CHAPTER 1

INTRODUCTION

1.1 STOCK MARKET

Stock Market is the aggregation of buyers and sellers who buys and sell stock. A stock represent share in a company but that does not mean that you own the power to influence decision of the top management and leaders in a company. A Stock Exchange is a place where trading of stocks take place. Investors buy stocks for making profits and companies sell stocks to raise funds for finance of the company. Investors buy stock at a price and then sell it for a higher price and make profit . Smart Investors look for trends, fundamental data, finance of company, history of the company, technical charts etc. and then decide whether to invest or not. Trading strategies are divided into two categories (i) fundamental analysis (ii) technical analysis . Fundamental analysis is the study of financial statements and determining the true value of the stock and then taking decision . Various technical tools like SMA, EMA, RSI, MACD, Bollinger Bands etc. have been in use for a long time. They are used by technical analysts. This project has also made an attempt of making a smart decision based on indicators, oscillators. An attempt to predict stock price has also made. Statistical Regression is used for determining relationship between index and stock price.

1.2 SYSTEM OVERVIEW

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable .The buy or sell signal is generated by entering the valid ticker and the next day's price is also predicted.

1.3 SCOPE OF THE PROJECT

The main objective is to generate the buy or sell signals of the stocks by entering the valid Ticker of the required stock. Here the indicators and the oscillators are being much useful to calculate the signals. The goal of these indicators and the oscillators is to generate the buy and sell signals. Statistical regression is used to predict the future price of the stocks. This method is more advanced and abrupt than the fundamental and the technical analysis. The stock details will not be displayed by entering the invalid Ticker

CHAPTER 2

LITERATURE SURVEY

[1] Abidatul Izzah,Yuita Arum Sari,Ratna Widyastuti,Toga Aldila ,"Moblie app for stock prediction using Improved Multiple Linear Regression", International Conference on Sustainable Information Engineering and Technology (SIET), 2017.

The Improved Multiple Linear Regression (IMLR) has been built into a mobile application based android platform for stock price prediction. IMLR is a hybrid Multiple Linear Regression with Moving Average technique. The app has been built in several steps, which are requirement analysis, system design, implementation, and testing. Data have been collected from the finance.yahoo.com page with category "Jakarta Composite Index (^{A} JKSE)" which have automatically taken by using Yahoo Finance API.

[2] Abhinandan Gupta,Dev Kumar Chaudhary,Tanupriya Choudhury, "Stock Prediction Using Functional Link Artificial Neural Network" 3rd International Conference on Computational Intelligence and Networks (CINE), 2017.

Neural Networks once again have become famous for prediction of stock due to their ability to deal with non-linear data. The use of Artificial Neural Networks to for predicting the stock prices has been used .The input features to the model sometimes can be non-related to the output. Hence, Functional Link Artificial Neural Networks have been used here to increase the number of related features in the form of inputs. The data has been taken from NSE and converted into a suitable form for FLANN and then prediction has been carried out using Multi-layer feed forward Perceptron model.

[3] Rupesh A.Kamble, "Short and long term stock prediction using decision tree",International Conference on Intelligent Computing and Control Systems (ICICCS), 2017.

The optimization of the stock price trend prediction for short term using some oscillators and indicators like Moving Average Convergence Divergence (MACD), the Relative Strength Index (RSI), the Stochastic Oscillator (KDJ) and Bollinger Band (BB). It has been observed that using appropriate pre-processing technique and Machine learning model, possible to improve accuracy rate of short-term trend prediction. Applying Preprocessing and then using combination of data yielded a better Accuracy rate in Short term Trades, while predicting for Long-term Trend of Stock this Technical indicators were not sufficient.

[4] Mingying Wei, Shuhong Zhang, "Stock market prediction using data mining techniques", International Conference on Intelligent Sustainable Systems (ICISS), 2017.

Stock market prediction has been an area of interest for investors as well as researchers for many years due to its volatile, complex and regularly changing in nature, making it difficult to make reliable predictions The method proposes an approach towards prediction of stock market trends using machine learning models like Random Forest model and Support Vector Machine. The Random Forest model is an ensemble learning method that has been an exceedingly successful model for classification and regression. Support vector machine is a machine learning model for classification. However, this model is mostly used for classification. These techniques have been used to forecast whether the price of a stock in the future will be higher than its price on a given day, based on historical data while providing an in-depth understanding of the models being used.

[5] Si-Shu Luoo,Yang Weng,Wei Wei Wang,Wei Xing Hong, "L1- regularised logistic regession for event-driven stock market prediction", 12th International Conference on Computer Science and Education (ICCSE) , 2017.

A machine learning method for event-driven stock prediction, using L1 regularized Logistic regression model. It has been studied the stock price movement after listed companies make announcements. The model used specific events extracted from these announcements and combine with financial indicators of listed companies, macro indicators, and technical indicators as dependent variables. The listed companies have been divided into sample sets based on market value size and industry. Experiments show that this model can be a good predictor of stock within one week after events occur. In addition, compared with commonly used machine learning methods, our model has a better overall ability.

[6] David M.Q.Nelson,Adriana C.M.Pereira,Renato A deOliveira, "Stock market's movement prediction with LSTM neural networks", International Joint Conference on Neural Networks (IJCNN), 2017.

The predict future trends of stock prices based on the price history, alongside with technical analysis indicators. For that goal, a prediction model was built, and a series of experiments have been executed and theirs results analyzed against a number of metrics to assess if this type of algorithm presents and improvements when compared to other Machine Learning methods and investment strategies. The results have been obtained promising, getting up to an average of 55.9% of accuracy when predicting if the price of a particular stock is going to go up or not in the near future.

[7] Naina Jamal,Akshay Malhotra, "Identification of persisting trend in the Indian stock markets using hurst exponent", IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016.

Efficient market hypothesis states that the current share prices reflect all relevant information about the underlying company. According to this widely accepted hypothesis, it is impossible for investors to consistently outperform the indices in the long run. Furthermore, the Random Walk Hypothesis, which is consistent with the Efficient Market Hypothesis, states that stock prices evolve in a random manner and price changes are independent of each other.The results showed that Indian stock markets do not conform to EMH, especially on consideration of larger time scales

[8] Xi Chen,Zhi-Jie He, "Prediction of Stock Trading Signal Based on Support Vector Machine", 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), 2015.

The usage of SVM to construct a prediction model to find the stock trading signal. In addition, Piecewise linear representation (PLR) has been good at extracting valuable information from a time sequence. PLR has been used for checking of turning points in this study. The experiments on some real stocks show that SVM obtains a better result in prediction accuracy and profitability than traditional Back Propagation neural network does.

