Visvesvaraya Technological University

Belgaum, Karnataka- 590014

A Project Report On

“Real time share market analysis using machine learning”

Submitted in the partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF ENGINEERING

In

INFORMATION SCIENCE AND ENGINEERING

ACCREDITED BY NBA

Submitted by

Basavaraj savalagi (1DS17IS402) **Deepak kumar N (1DS17IS405)**

Girish babu JN (1DS17IS407) **Mohammed Sufiyaan baig (1DS17IS413)**

Under the Guidance of

Mrs. Vaidehi M

Asst. Professor, Dept. of ISE

2019-2020

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DAYANANDA SAGAR COLLEGE OF ENGINEERING

SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE-78

DAYANANDA SAGAR COLLEGE OF ENGINEERING

Shavige Malleshwara Hills, Kumaraswamy Layout
Bangalore-560078

Department of Information Science and Engineering

ACCREDITED BY NBA

2019-2020

Certificate

This is to certify that the Project Work entitled -
“Real Time Share market analysis using machine learning” is a bonafide work carried out by **Basavaraj Savalagi** (1DS17IS402), **Deepak kumar N** (1DS17IS405), **Girish babu JN** (1DS17IS407), and **Mohammed Sufiyaan Baig** (1DS17IS413) in partial fulfilment for the 8th semester of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-2020. The Project Report has been approved as it satisfies the academics prescribed for the Bachelor of Engineering degree.

________________	___________________	________________
Signature of Guide	Signature of HOD	Signature of Principal
[Mrs. Vaidehi M]	[Dr. K.N. Rama Mohan Babu]	[Dr. C.P.S Prakash]

Name of the Examiners	Signature with Date
1. ______________________	______________________
2. ______________________	______________________

ACKNOWLEDGEMENT

It is great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project.

We take this opportunity to express our sincere gratitude to **Dayananda Sagar College of Engineering** for having provided us with a great opportunity to pursue our Bachelor Degree in this institution.

In particular we would like to thank **Dr. C. P. S Prakash,** Principal, Dayananda Sagar College of Engineering for his constant encouragement and advice.

Special thanks to **Dr. K.N. Rama Mohan Babu,** HOD, Department of Information Science & Engineering, Dayananda Sagar College of Engineering for his motivation and invaluable support well through the development of this project.

We are highly indebted to our internal guide **Mrs. Vaidehi M,** Asst. Professor, Department of Information Science & Engineering, Dayananda Sagar College of Engineering for her constant support and guidance. she has been a great source of inspiration throughout the course of this project.

Finally, we gratefully acknowledge the support of our families during the completion of the project.

Basavaraj savalagi	**(1DS17IS402)**
Deepak kumar N	**(1DS17IS405)**
Girish babu JN	**(1DS17IS407)**
Mohammed Sufiyaan baig	**(1DS17IS413)**

ABSTRACT

Share market analysis or prediction is demonstration of attempting to decide the real time value of a Organization stock or other financial instrument traded on a financial exchange using tools and techniques of Machine Learning. In this paper we propose a Machine Learning approach that will be trained from the accessible stocks data and gain intelligence and then uses the acquired knowledge for an exact prediction. This model predicts the stock prices using different datasets (for example: nasdaq finance, yahoo finance and google finance) and regression technique. The goal of this model is to analyze the historical data and predict the real time stock prices accurately.

Key Words: – Machine Learning datasets, yahoo finance, google finance.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview:

Predicting stocks correctly can cause heavy and large amount of profits for the vendor and therefore the dealer, it’s been said that prediction is disordered instead of irregular, that means it’s possible to predict stocks by analysis of previous data of respective stock market carefully.

With the help of Machine learning it’s possible to make such kind of process. It forecast a advertise cost on the precarious edge of substantial worth therefore increases the exactness. The indispensable a piece of machine learning is that the collection of data utilized.

The collection of data should be as real and definite as possible because touch change inside the information can propagate enormous changes inside the result. Determining the stock trade has been problematic for financial specialists since market's presence.

Stock Market and stock trade is where forecast doesn't adhere to specific standards to inquire about the value of an offer inside the stock market.

The two financial specialists and industry are engaged with accessible market and need to comprehend whether a particular offer would rise or go over a specific time of your time. The effective forecast of an offer's cost by its investigation could lead on to a major benefit. This is frequently being through with the help of amazingly enormous noteworthy informational collections to delineate changing conditions and therefore, reaffirming the supposition that the measurement designs have huge prescient force with a high likelihood to get productive exchanges and significant yields for interest in business.

1.2 Problem Statement

For developing any share market prediction model in machine learning, the main challenge is to collect the appropriate and correct datasets for accurate analysis and prediction in machine learning model and selection of correct machine learning model is very much important to predict the correct values of stocks. if the collected datasets are not accurate and incorrect then the stock market analysis and prediction goes wrong.

1.3 Objectives

1. To collect accurate datasets of various organizations in order to analyze and predict the stocks of those organization.
2. Systematic analysis of advantages and disadvantages of existing machine learning model for share market analysis.
3. Developing an machine learning model to train data extracted from raw data and testing the raw data and making it trained data results to predict the stock price .
4. Deployment of the efficient model for share market analysis using machine learning.

CHAPTER 2

LITERATURE SURVEY

Various Books and information materials from the web regarding share market analysis using Machine learning have been studied through in order to achieve the required information concern to this project. Among them, following are the key points extracted through:

i) The **Research paper** on **share market prediction** by **mehak usmani, syed hasan adil, and kamran raza** gave us a brief knowledge of Real time share market analysis using machine learning helped us to choose the appropriate approach.

This paper introduces a consolidating results from chronicled data, news data investigation using ARIMA (Autoregressive Integrated Moving Average) and SMA(Simple Moving Average) algorithms.

ii) **The Research paper** on **Future predictions in Indian stock market through linguistic-temporal approach** by **priti saxena and bhaskar pant** gave us a brief knowledge analyzing the forecast uses recorded information to define future forecasts

This paper presents analyzing the forecast uses recorded information to define future forecasts . These forecasts once in a while get the structure of absolute results, and are depicted to show the conduct that compares to the behavior taking place in the future. forecast is one of the significant situation in transient information mining in stock market analysis.

iii) The **Research paper " Stock Market Forecasting Using Hidden Markov Model " by md rafiul hassan, and baikunth nath** gives us an introductory knowledge on to forecast the stock prices for interrelated market HMM.
to forecast the stock prices for interrelated market HMM (Hidden Markov Model) approach used pattern recognization and classification problems.

iv) From the **paper "The impact of microblogging data for stock market prediction"** by **Tiffany Hui-Kang** and gives us an introductory knowledge, the Capability in handling non-linear relationship and furthermore execute another fluffy time arrangement model to improve forecasting. He uses fuzzy modeling for stock price prediction

v) The https://bseindia.com/ website provided us the required real time datasets to train the model in order to predict the stocks.

vi) The https://towardsdatascience.com/ websites helped us understand many concepts related to machine learning that helped us to choose appropriate algorithms for the work.

CHAPTER 3

REQUIREMENTS

Functional Requirements

- The model should be able to pick up raw data from datasets which it is trained on.

Non Functional Requirements

- The accuracy of the predicted value must be precise.
- The model should never fail in the middle of operation.
- The model should work consistently across various platforms.

Software Requirements

- OS Version: Windows 7(64 bit) or newer versions.
- Coding Language: Python 3.6
- Platform: Jupyter Notebook

Hardware Requirements

- Processor: i5 or i7 Intel Processor
- Primary Storage: 8 GB RAM or above (Recommended 16 GB)
- Secondary Storage: Any standard HDD or SDD

CHAPTER 4

SYSTEM ANALYSIS & DESIGN

Analysis

The procedure of breaking a difficult topic or substance into small parts to gain a better knowledge of the problem is known as analysis. Analysts in the field of engineering look at the structures and requirements, mechanisms, and systems dimensions. Analysis is an activity of exploration. The project lifecycle begins in the analysis phase.

System Design

The definition of the architecture of a system, components, modules, interfaces, and data for a system to fulfil specified requirements is system design. Systems design could be seen as the application of systems theory to product development.

