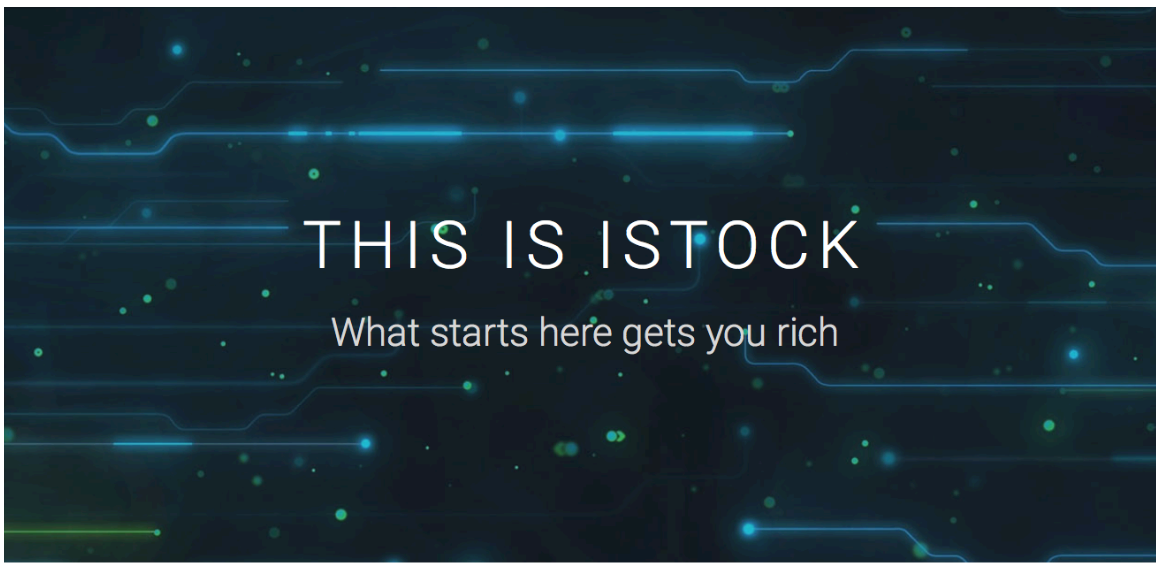
Web Stock Forecaster



D. Yao (T. L.), Z. Zheng, W. Fang, X. Zhang, K. Dong

Group 7
May 1, 2015

Individual Contributions Breakdown

All team members contribute equally

Table of Contents

1 Introduction and Background Information

1.1 Background

Stock market established in 1811 and attracts so many people ever since. People can buy or sell stock of different companies and earn or lose a lot of money.

To make money in the stock market, people need to analysis and make prediction about the stocks. There are two major analysis method that is widely used. One is called the Fundamental Analysis, the other is called Techinical Analysis. Fundamental Analysis is to get information about a company for every suspects, including the profit-an-loss-statements, the quarterly balance sheets, the dividend records, the policies of the company, sales data, managerial ability, plant capacity, bank and treasury reports, production indexes, price statistics, crop forecasts and daily news [1]. Then after summarizing all the finance information, also concerning about the daily news and other parts, a fundamental analyst would give an estimate price of the company's stock. If the estimate price is higher than the company's real stock price, it is a good chance to buy, in contrary, if the estimate price is lower than the real stock price, it is a good chance to sell. It is hard to be a fundamental analyst as there is too much to concern and need a person's instinct and some subjective judgement.

The Techinical Analysis is a method to study about the stock market itself. It mainly concern the stock price, with other parameters like the trading volume as references to use the history data for future predicting. The Technical Analysis starts from Dow's Theory and one of the most important principle of Dow's Theory is that the stock price itself can reflect everything about a company. The price can reflect the finance situation and also people's perspective of the company. So unlike Fundamental Analysis, Technical Analysis concern restricted parameters, and do not need some subjective

[1] *Technical Analysis of Stock Trends*, 9th Edition, CRC Press, 2007, chapter one, page 3

judgement. In this way, it gives us a chance to use computer to help us do the analysis.

So in modern analysis of stock market, several methods are used for different stock prediction companies. For example, regression analysis prediction like the Bayesian method, Support Vector Regression(SVR) method, and Artificial Neural Network(ANN).

1.2 Introduction

In our project, our goal is to set up a system, based on the data from the Yahoo Finance database, use several methods to predict the stock price on the market and then present the predict result on a website. The core of our system is the predict methods, we plan to do both the short term prediction and the long term prediction.

For the short term prediction, we would use four methods to implement, the Bayesian Curve Fitting Method, the Stochastic Oscillator (STS), the Support Vector Regression (SVR) and the Aritificial Neural Network(ANN). For each method, we would use the historical database as the input of the method. SVR and ANN would need some training process in advance. These method would give the user the prediction stock close price for the next two days.

For the long term prediction, we would use the Bayesian and STS method, instead of conituously daily data, we would use the average close price of every ten days. So we can get the trend for the next month (20 working days).

The project would finally set up a fully website that can allow users to search and check stock data based on the Yahoo Finance database. It would also provide the short term and long term prediction for a given stock (for 10 stocks, in our project, we choose Google, Facebook, Twitter, Yahoo, Microsoft, Apple, Amazon, Walmart, Sony, and Canon). The system can allow the user to check for the history of short term prediction and the relative error.

The website includes a game system that users can have an origin amount of virtual money and use them to buy/sell stocks in the system's virtual stock market based on the real price. The system would also present the result when following each short term method recommendation since July 10th, 2014. The use can also check his or her stock information and total value information.

2 Customer Statement of Requirement

The customer would want a stock forecaster website has the following characteristics:

Prediction Method including famous company, we choose ten company for our prediction part and they are:

Google (GOOG)
Yahoo (YHOO)
Apple (APPL)
Facebook (FB)
Microsoft (MSFT)
Amazon (AMZN)
Sony (SNE)
Walmart (WMT)
Canon (CAJ)
Twitter (TWTR)

The user can get the short term prediction and long term predction of these 10 companies. The data and prediction can be represent clearly in the graphs. The user want to get more prediction result by using different method as references. The more methods, the better.

The user want to get relevant information in news or twitter for a given company, the company may not be in the list below.

The user could log in or sign up to experiment the game section. The user would be given a startup amount of money in the virtual market and try to earn more money in the virtual market.

3 Glossary of Terms

Trend Day:

Trend Day is one day that the stock market is opening, ususally from Monday to Friday.

Close Price:

The stock price at the end of the working hours of stock market in a Trend Day.

Day's Low

Today's low is the lowest price at which a stock trades over the course of a trading day. Today's low is typically lower than the opening or closing price.

Day's High

Today's high is the Highest price at which a stock trades over the course of a high day. Today's high is typically lower than the opening or closing price.

Short Term Prediction

Predict the stock market in the following day, the predict value is important, we want to get an accuracy short term prediction.

Long Term Prediction

Predict the stock market in the following months or even years, the predict value is not important because there are too many factors may influent the result, the trend would be much important than the actual value.

Daily Volume

Daily volume is the trading volume of one certain stock or the whole market in one day.

4 System Requirement

4.1 Enumerated Functional Requirements

Table 1. Functional Requirements

Identifier	Description	PW[2]
REQ – 1	The system shall compare users' input information with the database to apply or reject a user login	3
REQ – 2	The system shall allow new users to create new accounts.	3
REQ – 3	The system shall load users' profile once they logged in.	3
REQ – 4	The system shall download historical data / real time data from yahoo finance and store in the local database	5
REQ – 5	The system shall allow users to search or choose the stock based on the stock ticker or the company name	4
REQ – 6	The system shall predict the 10 stocks's close price in the next two days as the short term prediction. The system shall also display the history of short term prediction including the relative differences.	5
REQ –7	The system shall predict the 10 stocks's trend in the following 20 days (a month) as the long term prediction. The system shall use graph to display the trend.	5
REQ – 8	The system shall allow users in a game mode, in which user would have virtual money to purchase and sell stocks on the market.	4
REQ – 9	The system shall implement results of strictly following each day's recommendation of those short term prediction.	4
REQ – 10	The system shall store and display the user's stock information and total value information in the game mode.	4
REQ – 11	The system should query related finance news based on user's search, and provid the result to the user.	3

[2] Priority Weight

REQ – 12	The system shall do some some specific query of the database.	5

4.2 Non – Functional Requirements

Table 2. Non - Functional Requirements

Identifier	Description	PW
REQ – 13	The system shall download data periodically in the background, for the real time data, it will upload in every 5 seconds.	4
REQ – 14	The system shall update the short term prediction every day	4
REQ – 15	The system shall include an android app to check all the charts on the website.	4

5 Functional Requirements Specification

5.1 Stakeholders

The mainly stake holders for our project would be share holder, or we called “investors”. If our system can have a precise prediction of the stock market, either the short term or the long term, it would be great help for the inverstors to rational distribute their fortune on the stock market and also help them to make the sell/buy decision.

5.2 Actors and Goals

User: (Initializing type):
Interacts with the system, gets the stock information and prediction they need, takes part in the game experience. A registered user can have access to all the information and services of the website.

Administrator: (Initializing type):
The administrator is a special user and has top priority to access and change our database and all other user information.

Database: (Participating type)
Records all the stock data including the historical data, the real-time data and also the prediction data.

Yahoo Finance Database: (Participating type)
We get all the “real data”, including the historical data and the real-time data from the Yahoo Finance database.

5.3 Use Case Casual Description

Table 3. Use Cases

UC-No.	Casual Description
UC-1	***Sign up/ Log in***: User can create a new accout or log in to participate in the game system
UC-2	***Search Stock***: User can search stock by the stock ticker or company names and get the relevant information.
UC-3	***Get Sepcific Query***: User can get the Specific Query from the local database
UC-4	***Get Short Term Prediction***: User can get the short term prediction among the 10 companies, user can also get the historical data of prediction and the relative difference statistics.
UC-5	***Get Long Time Prediction***: User can get the long term prediction among ten companies.
UC-6	***Check News***: User can check the finance news based on the input keyword
UC-7	***Game Operation***: User can purchase or sell stocks on the virtual stock market with the given amount of money.
UC-8	***Strategy Display***: User can check the result of strictly following the recommendation of each short term prediction since July 10th, 2014.

5.4 Use Case Analysis and Diagram

The Figure 1 is the Use Case Diagram for the whole system. The user can check the news from the website (UC-6), and the system would query the news from the Yahoo Finance Database, the Twitter Database and so on. The user can also search for a specific stock (UC-2), and in that way, he or she can get the short term prediction (UC-5), or the long term prediction (UC-5). The prediction result all based on the data from the local database, but the local database would query from the Yahoo Finance Database on the background. After the user login or sign up on the website (UC-1), he or she can take part in the game operation (UC-7), or check the strategy (UC-8). The users' operation and account information would be stored in the local database.