[9] Elena Bautu,Sun Kim,Andrei Bautu, "Evolving hypernetwork models of binary time series for forecasting price movements on stock markets", IEEE Congress on Evolutionary Computation , 2009.

A hypernetwork-based method for stock market prediction through a binary time series problem. Hypernetworks were a random hypergraph structure of higher-order probabilistic relations of data. The problem they tackled concerned the prediction of price movements (up/down) on stock markets. Compared to previous approaches, the method discovers a large population of variable subpatterns, i.e. local and global patterns, using a novel evolutionary hypernetwork. An output was obtained from combining these patterns. Here described two methods for assessing the prediction quality of the hypernetwork approach.

[10] R.S.T.Lee, "iJADE stock advisor:an intelligent agent based stock prediction system using hybrid RBF recurrent network", IEEE Transactions on Systems, Man, and Cybernetics¸, 2004.

The iJADE Stock Advisor-an intelligent agent-based stock prediction system using our proposed hybrid radial basis-function recurrent network (HRBFN). By using ten-year stock pricing information (1990-1999), consisting of 33 major Hong Kong stocks for testing, the iJADE Stock Advisor has achieved promising results in terms of efficiency, accuracy, and mobility as compared with other contemporary stock prediction models. Also, various analyzes on this stock advisory system have been performed: including round trip time (RTT) analysis, window-size evaluation test (for both long-term trend and short-term prediction), and stock prediction performance test.

CHAPTER 3

SYSTEM ANALYSIS

3.1EXIXTING SYSTEM

The fundamental analysis involves the in-depth analysis of a company’s performance and the profitability to measures it’s intrinsic value by studying the company physically in terms of its product sales, man power quality, infrastructure, profitability on investment. It uses revenues, earnings, future growth, return on equity, profit margins, and other data to determine a company's underlying value and potential for future growth. To a fundamentalist, the market price of a stock tends to move towards its “real value” or “intrinsic value”. If this value of a stock is above the current market price, the investor can decide to purchase the stock because the stock pricewill bound to rise and move towards its “intrinsic or real value”. If this value of a stock is below the market price, the investor may decide to sell the stock because the stock price is bound to fall and come closer to its intrinsic value. To start finding out the intrinsic value, the fundamentalist analyzer makes an examination of the current and future overall health of the economy as a whole.

Technical analysis is a method of evaluating stocks by analyzing statistics generated by market activity, past prices, and volume. It looks for peaks, bottoms, trends, patterns, and other factors affecting a stock's price movement.Future values of stock prices often depend on their past values and the past values of other correlated variables. Technical analysis looks for patterns and indicators on stock charts that will determine a stocks future performance .However, it is used by approximately 90% of the major stock traders. Despite its widespread use, technical analysis is criticized because it is highly subjective. Different individuals can interpret charts in different manners.Recently, neural networks have been successfully applied in time-series problems to improve multivariate prediction ability. Neural networks have good generalization capabilities by mapping input values and output values of given patterns. Neural networks are usually robust against noisy or missing data, all of which are highly desirable properties in time series prediction problems. Various neural network models have already been developed for the stock market analysis.

3.1.1 DISADVANTAGES OF THE EXISTING SYSTEM

- It becomes harder to formalize all this knowledge for purposes of automation (with a neural network for example), and interpretation of this knowledge may be subjective.
- It is hard to time the market using fundamental analysis.
- Despite its widespread use, technical analysis is criticized because it is highly subjective.
- Different individuals can interpret charts in different manners.

3.2 PROPOSED SYSTEM

- **Data Collection**

Quandl API is used to fetch the stock quote of any company under NSE. Stock Quote consist of Close Price, High, Low, Volume, Open etc. as a data-frame. A data-frame is a labelled two dimensionnal data structure that stores data of different types in different columns. A data-frame consists of columns and a single column represent a series.Series of Close price is extracted and is used as data source for every indicator, oscillator and neural network. Date is the index of the series and is also used as data for indicators and oscillators. This series represents the past closing prices of a company under NSE.

- **Implementation of Indicators and Oscillators**

The following are implemented:

SMA

EMA

MACD

RSI

BollingerBand

Statiscal Regression(Not an indicator/oscillator)

- **SMA (Simple Moving Average)**

It is a type of moving average in which mean is calculated over the price of n-periods. Using it, we get a series of averages of different subsets of full dataset.

$$SMA = (p_m + p_{m-1} + p_{m-2} + \ldots\ldots\ldots\ldots.. + p_{m-(n-1)})/n$$

n = 20 but it can have any value.

Where numerator is the summation of previous n-periods prices and n is the length of period. If the data used are not centered around the mean, a simple moving average lags behind the latest datum point by half the sample width. An SMA can also be disproportionately influenced by old datum points dropping out or new data coming in. One characteristic of the SMA is that if the data have a periodic fluctuation, then applying an SMA of that period will eliminate that variation. But a perfectly regular cycle is rarely encountered which is shown in Figure 3.1.

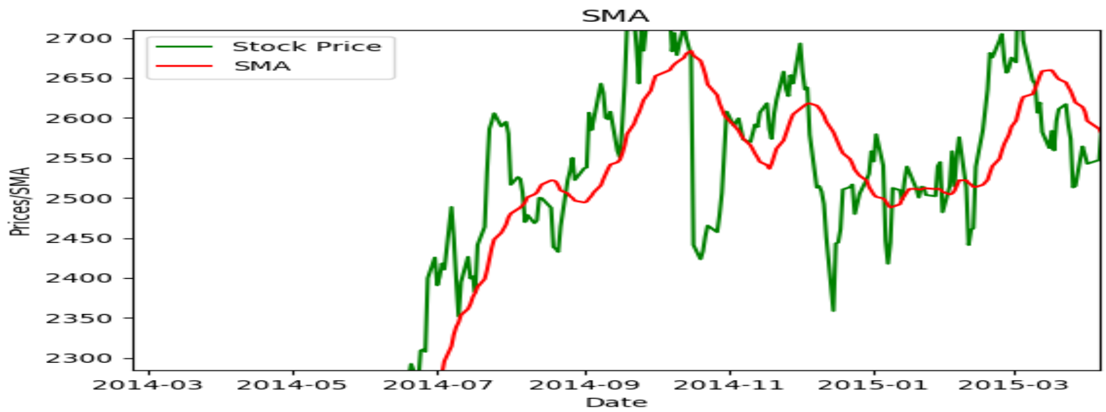


Figure 3.1 SMA Curve

- **EMA (Exponential Moving Average)**

EMA or EWMA is a type of moving average that is similar to a simple moving average, except that more weight is given to the latest data. It's also known as the exponentially weighted moving average. This type of moving average reacts faster to recent price changes than a simple moving average is shown in Figure 3.2.

