MarketPulse - An App that Generate Recommendations for the Stock Market

Chandrakishor Singh
Dept. of Computer Engineering
Vidyavardhini's College of Engineering and Technology
Palghar, India

Ronith Sinha
Dept. of Computer Engineering
Vidyavardhini's College of Engineering and Technology
Mumbai, India

Akshay Valaki
Dept. of Computer Engineering
Vidyavardhini's College of Engineering and Technology
Palghar, India

***Abstract*—People have invested in the stock market since its inception. Though it's very hard to keep up with the ups and downs of the stock market, history has shown that over time the stock market has given profits. It's a good choice for long term investments. However, recently new strategies of investment have emerged which involves trading of stocks via computer algorithms. This paper describes 3 measures of assessing the performance of a stock and finally we will list 3 algorithms which use one or more of these 3 measures to calculate the time to buy or sell any stock with a potential to generate profit. To ensure that our algorithm actually generates profits, we have tested our algorithm over the stock prices of the last 8 years of Nifty100 and the algorithm successfully generates profit in the range of 10-18% over the year.**

We have developed a cloud based system which stores near real time stock prices of Nifty50 companies that are listed in NSE every 30 minutes. This data is then analysed by the algorithm that we have developed and it may generate buy or sell signals. We have also developed a client application in the form of a Hybrid Android App which anyone with a google account can download. Using this app, users can buy or sell stocks of Nifty50 companies. However, due to complex compliance, security standards and scope of our project, we have not provided an actual payment mechanism. Instead users buy or sell stocks via virtual money. They have two modes of buying/selling stocks namely auto and normal. In auto mode the system will always buy/sell on behalf of users whenever buy/sell signal is generated and in normal mode, user will decide whether it wants to buy/sell upon receiving the notification.

Keywords—Algorithmic trading, technical indicators, cloud, AWS, ionic, android, hybrid app, real time, stock, NSE, Nifty50, Nifty100

I. INTRODUCTION

Our system is a cloud based platform which estimates the correct time to buy or sell stocks of Nifty50 companies that are listed on the National Stock Exchange(NSE) via our hybrid android app. The algorithm finds the most promising time of buying/selling a stock and then informs the users via mobile notification.

On a high level, our system works in the following manner.

- Get the real time stock prices of all Nifty50 companies and store it in the database
- Retrieve the stock prices from database and analyse it to determine whether it is correct time to buy/sell stocks
- Notify users about buy/sell signal via android notification mechanism
- Repeat the above steps after every 30 minutes from 9AM-3PM through MON-FRI

The most important component of our system is the algorithm which analyzes the stock data and determines whether it is the right time to buy or sell a particular stock. By 'right time' we mean the time which seems very promising in generating profit by that transaction. We have used 3 technical indicators namely Moving Average (MA), Moving Average Convergence Divergence (MACD) and Exponential Moving Average (EMA) to help us in developing the algorithm. We have used these technical indicators in various ways to develop three algorithms which attempt to estimate a promising time to buy/sell stocks which potentially gives profit. These algorithms are then tested on a dataset which contains the stock prices of Nifty100 companies of NSE. This dataset was obtained through Yahoo Finance. It has the data about the open, high, low, close price of stocks of each day. After testing all three algorithms on the same dataset, the one which gives the highest profit is used in the production environment. The calculation and detailed discussion on these technical indicators are given in the literature review section.

II. SCOPE

We aim to create a platform where users can buy/sell stocks of NIFTY50 companies, however, these transactions are not real. The transactions are not made using actual money because for that the system needs to comply with many complex policies, regulations and processes which are out of the scope of this project. However, this application could be used as a simulation environment where users can buy/sell

virtual stocks (though stock prices are real). It could be used to create awareness about stock market investment in India. Many people still don't invest in stock markets because they find it difficult to understand all the policies and jargon. Also the unpredictable nature of the stock market makes it very difficult for the first time investors to enter in it. This makes stock market investment seems more risky than it actually is. Using this application users can get a first hand experience about stock market investment. Since the money is also virtual there is no loss in using it. After some time when users get comfortable in stock market investment, they can try actual trading by themselves or through some other established firm.