The design phase produces the overall design of the software. The goal of design phase is to figure out the modules that should be in the system to fulfil all the system requirements in an efficient manner. It will contain the details of all these modules, their working with other modules and the desired output from each module. The output of the design process is a description of the software architecture.

System Architecture Diagram

The definition of the structure and operation and more views of a system is known as system architecture. A formal description and rendition of a system, organized in a way so that it supports reasoning about the working and behaviors of the system is called architecture description.

System architecture comprises of system components that work together and implement the overall system.

The below figure shows a general block diagram describing the activities performed by this project.

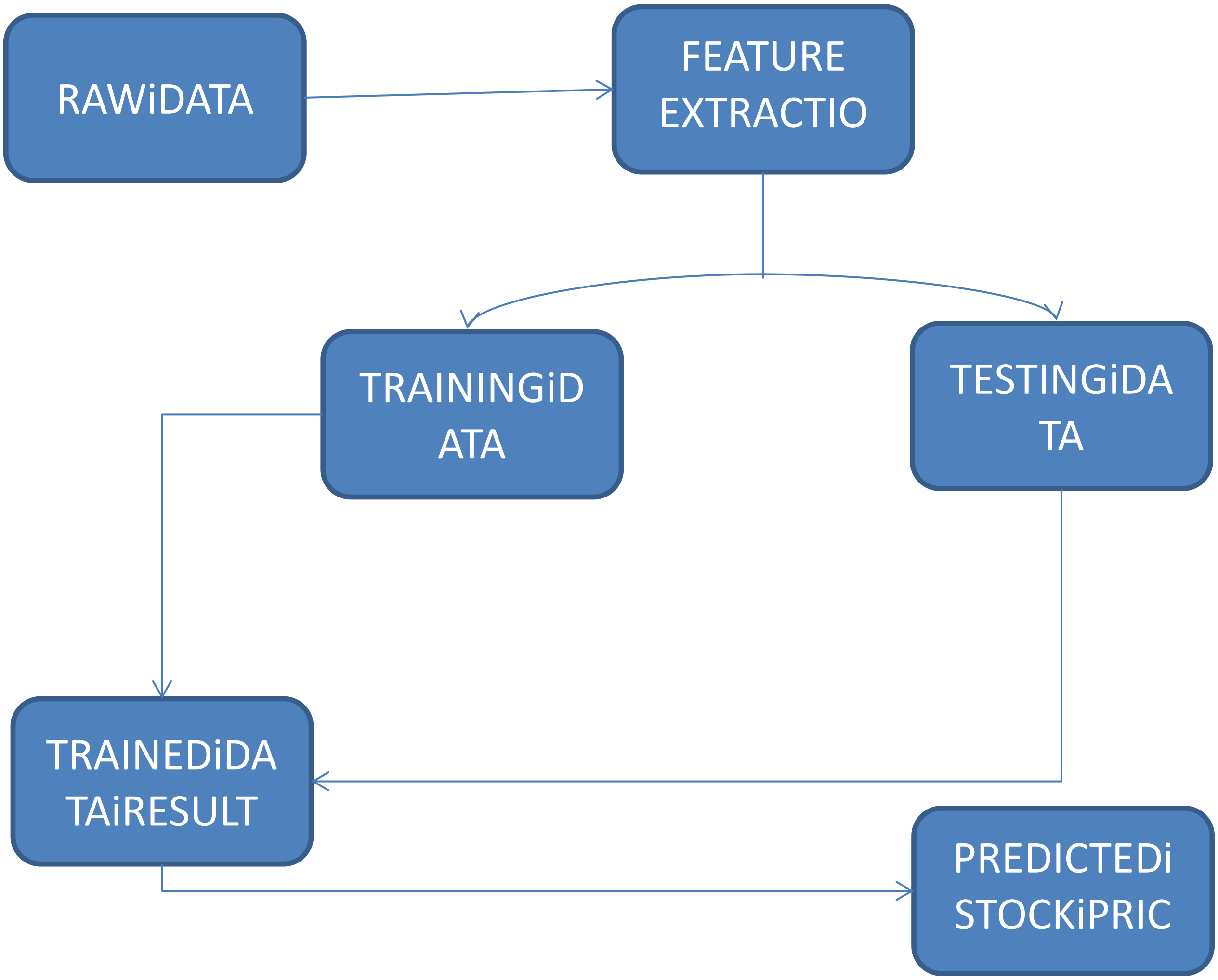


Fig. 4.2.1: System Architecture Diagram

Key Components of this Project are:

i CSV data set

ii LSTM.py

CSV Data set

A **comma-separated values** (**CSV**) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields.

The CSV file format is not fully standardized. The basic idea of separating fields with a comma is clear, but that idea gets complicated when the field data may also contain commas or even embedded line breaks. CSV implementations may not handle such field data, or they may use quotation marks to surround the field. Quotation does not solve everything: some fields may need embedded quotation marks, so a CSV implementation may include escape characters or escape sequences.

In addition, the term "CSV" also denotes some closely related delimiter-separated formats that use different field delimiters, for example, semicolons. These include tab-separated values and space-separated values. A delimiter that is not present in the field data (such as tab) keeps the format parsing simple. These alternate delimiter-separated files are often even given a .csv extension despite the use of a non-comma field separator. This loose terminology can cause problems in data exchange. Many applications that accept CSV files have options to select the delimiter character and the quotation character. Semicolons are often used in some European countries, such as Italy, instead of commas.

Data Flow Diagram

A data flow diagram is the graphical representation of the flow of data through an information system. DFD is very useful in understanding a system and can be efficiently used during analysis. A DFD shows the flow of data through a system. It views a system as a function that transforms the inp uts into desired outputs. Any complex systems will not perform this transformation in a single step an d a data will typically undergo a series of transformations before it becomes the output.

With a data flow diagram, users are able to visualize how the system will operate that the syst em will accomplish and how the system will be implemented, old system data flow diagrams can be d rawn up and compared with a new systems data flow diagram to draw comparisons to implement a m ore efficient system.

Data flow diagrams can be used to provide the end user with a physical idea of where they inp ut, ultimately as an effect upon the structure of the whole system.

In the perspective of application development Data Flow Diagram (DFD) is a special chart ty pe which lets graphically illustrate the "flow" of data through various application component. So the Data Flow Diagrams can be successfully used for visualization of data processing or structured design .

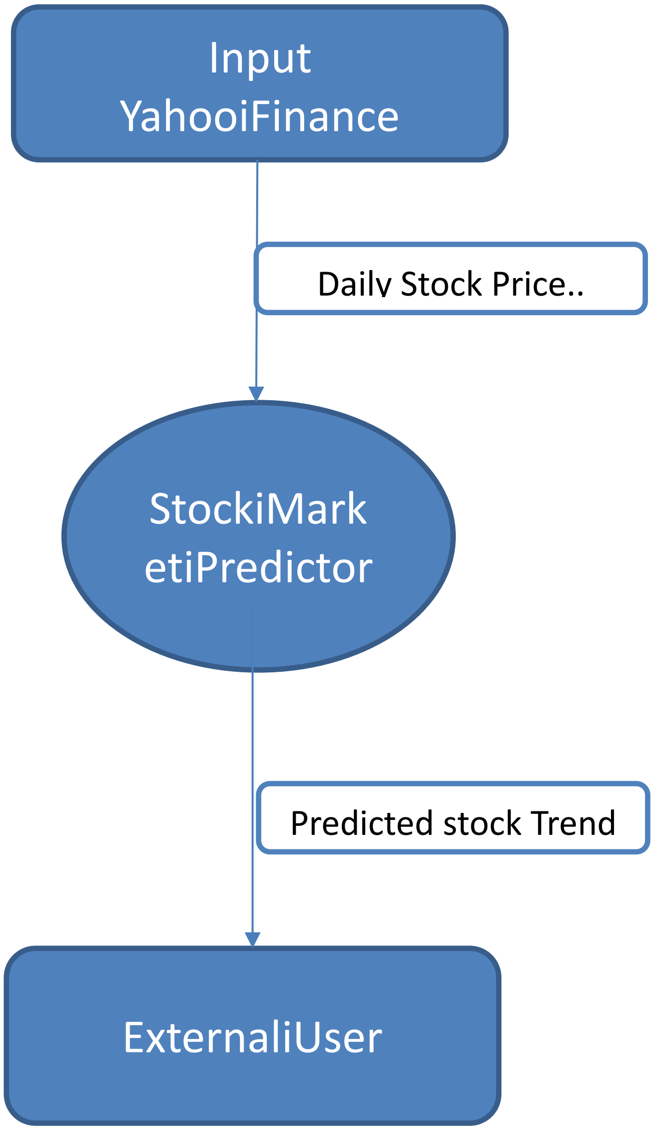


Fig. 4.2.1.2: Data Flow Diagram

Flow Chart

Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

- A processing step, usually called activity, and denoted as a rectangular box.
- A decision usually denoted as a diamond.