Calculating EMA is a 3 step process:

Calculate the SMA

Calculate the multiplier

Calculate the current EMA

Multiplier = 2 / (n+1),

EMA = (Closing Price – EMA (previous day))*multiplier + EMA (previous day))

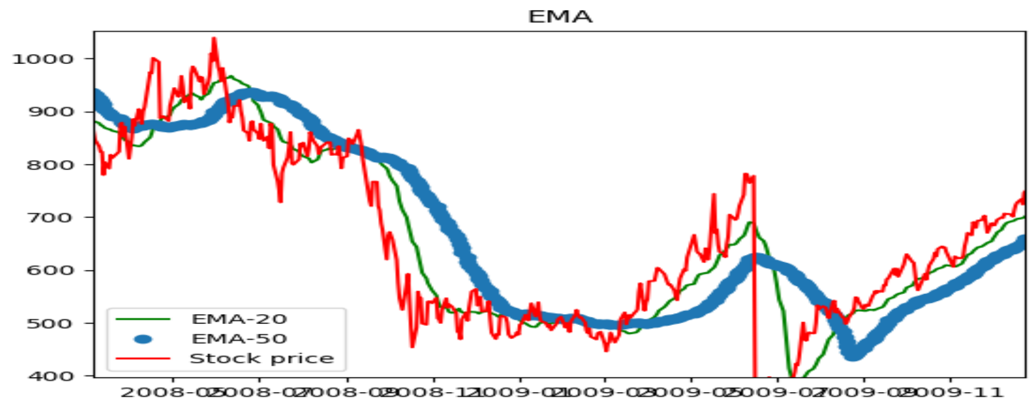


Figure 3.2 EMA Curve

- **MACD (Moving Average Convergence Divergence)**

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of prices. The MACD is calculated by subtracting the 26-day EMA from the 12-day EMA. In Figure 3.3 a nine-day EMA of the MACD, called the "signal line", is then plotted on top of the MACD, functioning as a trigger for buy and sell signals.

MACD = ((12-day EMA) – (26-day EMA))

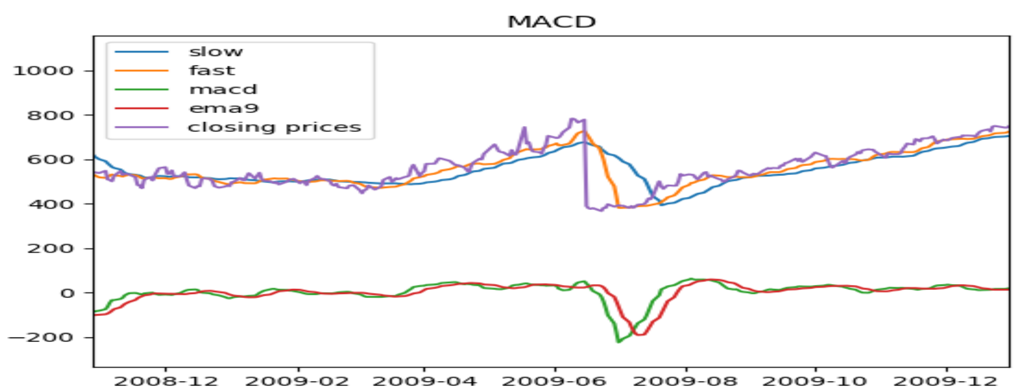


Figure 3.3 MACD Curve

- **RSI (Relative Strength Index)**

The relative strength index (RSI) is a momentum indicator that compares the magnitude of recent gains and losses over a specified time period to measure speed and change of price movements of a security. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset.

RSI = 100 - 100 / (1 + RS)

Where RS = Average gain of up periods during the specified time frame / Average loss of down periods during the specified time frame

RSI values range from 0 to 100. The default time frame for comparing up periods to down periods is 14, as in 14 trading days. Traditional interpretation and usage of the RSI is that RSI values of 70 or above indicate that a security is becoming overbought or overvalued, and therefore may be primed for a trend reversal or corrective pullback in price as shown in Figure 3.4. On the other side of RSI values, an RSI reading of 30 or below is commonly interpreted as indicating an oversold or undervalued condition that may signal a trend change or corrective price reversal to the upside.

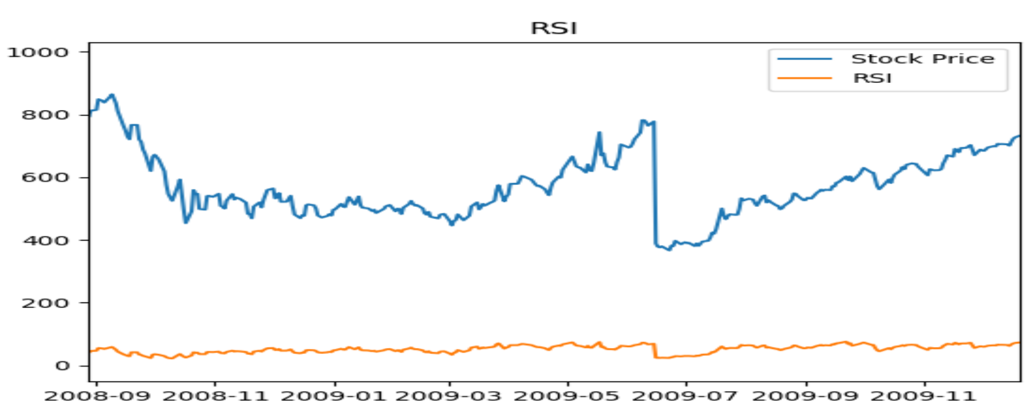


Figure 3.4 RSI Curve

- **Bollinger Band**

A Bollinger Band, developed by famous technical trader John Bollinger, is plotted two standard deviations away from a simple moving average. Many traders believe the closer the prices move to the upper band, the more overbought the market, and the closer the prices move to the lower band, the more oversold the market as shown in Figure 3.5. John Bollinger has a set of 22 rules to follow when using the bands as a trading system.