III. LITERATURE REVIEW

A. *Technical Indicators: Moving Average*

In statistics, a moving average (rolling average or running average) is a calculation to analyze data points by creating a series of averages of different subsets of the full dataset. It is also called a moving mean (MM) or rolling mean and is a type of finite impulse response filter.[4] Given a series of numbers and a fixed subset size, the first element of moving average is obtained by taking the average of the initial fixed subset of the number series. Then the subset is modified by "shifting forward" that is, excluding the first number of the series and including the next value in the subset. We are calculating the moving average by adding a stock's prices over a certain period and dividing the sum by the total number of periods. A five-day moving average adds up the five most recent daily closing prices of stocks and divides it by five to create a new average each day. Each average is connected to the next, creating the singular flowing line.[2]

For example, if the stock prices for 8 consecutive days are as follows: 100, 103, 104, 108,112, 124, 115, 118. Then, the moving average for these prices is calculated as: 105.4, 110.2, 112.6, 115.4.

Moving averages lag behind current price action because they are based on past prices; the longer the time period for the moving average, the greater the lag. Thus, a 200-day MA will have a much greater degree of lag than a 20-day MA because it contains prices for the past 200 days. The length of the moving average to use depends on the trading objectives, with shorter moving averages used for short-term trading and longer-term moving averages more suited for long-term investors.[4] The 50-day and 200-day MAs are widely followed by investors and traders, with breaks above and below this moving average considered to be important trading signals. Moving averages also impart important trading signals on their own, or when two averages cross over. A rising moving average indicates that the security is in an uptrend, while a declining moving average indicates that it is in a downtrend. Similarly, upward momentum is confirmed with a bullish crossover, which occurs when a short-term moving average crosses above a longer-term moving average.[3] Downward momentum is confirmed with a bearish crossover, which occurs when a short-term moving average crosses below a longer-term moving average. Another technical indicator used in conjunction with moving average is Exponential Moving Average (EMA). An exponential moving average (EMA) is a type of moving average (MA) that places a greater weight and significance on the most recent data points.[7]

Moving Average Illustration

B. *Technical Indicators: Exponential Moving Average (EMA)*

An exponential moving average (EMA) is a type of moving average (MA) that places a greater weight and significance on the most recent data points. The exponential moving average is also referred to as the exponentially weighted moving average. An exponentially weighted moving average reacts more significantly to recent price changes than a simple moving average (SMA), which applies an equal weight to all observations in the period.[5]

The formula for EMA is = ($value_{today}$ * (*Smoothing* / (1 + *days*))) + $EMA_{yesterday}$ * (1 - (*Smoothing* / (1 + *days*)))[6] The three basic steps to calculating the EMA are:

1. Calculate the SMA.
2. Calculate the multiplier for smoothing/weighting factor for the previous EMA.
3. Calculate the current EMA.

The EMA gives a higher weighting to recent prices, while the SMA assigns equal weighting to all values. The weighting given to the most recent price is greater for a shorter-period EMA than for a longer-period EMA. For example, an 18.18% multiplier is applied to the most recent price data for a 10-period EMA, whereas for a 20-period EMA, only a 9.52% multiplier weighting is used.[8]

EMA Illustration

The 12- and 26-day exponential moving averages (EMAs) are often the most popularly quoted or analyzed short-term averages. The 12- and 26-day are used to create indicators like the moving average convergence divergence (MACD) and the percentage price oscillator (PPO).[9]

Like all moving average indicators, they are much better suited for trending markets. When the market is in a strong and sustained uptrend, the EMA indicator line will also show an uptrend and vice-versa for a down trend. A vigilant trader will not only pay attention to the direction of the EMA line but also the relation of the rate of change from one bar to the next.[6] For example, as the price action of a strong uptrend begins to flatten and reverse, the EMA's rate of change from one bar to the next will begin to diminish until such time that the indicator line flattens and the rate of change is zero. Because of the lagging effect by this point, or even a few bars before, the price action should have already reversed. It follows, therefore, that observing a consistent diminishing in the rate of change of the EMA could itself be used as an indicator that could further counter the dilemma caused by the lagging effect of moving averages.