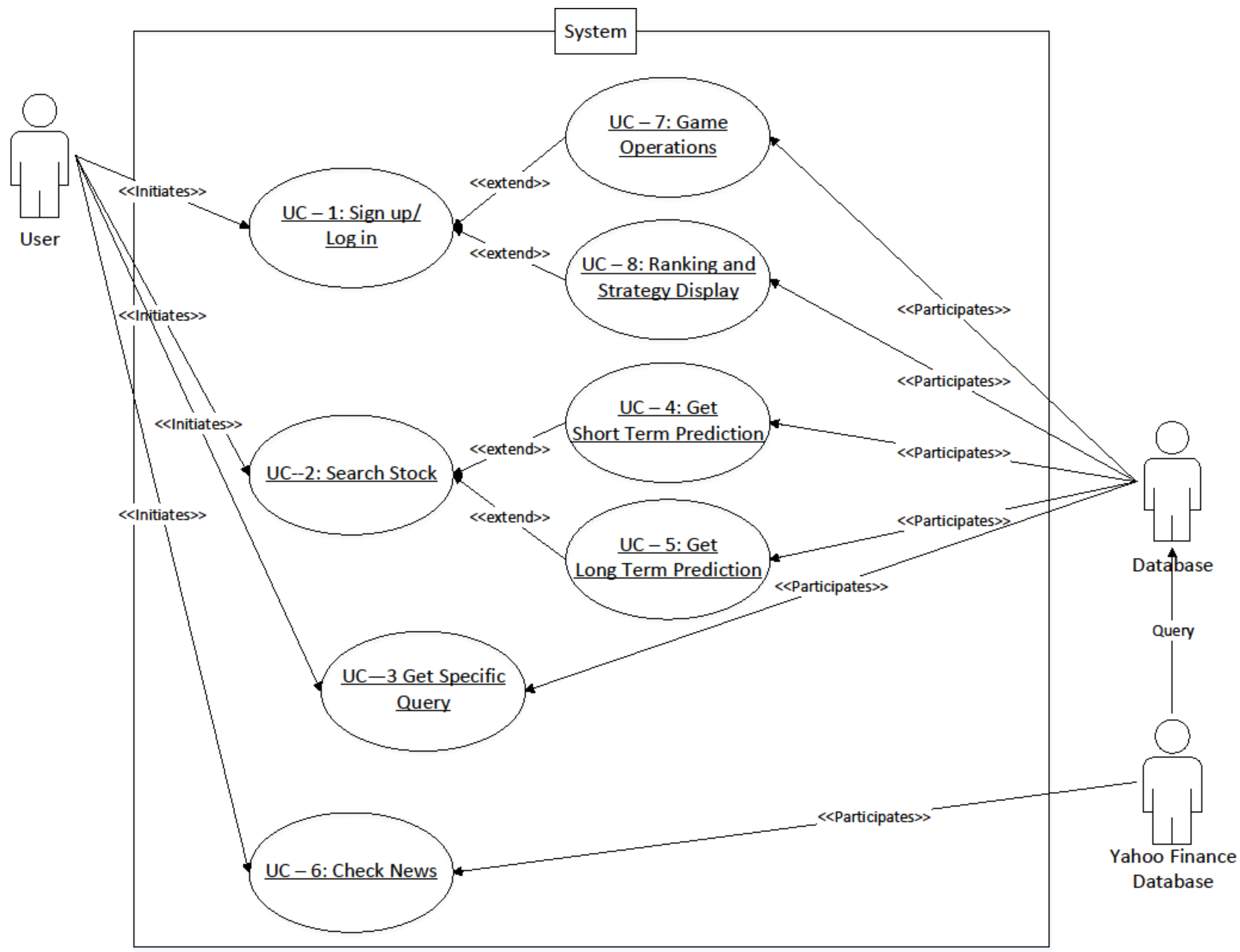


Figure 1. The Use Case Diagram for the Entire System

6 Sequence Diagram

The sequence diagram here include the important UCs in section 5.3

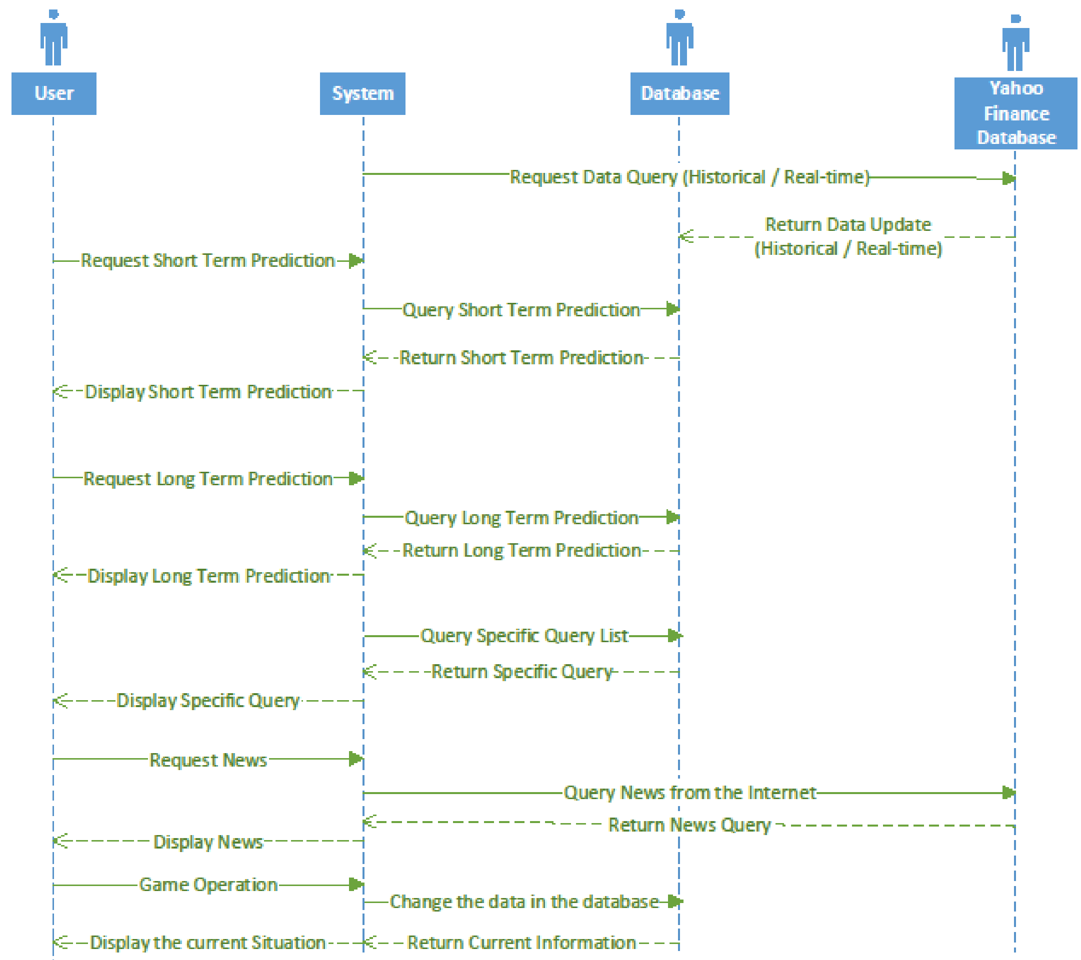


Figure 2. System Sequences Diagram

From Figure 2, we can see how system deal with the user requests and how the database would interact with the system. For getting the short term/long term prediction. The user send a request by choosing one stocks and then the system would query the prediction result from the local database, then the system would return to the user by graphing or numbers. The local database would automatically update the data from yahoo finance database. When user asked for news about any company, the system would query directly from yahoo finance database, twitter database and so on, and do not need to go through the local database. When user make a game operation (Buy or Sell), the system would modify his or her current information in the local database and display the current infos back to user.

7 System Architecture and System Design

7.1 Block Diagram

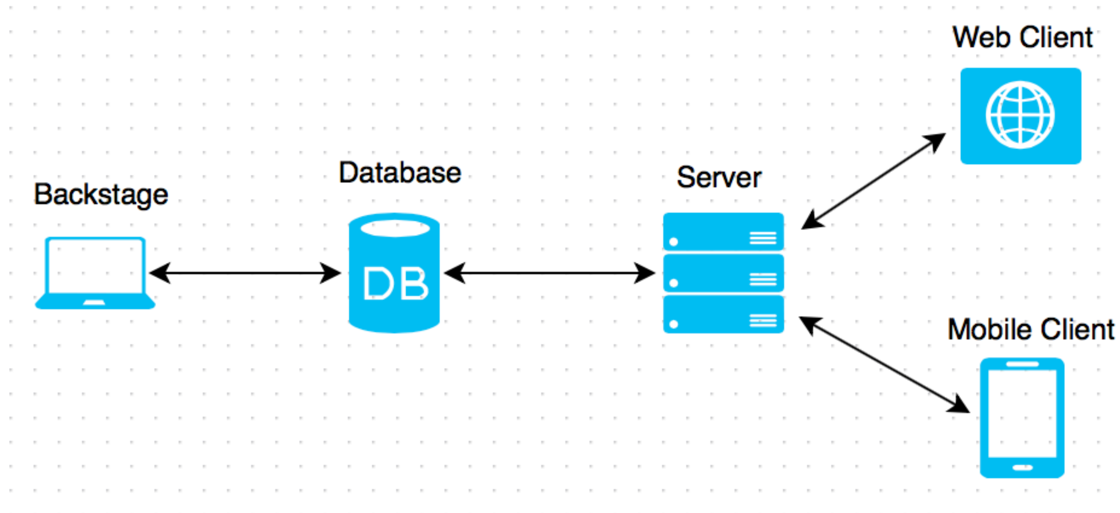


Figure 3. System Block Diagram

Our physical system is composed of a backstage, a database, a server and web or mobile clients. The architecture of our system is based on LAMP (Linux, Apache, MySQL, PHP). In Figure 3 , the backstage is a mac book computer and continuously runs the backstage programs. These programs will connect to Yahoo server and download real time, day's stock data from its database then store the data into our own database. Our system employs relational database MySQL 5.6.22, thus data is organized into records with several attributes in a table. The backstage programs also update the stock prediction in database when new stock data received. This is triggered by timer threads, one minute timer for real time data, and one day timer for day's data. The server in our system is Apache 2.2.26. The task of server is communicating with web clients and mobile clients. We designed user interface (UI) in the two clients, and users will interact with UI to send request to our server. The request may contain the information in database, so the server also need to communicate with the database, and rearrange the information then response to the users.

7.2 Programming Technologies and Web Sources

The backstage programs are written in Java language. Several application programming interfaces (API) such as Jama, libsvm, JDBC, neuroph and YhooFinance are imported in order to support our system's functions. YhooFinance API is used to connect and download stock data from Yhoo. Java database connectivity (JDBC) is a interface for backstage programs inserting, deleting, or updating data in our own database. Jama is a matrix computation library and is imported by our mathematical prediction methods. libsvm and neuroph are APIs for SVM and ANN machine learning algorithms, which also used for prediction computation. MySQL database language is a SOL like programming language and can be embedded in other languages such as java. On the server side, we use PHP as server side language, which is mainly responsible for accepting user's requirement, obtaining and re-organizing the data from database to display in clients. On web clients, they will download client files such HTML, CSS, JavaScript from our server to browser the UI. The communication between server and clients is implemented by asynchronous JavaScript and XML (AJAX) technology which could update parts of the websites instead of refreshing the whole page. The data format in communication we employed JavaScript object notation (JSON) instead of XML. In the communication process, PHP on server side will firstly receive the request from the JavaScript on client side and get data from database as required then package the data into JSON format then send back to client. The client will parse the JSON data and used for displaying.

8 Class Diagram and Interface Specification

In class diagram section, we introduce the programs running on the backstage not including programs running on the server and clients. Backstage programs are mainly responsible for data collection, processing and storing in database. An overview class diagram is shown in Figure 4.

StockForecaster

StockForecaster is the entry or main class of programs. In StockForecaster, SYMBOLS is a global array storing 10 stocks' symbol names and can be seen in other classes. StockForecaster will start two timers (threads) for real-time and historical stock data collection. REALTIME_STOCK_INTERVAL is set a number less than one minute, and HISTORICAL_STOCK_INTERVAL is set one day. In the historical stock thread, after inserting a new day's stock data in database, it also updates all calculation result tables for display.

DatabaseManager

This class is a static class and mainly designed for storing the database configurations such as database name and password for global use. Any class which needs to link to database for inserting or updating will use the members in DatabaseManager. By using DatabaseManager, we can easily change the database configurations when switching to a new database.

RealtimeStock & HistoricalStock

These two classes import YHOO API to connect YHOO server and download stock data from YHOO database. For RealtimeStock, we continuously receive the real time stock price and volume for 10 stocks and store them with the time information into database. For HistoricalStock, we download open price, close price, day's lowest price, day's highest price and volume and calculate the difference between today's close price and yesterday's close price. The difference describes the variation of the stock.

BayesianPredictor & STSPredictor & MACDPredictor & MovingAverage

These three classes depend on the Bayesian, STS and MACD classes respectively. The mathematical computations are packaged into Bayesian, STS and MACD classes, and predictors will get new historical data from database, send the formatted historical data as arguments in Bayesian, STS and MACD classes, finally store the prediction result into prediction tables. BayesianPredictor and STSPredictor both have methods for short term prediction and long term prediction while MACDPredictor is only designed for long term prediction. The results of short term prediction are a predict stock price and the corresponding suggestion for buy, sell or hold for that day. The results of long term prediction are a predict trend and a suggestion for buy, sell or hold for a long time. As for MovingAverage, this class contains the method to compute SMA and EMA sequence based on all historical data. As we know, SMA and EMA will smooth the stock price variation, and when the number of stock prices used for computation per step increased, the plot will become more smooth and it indicates the trend for longer time.

SVM & ANN

The SVM and ANN classes both use external libraries to implement the SVM and ANN algorithms. They both pick up latest several month stock prices as examples to train a mathematical model and predict the next day stock price. They are both implemented for short term prediction. Besides, some important parameters which may influence the prediction accuracy are calculated or predefined in the classes. Finally, SVM and ANN will store the prediction stock price into their responding tables.

AI

AI class is used in virtual stock exchange market of our project. AIs are like virtual players, they will follow the action suggestion yielded by their prediction methods for each stock every day. AI class computes four AIs for Bayesian, STS, SVM and ANN short term prediction methods and records the total value (balance plus stock values) they have every day. This demonstrates that users will gain profits if they follow our suggestions.

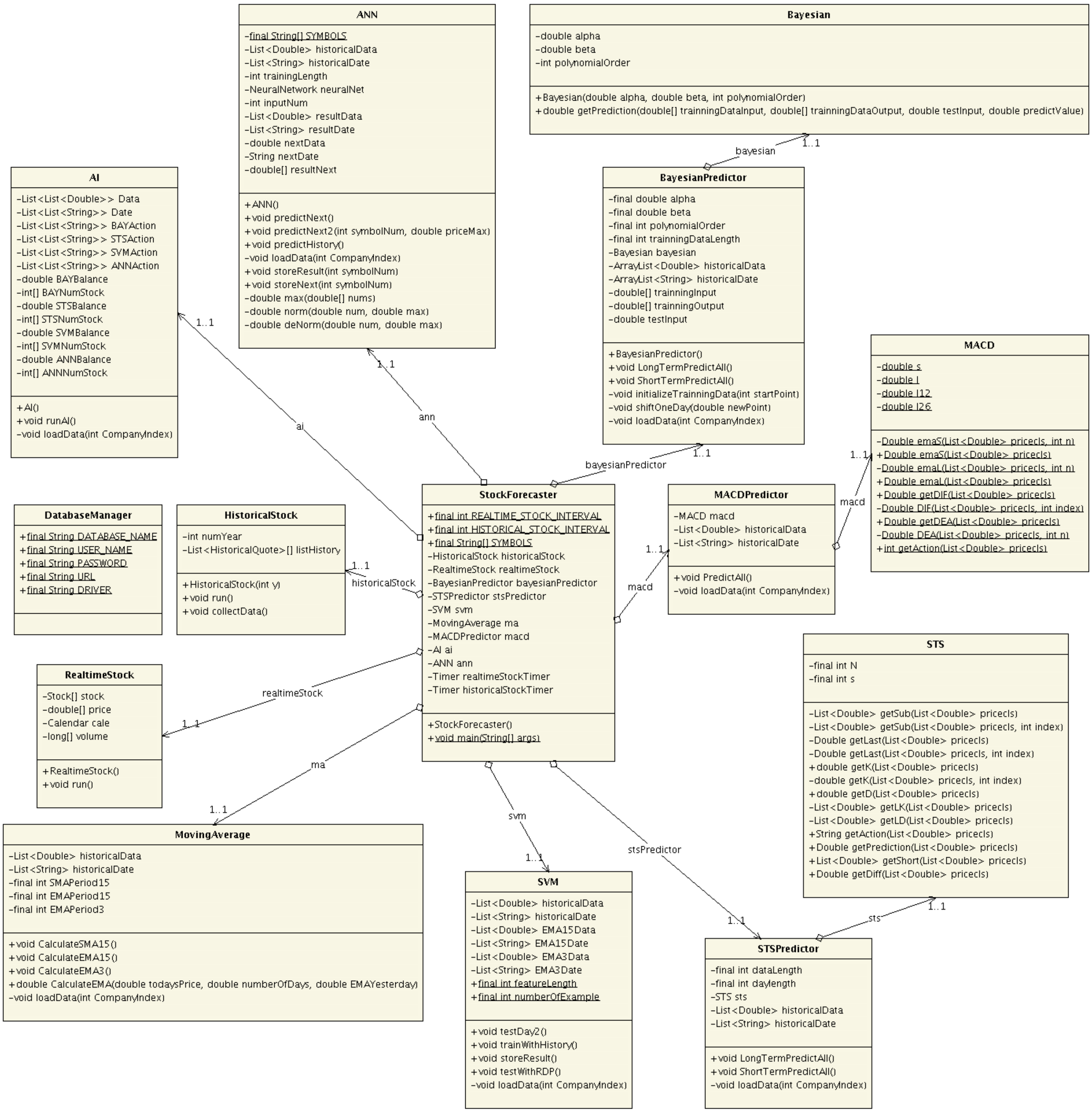


Figure 4. Class Diagram for Backend

9 Database Design and Implementation

9.1 Database Design

9.1.1 Description

When implement data collection module, it's important to retrieve stock information and store into a local relational database.

In this task, we were asked to store 10 stocks information.in the local database

- Stocks should be able to identify by a unique 'Symbol' name. And every stock belongs to a corresponding type (e.g. industrial).
- For each stock, their real-time data (at least one day) should contain the price, time and volume. The time slice between two real-time points should be no more than one minute.
- Except real-time data, database should store at least one year of the historical prices, which contain time, open, high, low, close, and volume.

9.1.2 Entity-relationship (ER) diagram

Based on the description, we developed the ER diagram for the database that contain the historical data and realtime data. Then based on this database, we extend it for storage the prediction results and the game information. The right part of the ER diagram contain the basic realtime data and historical data. The up part is the long prediction and it contain the predict result of three predict method, The down part is the place to store the game situation for the registed user. The complex part is the left part, which concern the short term prediction. Except the similar items in the long term prediction, this part also include four AIs that store the historical total value if you follow the recommendation of each short term prediction method.