Upper-Band= SMA-20 + 2 * standard deviation of 20 day closing

Lower-Band= SMA-20 + 2 * standard deviation of 20 day closing

%B = ((price – lower band) / (upper band – lower band))*100

%B quantifies a security's price relative to the upper and lower Bollinger Band. There are six basic relationship levels:

- %B equals 1 when price is at the upper band
- %B equals 0 when price is at the lower band
- %B is above 1 when price is above the upper band
- %B is below 0 when price is below the lower band

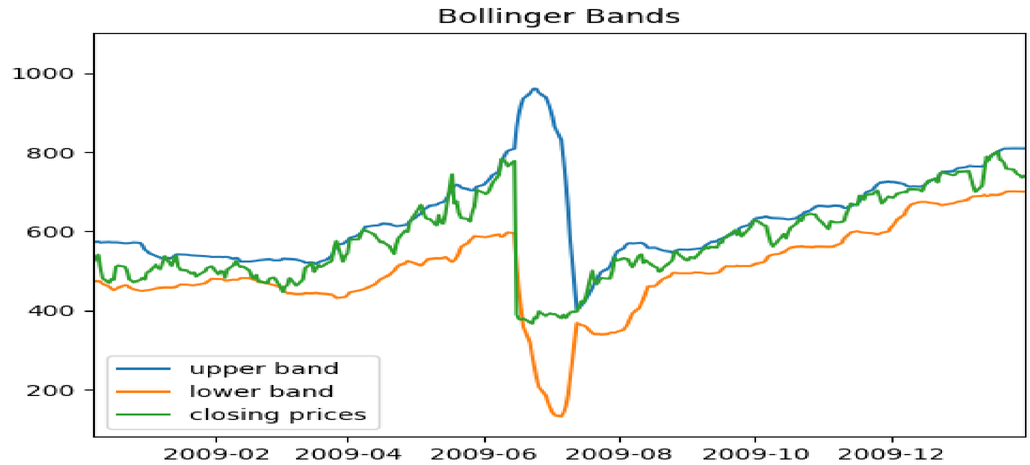


Figure 3.5 Bollinger Band

- **Statistical Regression**

It is a way of determining a relation between variables. There are 2 independent variables here.

Index

Stock Price

y = mx +c → trend line equation

where y = closing price , x = nth number, m =slope

m = (sum(x*y)-sum(y)*sum(x))/sum((x-mean(x))^2)

c = y – mx

- **Determining Buy/Sell Signals**

If previous SMA is lower than the previous Close Price and current SMA is greater than current Close Price then Buy Signal is generated. The same way when previous SMA is greater than the previous Close Price and current SMA is lower than current Close Price then Sell Signal is generated.If preceding EMA-20 is lower than the previous EMA-50 and current EMA-20 is greater than current EMA-50 then Buy Signal is generated. Similarly, if former EMA-20 is greater than the previous EMA-50 and usual EMA-20 is lower than current EMA-50 then Sell Signal is generated. If previous MACD value is less than previous EMA-9 and current MACD value is greater than current EMA9 then Buy Signal is generated. If previous MACD value is greater than previous EMA-9 and current MACD value is lower than current EMA-9 then Sell Signal is generated. If the difference between Close Price and lower band is greater than the difference between Upper Band and Close Price and both the differences are greater than 0 then buy Signal is generated. If the difference between the Upper band and Close Price is less than the difference between Close

Price and Lower Band]and both the differences are greater than 0 then Sell signal is generated

- **Predicting Stock Price**

RBF Neural Network is used for predicting stock price for the next day.
Previous 30 day stock price is used as training data.
For ex., for index n, output is X. where n can be anything between 1 – 30
For index 31^{st} , stock price is predicted which is the next day stock price
Sklearn Library is used for implementing RBFNN.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

- With the help of the indicators and oscillators the buy or sell signal is generated.
- The overbought and oversold stocks could be determined.
- The next day's closing price is predicted using the trend line equation .

3.3 REQUIREMENTS SPECIFICATION

3.3.1 Hardware Requirements

Processor	:	Pentium Dual Core 2.3 GHz
Hard Disk	:	250 GB or Higher
Ram	:	2 GB

3.3.2 Software Requirements

Operating System	:	Linux
Languages used	:	Python
Tools	:	Eclipse

3.4 LANGUAGE SPECIFICATION

3.4.1 PYTHON

Overview

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 type system and automatic memory management. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open sourcesoftware and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

3.4.2 ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP,Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge.The Eclipse software development kit (SDK). Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGi implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundlesEclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License. It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

CHAPTER 4

4.1 SYSTEM ARCHITECTURE

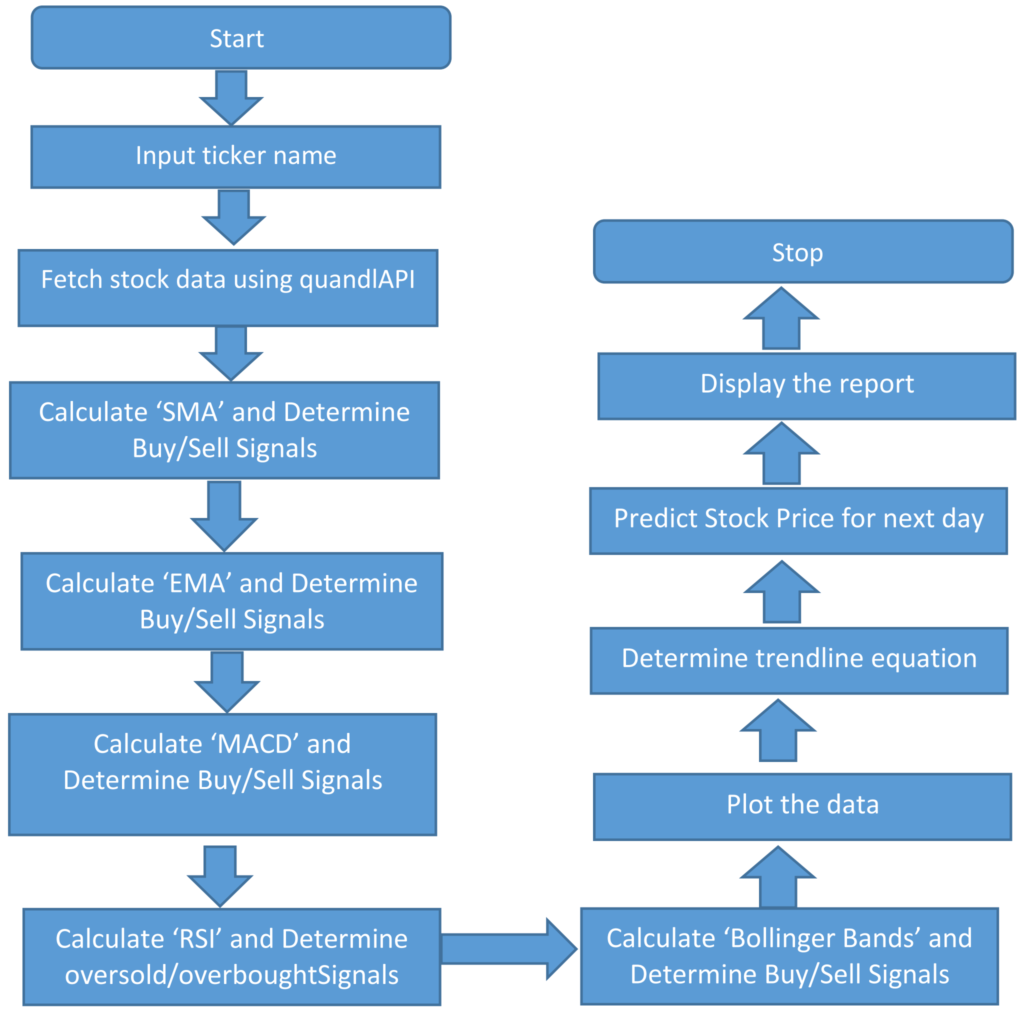


Figure 4.1 Architecture of proposed system

The ticker is given by the user as an input for the stock of interest, the entire data about the ticker will be retrieved . SMA, EMA, MACD, RSI, Bollinger Band, Statistical Linear Regression for trend-line are implemented .The Buy/Sell signal are generated using the above calculated indicators and oscillators.The next day's price is predicted