EMAs are commonly used in conjunction with other indicators to confirm significant market moves and to gauge their validity. For traders who trade intraday and fast-moving markets, the EMA is more applicable. Quite often, traders use EMAs to determine a trading bias. For example, if an EMA on a daily chart shows a strong upward trend, an intraday trader's strategy may be to trade only from the long side on an intraday chart.[7]

C. *Technical Indicators: Moving Average Convergence Divergence (MACD)*

Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA.[8] The result of that calculation is the MACD line. A nine-day EMA of the MACD called the "signal line," is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals. Traders may buy the security when the MACD crosses above its signal line and sell - or short - the security when the MACD crosses below the signal line.[9] Moving Average Convergence Divergence (MACD) indicators can be interpreted in several ways, but the more common methods are crossovers, divergences, and rapid rises/falls. The MACD has a positive value whenever the 12-period EMA (blue) is above the 26-period EMA (red) and a negative value when the 12-period EMA is below the 26-period EMA.[5] The more distant the MACD is above or below its baseline indicates that the distance between the two EMAs is growing. There are a number of calculations involved in the creation of the total (MACD) indicator, all involving the use of exponential moving averages. An EMA is calculated as follows:[9]

- Calculate the simple moving average (SMA) for the chosen number of time periods. (The EMA uses an SMA as the previous period's EMA to start its calculations.) To calculate a 12-period EMA, this would simply be the sum of the last 12 time periods, divided by 12.[3]
- Calculate the 12 EMA itself as:
  - $(\text{Close} - \text{EMA}_{\text{previous period}}) * 0.1538 + \text{EMA}_{\text{previous period}}$

Putting together the MACD requires simply doing all of the following EMA calculations for any given market instrument (a stock, future, currency pair, or market index):

1. Calculate a 12-period EMA of the price for the chosen time period.
2. Calculate a 26-period EMA of the price for the chosen time period.
3. Subtract the 26-period EMA from the 12-period EMA.
4. Calculate a nine-period EMA of the result obtained from step 3.

This nine-period EMA line is overlaid on a histogram that is created by subtracting the nine-period EMA from the result in step 3, which is called the MACD line, but it is not always visibly plotted on the MACD representation on a chart.[7]

MACD Illustration

IV. METHODOLOGY

A. *Analysis & Results*

We have developed three algorithms that use one or more of the above listed technical indicators namely Moving Average, Moving Average Convergence Divergence and Stochastic RSI. The algorithms use these technical indicators in various ways to estimate the most promising time of buying/selling a stock. These algorithms are then tested on a dataset which contains the stock prices of Nifty100 companies of NSE over a period of 8 years (october 2010 to october 2018). This dataset was obtained through Yahoo Finance. It has the data about the open, high, low, close price of stocks of each day. The explanation of all three algorithms and their performance (in terms of rate of return) is given as follows.

1. MACD Crossover
   Buy: when the signal line crosses the MACD line from above.
   Sell: when the MACD line crosses the signal line.

Profit percentage = 20.23%

2. EMA Crossover
   Buy: 9EMA crosses 14EMA and 20EMA from below to above
   Sell: 9EMA crosses 14EMA from above to below Profit percentage = 13.28%
3. EMA + MACD Crossover
   Buy: 9EMA crosses 14EMA and 20EMA from below to above AND MACD line crosses Signal line from below to above
   Sell: MACD line crosses Signal line from above to below
   Profit percentage = 17.45%

From the above analysis, it is clear that MACD Crossover condition is giving the highest rate of return of 20.23%. Therefore we will choose this algorithm to be used in the production environment.