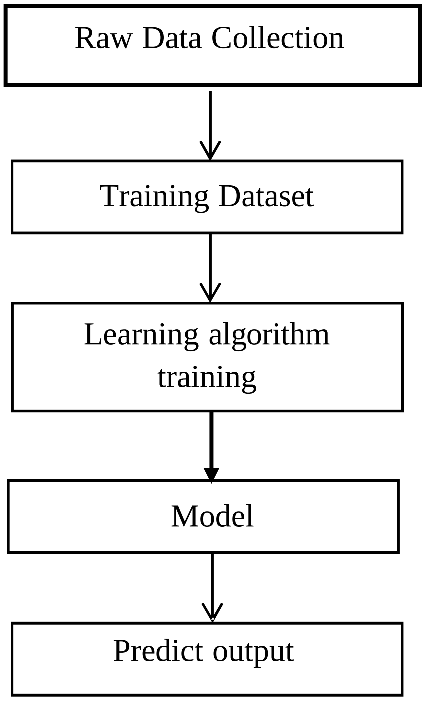


Fig. 4.2.1.3: Flowchart

Use Case Diagram

The external objects that interact directly with the system are called actors. Actors include hu mans, external devices and other software systems. The important thing about actors is that they are n ot under control of the application. In this project, user of the system is the actor. To find use cases, f or each actor, list the fundamentally different ways in which the actor uses the system. Each of these ways is a use case.

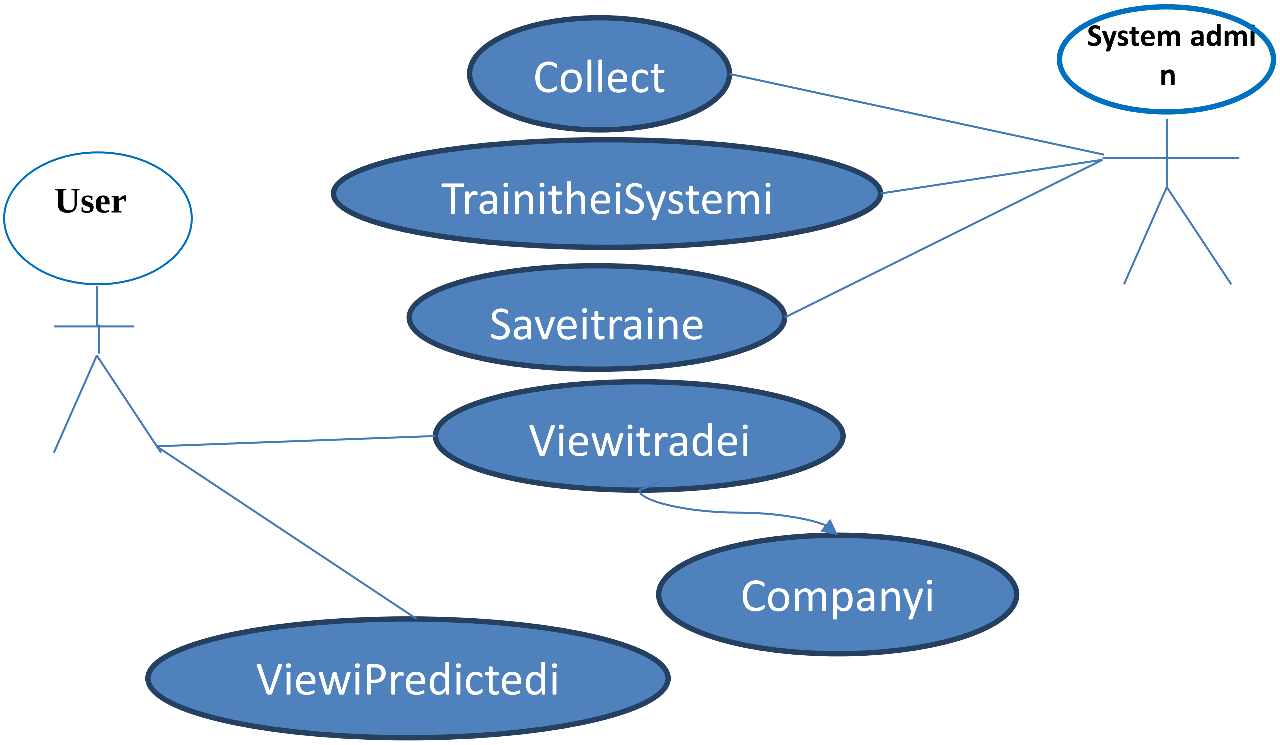


Fig. 4.2.1.4 Use Case Diagram

Sequence Diagram

A sequence diagram in a Unified Modelling Language (UML) is a kind of inte raction diagram that shows how processes operate with one another and in what order. It shows the participants in an interaction and the sequence of messages among them; each participant is assigned a column in a table.