4.2 USE CASE DIAGRAM

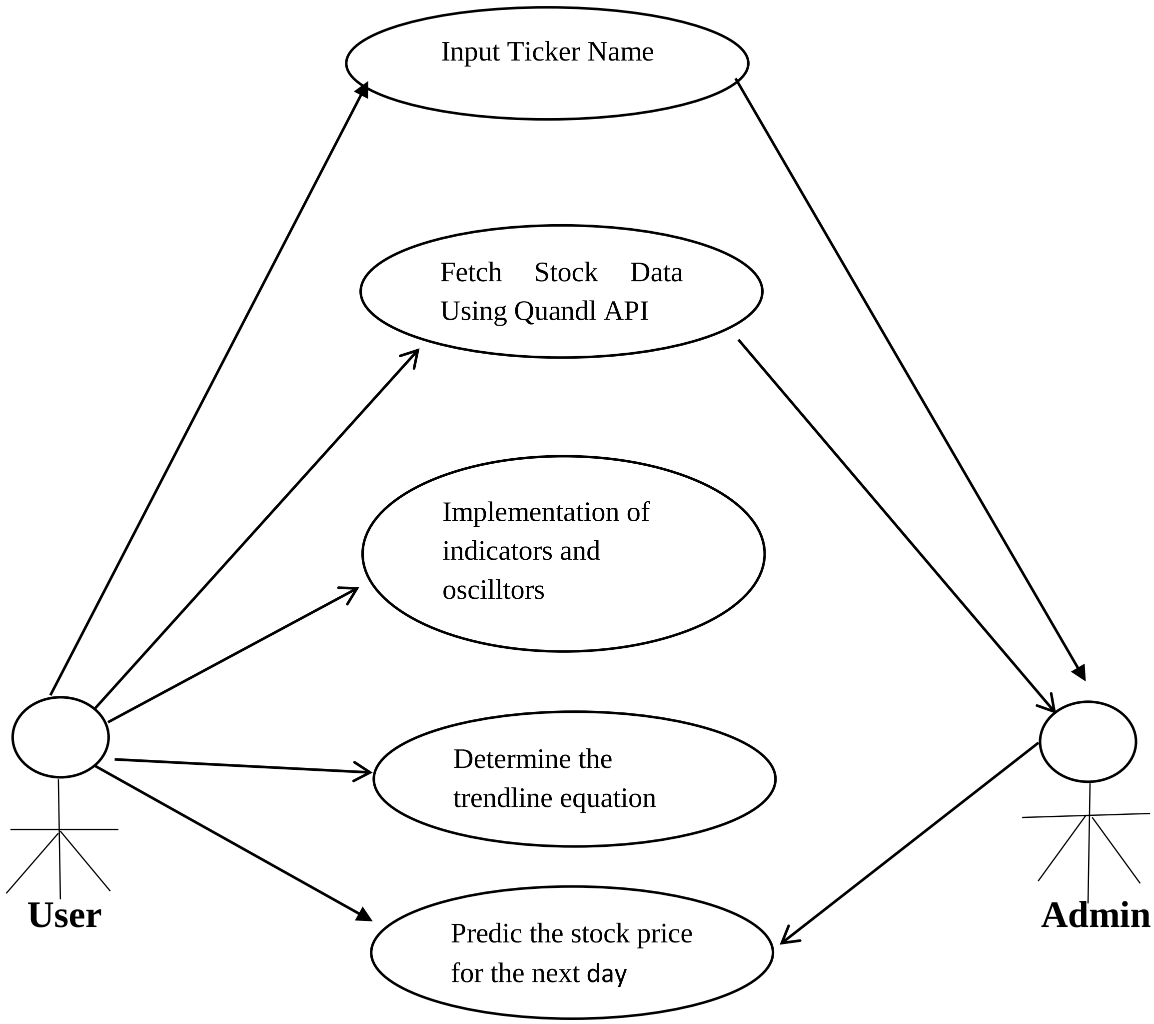


Figure 4.2 Use case diagram

A Valid Ticker is entered that would fetch the required details of the required stock. The indicators and oscillators are implemented and the buy/sell signal would be generated. The next day stock price is also predicted using the trend line equation.

CHAPTER 5

5.1 MODULES

The modules are

- Data collection
- Calculation of indicators and oscillators
- Determining buy/sell signal
- Prediction of stock details

5.1.1 DATA COLLECTION

When the ticker is given by the user as an input for the stock of interest, the entire data about the ticker will be retrieved. Quandl API is used to fetch the past stock data in JSON format. For the ticker provided, it returns Closing Price, Volume, Open, High, Low etc. as a data-frame. Close Price series is formed which is used as the data for further process.

5.1.2 CALCULATION OF INDICATORS AND OSCILLATORS

SMA, EMA, MACD, RSI, Bollinger Band, Statistical Linear Regression for trend-line are implemented. SMA is a type of moving average in which mean is calculated over the price of n-periods. Using SMA, we get a series of averages of different subsets of full dataset. EMA or EWMA is a type of moving average that is similar to a simple moving average, except that more weight is given to the latest data. MACD is a trend-following momentum indicator that shows the relationship between two moving averages of prices. RSI is a momentum indicator that compares the magnitude of recent gains and losses over a specified time period to measure speed and change of price movements of a security. A Bollinger Band, _is plotted two standard deviations away from a simple moving average. Statistical Regression is a way of determining a relation between variables. There are 2 independent variables that includes Index and Stock Price.