B. Implementation

We have used Serverless architecture for building the system. In a serverless approach, all the hardware and most of the software needed for building the system resides on the cloud. The cloud provider is responsible for the maintenance, security of the hardware. This enables the developer to focus more on the business logic and less on the things which are not directly related to the application like maintenance of hardware, applying the latest security patches, backup etc. We have used Amazon Web Services(AWS) as our cloud provider because of its competitive price and generous free tier plan.

AWS has more than 90 services related to computing, storage, network, analytics, management, deployment etc. To develop our application we have used few but core features of AWS. These include API Gateway, DynamoDB, Lambda, Cognito and CloudWatch. Below is a brief summary of each service/library that we have used in our backend system.

1. API Gateway
   API Gateway is a GUI based web API (Application Programming Interface) development service. We have used it to create web APIs for the client app. These include API for authentication of users, getting the list of stocks that are already purchased by the user, buying or selling any stock and so on.
2. DynamoDB
   DynamoDB is used to store the data of stock prices that are collected every 5 minutes. It is a fully managed proprietary NoSQL database service that supports key-value and document data structures and is offered by AWS.
3. Lambda
   AWS Lambda is used to do all the computing processes. It is an event driven computing platform. Basically it is a computing service that runs code in response to events and automatically manages all the resources required to run that code (like memory, CPU time etc.). Lambda is used to run all server side logic. It is responsible for scraping the stock prices from the web, storing the data in DynamoDB, performing analysis on the data, responding to users via REST APIs.
4. Cognito
   Cognito is an AWS service that lets you add user sign-up, sign-in and access control for your web and mobile apps quickly and easily. We have used this service to authenticate users in our system and to synchronize their data. Users are given the functionality of authenticating themselves by using a Google account.
5. CloudWatch
   CloudWatch is a monitoring service which gives the live status of all of the AWS resources and infrastructure. It is used to periodically trigger execution of code that is responsible for scrapping stock prices from the web. We are also using this service as an analytic tool to get insight on the health of our system. It gives data about the execution time of lambda functions, access frequency of DynamoDB, number of users etc.
6. Some Important Third Party Libraries
   We have used many third party open source libraries to develop the functionality of our system. Some of the critical ones are follows:
   - nsetools : nsetools is a library for collecting real time data from the National Stock Exchange (India). It can be used in various types of projects which require fetching live quotes for a given stock or index or building large data sets for further data analytics. It can also be used to build cli applications which can provide you live market details at a blazing fast speeds, much faster than any browser. The accuracy of data is only as correct as provided on the website of NSE http://www.nseindia.com[11]
   - pandas: pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. We have used this library for analysing the historical data of Nifty100 companies to rank the best algorithm.[10]
   - TA-Lib: TA-Lib is widely used by trading software developers requiring to perform

technical analysis of financial market data. It includes many indicators such as ADX, MACD, RSI etc.[12]

- NumPy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.[11]

The front end or client application is a Hybrid Android Application. We are using the following technology for building the mobile application.We are using Ionic to develop the android app. Ionic is an open source, front-end SDK for developing Hybrid Mobile Applications using web technologies such as HTML, CSS and JavaScript. It provides mobile optimised web technology based components as well as native APIs using Cordova and Ionic Native.