Below section shows the sequence diagrams in this application

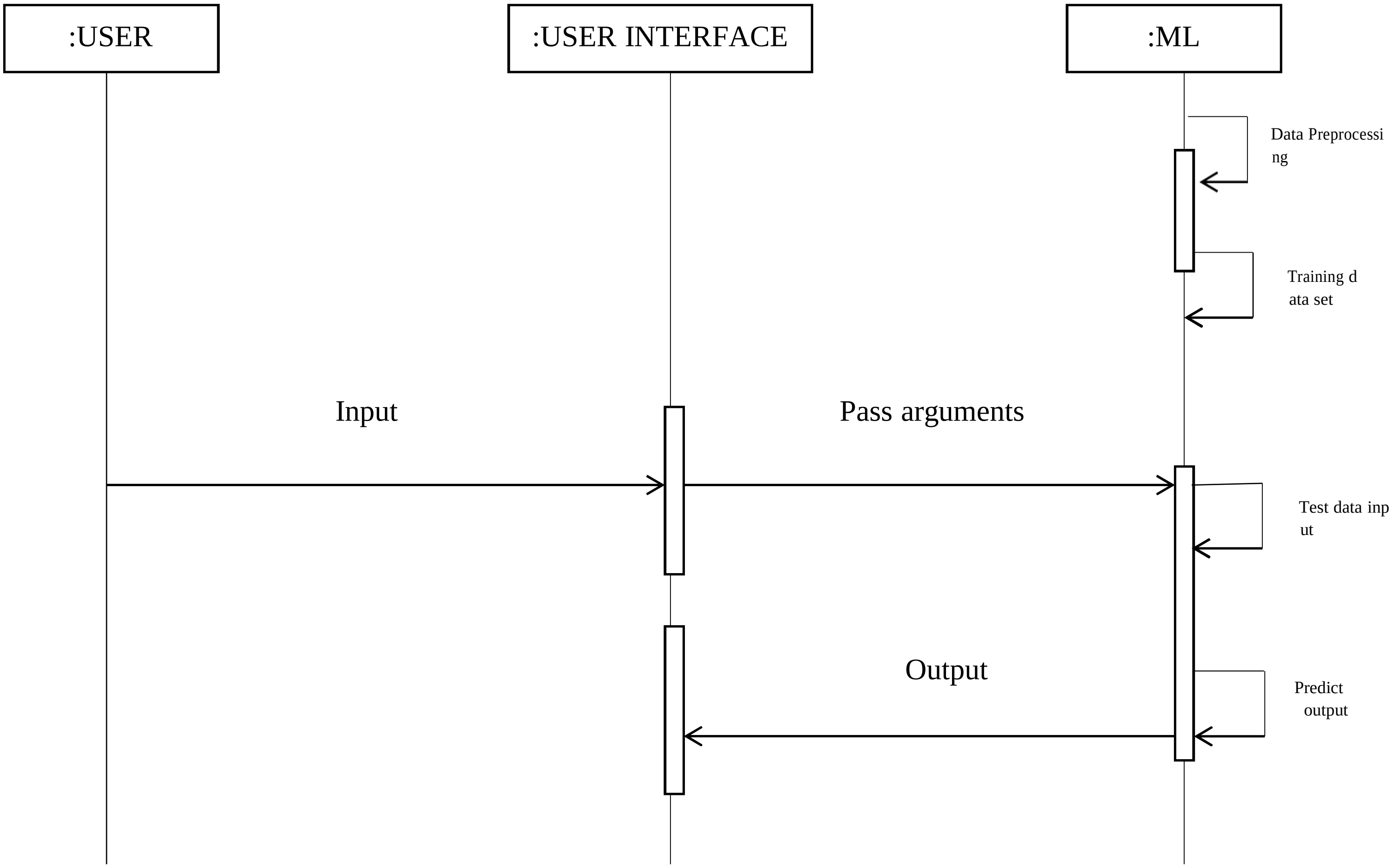


Fig. 4.2.1.5 Sequence Diagram

CHAPTER 5

IMPLEMENTATION

Introduction

The realizing of an application or execution of a plan, idea, model, design, specification, standard, algorithm, or policy is known as implementation. In other words, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

Implementation is one of the most important phases of the Software Development Life Cycle (SDLC). It encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. Specifically, it involves coding the system using a particular programming language and transferring the design into an actual working system.

Overview of System Implementation

This project is implemented considering the following aspects:

1. Usability Aspect.
2. Technical Aspect.

Usability Aspect

The usability aspect of implementation of the project is realized using two principles:

The project is implemented using PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic tape system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open-source software and has a community-based development model. Python and CPython are managed by the non-profit Python Software Foundation.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Technical Aspect

The technical aspect of implementation of the project is realized using following principle:

Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science

, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 13 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Tensorflow

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

Originally developed by researchers and engineers from the Google Brain team within Google's A I

organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

Implementation Support

Installation of Jupyter Notebook

Following are the requirements for installation of Jupyter Notebook on Windows Operating System:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 2 GB RAM minimum, 4 GB RAM recommended
- 1 GB of available disk space minimum, 2 GB Recommended
- 1280x800 minimum screen resolution
- Python 3.3 or greater or Python 2.7

To install Jupyter Notebook on Windows, we should proceed as follows:

1. Download Anaconda.
2. Install the version of Anaconda which you downloaded, following the instructions on the downloa d page.
3. Jupyter Notebook is automatically installed. To run the notebook: jupyter notebook

CHAPTER 6

PSEUDO CODE

Pseudo Code uses the structural conventions of a normal programming language but the intention is for human reading rather than machine reading. Omission of details that are essential for machine understanding of the algorithm such as variable declarations, specific system code and some sub-routines is pseudocode. Augmentation of programming language is done with natural language description details where convenient or with compact mathematical notation. The purpose of using pseudocode is that it is easier for people to understand than conventional programming language code and that the key principles of an algorithm are efficiently and environment independently described. Pseudocode is commonly used in textbooks and scientific publications that are documenting various algorithms and also in plan of computer program development, for sketching the structure of the program before the actual coding takes place.

Steps:

1. Firstly, we imported Python libraries.
2. Next, we created a web page to predict the output.
3. Loading of trained model was done.
4. Finally model is run real-time.

Front-End – index.html

```
<html>
<head>
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link href="css/materialize.min.css" type="text/css" rel="stylesheet" media="screen,projection"/>
  <link href="css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>
  <style>
  .close-first{
    display: none;
  }
  </style>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="style.css">
  <script src="https://kit.fontawesome.com/a076d05399.js"></script>
</head>
<body><br>
  <div class="back">
  <input type="checkbox" id="check">
  <label for="check">
    <i class="fas fa-bars" id="btn"></i>
    <i class="fas fa-times" id="cancel"></i>
  </label>
  <div class="sidebar">
    <header><b>Menu</b></header>
    <ul>
      <li><a href="#"><i class="fas fa-qrcode"></i>Companies</a></li>
      <li><a href="#"><i class="fas fa-link"></i>Graphs</a></li>
      <li><a href="#"><i class="fas fa-stream"></i>Overview</a></li>
      <li><a href="#"><i class="fas fa-calendar-week"></i>Events</a></li>
      <li><a href="#"><i class="fas fa-envelope"></i>About</a></li>
    </ul>
  </div>
  <section>
  <div class="row" style="padding-left:10px;padding-right:10px">
  <ul class="collapsible" data-collapsible="accordion">
```

```
<li>
  <div class="collapsible-header"><i class="material-icons" style="font-size:3rem">settings</i>
    <div class="row" style="margin-bottom:10px;margin-top:10px">
      <div class="col s3 m1">
        Settings
      </div>
      <div class="input-field col s12 m1 right" style="margin-top:5px; width:160px">
        <button id="trainbutton" class="waves-effect waves-light btn red lighten-2">Train</button>
      </div>
      <div class="input-field col s12 m1 right" style="margin-top:5px; width:160px">
        <button id="suggestbutton" class="waves-effect waves-light btn blue lighten-2">Suggest</button>
      </div>
      <div class="file-field input-field col s12 m1 right" style="margin-top:5px; width:160px">
        <div class="btn blue lighten-2" style="height:36px; line-height:2.5rem">
          <span>Pick CSV</span>
          <input id="uploadcsv" type="file">
        </div>
      </div>
    </div>
  </div>
  <div class="collapsible-body"><span>
    <div class="row center">
      <div class="input-field col m2 offset-m1" style="margin-left:5.33%">
        Neural Network settings
      </div>
      <div class="input-field col s12 m1">
        <input id="learningrate" type="number" placeholder="Eg: 0.001" class="validate tooltipped" data-position="bottom" data-delay="50" data-tooltip="learning rate during training">
        <label class="active">Learning rate</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="inputdropoutrate" type="number" placeholder="Eg: 0.9" class="validate tooltipped" data-position="bottom" data-delay="50" data-tooltip="dropout rate for LSTM input">
        <label class="active">Input dropout rate</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="outputdropoutrate" type="number" placeholder="Eg: 0.9" class="validate tooltipped" data-
```

```
position="bottom" data-delay="50" data-tooltip="dropout rate for LSTM output">
        <label class="active">Output dropout rate</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="timestamp" type="number" class="validate tooltipped" placeholder="Eg: 5" data-position="bottom" data-delay="50" data-tooltip="Trends for every minibatch">
        <label class="active">Timestamp per training</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="sizelayer" type="number" class="validate tooltipped" placeholder="Eg: 64" data-position="bottom" data-delay="50" data-tooltip="LSTM size">
        <label class="active">Size layer</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="epoch" type="number" class="validate tooltipped" placeholder="Eg: 10" data-position="bottom" data-delay="50" data-tooltip="Total epoch">
        <label class="active">Training Iteration</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="future" type="number" class="validate tooltipped" placeholder="Eg: 10" data-position="bottom" data-delay="50" data-tooltip="number of days forecast">
        <label class="active">Future days to forecast</label>
      </div>
      <div class="input-field col s12 m1">
        <input id="smooth" type="number" class="validate tooltipped" placeholder="Eg: 10" data-position="bottom" data-delay="50" data-tooltip="Rate anchor smoothing for trends">
        <label class="active">Smoothing weights</label>
      </div>
    </div>
    <div class="row center">
      <div class="input-field col m2 offset-m1" style="margin-left:5.33%">
        Buying & Selling simulation
      </div>
      <div class="input-field col s12 m2">
        <input id="initialmoney" type="number" placeholder="Eg: 10000" class="validate tooltipped" data-position="bottom" data-delay="50" data-tooltip="Money in for simulation">
        <label class="active">Initial money(usd)</label>
      </div>
```

```
          <div class="input-field col s12 m2">
            <input id="maxbuy" type="number" placeholder="Eg: 5" class="validate tooltipped" data-position="bottom" data-delay="50" data-tooltip="Max unit to buy">
            <label class="active">Max buy(unit)</label>
          </div>
          <div class="input-field col s12 m2">
            <input id="maxsell" type="number" class="validate tooltipped" placeholder="Eg: 10" data-position="bottom" data-delay="50" data-tooltip="Max unit to sell">
            <label class="active">Max sell(unit)</label>
          </div>
          <div class="input-field col s12 m2">
            <input id="history" type="number" class="validate tooltipped" placeholder="Eg: 5" data-position="bottom" data-delay="50" data-tooltip="MA to compare of">
            <label class="active">Historical rolling</label>
          </div>
        </div>
      </span></div>
    </li>
  </ul>
</div>

<h6 class='header center light'>WARNING, This website may hang during training, and do not use this website to buy real stock!<br><br>Default stock is Google 2018, you can try upload any stock CSV</h6>
<div class="row" style="padding-left:10px;padding-right:10px">
  <div class="col s12 m12">
    <div id="div_output" style="height: 500px;"></div>
  </div>
</div>
<br>
<div class="row close-first" style="padding-left:10px;padding-right:10px">
  <div class="col s12 m8">
    <div id="div_dist" style="height: 450px;"></div>
  </div>
  <div class="col s12 m4">
    <div class="row">
      <div id="div_loss" style="height: 250px;"></div>
    </div>   <div class="row" id="log" style="height: 150px; overflow:auto;">
    </div>
  </div>
</div>
<div class="row" style="padding-left:10px;padding-right:10px">
  <ul class="collapsible" data-collapsible="accordion">
    <li>
      <div class="collapsible-header"><i class="material-icons">archive</i>Simulation log</div>
      <div class="collapsible-body"><span>
        <table class="bordered highlight">
          <thead>
            <tr>
              <th>Date</th>
              <th>Action</th>
              <th>Price</th>
              <th>Investment</th>
              <th>Balance</th>
            </tr>
          </thead>
          <tbody id='table-body'>
          </tbody>
        </table><br>
        <span id="log-invest"></span>
      </span></div>
    </li>
  </ul>
</div>
<div class="row center" id="color-investment">
</div>
<script src="js/tf.js"></script>
<script src="js/jquery-3.3.1.min.js"></script>
<script src="js/materialize.min.js"></script>
<script src="js/d3.v3.min.js"></script>
<script src="js/numeric-1.2.6.min.js"></script>
<script src="js/numjs.min.js"></script>
<script src="js/utils.js"></script>
<script src="js/echarts.min.js"></script>
```

```
<script src="js/echarts-gl.min.js"></script>
<script src="js/papaparse.min.js"></script>
<script src="data/google.js"> </script>
<script src="init.js"> </script>
</section>
</body>
</html>
```