5.1.3 DETERMINING BUY/SELL SIGNALS

If previous SMA is lower than the previous Close Price and current SMA is greater than current Close Price then Buy Signal is generated. The same way when previous SMA is greater than the previous Close Price and current SMA is lower than current Close Price then Sell Signal is generated.If preceding EMA-20 is lower than the previous EMA-50 and current EMA-20 is greater than current EMA-50 then Buy Signal is generated. If previous MACD value is less than previous EMA-9 and current MACD value is greater than current EMA9 then Buy Signal is generated. If previous MACD value is greater than previous EMA-9 and current MACD value is lower than current EMA-9 then Sell Signal is generated. If the difference between Close Price and lower band is greater than the difference between Upper Band and Close Price and both the differences are greater than 0 then buy Signal is generated. If the difference between the Upper band and Close Price is less than the difference between Close Price and Lower Band]and both the differences are greater than 0 then Sell signal is generated.

5.1.4 PREDICTION OF STOCK DETAILS

RBF Neural Network is used for predicting stock price for the next day. Previous 30-day stock price is used as training data. For ex., for index n, output is X. where n can be anything between 1 – 30. For index 31^{st} , stock price is predicted which is the next day stock price. Sklearn Library is used for implementing RBFN.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

- Technical Analysis of stock market is a method of evaluating securities by analyzing statistics generated by market activity. Chartist uses these data to determine whether they are going to invest in that stock. They look for the buy and sell signals. This project has also performed technical analysis over stock prices by using some of the mostly used indicators and oscillators. We have also tried to predict stock price for the next day.
- Stock price is influenced by many factors including –internal information, news, speeches of ministers etc. So this does not give accurate results all the time. But it is better than blindly investing
- Hence, it can be concluded that a smart investing decision can be made if one uses technical indicators and oscillators and other sorts of information like news etc. before making an investment

6.2 FUTURE ENHANCEMENT

- More Indicators and oscillators will be added.
- Sentiment Analysis will be performed on the data fetched from twitter.
- Buy/Sell Signals will be generated according to the sentiments with combined results of indicators and oscillators.

APPENDIX 1

SAMPLE CODE

```
import quandl
import requests
import matplotlib.pyplot as plt
import pandas
# from pandas import DataFrame
# import datetime
import numpy as np
# import matplotlib.patches as mpatches
# from math import sqrt
from sklearn.svm import SVR
quandl.ApiConfig.api_key = 'Hqu7HLNnBU4bxBU4jLaZ'
def regression(price):
    price = price.tolist()
    lis = list()
    lis2 = list()
    x = len(price)
    a = x - 20
    X = cal_X_mean(x)
    Y = cal_Y_mean(price, x)
    # y-Y
    for i in range(0, 20):
        lis.append(price[a] - Y)
        a = a + 1
    # x-X
    a = 0
    for i in range(x - 20, x):
        lis2.append(i - X)
        a = a + 1
    lis3 = []
    for i in range(0, 20):
        lis3.append(lis[i] * lis2[i])
    lis4 = []
    for i in range(0, 20):
        lis4.append(lis2[i] * lis2[i])
    sum1 = 0
    sum2 = 0
```

```
    for i in range(0, 20):
        sum1 = sum1 + lis3[i]
        sum2 = sum2 + lis4[i]
    m = sum1 / sum2
    m = float('{0:.2f}'.format(m))
    b = Y - m * X
    b = float('{0:.2f}'.format(b))
    print('TrendLine equation is : y = ', m, '* x +(', b, ')')
    return 0
def cal_X_mean(x):
    sum = 0
    for i in range(x - 20, x):
        sum = sum + i
    return sum / 20
def cal_Y_mean(close, x):
    sum = 0
    for i in range(x - 20, x):
        sum = sum + close[i]
    return sum / 20
def plotMACD(
    close,
    ema9,
    emaslow,
    emafast,
    macd,
    date,
):
def drawChartSMA(close, date, sma):
    y = close
    x = date
    plt.plot(x, y, 'g', label='Stock Price')
    plt.plot(x, sma, 'r', label='SMA')
    plt.xlabel('Date')
    plt.ylabel('Prices/SMA')
    plt.title('SMA')
    plt.legend()
    l = len(close)
    k = l - 20
    min = close[k]
    max = close[k]
```

```
    for t in range(k, l):
        if close[t] < min:
            min = close[t]
        if close[t] > max:
            max = close[t]
    if sma[len(sma) - 1] < close[len(close) - 1]:
        print('SMA says Uptrend')
        print('Support is Rs.', min)
        print('Resistance is Rs.', max)
    elif sma[len(sma) - 1] > close[len(close) - 1]:
        print('SMA says Downtrend')
        print('Support is Rs.', min)
        print('Resistance is Rs.', max)
    else:
        print('Trend Reversal may occur')
        print('Support is Rs.', min)
        print('Resistance is Rs.', max)
    if sma[len(sma) - 2] >= close[len(close) - 2] and sma[len(sma) - 1] \
            < close[len(close) - 1]:
        print('SMA:Buy Signal Is Generated.')
    elif sma[len(sma) - 2] <= close[len(close) - 2] and sma[len(sma)
                                                           - 1] > close[len(close) - 1]:
        print('SMA:Sell Signal is generated.')
    else:
        print('SMA:No Buy/Sell Signal Generated.')

    plt.show()
def movingaverage(values, window):
    weights = np.repeat(1., window) / window
    smas = np.convolve(values, weights, 'valid').tolist()
    listt = list()
    for c in range(0, window - 1):
        listt.append(0)
    smas = listt + smas
    for m in range(0, len(smas)):
        smas[m] = float('{0:.2f}'.format(smas[m]))
    return smas
def ExpMovingAverage(values, window):
    weights = np.exp(np.linspace(-1., 0., window))
    weights /= weights.sum()
```

```
    a = np.convolve(values, weights)[:len(values)]
    a[:window] = a[window]
    return a
def cal_X_mean(x):
    sum = 0
    for i in range(x - 20, x):
        sum = sum + i
    return sum / 20
def cal_Y_mean(close, x):
    sum = 0
    for i in range(x - 20, x):
        sum = sum + close[i]
    return sum / 20
def computeMACD(x, slow=26, fast=12):
    emaslow = ExpMovingAverage(x, slow)
    emafast = ExpMovingAverage(x, fast)
    return (emaslow, emafast, emafast - emaslow)
def rsiFunc(close, n, date):
    for i in range(0, 20):
        lis.append(price[a] - Y)
        a = a + 1
    # x-X
    a = 0
    for i in range(x - 20, x):
        lis2.append(i - X)
        a = a + 1
    lis3 = []
    for i in range(0, 20):
        lis3.append(lis[i] * lis2[i])
    lis4 = []
    for i in range(0, 20):
        lis4.append(lis2[i] * lis2[i])

    deltas = np.diff(close)
    seed = deltas[:n + 1]
    up = seed[seed >= 0].sum() / n
    down = -seed[seed < 0].sum() / n
    rs = up / down
    rsi = np.zeros_like(close)
    rsi[:n] = 100. - 100. / (1 + rs)
```

```
    for i in range(n, len(close)):
        delta = deltas[i - 1]
        if delta > 0:
            upval = delta
            downval = 0.
        else:
            upval = 0.
            downval = -delta
        up = (up * (n - 1) + upval) / n
        down = (down * (n - 1) + downval) / n
        rs = up / down
        rsi[i] = 100. - 100. / (1. + rs)
        rsi[i] = float('{0:.2f}'.format(rsi[i]))
    plt.plot(date, close.tolist(), label='Stock Price')
    plt.plot(date, rsi.tolist(), label='RSI')
    plt.legend()
    plt.title('RSI')
    plt.show()
    return rsi
def bollinger_band(close, window):
    rolling_mean = close.rolling(window=20).mean()
    rolling_std = close.rolling(window=20).std()
    upper_band = rolling_mean + rolling_std * 2
    lower_band = rolling_mean - rolling_std * 2
    cl = close.tolist()
    lo = lower_band.tolist()
    up = upper_band.tolist()
    b = (cl[len(cl) - 1] - lo[len(lo) - 1]) / (up[len(up) - 1]
                                              - lo[len(lo) - 1])
    print('%B is:', float('{0:.2f}'.format(b * 100)), '%')
    q = cl[len(cl) - 1] - lo[len(lo) - 1]
    p = up[len(up) - 1] - cl[len(cl) - 1]
    if p > q and cl[len(cl) - 1] > lo[len(lo) - 1]:
        print('Bollinger Band: Buy Signal Generated')
    elif p < q and cl[len(cl) - 1] < up[len(up) - 1]:
        print('Bollinger Band: Sell Signal Generated')

    return (rolling_mean, upper_band, lower_band)
def regression(price):
    price = price.tolist()
```

```
    lis = list()
    lis2 = list()
    x = len(price)
    a = x - 20
    X = cal_X_mean(x)
    Y = cal_Y_mean(price, x)
    # y-Y
    for i in range(0, 20):
        lis.append(price[a] - Y)
        a = a + 1
    # x-X
    a = 0
    for i in range(x - 20, x):
        lis2.append(i - X)
        a = a + 1
    lis3 = []
    for i in range(0, 20):
        lis3.append(lis[i] * lis2[i])
    lis4 = []
    for i in range(0, 20):
        lis4.append(lis2[i] * lis2[i])
    sum1 = 0
    sum2 = 0
    for i in range(0, 20):
        sum1 = sum1 + lis3[i]
        sum2 = sum2 + lis4[i]
    m = sum1 / sum2
    m = float('{0:.2f}'.format(m))
    b = Y - m * X
    b = float('{0:.2f}'.format(b))
    print('TrendLine equation is : y = ', m, '* x +(', b, ')')
    return 0
def cal_X_mean(x):
    sum = 0
    for i in range(x - 20, x):
        sum = sum + i
    return sum / 20
def cal_Y_mean(close, x):
    sum = 0
    for i in range(x - 20, x):
```

```
        sum = sum + close[i]
    return sum / 20
def plotMACD(
    close,
    ema9,
    emaslow,
    emafast,
    macd,
    date,
):
    plt.plot(date, emaslow.tolist(), label='slow')
    plt.plot(date, emafast.tolist(), label='fast')
    plt.plot(date, macd.tolist(), label='macd')
    plt.plot(date, ema9.tolist(), label='ema9')
    plt.plot(date, close, label='closing prices')
    plt.title('MACD')
    plt.legend()
    ema9 = ema9.tolist()
    macd = macd.tolist()
    if ema9[len(ema9) - 1] < macd[len(macd) - 1] and ema9[len(ema9)
                                                    - 2] > macd[len(macd) - 2]:
        print('MACD:Buy Signal Generated')
    elif ema9[len(ema9) - 1] > macd[len(macd) - 1] and ema9[len(ema9)
                                                      - 2] < macd[len(macd) - 2]:
        print('MACD:Sell Signal Generated')
    else:
        print('MACD: No Buy/Sell Signal Generated')
    plt.show()
def plotB_band(
    b,
    c,
    date,
    close,
):
    plt.title('Bollinger Band')
    plt.plot(date, b, label='upper band')
    plt.plot(date, c, label='lower band')
    plt.plot(date, close, label='closing prices')
    plt.title('Bollinger Bands')
    plt.legend()
```