1. Ionic
   Ionic is an open source, front-end SDK for developing Hybrid Mobile Applications using web technologies such as HTML, CSS and JavaScript. It provides mobile optimised web technology based components as well as native APIs using Cordova and Ionic Native. Ionic provides tools and services for developing hybrid mobile, desktop, and Progressive Web Apps based on modern web development technologies and practices, using Web technologies like CSS, HTML5, and Sass
2. Angular
   Angular is a TypeScript-based open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. Ionic uses angular behind the scenes to define web components required to give a native app experience while using the standard web technologies like HTML, CSS and JavaScript.
3. Cordova
   Apache Cordova (formerly PhoneGap) is a mobile application development framework. Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone. It enables wrapping up of CSS, HTML, and JavaScript code depending upon the platform of the device. Using the cordova device level APIs like camera, location services can be accessed by JS based code.

V. DESIGN

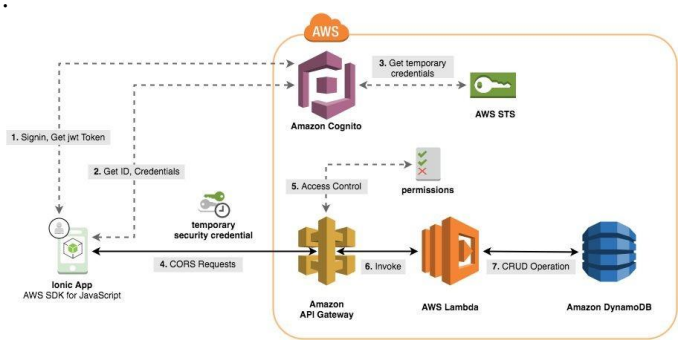
Our system is further divided into 2 modules which are as follows:

A. Client Module

This Module is the Client side of the Android application. This application will be installed on the Android platform smartphone. It requires reliable internet connectivity. The user will be notified whenever the server generates a buy signal on the stock of any company. But the sell signal will only be sent to users who have already bought the stock of that company in the past. After receiving the notification of buy/sell the user will have a 5 minute window to accept that signal. After that the option will become invalid. If the user doesn't want to miss any buy/sell signal then he/she can enable Auto mode of Trading option in settings which will automatically accept all buy/sell signals.

- Ionic will be used for developing the client side Hybrid Android App.
- Angular will be used as the User Interface Framework for developing the user interface of the hybrid android app. Also it will be responsible for fetching and posting the user data from/to the server.
- The web application developed using Ionic and Angular will be converted to Hybrid Android App via the cordova framework.
- This android app will use web APIs, developed and managed in AWS API Gateway, to authenticate users, buy/sell stocks.

.



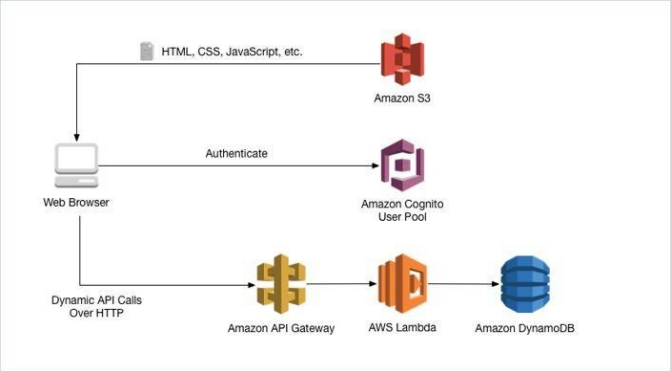
Client - Server Architecture Using Serverless Approach

B. Server Module

This module is the server side (or the backend) of the system where most of the processing is done.

- AWS Lambda is used to execute the code which is responsible for scraping the stock prices from the web. A library called nsetools is used to scrape the prices of the stocks from the website. It is also the service which runs the code that contains the algorithm for generating buy/sell signals.
- DynamoDB will be used for storing recent stock prices, buying/selling signals and user data like it's username, portfolio details etc.
- API Gateway is used for providing REST APIs for the client app (Android app). Using these APIs, clients can authenticate themselves, fetch and post user data, buy/sell stocks.

- CloudWatch is used to set an event which is basically a trigger that is used to run the scrapping and analysis code every 30 minutes.
- AWS Cognito is used to authenticate the users via Google Account.