Lstm.py

```
import sys
import warnings

if not sys.warnoptions:
    warnings.simplefilter('ignore')

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from datetime import datetime
from datetime import timedelta
from tqdm import tqdm
sns.set()
tf.compat.v1.random.set_random_seed(1234)
df = pd.read_csv('../dataset/GOOG-year.csv')
df.head()
minmax = MinMaxScaler().fit(df.iloc[:, 4:5].astype('float32')) # C
lose index
df_log = minmax.transform(df.iloc[:, 4:5].astype('float32')) # Clos
e index
df_log = pd.DataFrame(df_log)
df_log.head()
test_size = 30
simulation_size = 10
df_train = df_log.iloc[:-test_size]
df_test = df_log.iloc[-test_size:]
df.shape, df_train.shape, df_test.shape
```

```
class Model:
    def __init__(
        self,
        learning_rate,
        num_layers,
        size,
        size_layer,
        output_size,
        forget_bias = 0.1,
    ):
        def lstm_cell(size_layer):
            return tf.nn.rnn_cell.LSTMCell(size_layer, state_is_tuple
= False)
               rnn_cells = tf.nn.rnn_cell.MultiRNNCell(
            [lstm_cell(size_layer) for _ in range(num_layers)],
            state_is_tuple = False,
        )
        self.X = tf.placeholder(tf.float32, (None, None, size))
        self.Y = tf.placeholder(tf.float32, (None, output_size))
        drop = tf.contrib.rnn.DropoutWrapper(
            rnn_cells, output_keep_prob = forget_bias
        )
        self.hidden_layer = tf.placeholder(
            tf.float32, (None, num_layers * 2 * size_layer)
        )
        self.outputs, self.last_state = tf.nn.dynamic_rnn(
            drop, self.X, initial_state = self.hidden_layer, dtype = tf.fl
oat32
        )
        self.logits = tf.layers.dense(self.outputs[-1], output_size)
        self.cost = tf.reduce_mean(tf.square(self.Y - self.logits))
        self.optimizer = tf.train.AdamOptimizer(learning_rate).mini
mize(
            self.cost
        )
    def calculate_accuracy(real, predict):
    real = np.array(real) + 1
    predict = np.array(predict) + 1
    percentage = 1 - np.sqrt(np.mean(np.square((real -
 predict) / real)))
    return percentage * 100
    def anchor(signal, weight):
```

```
    buffer = []
    last = signal[0]
    for i in signal:
        smoothed_val = last * weight + (1 - weight) * i
        buffer.append(smoothed_val)
        last = smoothed_val
    return buffer

num_layers = 1
size_layer = 128
timestamp = 5
epoch = 300
dropout_rate = 0.8
future_day = test_size
learning_rate = 0.01
def forecast():
    tf.reset_default_graph()
    modelnn = Model(
        learning_rate, num_layers, df_log.shape[1], size_layer, df_log.shape[1], dropout_rate
    )
    sess = tf.InteractiveSession()
    sess.run(tf.global_variables_initializer())
    date_ori = pd.to_datetime(df.iloc[:, 0]).tolist()

    pbar = tqdm(range(epoch), desc = 'train loop')
    for i in pbar:
        init_value = np.zeros((1, num_layers * 2 * size_layer))
        total_loss, total_acc = [], []
        for k in range(0, df_train.shape[0] - 1, timestamp):
            index = min(k + timestamp, df_train.shape[0] - 1)
            batch_x = np.expand_dims(
                df_train.iloc[k : index, :].values, axis = 0
            )
            batch_y = df_train.iloc[k + 1 : index + 1, :].values
            logits, last_state, _, loss = sess.run(
                [modelnn.logits, modelnn.last_state, modelnn.optimizer, modelnn.cost],
                feed_dict = {
                    modelnn.X: batch_x,
                modelnn.Y: batch_y,
                modelnn.hidden_layer: init_value,
            },
        )
        init_value = last_state
        total_loss.append(loss)
        total_acc.append(calculate_accuracy(batch_y[:, 0], logits[:, 0]))
    pbar.set_postfix(cost = np.mean(total_loss), acc = np.mean(total_acc))

future_day = test_size

output_predict = np.zeros((df_train.shape[0] + future_day, df_train.shape[1]))
output_predict[0] = df_train.iloc[0]
upper_b = (df_train.shape[0] // timestamp) * timestamp
init_value = np.zeros((1, num_layers * 2 * size_layer))

for k in range(0, (df_train.shape[0] // timestamp) * timestamp, timestamp):
    out_logits, last_state = sess.run(
        [modelnn.logits, modelnn.last_state],
        feed_dict = {
            modelnn.X: np.expand_dims(
                df_train.iloc[k : k + timestamp], axis = 0
            ),
            modelnn.hidden_layer: init_value,
        },
    )
    init_value = last_state
    output_predict[k + 1 : k + timestamp + 1] = out_logits

if upper_b != df_train.shape[0]:
    out_logits, last_state = sess.run(
        [modelnn.logits, modelnn.last_state],
        feed_dict = {
            modelnn.X: np.expand_dims(df_train.iloc[upper_b:], axis = 0),
            modelnn.hidden_layer: init_value,
        },
    )
```

```
        output_predict[upper_b + 1 : df_train.shape[0] + 1] = out_logits
        future_day -= 1
        date_ori.append(date_ori[-1] + timedelta(days = 1))

    init_value = last_state

    for i in range(future_day):
        o = output_predict[-future_day - timestamp + i:-future_day + i]
        out_logits, last_state = sess.run(
            [modelnn.logits, modelnn.last_state],
            feed_dict = {
                modelnn.X: np.expand_dims(o, axis = 0),
                modelnn.hidden_layer: init_value,
            },
        )
        init_value = last_state
        output_predict[-future_day + i] = out_logits[-1]
        date_ori.append(date_ori[-1] + timedelta(days = 1))

    output_predict = minmax.inverse_transform(output_predict)
    deep_future = anchor(output_predict[:, 0], 0.3)
    return deep_future[-test_size:]
results = []
for i in range(simulation_size):
    print('simulation %d'%(i + 1))
    results.append(forecast())
accuracies = [calculate_accuracy(df['Close'].iloc[-test_size:].values, r) for r in results]

plt.figure(figsize = (15, 5))
for no, r in enumerate(results):
    plt.plot(r, label = 'forecast %d'%(no + 1))
plt.plot(df['Close'].iloc[-test_size:].values, label = 'true trend', c = 'black')
plt.legend()
plt.title('average accuracy: %.4f'%(np.mean(accuracies)))
plt.show()
```