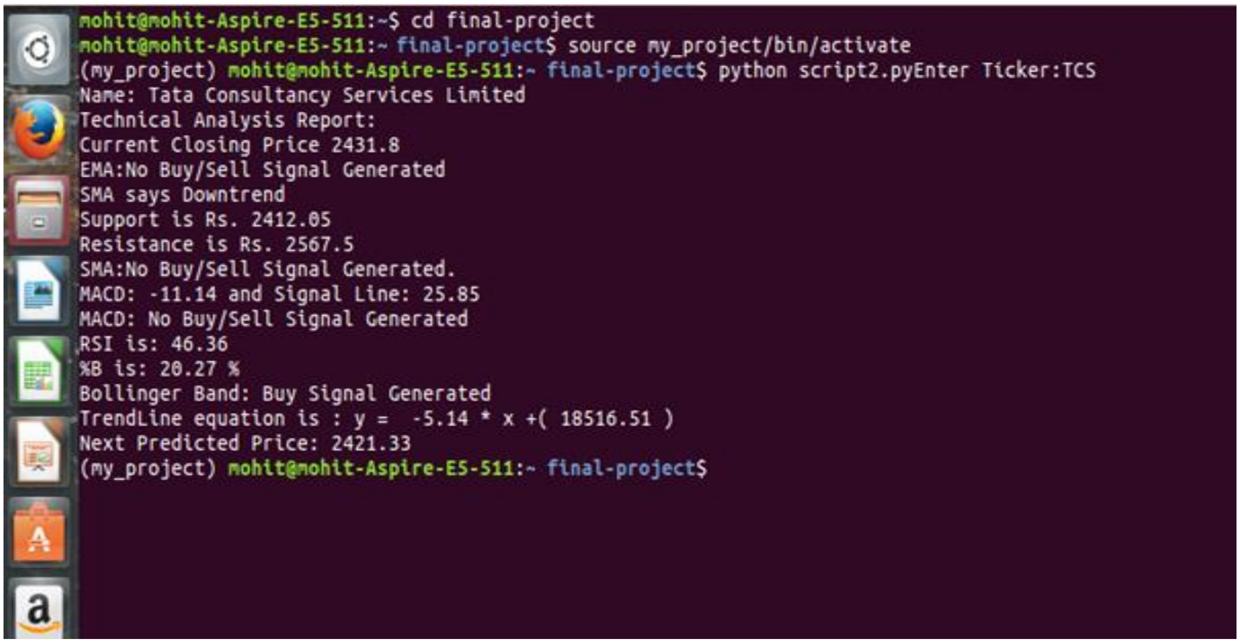
```
    plt.show()
def plotEMA(
    ema50,
    ema20,
    date,
    close,
):
    plt.plot(date.tolist(), ema20, 'g', label='EMA-20')
    plt.plot(date.tolist(), ema50, 'o', label='EMA-50')
    plt.plot(date.tolist(), close, 'r', label='Stock price')
    plt.legend()
    plt.title('EMA')
    plt.show()
def predict_price(date, close, x):
    close = close.tolist()
    x = 31
    y = len(close) - x
    closed = []
    date[0] = 0
    for i in range(0, 30):
        if y < len(close):
            closed.append(close[y])
            y = y + 1
    date = np.array(date).reshape((len(date), 1))
    svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)
    svr_rbf.fit(date, closed)
    # plt.scatter(date,closed,color='black',label='data')
    # plt.plot(date,svr_rbf.predict(date),color='red',label='RBF Model')
    # plt.plot(date,svr_lin.predict(date),color='green',label='Linear model')
    # plt.plot(date,svr_poly.predict(date),color='blue',label='Polynomial model')
    # plt.xlabel('Date')
    # plt.ylabel('Price')
    # plt.title('Support Vector Regression')
    # plt.legend()
    # plt.show()
    return svr_rbf.predict(x)[0]
try:
    name = input('Enter Ticker:')
    url = 'https://www.quandl.com/api/v3/datasets/NSE/' + name \
        + '/metadata.json'
```

```
try:
    meta_data = requests.get(url)
    parsed_meta_data = meta_data.json()
    print('Name: {}'.format(parsed_meta_data['dataset']['name']))
    print('Technical Analysis Report:',
          ''.format(parsed_meta_data['dataset']['name']))
    searched_data = quandl.get('NSE/' + name)
    close = searched_data.Close
    n = 20
    k = close.tolist()
    print('Current Closing Price', k[len(k) - 1])
    sma = movingaverage(close.tolist(), n)
    ema50 = ExpMovingAverage(close, 50)
    ema20 = ExpMovingAverage(close, 20)
    ema50 = ema50.tolist()
    ema20 = ema20.tolist()
    for x in range(0, len(ema50)):
        ema50[x] = float('{0:.2f}'.format(ema50[x]))
        ema20[x] = float('{0:.2f}'.format(ema20[x]))
    l = close.tolist()
    if ema20[len(ema20) - 2] < ema50[len(ema50) - 2] \
            and ema20[len(ema20) - 1] > ema50[len(ema50) - 1]:
        print('EMA:Buy Signal Generated.')
    elif ema20[len(ema20) - 2] > ema50[len(ema50) - 2] \
            and ema20[len(ema20) - 1] < ema50[len(ema50) - 1]:
        print('EMA:Sell Signal Generated')
    else:
        print('EMA:No Buy/Sell Signal Generated')
    date = pandas.to_datetime(searched_data.index)
    drawChartSMA(close.tolist(), date.tolist(), sma)
    plotEMA(ema50, ema20, date, close)
    nema = 9
    (emaslow, emafast, macd) = computeMACD(close)
    ema9 = ExpMovingAverage(macd, nema)
    print('MACD:', float('{0:.2f}'.format(macd[len(macd) - 1])),
          'and Signal Line:',
          float('{0:.2f}'.format(ema9[len(ema9) - 1])))
    plotMACD(
        close,
        ema9,
```

```
            emaslow,
            emafast,
            macd,
            date.tolist(),
        )
        rsi = rsiFunc(close, 14, date.tolist())
        print('RSI is:', rsi[len(rsi) - 1])
        if rsi[len(rsi) - 1] > 70:
            print('Stock may be Overbought')
        if rsi[len(rsi) - 1] < 30:
            print('Stock may be Oversold')
        (a, b, c) = bollinger_band(close, 20)
        a[:19] = 0
        b[:19] = 0
        c[:19] = 0
        plotB_band(b.tolist(), c.tolist(), date.tolist(),
                   close.tolist())
        regression(close)
        dates = []
        for x in range(0, 30):
            dates.append(x)
        f = predict_price(dates, close, 30)
        f = float('{0:.2f}'.format(f))
        print('Next Predicted Price:', f)
    except Exception:
        print('May be No. Of Attempts for today by the key is finished')
except Exception:
    print('Check your Internet Connection')
```

