(stock_close_dataset)

    StockTrainData = scaledData[0:trainingDataLength , :]

    Xtrain = []
    Ytrain = []
```

LMS

```
def LMS(df, pred_col, next_days, epochs, updateEpochs):
    print("LMS Training for", pred_col)

    ndf, omax, omin = minmaxscaler(df[pred_col], 1000, 2000)
    x = ndf.values

    tmp = []
    for i in x: tmp.append(i)

    x = np.array(tmp)
```

```
def lmsPred(x,l,u,N):
    xd = np.block([1, x]).T
    y=np.zeros((len(xd),1))

    xn = np.zeros((N+1,1))
    xn = np.matrix(xn)

    wn=np.random.rand(N+1,1)/10

    M=len(xd)
    for epoch in range(epochs):
        updateEpochs(epoch)
        print("epoch ", epoch+1, "/", epochs, sep='')

        for n in range(0,M):
            xn = np.block([[xd[n]], [xn[0:N]]])
            y[n]= np.matmul(wn.T, xn)

            if(n>M-l-1): e = 0;
            else: e=int(x[n]-y[n])

            wn = wn + 2*u*e*xn

    return y,wn;
x_train = x[:-next_days]
u = 2**(-30);

l=next_days;
N=100;

y,wn = lmsPred(x_train,l,u,N)

x = inverse_scalar(ndf, omax, omin, 1000, 2000)
y = inverse_scalar(y, omax, omin, 1000, 2000)

# plotGraph(cols=[x, y], title=pred_col, colors=['black', 'red'])

json = {
    "inputs": x,
    "outputs": y,
    "actual": x[-l:].values,
    "predicted": y[-l:]
}

return json
```

Chapter 5

RESULT

5.1 Comparative result to existing system

epochs	Accuracy	MSE	RMSE
10	97.154	201835	449.26
20	95.2489	438515	662.204
30	97.4571	159737	399.671
40	97.8633	95549	309.11
50	97.9285	106734	326.702

epochs	Accuracy
100	97.97517139027555
200	98.30422315197787
300	92.95956921171869
400	98.67452775485742

-Epochs for Nifty50 Dataset using LSTM

epochs	Accuracy	MSE	RMSE
10	90.14171	1.50E+06	1225.58
20	94.41587	5.52E+05	742.9043
30	94.54524	5.47E+05	739.3578
40	96.65602	2.68E+05	517.9008
50	96.79688	2.50E+05	500.3757

epochs	Accuracy
100	97.67512047219317
200	91.36568721761229
300	92.21746083762834
400	88.3763795882347
500	91.26283879244312

- Epochs for Nifty50 Dataset using LSTM with LMS

Chapter 6
CONCLUSION

6.1 CONCLUSION:

We are predicting the closing stock price of any given organization, we have developed an application for predicting close stock price using LSTM algorithm. We have used datasets belonging to Google, Nifty50, TCS, Infosys and Reliance Stocks and achieved above 93% accuracy for these datasets. In the future, we can extend this application for predicting cryptocurrency trading and also, we can add sentiment analysis for better predictions.

6.2 FUTURE SCOPE:

Future scope of this project will involve adding more parameters and factors like the financial ratios, multiple instances, etc. The more the parameters are taken into account more will be the accuracy. The algorithms can also be applied for analyzing the contents of public comments and thus determine patterns/relationships between the customer and the corporate employee. The use of traditional algorithms and data mining techniques can also help predict the corporation`s performance structure as a whole.

REFERENCES

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, GiridharMaji, Narayan C. Debnath, Soumya Sen

[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman - Stock price predictionusing LSTM, RNN and CNN-sliding window model - 2017.

[3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.

[4] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.

[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.

[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.

[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning

[9] Asset Durmagambetov currently works at the mathematics, CNTFI- 'The Riemann HypothesisMillennium Prize Problems' - stock market predictions.

[10] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.

[11] Manh Ha Duong Boriss Siliverstovs June 2006 - The Stock Market and Investment.

[12] Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India - Indian stock market prediction using artificial neural networks on tick data.

[13] Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering - Forecasting the Stock Market Index Using Artificial Intelligence Techniques.

[14] Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, Junwen Duan Research Centre for Social

Computing and Information Retrieval Harbin Institute of Technology, China - Event Representation Learning Enhanced with External Common-sense Knowledge.

[15] Huicheng Liu Department of Electrical and Computer Engineering Queen's University,

Canada - Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network

[16] Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea = Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model.

[17] M. Nabipour Faculty of Mechanical Engineering, Tarbiat Modares University, 14115-143 Tehran, Iran;Mojtaba.nabipour@modares.ac.ir - Deep Learning for Stock Market prediction.

[18] Lavanya Ra SRM Institute of Science and Technology | SRM · Department of Computer Science - Stock Market Prediction.

[19] M. Mekayel Anik · M. Shamsul Arefin (B) Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh - An Intelligent Technique for Stock Market Prediction.

[20] Osman Hegazy, Omar S. Soliman and Mustafa Abdul Salam Faculty of Computers and

Informatics, Cairo University, Egypt - A Machine Learning Model for Stock Market Prediction