Backend Architecture

VI. CONCLUSION

We have developed three different algorithms that estimate the best time to buy/sell a stock of Nifty50 companies by using three technical indicators namely Moving Average, Moving Average Convergence Divergence and Exponential Moving Average. We have gathered the stock market data for the past 8 years of Nifty100 companies and tested the performance (amount of profit generated) of all three algorithms. Based on the amount of profit generated by individual algorithms we have selected the one which generates the maximum profit.

After the development and selection of the final algorithm, we developed our system. Our system is a cloud based system i.e., all of the hardware required to build the system is present as cloud infrastructure. We have chosen AWS as our cloud provider. Various services of AWS like Lambda, DynamoDB, CloudWatch, API Gateway etc. are used to develop the backend of the system. Ionic (with Angular) was used to develop the hybrid android app along with Cordova.

As we have mentioned in our Analysis section, the profit percentage of different algorithms is different. Stock market is extremely unpredictable and obviously designing algorithms that give relatively high rates of return (> 18%) is quite difficult. However, in the present time most of the stock trading is already done by automated systems and this trend is just increasing day by day. Even if the profit percentage is normal, the algorithmic trading can give relatively large amount of profit by doing frequent tradings i.e., more than 100 or 1000 transactions per second. Here is a statement by Robert Greifeld, NASDAQ CEO in April 2011 “It is over. The trading that existed down the centuries has died. We have an electronic market today. It is the present. It is the future.”

individuals and organizations. We would like to extend our sincere thanks to all of them.

We also like to express our gratitude to our Head of Department, Dr. Megha Trivedi for providing assistance for the project. We are highly indebted to Prof. Swapna Borde for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards our parents & member of Vidyavardhini’s College of Engineering and Technology for their kind cooperation and encouragement which helped us in completion of this project.Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the support and help of many

REFERENCES

[1] J. Welles Wilder, New Concepts in Technical Trading Systems, ISBN 0-89459-027-8 Trend Research, 1978.
[2] John J. Murphy (2009). The Visual Investor: How to Spot Market Trends (2nd ed.). John Wiley and Sons. p. 100. ISBN 9780470382059..
[3] Appel, Gerald (2005). Technical Analysis Power Tools for Active Investors. Financial Times Prentice Hall. p. 166. ISBN 0-13-147902-4.
[4] Moving Average Convergence Divergence". Investopedia. Investopedia LLC. https://www.investopedia.com/terms/m/macd.asp Accessed 29 Nov 2019.
[5] Using the MACD". Investors Underground. Accessed 29 Nov 2019. https://www.investorsunderground.com/technical-indicators/macd-movi ng-average-convergence-divergence/
[6] Weighted Moving Averages: The Basics". Investopedia. Accessed 29 Nov 2019 https://www.investopedia.com/articles/technical/060401.asp
[7] Marek, Patrice; Šedivá, Blanka (2017). "Optimization and Testing of RSI". 11th International Scientific Conference on Financial Management of Firms and Financial Institutions.
[8] RSI. StockCharts. StockCharts.com. Accessed 29 Nov 2019. http://stockcharts.com/school/doku.php?id=chart_school:technical_indic ators:relative_strength_index_rsi
[9] Stochastic Indicator [ChartSchool]". Accessed 29 Nov 2019. https://school.stockcharts.com/doku.php?id=technical_indicators:stochas tic_oscillator_fast_slow_and_full
[10] pandas - Python Data Analysis Library. https://pandas.pydata.org Accessed 5 Feb 2020.
[11] Welcome to nsetools’s documentation! — nsetools 1.0.4 documentation.https://nsetools.readthedocs.io/en/latest/ Accessed 5 Feb 2020.
[12] Numpy. Accessed 5 Feb 2020. https://numpy.org
[13] TA-Lib : Technical Analysis Library. Accessed 5 Feb 2020. https://ta-lib.org