APPENDIX 2

SCREENSHOTS



TECHNICAL ANALYSIS REPORT

REFERENCES

[1] Abidatul Izzah,Yuita Arum Sari,Ratna Widyastuti,Toga Aldila ,"Moblie app for stock prediction using Improved Multiple Linear Regression", International Conference on Sustainable Information Engineering and Technology (SIET), 2017.
[2] Abhinandan Gupta,Dev Kumar Chaudhary,Tanupriya Choudhury, "Stock Prediction Using Functional Link Artificial Neural Network" ,3rd International Conference on Computational Intelligence and Networks (CINE), 2017.

[3] Rupesh A.Kamble, "Short and long term stock prediction using decision tree",International Conference on Intelligent Computing and Control Systems (ICICCS), 2017.

[4] Mingying Wei, Shuhong Zhang, "Stock market prediction using data mining techniques", International Conference on Intelligent Sustainable Systems (ICISS), 2017.
[5] Si-Shu Luoo,Yang Weng,Wei Wei Wang,Wei Xing Hong, "L1- regularised logistic regession for event-driven stock market prediction", 12th International Conference on Computer Science and Education (ICCSE) , 2017.
[6] David M.Q.Nelson,Adriana C.M.Pereira,Renato A deOliveira, "Stock market's movement prediction with LSTM neural networks", International Joint Conference on Neural Networks (IJCNN), 2017.
[7] Naina Jamal,Akshay Malhotra, "Identification of persisting trend in the Indian stock markets using hurst exponent", IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016.
[8] Xi Chen,Zhi-Jie He, "Prediction of Stock Trading Signal Based on Support Vector Machine", 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), 2015.
[9] Elena Bautu,Sun Kim,Andrei Bautu, "Evolving hypernetwork models of binary time series for forecasting price movements on stock markets", IEEE Congress on Evolutionary Computation , 2009.
[10] R.S.T.Lee, "iJADE stock advisor:an intelligent agent based stock prediction system using hybrid RBF recurrent network", IEEE Transactions on Systems, Man, and Cybernetics, 2004.