CHAPTER 7

TESTING

The discovery of errors is the purpose of testing. Discovery of every possible conceivable fault or weakness in a work product is called Testing. Testing provides a way to check the functionality of components, sub assemblies, assemblies and a finished product. Exercise of software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner is the process of testing.

7.1. Unit Testing :

Unit Testing Unit testing is carried out for testing modules constructed from the system design. Each part is compiled using inputs for specific modules. Every modules are assembled into a larger unit during the unit testing process.

Testing has been performed on each phase of project design and coding. The testing of module interface is carried out to ensure the proper flow of information into and out of the program unit while testing. The temporarily generated output data is ensured that maintains its integrity throughout the algorithm's execution by examining the local data structure. Finally, all error-handling paths are also tested.

7.2. Integration Testing :

We usually perform system testing to find errors resulting from unanticipated interaction between the sub-system and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, series of test cases must be developed which ultimately uncover all the possibly existing errors. Different software techniques can be used for this process. These techniques provide systematic guidance for designing test that exercise the internal logic of the software components and exercise the input and output domains of a program to uncover errors in program function, behavior and performance.

We test the software using two methods:

White Box testing: Internal program logic is exercised using this test case design techniques.

Black Box testing: Software requirements are exercised using this test case design techniques. Both techniques help in finding maximum number of errors with minimal effort and time.

7.3. Verification and Validation :

The testing process is a part of broader subject referring to verification and validation. We have to acknowledge the system specifications and try to meet the customer's requirements and for this sole p urpose, we have to verify and validate the product to make sure everything is in place. Verification an d validation are two different things. One is performed to ensure that the software correctly implemen ts a specific functionality and other is done to ensure if the customer requirements are properly met or not by the end product. Verification of the project was carried out to ensure that the project met all th e requirement and specification of our project. We made sure that our project is up to the standard as we planned at the beginning of our project development.

CHAPTER 8

RESULTS

This is the actual model that is used for the Stock Prediction. List of the datasets that is used t o get future stock trends.

Processing for a future stock prices is done and Plots the graph using histogram, echart, buy-sell graph. This prediction is done using LSTM algorithm, this avoid the long-term dependency problem. Remembering information for long periods of time is practically their defa ult behavior.

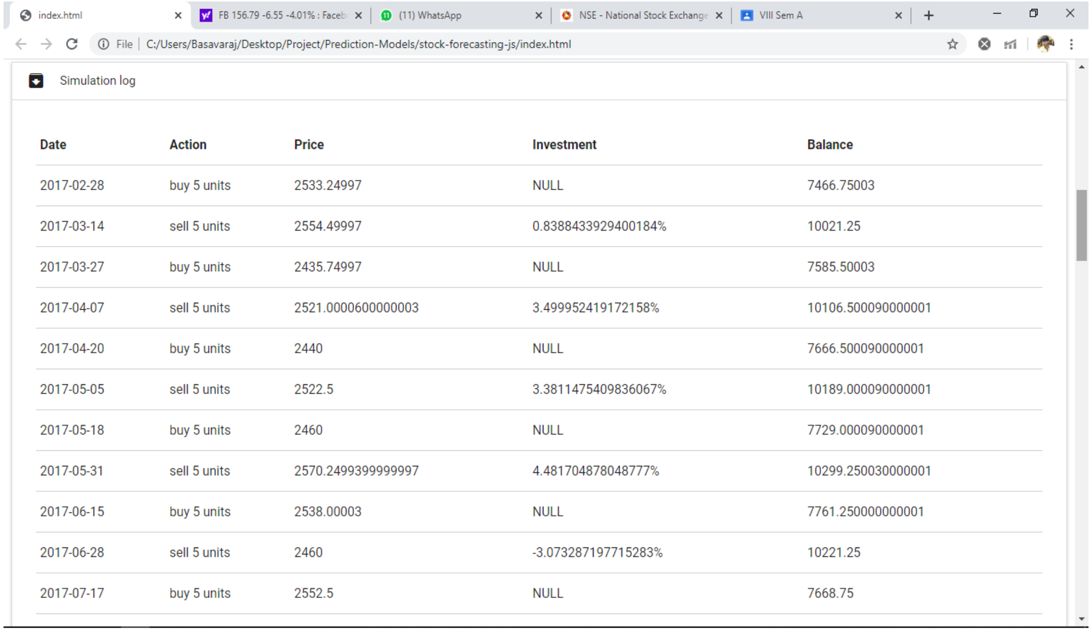
However in the case of LSTM, it uses information from previous lags to predict the future inst ances. Since stock market is a highly dynamical system,the patterns and dynamics existing with in the system will not always be the same.

Testing the data which is extracted from the data pre-processing technique and validates the data to the training the machine.

Training the machine is similar to feeding the data to the algorithm to touch up the test data.

Predicting the stock prices of a company of desired days using the machine learning models. Plotting the predicted stock price result of a company in the graphs and histograms, as output for the users.

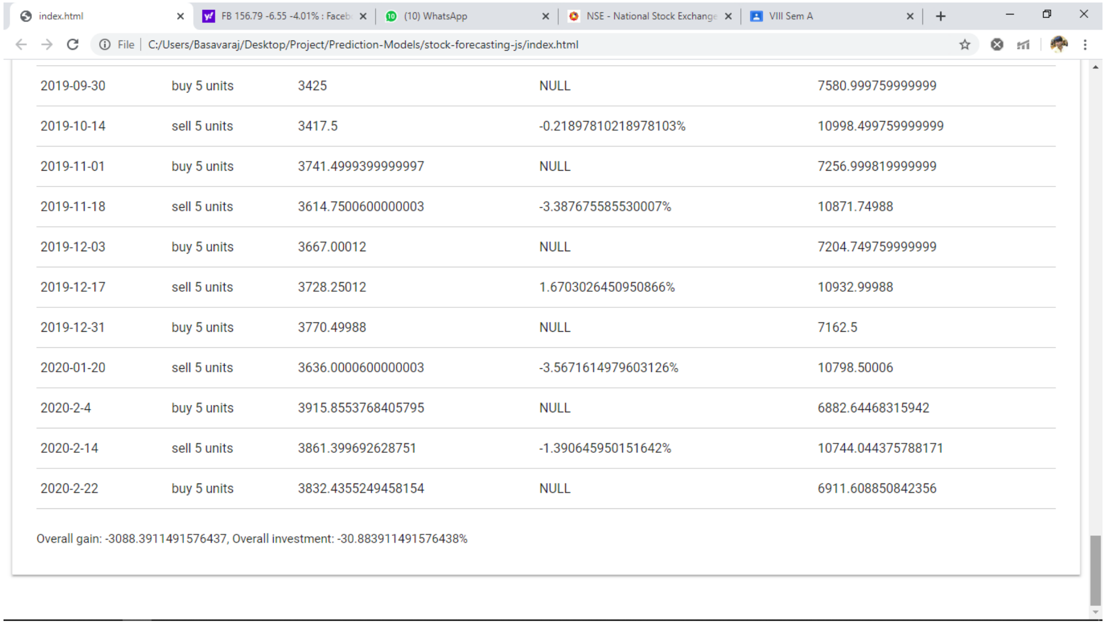
Simulation Log :


index.html
FB 156.79 -6.55 -4.01% : Faceb
(11) WhatsApp
NSE - National Stock Exchange
VIII Sem A
File
C:/Users/Basavaraj/Desktop/Project/Prediction-Models/stock-forecasting-js/index.html

