Web Based

Stock Forecaster

Final report

Group 4

Jian Ren / Yan Kang / Zhan Chen / Chengguang Xu / Yue Song

16:332:568 S15
Software Engineering Web Application
Instructor : Shiyu Zhou

Index

4.4 Inquire for predicted price

Part V Reference

Contribution Breakdown

All members contributed equally.

Part I
General Picture

1.1 Introduction

The goal of this project is to help the individual investor make smarter investment decisions. In order to achieve this goal, we create a web-based system, which helps investor make trading decisions by predicting the future stock price based on the analysis of historical price. First thing to be implemented is extracting stock information from website like Yahoo Finance or Google finance. Then certain prediction models is built to analyze the trends of stock prices.

In this project, there are overall two models, Bayesian Curve Fitting model and Artificial Neural Networking model to estimate short term stock prices and long term stock prices respectively.

The goal in the curve fitting problem is to be able to make predictions for the target variable t given some new value of the input variable x on the basis of a set of training data comprising N input values x = (x1, . . . , xN)T and their corresponding target values t = (t1, . . . , tN)T.

In a fully Bayesian approach, given the prior distribution p(w|α), we should consistently apply the sum and product rules of probability, which requires integrating over all values of w.

The advantage of the usage of neural networks for prediction is that they are able to learn from examples only and that after their learning is finished, they are able to catch hidden and strongly non-linear dependencies, even when there is a significant noise in the training set. The disadvantage is that ANN can learn the dependency valid in a certain period only. The error of prediction cannot be generally estimated.

With this realization, this project will attempt to benefit a broad range of investors by providing calculated predictions as a tool for them to make their own decision on whether to buy, sell, or hold the stock. Our application periodically download stock information from Yahoo Finance and store them in the local database, after that, python scripts are called to analyze the data and use above two models to make the estimation. Also, a web page is created so that the users can easily choose the stock, view information and get results of predictions they want.

1.2 Background

1.2.1 Basic stock market information and stock prediction

This is a time that everyone want to earn money in the finance market especially stock market. Investing money in the stock market is relatively easy but investing successfully and earning a profit can be a huge challenge. Actually most non-professional investors lose money.[1] In spite of the fact that it is impossible to be perfectly estimate the stock price, some basic methodologies still exist which can be used to analysis the stock market.

In order to get an informed estimate, generally, two aspects need to be focused: fundamental analysis and technical analysis. [2]

Fundamental analysis looks the long run and it takes the true value of a certain corporation as the basis of estimation. It maintains that markets may misprice a stock in the short run but that the "correct" price will eventually be reached. Profits can be made by purchasing the mispriced stock and then waiting for the market to recognize its "mistake" and reprice the stock.

Technical analysts are not concerned with any of the company's fundamentals. They seek to determine the future price of a stock based solely on the (potential) trends of the past price (a form of time series analysis). Numerous patterns are employed such as the head and shoulders or cup and saucer. Alongside the patterns, statistical techniques are used such as the exponential moving average (EMA). Candle stick patterns are believed to be first developed by Japanese rice merchants, and nowadays widely used by technical analysts.

Basically, there are two types of companies active in the stock market, one of which do the analysis without considering the company’s true value, often they only do the technical analysis.Most of this type of companies focus on short-time profit. The other type of companies care about a company’s long time value and they combine both the technical analysis and their opinion towards the company to make prediction of stock price. This kind of companies care about the long-time profit.

With the advent of the digital computer, stock market prediction has since moved into the technological realm. The most prominent technique involves the use of artificial neural networks (ANNs) and Genetic Algorithms. ANNs can be thought of as mathematical function approximators. The use of ANN simulates how human brain functions, by feeding computers with massive data to mimic human thinking. The most common form of ANN in use for stock market prediction is the feed forward network utilising the backward propagation of errors

algorithm to update the network weights. These networks are commonly referred to as Backpropagation networks. Another form of ANN that is more appropriate for stock prediction is the time recurrent neural network (RNN) or time delay neural network (TDNN). Examples of RNNN and TDNN are the Elman, Jordan, and Elman-Jordan networks.[3]

1.2.2 Yahoo Finance & API

Yahoo! Finance is a web site sponsored by Yahoo! that provides financial information and commentary with a focus on US markets. The web site offers information such as stock quotes, stock exchange rates, corporate press releases and financial reports, original programming video clips, and message boards. It also offers some online tools for personal finance management.

According to com.Scoredata for May 2012, Yahoo! Finance was the top financial news and research website in the United States with more than 37.5 million unique visitors per month.

Also, Yahoo Finance provides very useful API to retrieve real-time and historical data for any stocks very easily. But in this project, we did not use the API to extract stock data but wrote our own script to do that. In two ways of CSV and YQL, the data collection can be completed based on creating connections and requesting their urls.

1.2.3 .PHP + Mysql + Apache Client-Server Model

Figure 1.1

Using PHP scripting and MySQL database enables programmers to create applications that'll run on just about any computer, regardless of operating system. PHP has thousands of programming functions to facilitate almost any task.