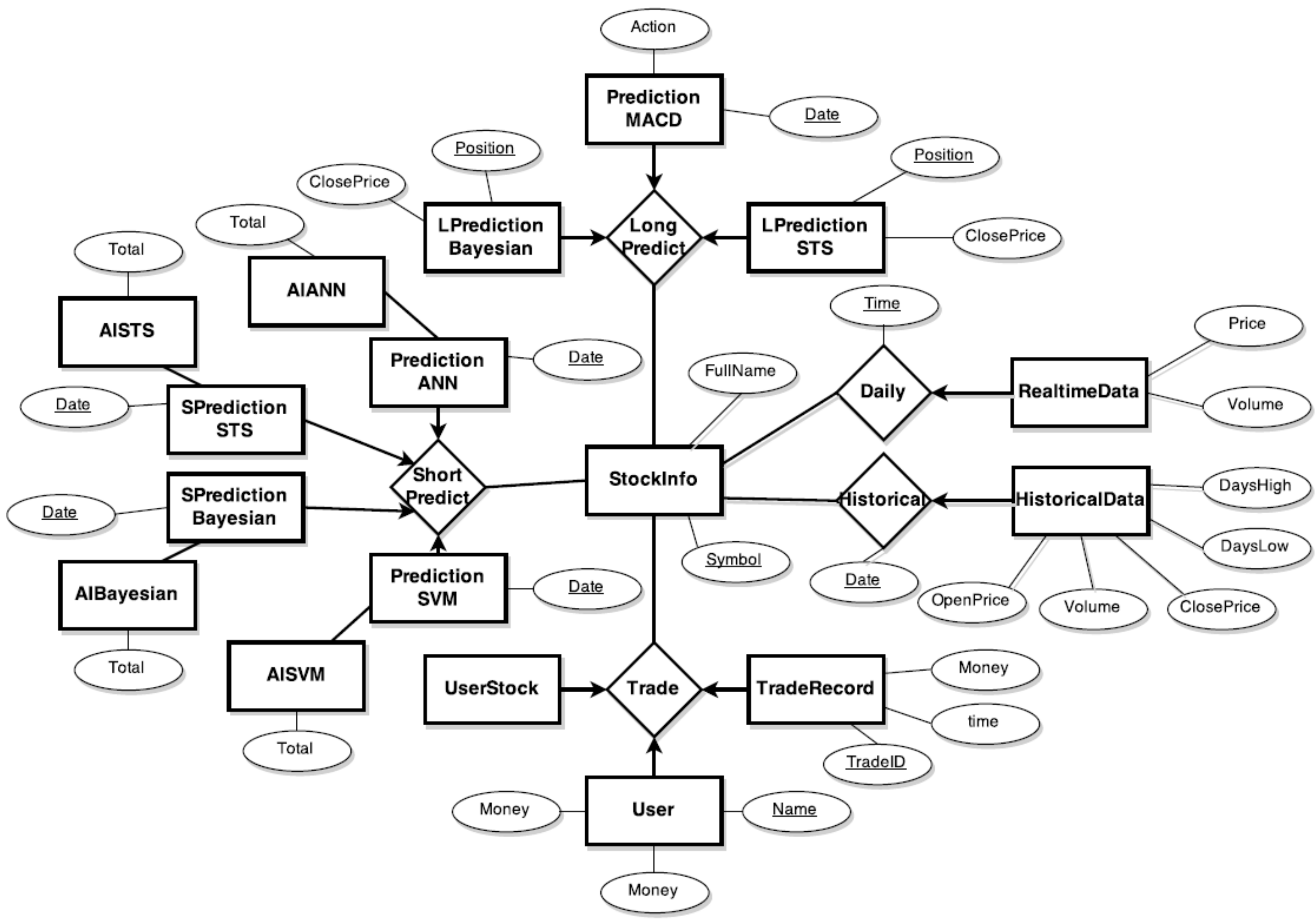


Figure 5. The Entity-Relation Diagram for the Database

9.1.3 Relationship between tables

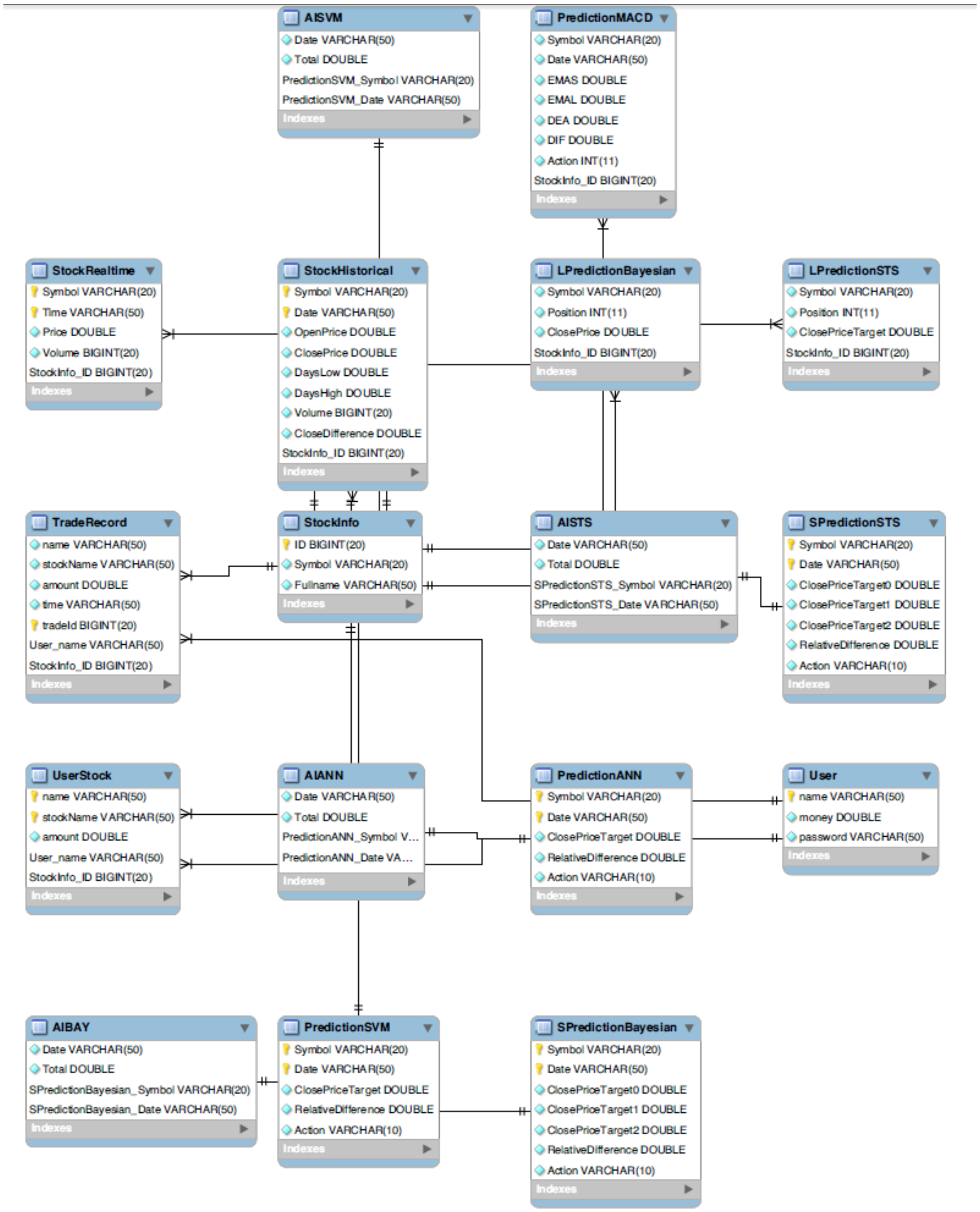


Figure 6. The Relationship between the Tables

Figure 6 is the diagram that shows the relationships between all the tables.

9.1.4 Database schema

Table 4. Table structure for table StockInfo

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Unique
StockType	varchar(20)	No	

Table 5. Table structure for table StockRealtime

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Foreign
Time	varchar(50)	No	Primary Key
Price	double	No	
Volume	double	No	

Table 6. Table structure for table StockHistorical

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Foreign
Date	varchar(50)	No	Primary Key
OpenPrice	double	No	
ClosePrice	double	No	
DaysLow	double	No	
DaysHigh	double	No	
Volume	double	No	

As the structure, primary key set of StockHistorical is <Symbol, Date>, and primary key set of StockRealtime is <Symbol, Time>. So, all historical data for five (at least) stocks is in one table. The same in real-time data.

Table 7. Table structure for table TradeRecord

Column	Type	Null	Default

Name	varchar(50)	No	
StockName	varchar(50)	No	
Amount	double	No	
Time	varchar(50)	No	
TradeID	Bigint(20)	No	Primary Key

Table 8. Table structure for table User

Column	Type	Null	Default
Name	varchar(50)	No	Primary Key, Unique
Money	double	No	
Password	varchar(50)	No	

Table 9. Table Structure for table UserStock

Column	Type	Null	Default
Name	varchar(50)	No	Primary Key, Foreign
StockName	varchar(50)	No	Primary Key
Amount	double	No	

Table 7, 8, 9 include the game part of the system, the User would contain the user's login information, the TradeRecord would record every trade from every user, and the UserStock would contain the current stock hold information of the user. For the amount, the minus is for buying and positive is for saling.

Table 10 .Table Structure for table SpredictionBayesian or SpredictionSTS

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Foreign
Date	varchar(50)	No	Primary Key
ClosePriceTarget0	double	No	
ClosePriceTarget1	double	No	
ClosePriceTarget2	double	No	

RelativeDifference	double	No	
Action	varchar(10)	No	

Table 11. Table Structure for table PredictionSVM or PredictionANN

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Foreign
Date	varchar(50)	No	Primary Key
ClosePriceTarget	double	No	
RelativeDifference	double	No	
Action	varchar(10)	No	

Table 12. Table Structure for table PredictionMACD

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Foreign
Date	varchar(50)	No	Primary Key
EMAS(12)	double	No	
EMAL(26)	double	No	
DEA	double	No	
DIF	double	No	
Action	int(11)	No	

Table 13. Table Structure for table LPredictionBayesian or LPrediction STS

Column	Type	Null	Default
Symbol	varchar(20)	No	Primary Key, Foreign
Position	Int(11)	No	
ClosePrice	double	No	

These are the tables that contain the prediction result, for short term prediction using Bayesian and STS, the ClosePriceTarget0 is the prediction close price of that day, we use that to calculate the relative difference. The ClosePriceTarget1 and 2 are the prediction value for the next day and the day after next day. The Action would be the string of "Buy", "Sell" or "Hold". In the table of PredictionSVM or PredictionANN, the other is similar, the ClosePriceTarget is the prediction close price of the day. For PredictionMACD, it includes several internal parameter that is useful for graph drawing and the action is an int. 0 means no action, 1 for "buy", 2 for "strong buy", 3 for "sell", 4 for "strong sell". The Table 13 is a little different from before, the position is that the 10 days before today concern the posion of 0, so next ten days would be 1, next 20 days would be two. And 20 days to 10 days ago would consider as -1. The ClosePrice is a historical data for the past, and prediction value for the future.

Table 14. Table Structure for table AIANN, AIBay, AISTS, and AISVM

Column	**Type**	**Null**	**Default**
Date	varchar(50)	No	Primary Key, Foreign
Total	double	No	

These are the AI table, they get the date from the prediction table, and the total is the total money of that date when following the recommendation of the method.

9.2 Result of Gathering Data

9.2.1 Real-time data

To run the real-time data collecting program automatically, we set up a Timer --- Class realtimeStockTimer. This Timer would trigger the specific codes --- Class RealtimeStock --- to run every 5 seconds. The codes could retrieve the real-time stock price and volume through Yahoo Finance API and store them into the table we created above. All the real-time data is stored into one table StockRealtime. The implement result is shown below.

Symbol	Time	Price	Volume
GOOG	2015-04-26_18-01-39	565.06	4919031
YHOO	2015-04-26_18-01-39	44.52	11281077
AAPL	2015-04-26_18-01-39	130.28	44525905
FB	2015-04-26_18-01-39	81.53	29660356
MSFT	2015-04-26_18-01-39	47.87	130933665
AMZN	2015-04-26_18-01-39	445.1	17176904
SNE	2015-04-26_18-01-39	31.34	3291552
WMT	2015-04-26_18-01-39	79.84	6867845
CAJ	2015-04-26_18-01-39	37.58	163884
TWTR	2015-04-26_18-01-39	50.82	14896860

Figure 7. Collected Real-Time Data

Besides, because the stock market would not open in weekend and some specific dates, the program would not be run in those time. Only the meaningful(activate) data would be stored into the database.

9.2.2 Historical data

To get the historical data, we created a class name HistoricalStock. This class has fields including the number of years before today of the historical prices, and a list of historical data in these years for each stock. Besides, we implemented a method name collectData() to extract the data for time, open price, high price, low price, close price and volume of every specific day in the past. After all these data were extracted from Yahoo API, we then store them into the "StockHistory" table in our database.

Because the "Symbol" and "Date" table has been set as the primary keys of the StockHistory table, if the collectData() is executed multiple times during the same day, duplicate data will not be added to this table repeatedly. On the other hand, when collectData() is executed in a different date, all new data corresponding to new days will be added into this table, thus keeping this table up-dated.

Symbol	Date ▲ 1	OpenPrice	ClosePrice	DaysLow	DaysHigh	Volume	CloseDifference
SNE	2015-04-24	31.45	31.34	31.18	31.53	3291100	0.62
AAPL	2015-04-24	130.49001	130.28	129.23	130.63	44103400	0.61
CAJ	2015-04-24	37.66	37.58	37.5	37.68	163900	0.25
MSFT	2015-04-24	45.66	47.87	45.65	48.14	127995200	4.53
TWTR	2015-04-24	51.85	50.82	50.24	51.96	14633400	-0.59
AMZN	2015-04-24	439	445.10001	439	452.64999	17031400	55.11
YHOO	2015-04-24	43.73	44.52	43.69	44.71	11267500	0.82
FB	2015-04-24	82.77	81.53	81.48	82.94	29576500	-0.88
GOOG	2015-04-24	566.09998	565.06	557.25	571.14001	4911800	18.06
WMT	2015-04-24	79.38	79.84	79.24	80.93	6738500	0.66
YHOO	2015-04-23	43.92	43.7	43.58	44.06	14274900	-0.28
TWTR	2015-04-23	51.87	51.41	51.22	52.2	11413500	-0.32
AMZN	2015-04-23	390.20999	389.98999	386.14999	391.88	6640600	0.19
GOOG	2015-04-23	541	547	540.22998	550.96002	3918000	7.64
AAPL	2015-04-23	128.3	129.67	128.14	130.42	45412000	1.05
FB	2015-04-23	84.1	82.41	82.41	85.59	73512600	-2.22

Figure 8. Collected Historical Data

9.2.3 User stock

name	stockName ▼ 1	amount
DongyangY	AAPL	300
mrhohn	AAPL	0
weiran	AAPL	0
mrhohn	AMZN	0
weiran	AMZN	0
DongyangY	AMZN	0
weiran	CAJ	0
mrhohn	CAJ	0
DongyangY	CAJ	0

Figure 9. UserStock Data

9.2.4 Short term Bayesian/STS prediction data

Symbol	Date ▲ 1	ClosePriceTarget0	ClosePriceTarget1	ClosePriceTarget2	RelativeDifference	Action
GOOG	2014-05-12	515.64	521.94	522.21	0.6	SELL
CAJ	2014-05-12	31.82	32.04	32.05	0.54	SELL
FB	2014-05-12	57.46	58.47	58.49	0.39	SELL
MSFT	2014-05-12	39.51	39.72	39.73	0.07	SELL
SNE	2014-05-12	17.46	17.45	17.45	0.45	HOLD
AAPL	2014-05-12	588.63	590.31	590.28	0.53	SELL
WMT	2014-05-12	78.77	78.95	78.97	0.55	SELL
TWTR	2014-05-12	32.21	32.92	32.93	0.51	SELL
YHOO	2014-05-12	34.25	34.29	34.27	1.45	SELL
AMZN	2014-05-12	292.69	297	297.1	0.15	SELL
YHOO	2014-05-13	34.29	34.32	34.31	0.47	SELL
GOOG	2014-05-13	521.94	527.17	527.56	1.51	SELL
CAJ	2014-05-13	32.04	32.19	32.21	0.78	SELL
MSFT	2014-05-13	39.72	40.04	40.06	0.63	SELL
FB	2014-05-13	58.47	59.1	59.14	2.28	SELL

Figure 10. SPredictionBayesian/ SPredictionSTS Table

9.2.5 SVM/ANN prediction data

Symbol	Date ▲ 1	ClosePriceTarget	RelativeDifference	Action
GOOG	2014-05-12	514.02	3	SELL
CAJ	2014-05-12	31.96	1.02	SELL
FB	2014-05-12	57.69	3.58	BUY
MSFT	2014-05-12	39.87	0.25	BUY
SNE	2014-05-12	17.4	0.29	BUY
AAPL	2014-05-12	587.86	0.84	BUY
WMT	2014-05-12	78.8	0.44	SELL
TWTR	2014-05-12	31.94	5.89	SELL
YHOO	2014-05-12	34.32	0.38	BUY
AMZN	2014-05-12	294.66	2.71	BUY
YHOO	2014-05-13	35.03	1.83	BUY

Figure 11. PredictionSVM / PredictionANN Table

9.2.6 Long term Bayesian/STS prediction data

Symbol	Position ▼ 1	ClosePrice
GOOG	2	543.86
WMT	2	80.72
CAJ	2	36.19
FB	2	82
AAPL	2	126.53
YHOO	2	44.52
TWTR	2	50.27
SNE	2	29.26
MSFT	2	42.33
AMZN	2	382.67
YHOO	1	44.51
MSFT	1	42.34
AMZN	1	382.18

Figure 12. LPredictionBayesian / LPredictionSTS Table

9.2.7 EMACD prediction data

Symbol	Date	EMAS	EMAL	DEA	DIF	Action ▼ 1
FB	2015-01-30	76.76	76.94	-0.17	-0.18	4
YHOO	2015-01-28	48.19	48.71	-0.43	-0.51	4
TWTR	2014-12-08	39.16	40.83	-1.65	-1.67	4
WMT	2015-04-02	81.86	82.61	-0.72	-0.75	4
GOOG	2014-11-18	543.32	546.71	-3.17	-3.39	4
CAJ	2014-09-09	32.92	32.97	-0.03	-0.05	4
GOOG	2015-01-06	520.54	524.88	-3.75	-4.34	4
MSFT	2015-03-06	43.32	43.62	-0.29	-0.3	4
AMZN	2014-10-24	308.02	314.11	-5.24	-6.09	4
CAJ	2014-09-26	32.93	32.95	-0.01	-0.02	4
FB	2015-01-27	76.8	77	-0.16	-0.2	4
CAJ	2014-09-17	32.9	32.94	-0.04	-0.04	4
MSFT	2015-03-27	41.98	42.47	-0.43	-0.48	4

Figure 13. PredictionMACD Table

10 Prediction Strategy

10.1 Short Term

In our project, short term prediction strategy are used to predict the stock price in the following two days. We use 4 methods to do the short term prediction including the Bayesian Prediction Method, the Stochastic Oscillator (STS) Method, the Support Vector Regression (SVR) Method and the Artificial Neural Network (ANN) Method.