Simulation log

Date	Action	Price	Investment	Balance
2017-02-28	buy 5 units	2533.24997	NULL	7466.75003
2017-03-14	sell 5 units	2554.49997	0.8388433929400184%	10021.25
2017-03-27	buy 5 units	2435.74997	NULL	7585.50003
2017-04-07	sell 5 units	2521.0000600000003	3.499952419172158%	10106.500090000001
2017-04-20	buy 5 units	2440	NULL	7666.500090000001
2017-05-05	sell 5 units	2522.5	3.3811475409836067%	10189.000090000001
2017-05-18	buy 5 units	2460	NULL	7729.000090000001
2017-05-31	sell 5 units	2570.2499399999997	4.481704878048777%	10299.250030000001
2017-06-15	buy 5 units	2538.00003	NULL	7761.250000000001
2017-06-28	sell 5 units	2460	-3.073287197715283%	10221.25
2017-07-17	buy 5 units	2552.5	NULL	7668.75

Fig 8.1.1 Simulation Log 1

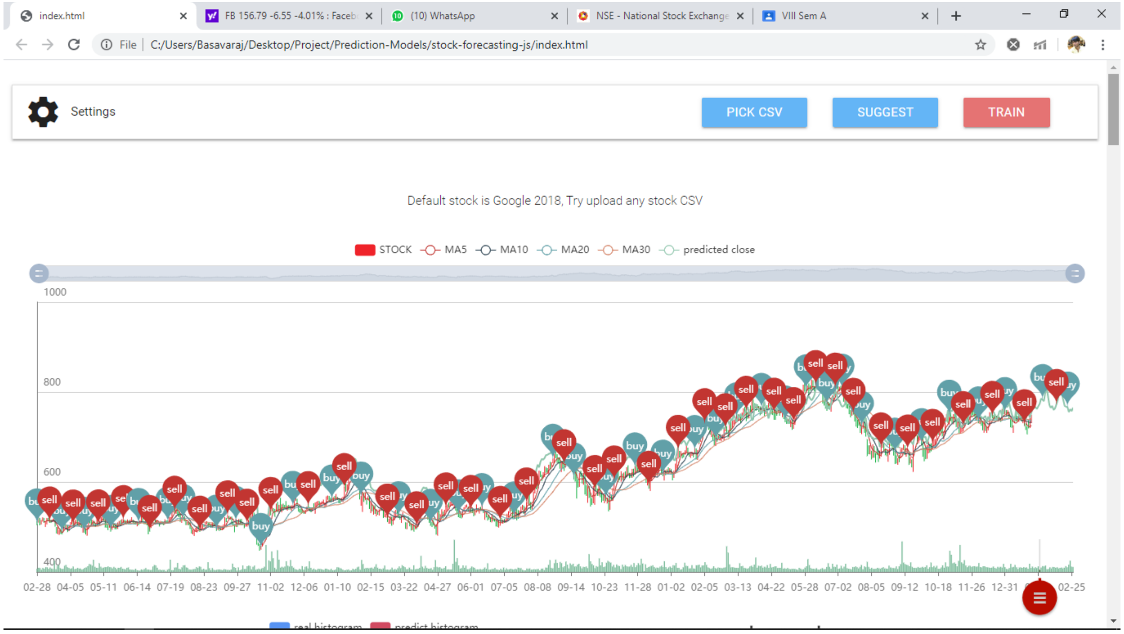

index.html
FB 156.79 -6.55 -4.01% : Faceb
(10) WhatsApp
NSE - National Stock Exchange
VIII Sem A
File
C:/Users/Basavaraj/Desktop/Project/Prediction-Models/stock-forecasting-js/index.html

2019-09-30	buy 5 units	3425	NULL	7580.999759999999
2019-10-14	sell 5 units	3417.5	-0.21897810218978103%	10998.499759999999
2019-11-01	buy 5 units	3741.4999399999997	NULL	7256.999819999999
2019-11-18	sell 5 units	3614.7500600000003	-3.387675585530007%	10871.74988
2019-12-03	buy 5 units	3667.00012	NULL	7204.749759999999
2019-12-17	sell 5 units	3728.25012	1.6703026450950866%	10932.99988
2019-12-31	buy 5 units	3770.49988	NULL	7162.5
2020-01-20	sell 5 units	3636.0000600000003	-3.5671614979603126%	10798.50006
2020-2-4	buy 5 units	3915.8553768405795	NULL	6882.64468315942
2020-2-14	sell 5 units	3861.399692628751	-1.390645950151642%	10744.044375788171
2020-2-22	buy 5 units	3832.4355249458154	NULL	6911.608850842356

Overall gain: -3088.3911491576437, Overall investment: -30.883911491576438%

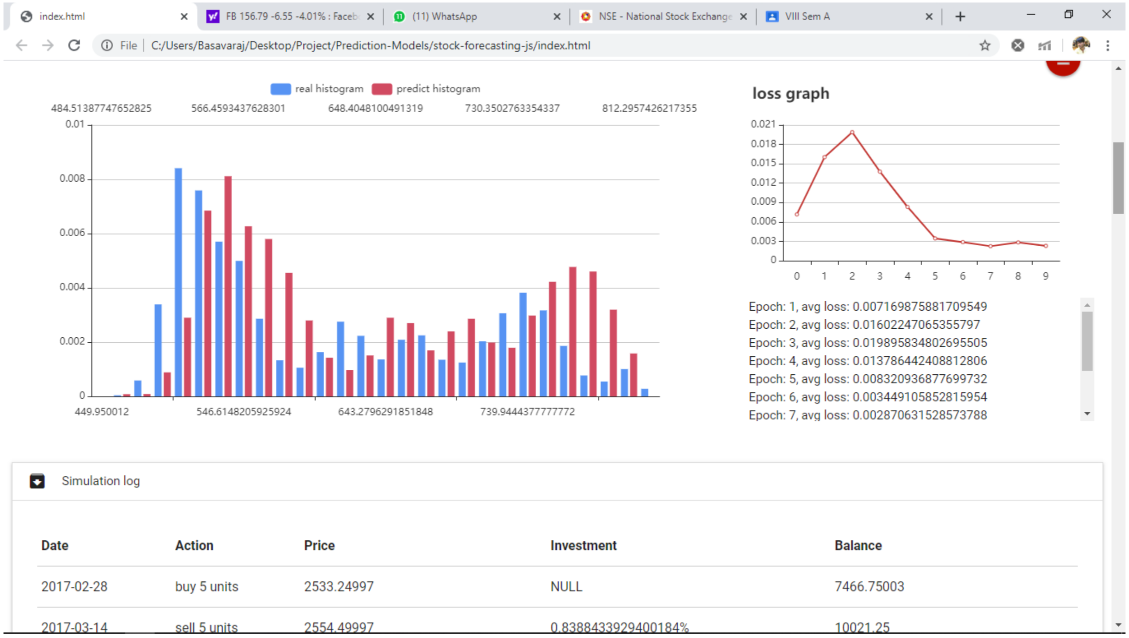
Fig. 8.1.2 Simulation Log 2

Results Using Graphs:



Default stock is Google 2018, Try upload any stock CSV

Fig. 8.2.1 Price Graph



```
Epoch: 1, avg loss: 0.007169875881709549
Epoch: 2, avg loss: 0.01602247065355797
Epoch: 3, avg loss: 0.019895834802695505
Epoch: 4, avg loss: 0.013786442408812806
Epoch: 5, avg loss: 0.008320936877699732
Epoch: 6, avg loss: 0.003449105852815954
Epoch: 7, avg loss: 0.002870631528573788
```