If the computer can run web server software, the PHP / MySQL™ application is portable across operating systems and environments ... PC, Mac, Linux, Unix, Windows, Lindows, Internet, Intranet, etc. This means I can develop a project on my Windows PC, and send it to my friend in Slovakia who can run it on his Linux box.

PHP / MySQL™ is most often used to create dynamic web sites. On this web site, one script drives about 60 pages of content. Additional scripts are used here to process form data ... but that too could be done through the main script using included files.

PHP / MySQL™ projects include forums or communities, organizers, project management tools, calendars, shopping carts, mailing lists, and all sorts of useful applications. Source code for many open source projects is free, while advanced projects often require a registration fee for commercial use.

Data can be exported from MySQL™ for use in spreadsheets or databases on your PC. Similarly, data residing in existing PC spreadsheets and databases can be imported to an online database. The portability of data opens up all sorts of uses, especially for workgroups and for those who need to access data from both home and work.

In our project we use the XAMPP platform. XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

1.2.4 Curve fitting

Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points, possibly subject to constraints. Curve fitting can involve either interpolation, where an exact fit to the data is required, or smoothing, in which a "smooth" function is constructed that approximately fits the data. A related topic is regression analysis, which focuses more on questions of statistical inference such as how much uncertainty is present in a curve that is fit to data observed with random errors. Fitted curves can be used as an aid for data visualization, to infer values of a function where no data are available, and to summarize the relationships among two or more variables. Extrapolation refers to the use of a fitted curve beyond the range of the observed data, and is subject to a degree of uncertainty since it may reflect the method used to construct the curve as much as it reflects the observed data.

We applied Bayesian curve fitting to obtain the stock price value for short term prediction and Artificial Neural Network(ANN) curve fitting for long term prediction. The details will be stated in the prediction algorithm part.

1.3 Customer Requirement

By using Yahoo Finance to get access to the online stock market quotes, our system can download and store stock data continuously, both the real time data and historical data. Based on the analysis of the stock data, our system would extract useful parameters and stock information into a local database, and would provide our customer following features.

1.3.1 Sign in

In order to serve our users better, our system let visitors to sign up and create a personal account, and login to get detailed stock information and individual service. Figure 1.2 would show an example.
Features clearly specified as below:

- Sign up and create users' own account to receive accurate and better service. Use username and password to log in the system.

1.3.2 Latest market performance and news summary

After user login, in our home page, the customer can see the latest news and well-known company stock real time performance directly. Customer can quickly glance over the latest price and volume, which could help to decide to look up which stock information in detail, since the change of price and volume of well-known companies sometimes could also influence the performance of the whinputole stock market.Figure 1.3 would show an example.
Features clearly specified as below:

- Show four latest finance news and link to the whole news in the original website.
- Show six well-known companies real time stock price and volume.(Yahoo, Google, Costco,Twitter, Baidu, Facebook)

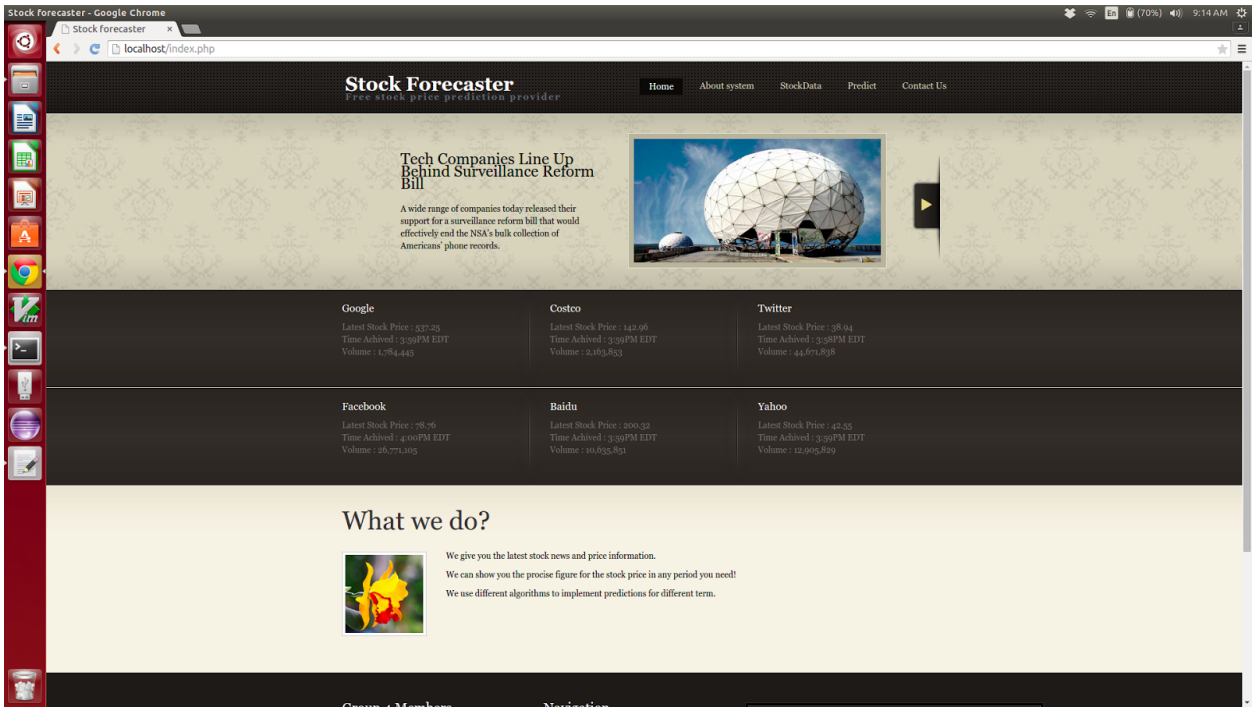


Figure 1.2

1.3.3 Stock Information and Prediction

Users can get detail stock information. User also can get the prediction of the stock by our system.Figure 1.4 would show an example.