10.1.1 Bayesian prediction

a. Bays' Theorem

In probability theory and statistics, Bayes' theorem, which relates current probability to prior probability, is important in the mathematical manipulation of conditional probabilities.

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

in which p(w) represents a prior probability, The effect of the observed data D = {t1, . . . , tN } is expressed through the conditional probability p(D|w).

As an important application of Bayes' Theorem, we can evaluate the uncertainty in w after we have observed D in the form of the posterior probability p(w|D). And the quantity p(w|D) is viewed as a function of parameter vector w if we take likelihood function into consideration.
Then the Bays' theorem in terms of the prior distribution and the likelihood function:

$$p(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})\,\mathrm{d}\mathbf{w}.$$

To fix the parameter w, a widely used estimator is maximum likelihood, in which w is set to the value that maximizes the likelihood function p(D|w). In the machine learning literature, maximizing the likelihood is equivalent to minimizing the error.

b. Curve fitting

The goal in the curve fitting problem is to be able to make predictions based on N training data {x, t}. Assume given the value of x, the corresponding value of t has a Gaussian distribution:

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right)$$

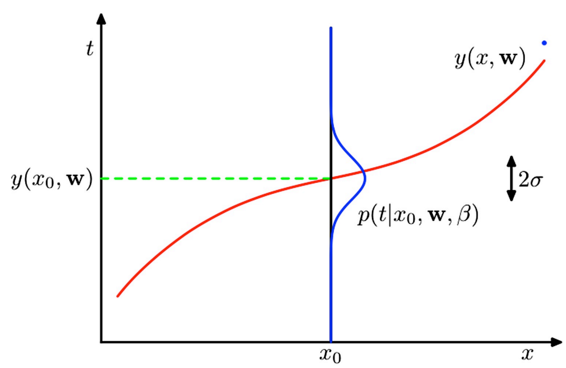


Figure 14. Schematic illustration of a Gaussian conditional distribution and the precision is given by the parameter β.

After using training data {x, t} to determine the values of the unknown parameters w and β by maximum likelihood, we can now make predictions for new values of x.

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha).$$

which can be written as

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}.$$

c. Bayesian Curve Fitting

A Bayesian treatment simply corresponds to a consistent application of the sum and product rules of probability, which allow the predictive distribution to be written in the form [1.1 Bishop]:

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w}) p(\mathbf{w}|\mathbf{x}, \mathbf{t}) \, \mathrm{d}\mathbf{w}.$$

As a curve-fitting example, this posterior distribution is a Gaussian and can be evaluated analytically.

$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$

In this case, we take m(x) as the most likely value for new point t.

$$m(x) \quad = \quad \beta \boldsymbol{\phi}(x)^{\mathrm{T}} \mathbf{S} \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) t_n$$

Here the matrix S is given by

$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x)^{\mathrm{T}}$$

d. Bayesian Stock Prediction

We use the historical data of the stock as the training output data in the Bayesian curve fitting, that is we use continuous 10 days' close price as the training data. We do the Bayesian curve fitting and get the predict close price of the 11th day. And to predict the next day, we use the close price from the 2nd day to the 10th day, and the prediction price of the 11th day as the training data.

10.1.2 Stochastic oscillator (STS)

The stochastic oscillator is another well-known momentum indicator, which is developed by George C. Lane in the late 1950s. It shows the location of the close relative to the high-low range over a set number of periods. According to an interview with Lane, the Stochastic Oscillator "doesn't follow price, it doesn't follow volume or anything like that. It follows the speed or the momentum of price. As a rule, the momentum changes direction before price."

Calculation Method:

$$\%K = 100 \times \frac{(\text{Close} - \text{Lowest Low for the period})}{\text{Highest High for the period} - \text{Lowest Low for the period}}$$

$$\%D = \text{3-period moving average of } \%K$$

where the period is set up for 14 days for most cases, but users can adjust themselves.

BUY/SELL Signal

The first line is the %K which is essentially the raw measure used to formulate the idea of momentum behind the oscillator. In an upward trend the price should be closing near the highs of the trading range and in a downward trend the price should be closing near the lows of the trading range.

The second line is the %D, which is plotted within a range of zero and 100 and signals overbought conditions ("SELL" signal) above 80 and oversold conditions ("BUY" signal) below 20.

Figure 15. STS Application on IBM

Price Prediction

Since D%, so called as Slow Stochastic, is the moving average of k% and it reacts more slowly to price changes, we can predict the next-day price by predicting the next K%.

Prediction_k = the average of latest three differences of D% + latest K%

Prediction_price = $\frac{pred\_k(highest-lowest)}{100} + lowest$

In order to illustrate the accuracy, there is an example in Table 15.

ClosePrice	K%	D%	Prediction Price	Error%
68.8				
71.57				
70.84				
69.8				
72.03				
70.1				
70.88				
68.83				
67.72				
68.74				
69.19				
68.24				
66.97				
67.24	5.33			
64.1	0			
64.89	9.96			
60.39	0	3.82		
60.97	4.98	3.73		
60.01	0	3.73		
60.24	2.11	1.77	60.16	0.13
62.62	28.4	8.88	62.77	0.24
62.72	29.5	15.01	63.06	0.54
59.49	0	15.01	59.91	0.70
56.75	0	14.49	56.98	0.40
56.95	1.74	7.81	56.67	0.49

Table 15. Accuracy is proved by STS prediction

10.1.3 Support vector regression (SVR)

The earliest support vector machine (SVM) is a simple binary classification mathematic model. It describes an optimal decision boundary to separate two categories targeted at the known examples trained. The training examples and test examples are represented as points in a space. The dimension of the space is associated with the number of features of the examples. The decision boundary is a line in 2-dimensional space while it becomes a hyper-plane in higher dimensional space. The basic idea of SVM is always choosing the decision boundary which has the largest distance from the nearest positive examples and negative examples. This can be described as a minimization or maximization problem in the geometry, and this a typical optimization problem. To test a new example, i.e., determine the new example's category, just to determine which side of decision boundary the new point is on.

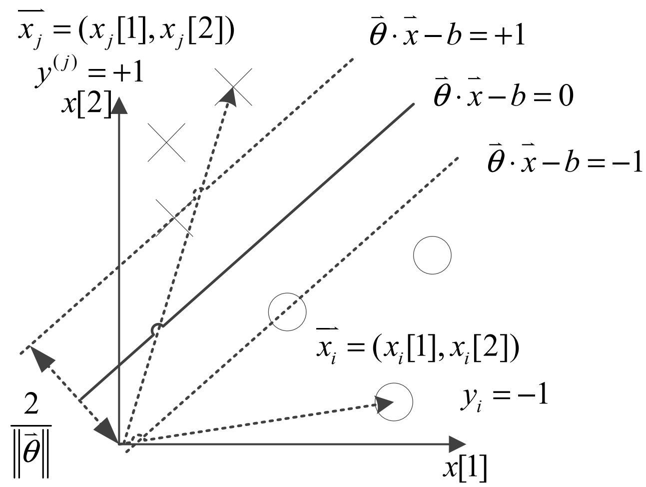


Figure 16. Decision boundary in 2-dimensional space for linear problem

Due to the difficulty of visualization of the hyper-plane in high dimensional space, take the simple 2-D linear problem as shown in Fig. 1 and assume the training examples are linearly separable. In Fig. 16, the cross represents for the positive example, and the cycle represents for the negative example. $\overline{x_j}$ is a 2-D vector, and has $x_j[1]$ and $x_j[2]$ two features. Here, 1 and 2 are index of features and j is the index of examples. y_j is example j's label which is +1 when example j is positive otherwise is -1 when example j is negative. Vector $\vec{\theta}$ is the coefficients of $\vec{x}$, and they have the same dimension. $\vec{\theta} \cdot \vec{x} - b = 0$ is the decision boundary which is a line here. $\frac{2}{\|\vec{\theta}\|}$ is the margin between two category regions. As mentioned before, what SVM do is to find the largest margin with the constraint - no examples fall into the margin. This optimization problem and constraint can be describes as the primal equation shown in Eq. 1.

$$\min_{\theta,b} \frac{1}{2}\|\vec{\theta}\|^2 \quad \text{s.t.} \quad y_i(\vec{\theta} \cdot \overline{x_i} - b) \geq 1, \forall i \tag{1}$$

However, there are several significant limitation for the solution above, thus modifications are made afterward. Soft margin is introduced by Cortes and Vapnik in

1995 [2], which increase the tolerance for some examples in the wrong side just as what regulation do to reduce over-fitting, in order to solve linearly inseparable problem. Another modification is the kernel function. Some functions such as Gaussian are regarded as similarity function to solve non-linear SVM problem. Besides, for computation purpose, the Eq. 1 is transformed to dual form by using lagrangian. Finally, the optimization problem is equal to Eq. 2.

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j K(\overline{x_i}, \overline{x_j})\alpha_i \alpha_j - \sum_{i=1}^{m}\alpha_i \tag{2}$$

$$\text{s.t. } 0 \le \alpha_i \le C, \forall i \text{ and } \sum_{i=1}^{m} y_i \alpha_i = 0$$

In Eq. 2, m is total number of training examples. C indicates the soft margin strategy. $K(\overline{x_i}, \overline{x_j})$ is the kernel function and can be replaced by different non-linear similarity functions or equal to $\overline{x_i} \cdot \overline{x_j}$ for linear. $\overline{\alpha}$ is the lagrange multipliers and it is a m-dimensional vector and becomes the variable needed to find to make the object function on the right minimum. Note that, only $\overline{\alpha}$ is the variable while the others all are known by training examples. b in Eq. 1 is solved by updating in the process of finding $\overline{\alpha}$ while $\vec{\theta}$ is solved by Eq. 3 when $\overline{\alpha}$ is known.

$$\vec{\theta} = \sum_{i=1}^{m} y_i \alpha_i \overline{x_i} \tag{3}$$

Based on the original SVM theory, SVM has been extended to regression problems, i.e., support vector regression (SVR) [3]. In SVR, the number of classes or categories of SVM is no longer limited by 2, but infinite. Thus, it is available to apply SVR in stock prediction area. In our strategy, we use the epsilon-SVR with radial basis function (RBF) Gaussian kernel. In order to improve the accuracy, we determine the parameter C and gamma by grid search method. A training example is composed of a known stock price as its class or category and four previous stock prices as its features. To predict a stock price, we use five examples just before the predict date to train a latest model in order to get rid of huge stock shock. The tool employed is LIBSVM [4)].

10.1.4 Artificial neural network (ANN)

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning algorithms inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs, and are capable of machine learning as well as pattern recognition thanks to their adaptive nature

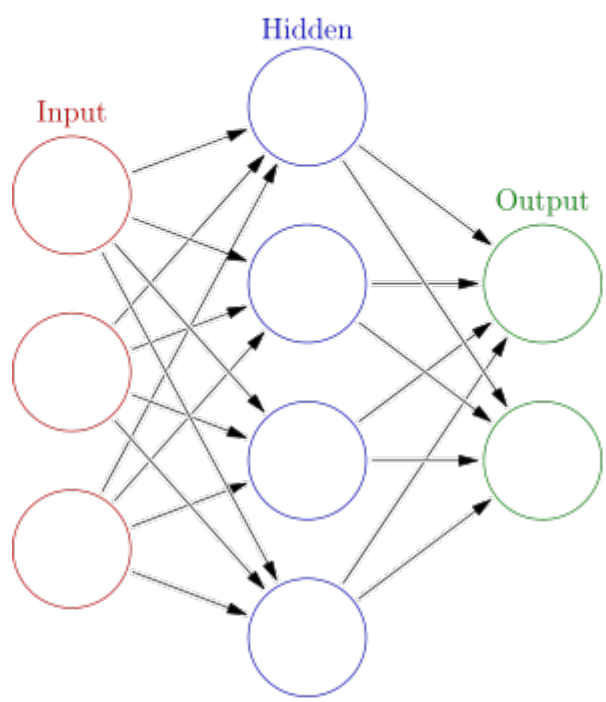


Figure 17. Pictorial ANN

As this Figure 17 indicates, an artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another [5].

There is no single formal definition of what an artificial neural network is. However, a class of statistical models may commonly be called "Neural" if they possess the following characteristics:

1. Consist of sets of adaptive weights, i.e. numerical parameters that are tuned by a learning algorithm, and
2. Are capable of approximating non-linear functions of their inputs.

The adaptive weights are conceptually connection strengths between neurons, which are activated during training and prediction.

Neural networks are similar to biological neural networks in performing functions collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned. The term "neural network" usually refers to models employed in statistics, cognitive psychology and artificial intelligence. Neural network models which emulate the central nervous system are part of theoretical neuroscience and computational neuroscience [5].

Actually it is extremely hard to design a useable and sophisticated Artificial Neuro Algorithm, which also could not be done in a short term of period. Hence we use a free-to-use resource on the internet called Neuroph. Neuroph is lightweight Java neural network framework to develop common neural network architectures. It contains well designed, open source Java library with small number of basic classes which correspond to basic NN concepts. Also has nice GUI neural network editor to quickly create Java neural network components [6].

Since, we in fact using some ANN package developed by others, in the following part we would focus on how to implement the Stock Price Prediction based on ANN and how to make it more efficient. Figure 18 shows the basic concepts in Neuroph framework.