Date	Action	Price	Investment	Balance
2017-02-28	buy 5 units	2533.24997	NULL	7466.75003
2017-03-14	sell 5 units	2554.49997	0.8388433929400184%	10021.25

Fig. 8.2.2 Histogram of Stock Price

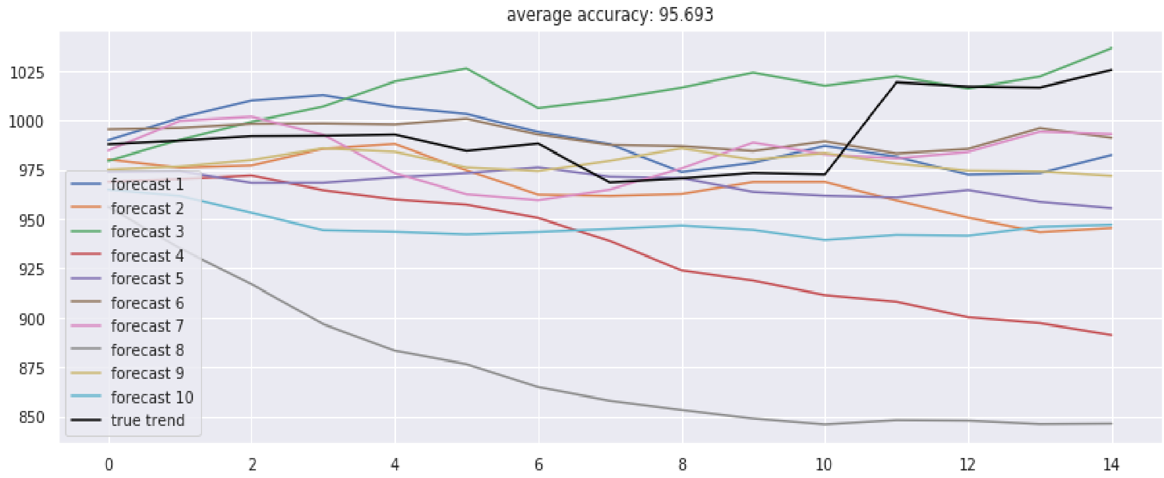


Fig. 8.2.3 Prediction Using LSTM

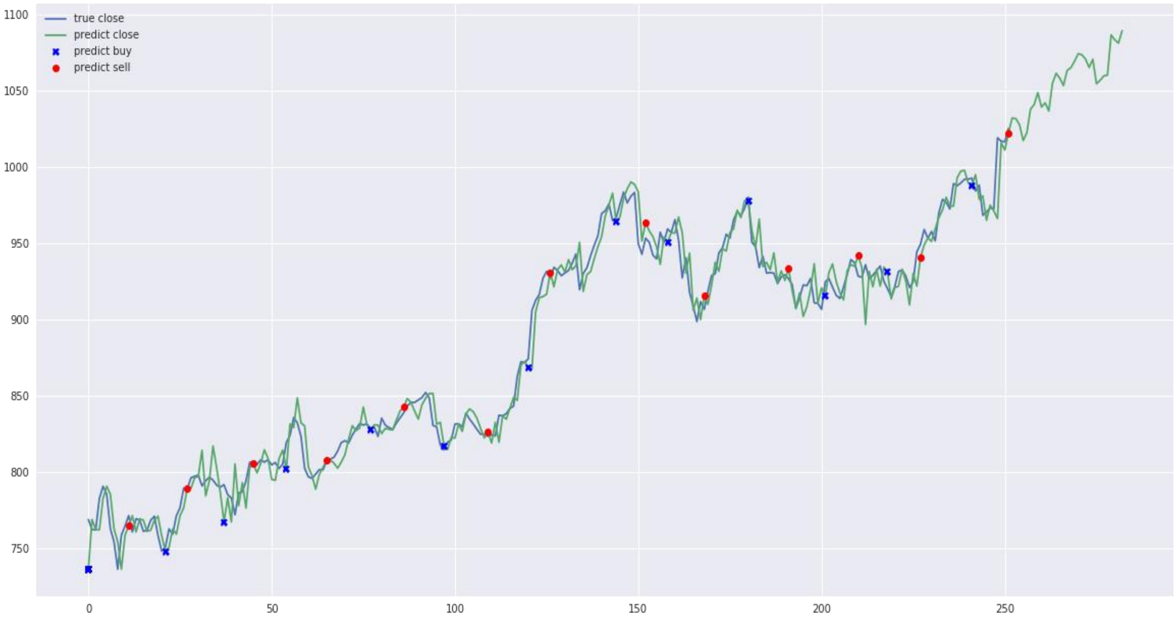


Fig. 8.2.4 Price-Prediction

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

The difficulty for a beginner to the field of Machine learning to know what type of network to use is high. There are so many types of networks to choose from and new methods being published and discussions are happening every day. Making things worse, most neural networks are flexible enough that they work and make a prediction even when used with the wrong type of data or prediction problem.

Upgrading the arrangement reports scale and time allocation can achieve better expectation. A trading model using the proposed way of thinking can be made to figure outright returns or interests dynamically. This can exhibit the precision of the model. This model can successfully endorse the stocks for investment.

Potential improvement can be made to our data collection and analysis method. Future research can be done with possible improvement such as more refined data and more accurate algorithm.

As is indent from fig 8.2.3, 8.2.4, the prediction is fairly accurate unless there is huge and sudden variation in the actual data.

Chapter 10

REFERENCES

Reference Papers:

1. Mehak Usmani, Syed Hasan Adil , Kamran Raza and Syed Saad Azhar Ali.“ Stock Market Predictio n
   Using Machine Learning Techniques”. In 2016.

2. Saxena P, Pant, B, Goudar, R.H , Srivastav, S, Garg, V., Pareek, S.” Future predictions in Indian stock market through linguistic-temporal approach.” In: 7th International Conference on Intelligent Systems and Control (ISCO) 2018

3. Paul. “Sentiment analysis in predicting the stock market trends” 2016.

4. Bohn, Tanner A. 2017 "Improving Long Term Stock Market Prediction with Text Analysis." (2017).

5. Md. Rafiul Hassan and Baikunth Nath. The University of Melbourne, Carlton 3010, Australia.
   “Stock Market Forecasting Using Hidden Markov Model” In 2005.

6. Oliveira, Nuno, Paulo Cortez, and Nelson Areal. "The impact of microblogging data for stock market prediction” In 2017.

7. Li, Xiaodong "Empirical analysis: stock market prediction via extreme learning machine. "*Neural Computing and Applications* In 2016.

8. Sorto, Max, Cheryl Aasheim, and Hayden Wimmer. "Feeling The Stock Market: A Study in the Prediction of Financial Markets Based on News Sentiment." In 2017.

9. Radu Jacomin “Stock Market Prediction”,9th International Conference on System Theory,

Control and Computing In 2015.

10. Farahmandian M, Hatamlou A“ Solving optimization problems using black hole algorithm” In 2015.

11. B. Liu, E. Blasch, Y. Chen, D.Shen, G. Chen, “Scalable Sentiment Classification for Big Data Analys is Using Naïve Bayes Classifier,” IEEE Intl Conf. on Big Data, In Oct 2013.

12. C.W. Tsai, C. H. Hsieh, and M.C. Chiang, "Parallel Black Hole Clustering Based on Map Reduce",*IE EE International Conference on Systems, Man, and Cybernetics (SMC)*, In 2015.

13. A. Navada, et al., "Overview of use of decision tree algorithms in machine learning," in Control and System Graduate Research Colloquium (ICSGRC), 2011 IEEE, In 2011.

Reference Websites

The https://towardsdatascience.com/ websites helped us understand many concepts related to machine learning that helped us to choose appropriate algorithms for the work.

Data has been collected from the following websites:

- https://www.bseindia.com/
- https://in.finance.yahoo.com/
- https://www.nasdaq.com/
- https://www.quandl.com/
- https://www.kaggle.com/