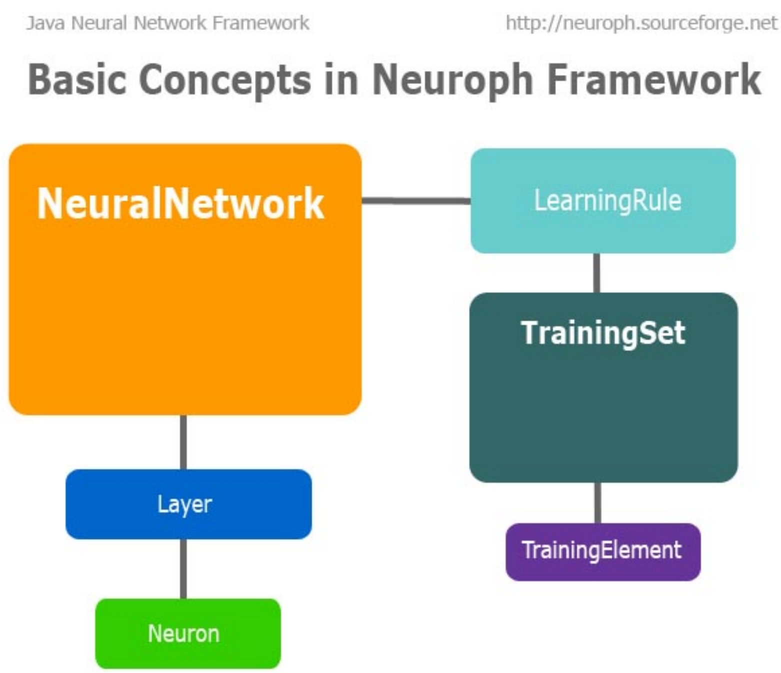


Figure 18. Neuroph Framework

To think about Stock Price prediction, we first need to learn something about time-series prediction. Neural networks have been applied to time-series prediction for many years from forecasting stock prices and sunspot activity to predicting the growth of tree rings. Also, Time series prediction plays a big role in economics. In essence all forms of time series prediction are fundamentally the same. Namely given data x = x(t) which varies as a function of time t, it should be possible to learn the function that maps x(t+1) = x(t). Consider a single variable x which varies with time, one common approach is to sample x at regular time intervals to yield a series of observations x(t-2), x(t-1), x(t) and so on. We can then take such observations and present them as the input vector to the network and use observation x(t+1) as the target value. By stepping along the time axis one sample at a time we can form the training set for the problem. In other words 'given the last three samples what is the next value?'. Once we have trained the network we should then be able to present a new vector x'(t-2), x'(t-1), x'(t) vector and predict x'(t+1). This is termed one step ahead prediction. We could also use the predicted value as part of the next input vector then depending on how many predicted values we allow to be fed back into the network we then have what is termed multi-step ahead prediction. Unfortunately the latter approach tends to diverge rapidly from the true pattern due to the accumulation of errors [7]. Hence, we choose to predict only the price of one or two days later to prevent the accumulation of errors.

For the stock price prediction, we will be using a standard multilayer back-propogation network often called a multilayer perceptron (MLP). Multilayer perceptron (MLP) is a feedforward neural network with one or more layers between input and output layer. Feedforward means that data flows in one direction from input to output layer (forward). This type of network is trained with the backpropagation learning algorithm. MLPs are widely used for pattern classification, recognition, prediction and approximation. Multilayer Perceptron can solve problems which are not linearly separable.

The first thing we need to do is to normalize the input stock data to the range [-1…1] which fits our problem data nicely. This could be achieved by following two steps:

1. To find the max value of DAX : maxPrice = max(days[k], k =0, days.length-1))
2. To calculate normalized values: normPrice[i] = (days[i][3] / maxPrice) * 0.8 + 0.1, where 0.8 and 0.1 will be used to avoid the very small (0.0...) and very big (0.9999) values.

The second aspect needs to be considered is the topology of the network: input, output and hidden layers, nodes. It is relatively easy to make a neural network learn a problem perfectly. However, we don't just want it to learn a given problem, we want it to be able to generalize the solution to data it has never seen before. Learning the problem perfectly but not being able to predict on data it has never been shown before is called over-fitting. The number of nodes is directly related to this balancing act between learning the problem but not generalizing, and conversely not even learning the problem. This is why the number and topology of the nodes should be considered. To avoid this situation, we basically employed the Baum-Haussler rule. This states that:

$$N_{hidden} \leq \frac{N_{train} E_{tolerance}}{N_{input} N_{output}}$$

Where N_{hidden} is the number of hidden nodes, N_{train} is the number of training patterns, $E_{tolerance}$ is the error we desire of the network, N_{input} and N_{output} are the number of input and output nodes respectively. This rule of thumb generally ensures that the network generalizes rather than memorizes the problem.

Also, a recommendation for topology is to have 2n+1 nodes for hidden-layer, where n is the number of the input nodes. The output layer has only one node in this case (for one output price). Finally, the parameters is set as follow: maxIteration = 10000, learningRate = 0.7, maxerror = 0.0001, inputNum = 4, outputNum = 1, hiddenNum = 9.

The sample of prediction including the output and the error is shown below as Figure 19, 20, 21.

Figure 19. Actual price and the predicted price of MSFT

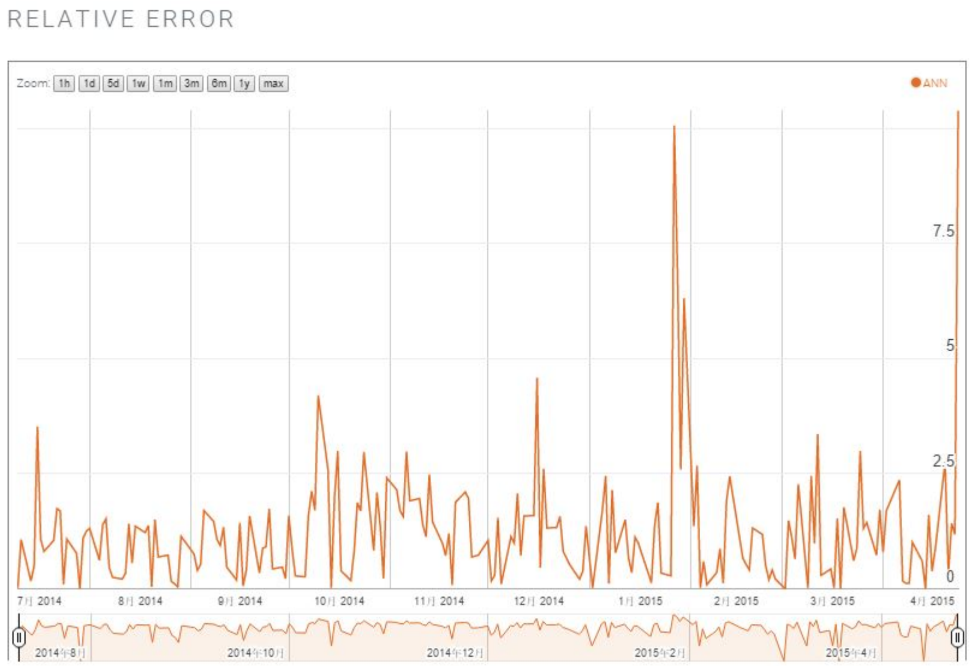


Figure 20. Absolute predicted error of MSFT(%)

AVERAGE RELATIVE ERROR

Symbol	Strategy	Error(%)
MSFT	Bayesian	0.62
MSFT	STS	1.04
MSFT	SVM	1.3
MSFT	ANN	1.17

Figure 21. Error table of predictions on MSFT

10.1.5 Short term strategies comparison

	Bayesian (%)	STS (%)	SVR (%)	ANN (%)
GOOG	0.64	1.04	1.58	1.67
YHOO	0.9	1.43	1.87	1.68
AAPL	3.29	6.14	8.81	7.9
FB	0.77	1.3	1.69	1.41
MSFT	0.62	1.04	1.32	1.17
AMZN	0.83	1.41	2.12	1.48
SNE	0.99	1.65	2.08	1.67
WMT	0.43	0.74	0.92	0.82
CAJ	0.45	0.73	0.92	0.78
TWTR	1.24	2.17	2.83	2.34
Average	1.02	1.77	2.41	2.09

Table 16. Average Relative Error of 4 strategies for 10 companies

As a shown in Table , the methods based on mathematical equations (Bayesian and STS) are better than machine learning algorithms (SVR and ANN), probably because the training examples are limited and we cannot feed all possible situations to the model.

10.2 Long Term

In our project, long term prediction strategy are used to predict the stock price trend in the following twenty days. We use 3 methods to do the long term prediction including the Bayesian Prediction Method, the Stochastic Oscillator (STS) Method, and the Moving Average Convergence Divergence (MACD) Method.

10.2.1 Bayesian and STS method

For these two kind of methods, when doing the long term prediction, the algorithms and methods are totally same with the short term prediction method. The only difference is that when doing the short term prediction, we use several continuously days as the training data. Now we do not use the daily close price, instead, we use the average close price for every ten days, and use them as the training data to calculate next two value. The two prediction value itself is meaningless, but they could tell the trend of the stock price for the next 20 days.

10.2.2 Moving average convergence divergence (MACD)

MACD, short for moving average convergence/divergence, is a trading indicator used in technical analysis of stock prices, created by Gerald Appel in the late 1970s. This index tries to predict market tendency changes before they happen. It is supposed to reveal changes in the strength, direction, momentum, and duration of a trend in a stock's price.

EMA, so called as exponential moving average, is calculated by taking an exponentially weighted for past data. The weighting for each older datum decreases exponentially, never reaching zero.

$$\mathrm{EMA}_{\mathrm{today}} = \mathrm{EMA}_{\mathrm{yesterday}} + \alpha \times (\mathrm{price}_{\mathrm{today}} - \mathrm{EMA}_{\mathrm{yesterday}})$$

where $\alpha=2/(n+1)$,   n is the number of days.

The MACD method is a collection consists of the 26-day EMA ("slow" line), 12-day EMA ("fast" line).

Define: DIF = (12-day EMA) – (26-day EMA)
DEA = 9-day EMA of DIF ("signal" line)

Signal Line Crossovers

Signal line crossovers are the most common MACD signals. The signal line is a 9-day EMA of the DIF, which trails the DIF and makes it easier to spot DIF turns.

A BUY signal occurs when the DIF turns up and crosses above the signal line.
A SELL signal occurs when the DIF turns down and crosses below the signal line. Crossovers can last a few days or a few weeks, it all depends on the strength of the move.

Centerline Crossovers

Centerline crossovers are the next most common MACD signals. A BUY signal occurs when the DIF Line moves above the zero line and turn positive, which indicates the 12-day EMA of the underlying security moves above the 26-day EMA. A SELL signal occurs when the DIF moves below the zero line and turn negative, which indicates the 12-day EMA moves below the 26-day EMA.

Figure 22. Signal line crossover example.

As we can observe in the example, only depend on centerline crossover or signal line crossover may cause inaccurate result.

So, our strategy is

➔ DIF turn up across DEA – BUY

➔ DIF turn down across DEA – SELL

➔ DIF turn up across DEA and DIF > 0 DEA > 0 – Strongly BUY

➔ DIF turn down across DEA and DIF < 0 DEA < 0 – Strongly SELL

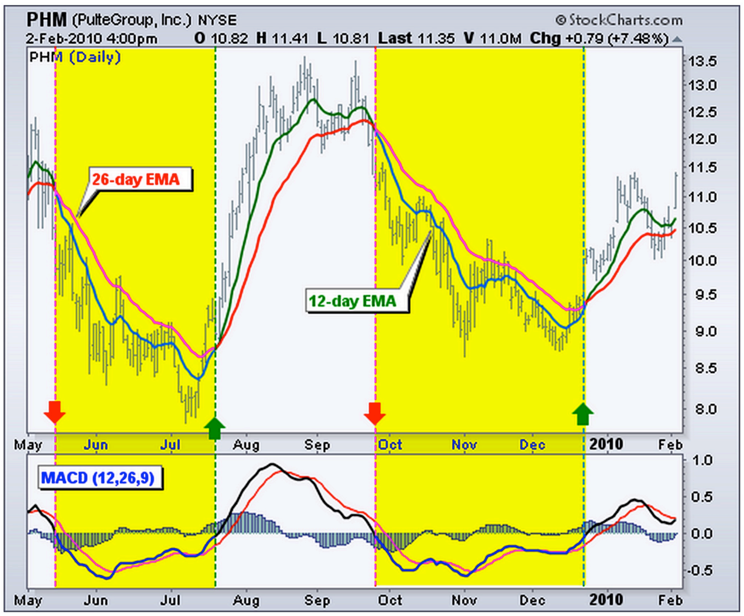


Figure 23. Centerline crossover example

11 Special Features – Virtual Stock Exchange Game

11.1 Introduction

In order to enhance the enjoyment of the Stock Forecaster System, we developed a virtual stock game and put it on the website. The logged on user could simply click on the VSE tab or click on PLAY YOUR STOCK button on the homepage to enter the virtual stock game page.

Within the game, we mimic a stock market where clients could purchase their own stock holdings and keep tracking what is going on. 100,000 dollars balance is initially allocated to every user. Due to the time limited, we only provide the prediction on ten stocks (GOOG, YHOO, AAPL, FB, MSFT, AMZN, SNE, WMT, CAJ, TWTR). Also, the user is allowed to do the operations with these ten stocks, such as buy, sell and checking their own holdings. Their exchange records would be displayed in the bottom of the VSE page.

One more interesting feature of the virtual stock exchange game is that we provide the action suggestion to the ten stocks mentioned above. These suggestions are calculated based on the four predict methods (STS, Bayesian, SVM, and ANN) we applied on the stock price prediction. The details of how these suggestion come out would be introduced in following section. The players could simply follow these suggestions to decide whether increase or decrease the holdings.

The most attractive part of the game is that there are four simulate Players along with you. We call them AIs because they are all controlled by the computer while doing the operation. As the normal user, the four AIs are allocated 100,000 dollars initially. In order to make sure the user could clearly see the profits gained by these four AIs, we plot their balance on the top of the VSE page. The details of AI would be also introduced later.

11.2 Procedure and action

The procedure of the whole virtual stock exchange game is simple. Just imaging the user wakes up in the early morning of a trading day. What he/she should do is to take a look at the suggestion actions provided by the system. After that the user should decide to buy, sell or hold the stocks.

To simplify the development, we do some assumptions with the stock market. First, we assume that the closing price today is the same as the opening price of the next trading day. Second, no handling charge would occur when the user do any operations with the stocks. Third, the smallest time unit of trading operation is one day, which means the system does not provide the operation in the middle of a day. Fourth, the system do not consider the potential delay of operation, hence the user could buy or sell the stock holding immediately. After announcing above several assumptions, we believe the procedure and the goal of the virtual stock exchange game is quite clear and straight forward --- the client needs only to buy, sell or hold the stocks today and see how his/her balance changed tomorrow.

Let’s come back to discuss the details about how those suggesting actions come out. For each prediction method, we give out a suggestion about buy, sell or hold the stock. These suggestions are basically calculated directly by comparing yesterday’s closing price and today’s predicted price of the specific stock. If today’s predicted price is high than yesterday’s closing price, it means the stock price would increase today (keep in mind the user just wake up in the early morning and the stock is not open yet). Hence the system would give out a buy suggestion. Otherwise, a sell suggestion would be displayed. For the hold action, the system would tell the user to hold only if the price is remain in the same price (it is apparently not make much sense, we will try to make the hold occurred when the prices are in the same level or in other meaningful moments).

Eventually, we display all the suggestion in the page of VSE, without doing any comparison and filter. Hence the client might see different suggestions on the same stock like sell versus buy, hold versus sell and so on. Of course, the client could click on the forecast tab to take a look at the performance of each prediction method and then decide which suggestion they should follow. Above is basically all the things about the normal part of the virtual stock exchange game.

11.3 AI strategy

Finally we come to the most interesting part of the virtual stock exchange game --- AI strategy. In fact, the AI we employed is not such smart. Due to the time limited, we have not combined all the prediction methods and all the useful information extracted from the original stock market (we might finish the SUPER AI after, which considers all the suggestion given by the prediction methods).

Now let's take a look at the simple but useful AI we developed for the game. Considering we have totally ten stocks and have all of the suggestions about them for more than one year. For the align purpose, we choose to begin placing the AI in the game at the date when all the prediction methods have their own prediction and suggestion. As what is mentioned above, the AI is given 100,000 dollars initially, and start do the operation according to the suggestion provided by each prediction methods. One important issue we could think about is what if the AI is running out of money and could not do any operation with the stock? To avoid this situation, we simply divide the whole balance into 10 parts, each for one stock. By doing so, we make sure that each stock's profit or deficit would not interrupt other stocks too much. For the buy action, we take 20% of the remaining balance for the specific stock (remember we divide the balance into ten parts now) and use this virtual money to buy in as much as we could. Consider the single unit price of a stock, we could not always spend all of the money on it. For this situation, we make a round down of the maximum possible buying amount (moneyUsedToBuy / unitPrice) to make sure there must be some remainder after the buy operation. Similarly, for the sell action, we take out 20% of the existing holding for the specific stocks and sell it (consider the increase trend recently, we believe it is not wise to sell too much). Finally, for the hold action, the AIs simply do not do anything. Below is the chart about how the balance of AIs changed in the past several months.

AI HISTORY

Figure 24. AI Prediction

Notice that although the error rate, which could be found in the former section, of Bayesian prediction methods is lowest, the profit of Bayesian AI is not the highest. It is the STS AI earn most money in the past few months.

12 User Interface Design and Implementation

12.1 Web Interface

12.1.1 Home

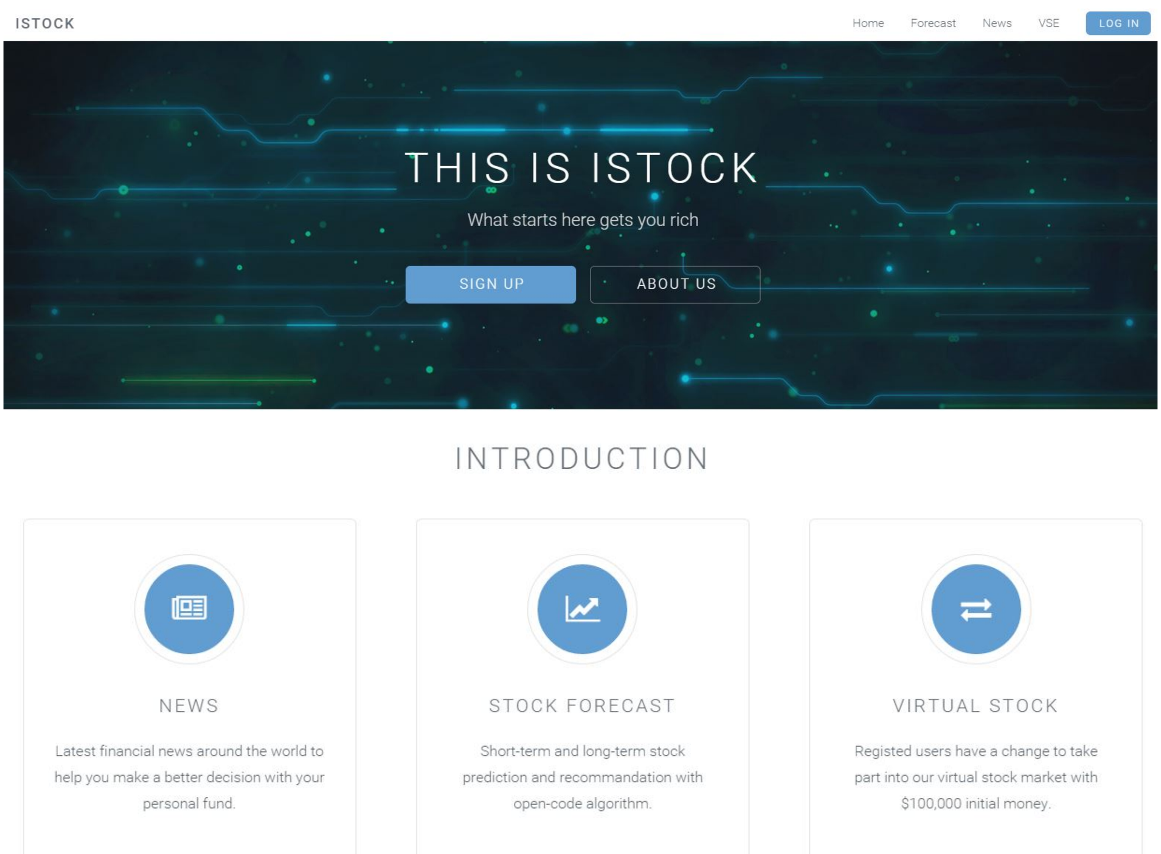


Figure 25. The Home Page of the Website

This is the hompage of the website, the news would lead the user to the news page which include sevral useful information and the specific query. The Stock Forecast would lead the user to the forecast page. The virtual stock would lead the user to check the information about the Virtual Stock Exchange market, and if you are a registed one, you can also take part in the game to buy or sell stocks.

12.1.2 News

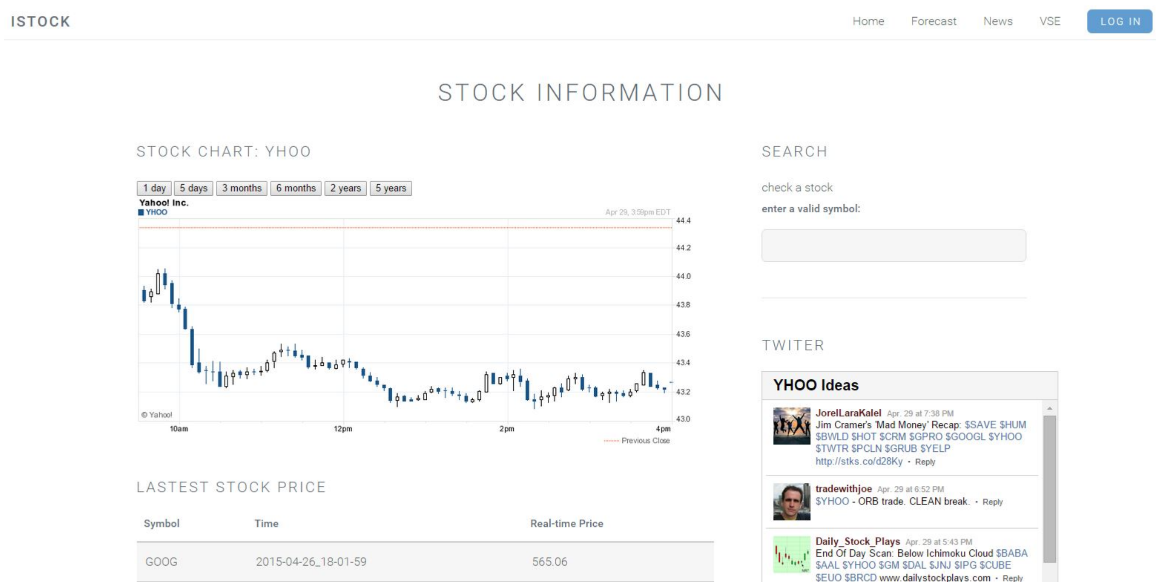


Figure 26. News Page

The News page including a lot of useful information. There is a search bar on the right of the webpage, and that is where the user can type the Sticker of the company, then it would have the Stock chart for this company, and also the query of Twitter and News for the same company like the Figure 26 shows below.

Here we put the specific query in this page and the result of those query is :

LASTEST STOCK PRICE

Symbol	Time	Real-time Price
GOOG	2015-04-26_18-01-59	565.06
YHOO	2015-04-26_18-01-59	44.52
AAPL	2015-04-26_18-01-59	130.28
FB	2015-04-26_18-01-59	81.53
MSFT	2015-04-26_18-01-59	47.87
AMZN	2015-04-26_18-01-59	445.1
SNE	2015-04-26_18-01-59	31.34

Figure 27. Specific Query 1 -- Latest Stock Price

HIGHEST PRICE IN LAST TEN DAYS

Symbol	Date	Days High Price
GOOG	2015-04-24	571.14001
YHOO	2015-04-16	46.13
AAPL	2015-04-24	130.63
FB	2015-04-23	85.59
MSFT	2015-04-24	48.14
AMZN	2015-04-24	452.64999

Figure 28. Specific Query 2 -- Highest price in last ten days

AVERAGE STOCK PRICE IN THE LATEST ONE YEAR

Symbol	Close Price
GOOG	549.83
YHOO	41.84
AAPL	166.89
FB	73.97
MSFT	44.07
AMZN	331.95

Figure 29. Specific Query 3 -- Average Stock Price in the latest One Year

LOWEST STOCK PRICE IN THE LATEST ONE YEAR

Symbol	Days Low Price
GOOG	487.56
YHOO	32.93
AAPL	89.65
FB	54.66
MSFT	38.51
AMZN	284

Figure 30. Specific Query 4 -- Lowest tock Price in the letest One Year

COMPANIES WHOSE AVERAGE PRICE LOWER THAN LOWEST PRICE OF GOOG IN THE LASTEST ONE YEAR

Id	Symbol	Full Name
2	YHOO	Yahoo!
3	AAPL	Apple Inc.
4	FB	Facebook, Inc.
5	MSFT	Microsoft Corporation
6	AMZN	Amazon.com, Inc.

Figure 31. Specific Query 5 -- Companies whose average prove lower than lowest price of Google in the latest one year

12.1.3 Forcaster

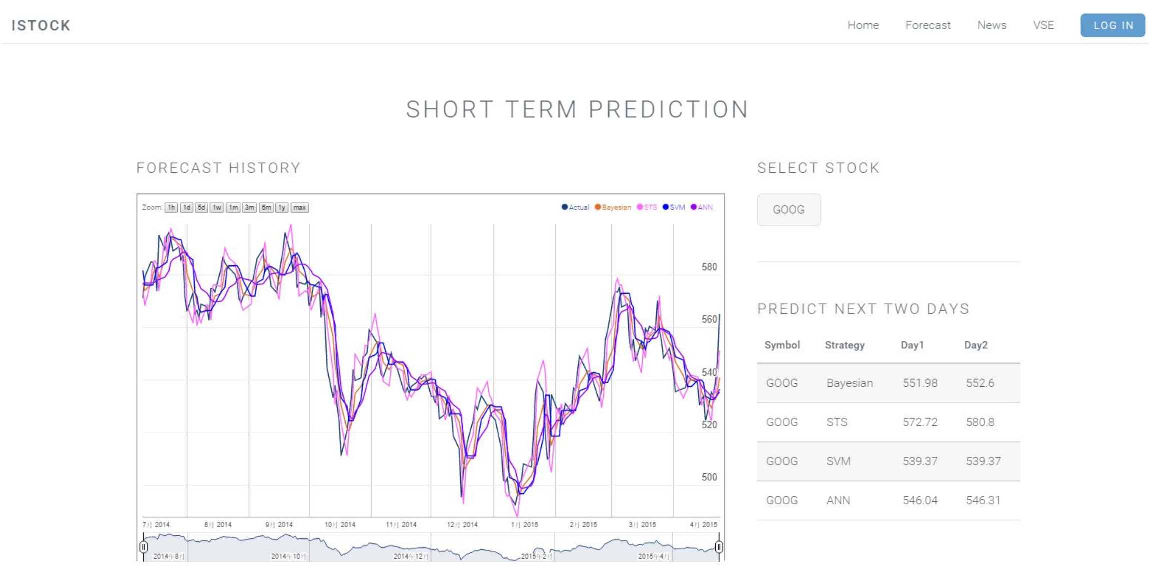


Figure 32. The Forcaster Page

This is the forcaster page of the website, the Select Stock is the place you choose stocks from ten companies. The Forecast History is the graph that show the short term prediction historical data. The graph can be changed by clicking the zoom bottoms at the left of the upper part of the graph, you can zoom the graph from 1 day to 1 year. The Prediction Nex Two Days contain all the short term prediction result.

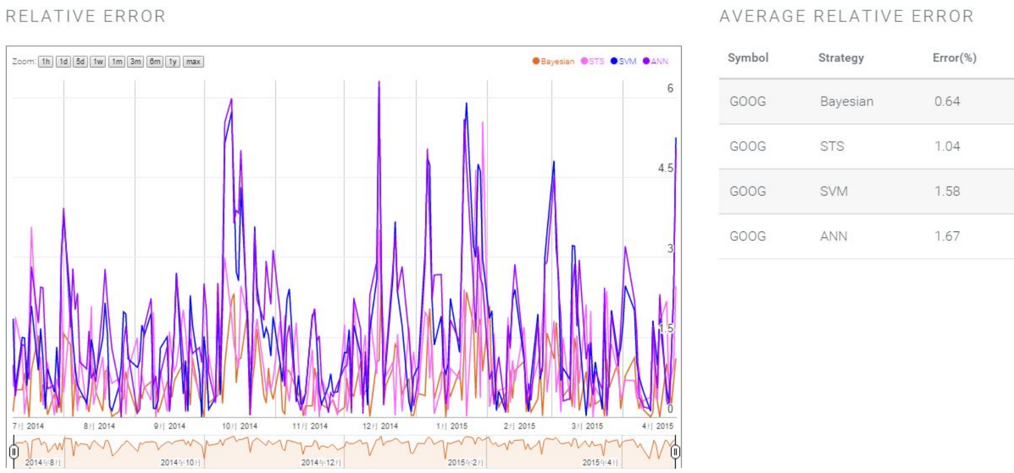


Figure 33. Relative Error

Under the forecast history graph is the Relative Error graph, the graph can also be zoomed. It also show the average relative error to all the short term methods.

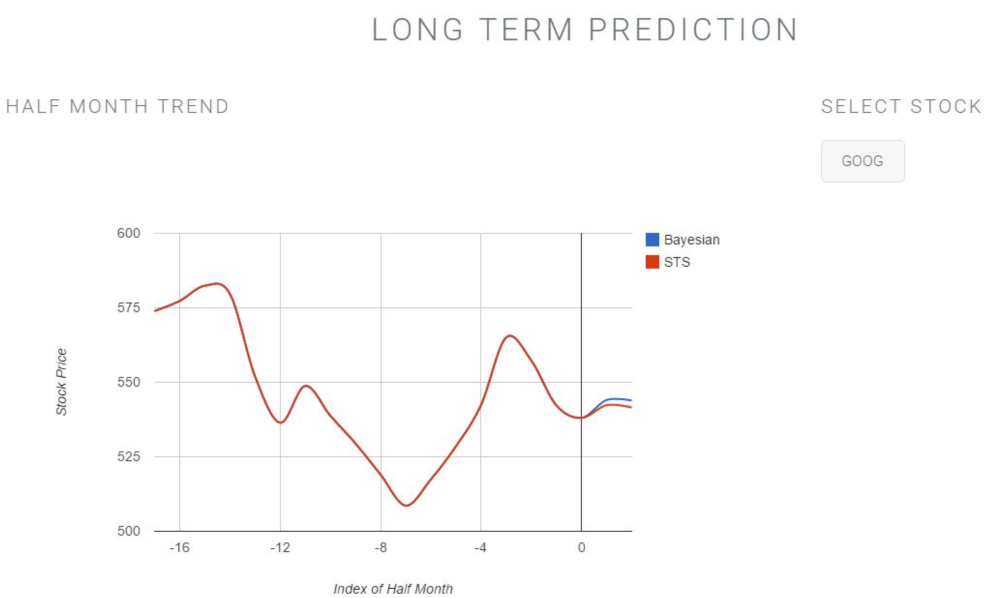
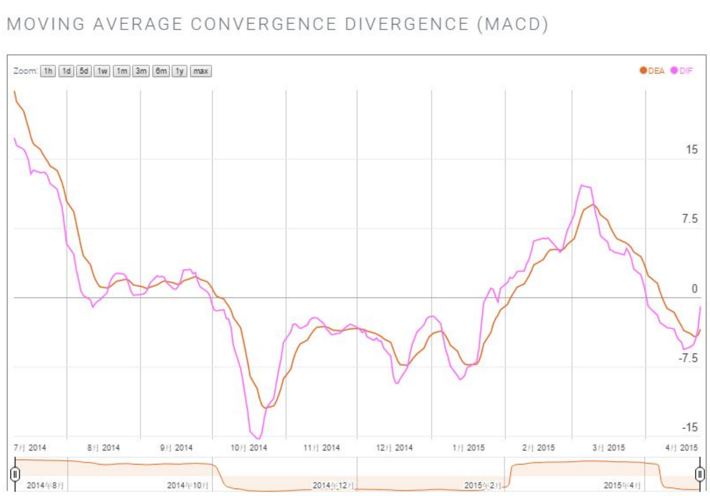


Figure 34. Long Term Prediction of Bayesian/STS

This is the long term prediction byusing the Bayesian/ STS methods. It shows the trend for the next two 10-days.

Figure 35. The EMA Graph



ACTION RECOMMENDATION

Symbol	Date	Action
GOOG	2015-04-23	BUY
GOOG	2015-03-11	SELL
GOOG	2015-02-25	Strongly BUY
GOOG	2015-02-24	SELL
GOOG	2015-01-20	BUY
GOOG	2015-01-06	Strongly SELL
GOOG	2014-12-23	BUY
GOOG	2014-12-02	Strongly SELL
GOOG	2014-11-25	BUY

Figure 36. The MACD Graph and the special action point in the historical data

EMA and MACD graph is for getting the MACD long term presentation. Secotion 10.2.2 has details of the MACD prediction method. We also list the action point in the historical data.

12.1.4 Virtual stock exchange (VSE)

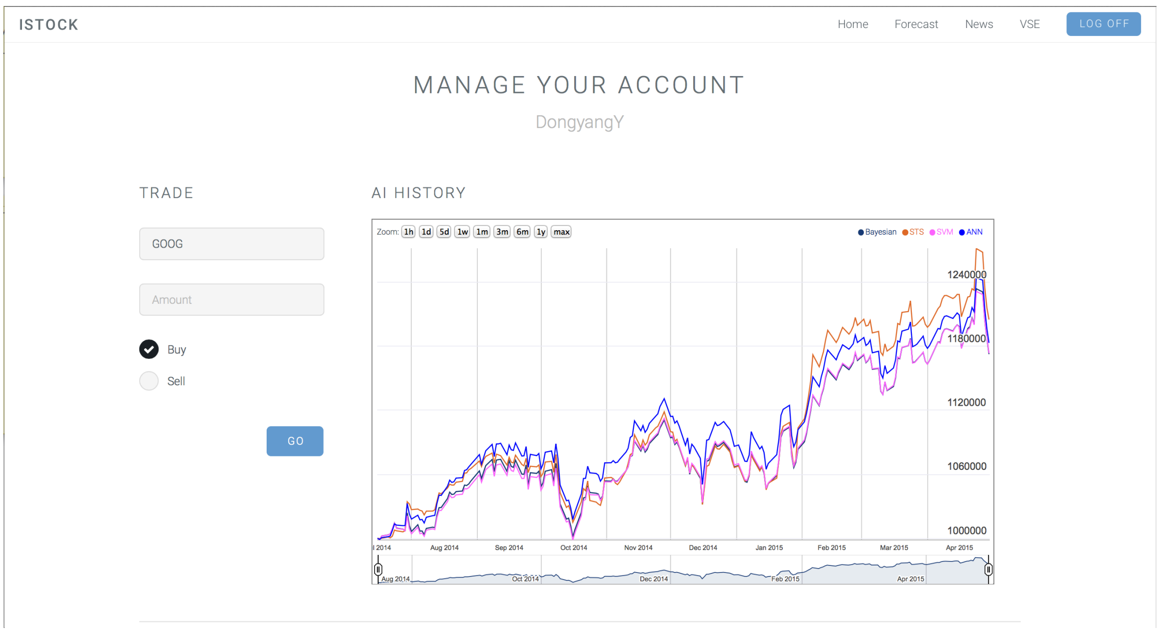


Figure 7. VSE Page

The VSE page contain a graph with AI history of all 4 short term prediction methods. The registed user can use the left part to buy or sell stocks in the virtual stock market. The user can also check the balance(current total value), the holdings and the record of every exchange in the VSE.

BALANCE

User	Balance	Total
DongyangY	955834	997275

HOLDINGS

Symbol	Holdings	STS Suggestion	Bayesian Suggestion	SVM Suggestion	ANN Suggestion
AAPL	300	SELL	BUY	BUY	BUY
AMZN	0	SELL	BUY	BUY	BUY
CAJ	0	BUY	BUY	BUY	BUY
FB	0	BUY	BUY	BUY	BUY
GOOG	0	HOLD	BUY	BUY	BUY
MSFT	0	SELL	SELL	SELL	SELL
SNE	0	HOLD	BUY	BUY	BUY
TWTR	100	BUY	BUY	BUY	SELL
WMT	0	HOLD	BUY	BUY	BUY
YHOO	0	HOLD	BUY	BUY	BUY

RECORD

NEGATIVE VALUE MEANS BUY; POSITIVE MEANS SELL

Symbol	Amount	Time
AAPL	-300	2015-04-29 16-13-54
TWTR	-100	2015-04-29 21-26-25

Figure 38. Balance Holding and Record of VSE

12.1.5 Log in

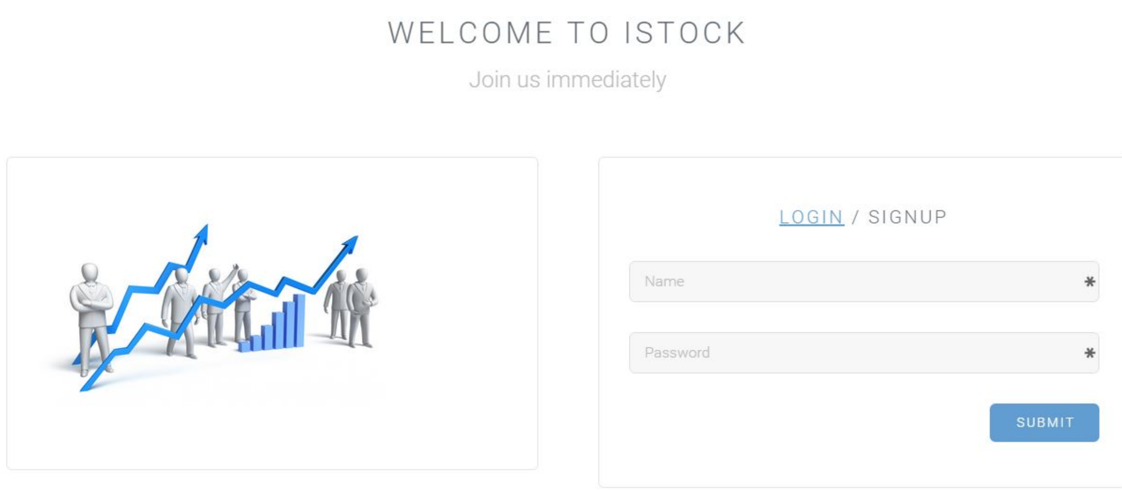


Figure 39. The log in interface of the website

12.2 Android Interface

The Android application is a simplify version of our website. It has no news information nor the VSE syste. But user can easily use it to check all the graphs for prediction and choose the company inside the 10 company list.

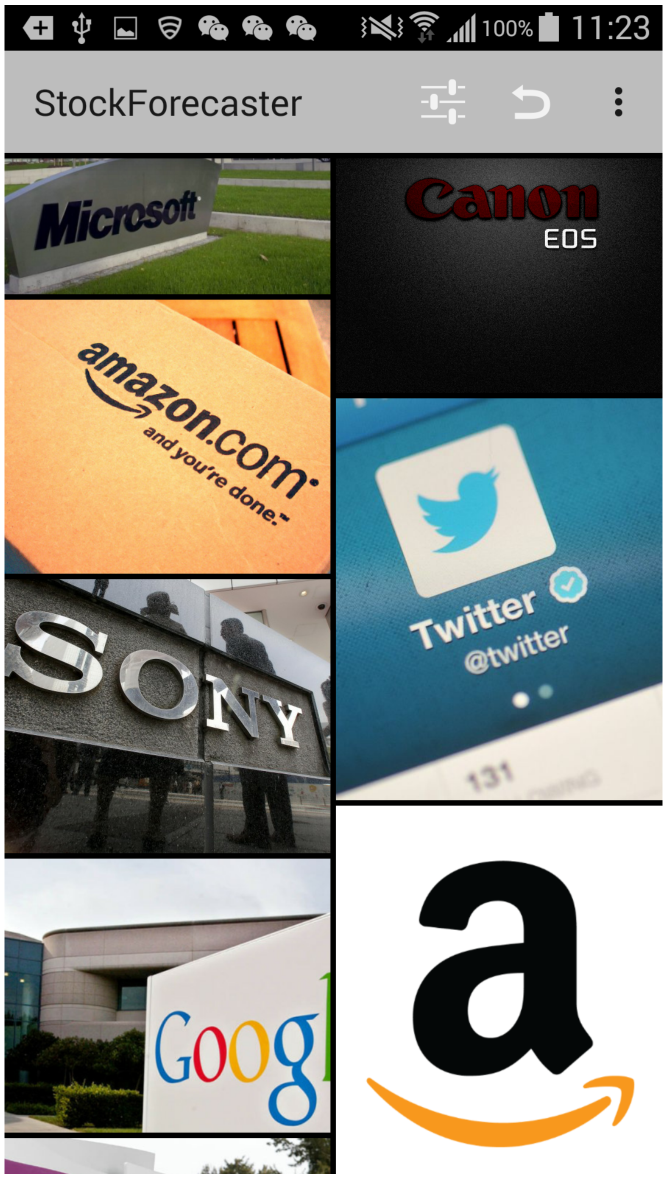


Figure 40. Andriod application: choose company

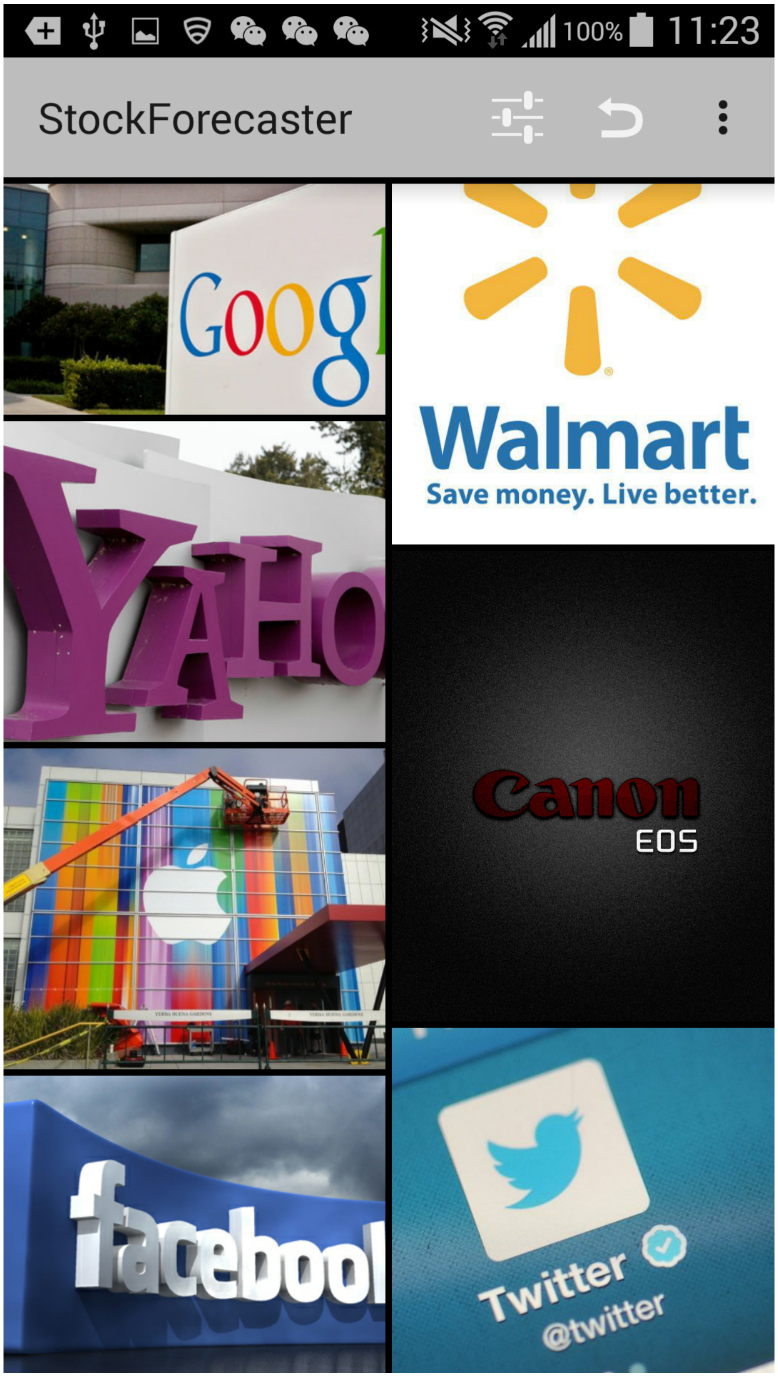


Figure 41. Andriod application: choose company(contd.)

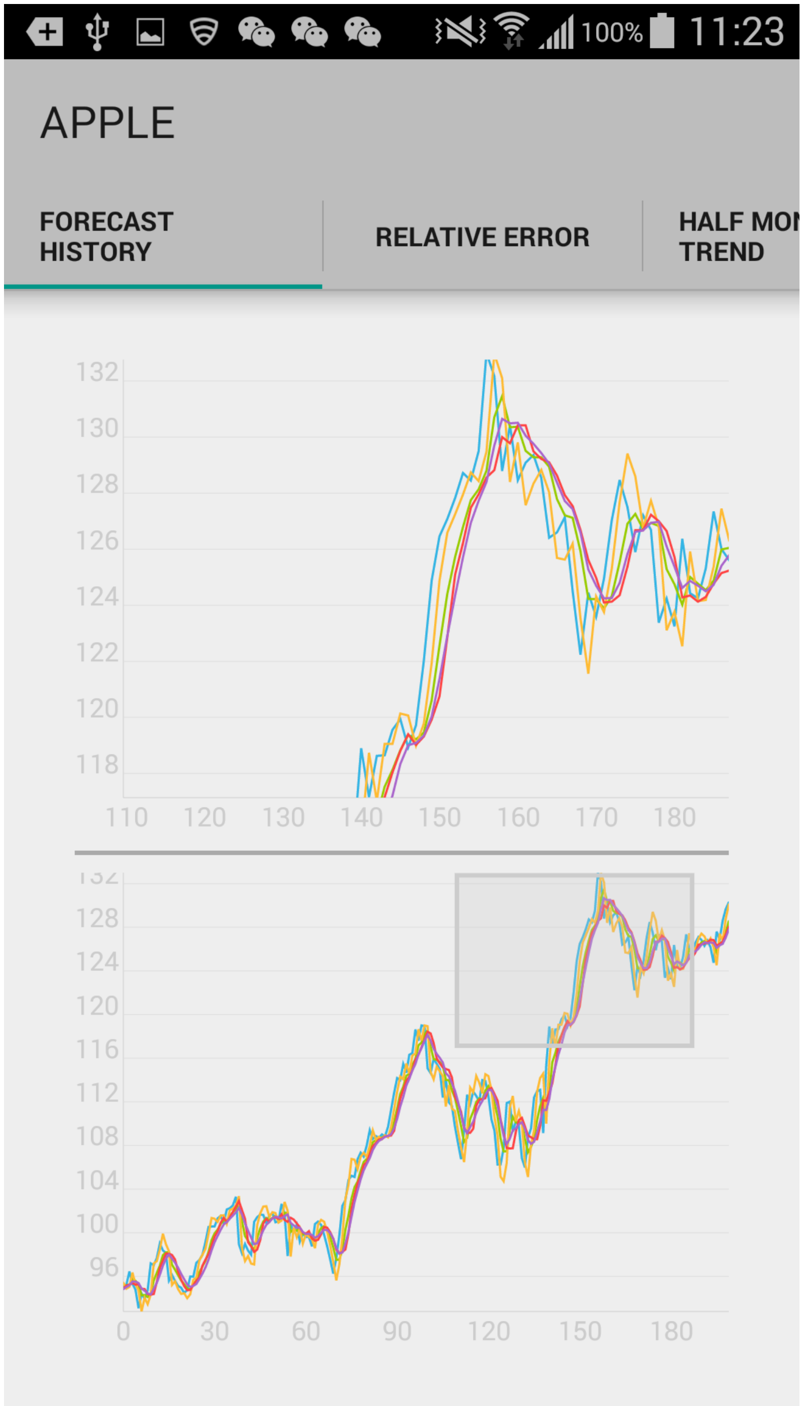


Figure 42. Short Term Prediction

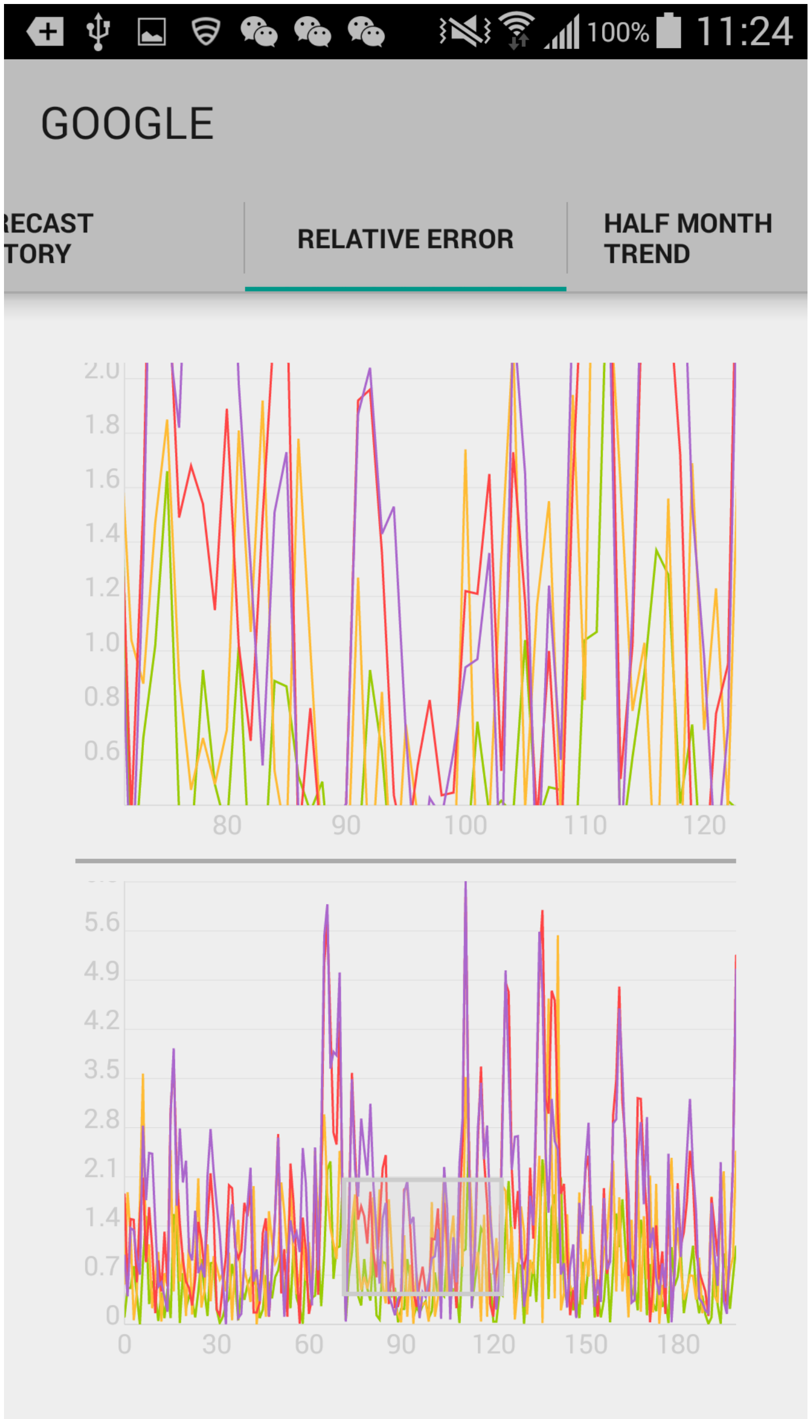


Figure 43. Read the Relative Difference

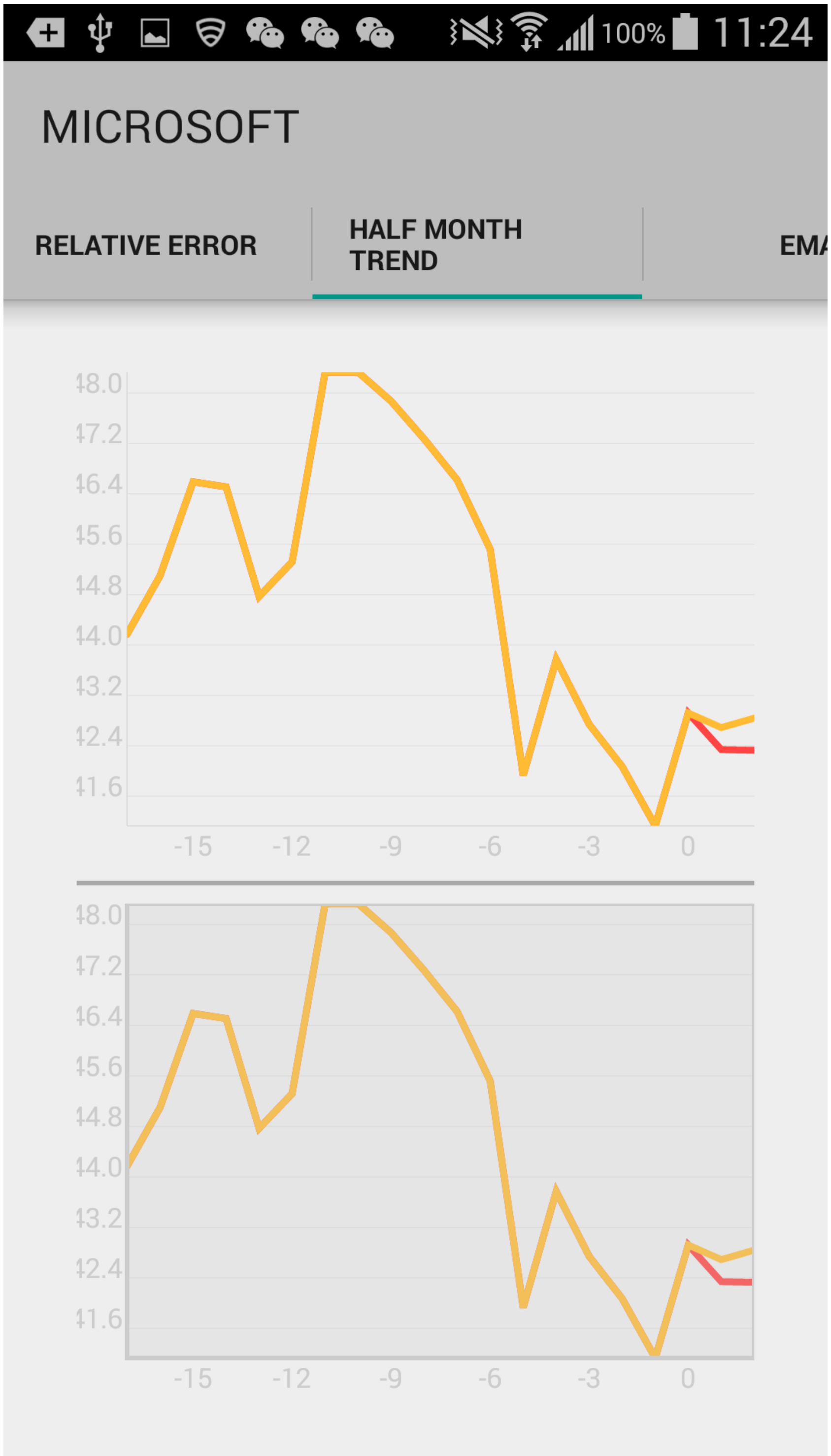


Figure 44. Get the long term prediction

Figure 45. Get the EMA graph

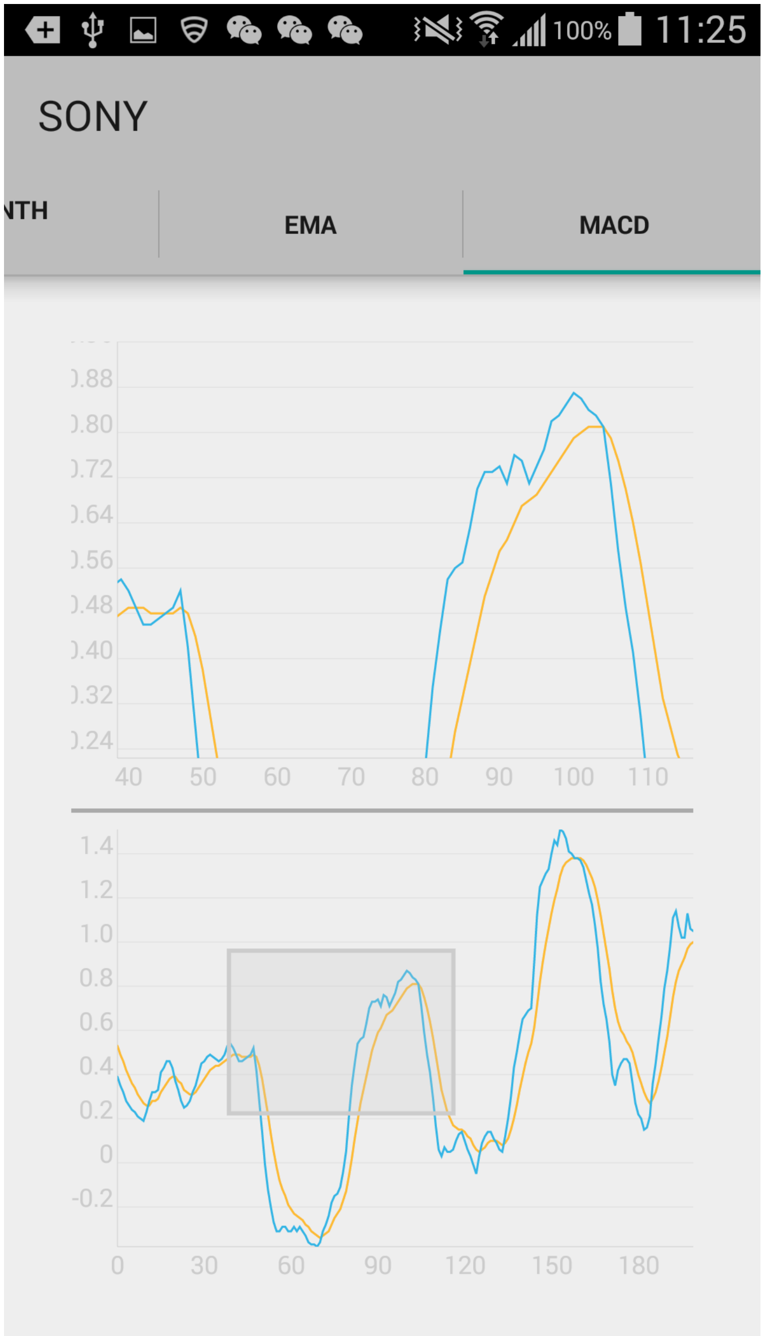


Figure 46. Get the MACD Graph

13 Schedule of Work

No.	Task	Start Time	End Time	Duration
1	Preliminary Presentation	Mar. 9th	Mar.13th	1 week
2	Set up the database for all the stocks	Mar. 9th	Mar.13th	1 week
3	Set up Bayesian Prediction Method for the system	Mar. 9th	Mar.13th	1 week
4	Set up the Linear SVM Prediction Method for the system	Mar.23rd	Apr.3rd	2 weeks
5	Set up other Short Term Prediction Methods for the system (STS & ANN)	Mar.23rd	Apr.3rd	2 weeks
6	Set up the Long Term Prediction Methods for the system (MACD, STS, Bayesian)	Mar.30th	Apr.10th	2 weeks
7	Set up the website for the system	Mar. 9th	Apr.3rd	4 weeks
8	Create a game platform on the website to allow user buy/sell on virtual stock market	Apr.6th	Apr.10th	1 week
9	Create a Strategy Display for each short term prediction method	Apr.13th	Apr.24th	2 weeks
10	Final Presentation and Report	Apr.27rd	May.1th	1 week

Table 18. Work Schedule

The Gantt Chart of the project plan is as following

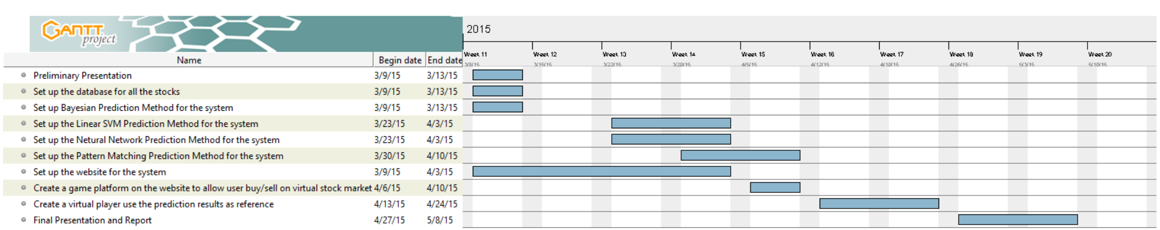


Figure 47. The Gantt Chart of the Project

Reference

[1] Robert D. Edwards, John Magee, W.H.C.Bassetti, *Technical Analysis of Stock Trends*, 9th Edition, CRC Press, (2007)

[2] C. Cortes, V. Vapnik, *Support Vector Networks, Machine Learning* 20, p. 273 (1995)

[3] Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola and Vladimir Vapnik (1997). *"Support Vector Regression Machines".* Advances in Neural Information Processing Systems 9, NIPS 1996, 155-161, MIT Press.

[4] R. -E. Fan, P. -H. Chen, C. –J. Lin, *Working set selection using the second order information for training SVM*, Journal of Machine Learning Research, 6, p. 1889 (2005)

[5] Wikipedia "Artificial Neural Network"
<http://en.wikipedia.org/wiki/Artificial_neural_network>

[6] Neuroph, website about a Java Neural Network Framework
<http://neuroph.sourceforge.net/index.html>
<http://neuroph.sourceforge.net/TimeSeriesPredictionTutorial.html>

[7] Stockcharts.com, website for financial charts
<http://stockcharts.com>

[8] Gardiner, C. W. *Stochastic methods*. Springer-Verlag, Berlin–Heidelberg–New York–Tokyo, 1985.

[9] Kannan, K. Senthamarai, et al. *"Financial stock market forecast using data mining techniques."* Proceedings of the International Multiconference of Engineers and computer scientists. Vol. 1. 2010.

[10] Appel, Gerald. *Technical analysis: power tools for active investors*. FT Press, 2005.

[11] *NIST/SEMATECH e-Handbook of Statistical Methods: Single Exponential Smoothing* at the National Institute of Standards and Technology

[12] Deng, Shangkun, et al. *"Combining technical analysis with sentiment analysis for stock price prediction."* Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on. IEEE, 2011.]

[13] *Web Based Stock Forecaster Report 1,* Group X, 2014 Spring
Web Based Stock Forecaster Report 3, Group 9, 2014 Spring
Web Based Stock Forecaster Report 3, Group 3, 2014 Spring