Features clearly specified as below:

- The recently price trend chart of the stock
- The short-term and long-term prediction of price trend of the stock
- The price, time and volume of the real-time stock
- The historical data like time, open, high, low, close, and volume of the stock at exact date

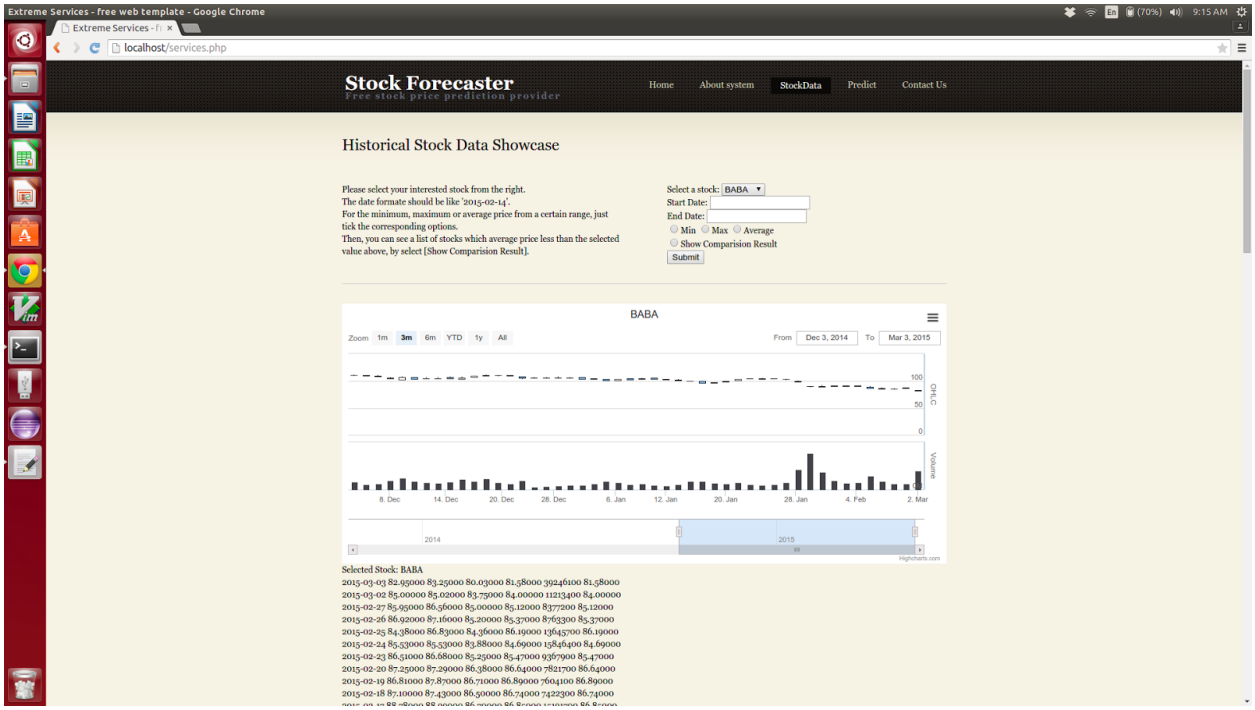


Figure 1.3

1.4 System requirement and design

1.4.1 System requirement details

Identifier	Priority	Requirement
REQ 1	5	The system shall download historical data from yahoo finance.
REQ 2	5	The system shall download real time data from yahoo finance on a daily basis.
REQ 3	5	All the stock data will then be placed into a stock database.
REQ 4	5	Implement the moving average prediction model or the decided prediction for the closing price of the stock prices

REQ 5	4	The system will allow for users to choose for the stock based on the stock ticker or the company name
REQ 6	3	The program will visualize the predict stock prices as a figure of price versus date.
REQ 7	5	The system shall display the stock information on a web page, each stock shall have its own page in the website with all of the stock information on the page.
REQ 8	4	All stocks in the database can be searched by its ticker symbols and accessed through the search bar.
REQ 9	5	This system MUST have internet access. Also user may need to enable JavaScript on his/her browser to accomplish certain function on their web browser.
REQ 10	3	The system must take the users input login information and match it to the system, then either allow access or deny it.
REQ 11	2	The website should display the current news articles from reputable sources to users in the home page.
REQ 12	2	The system should be easy to operate.

1.4.2 System Outline

The system is implemented as Web services for stock-prediction. It will track different stocks and issue a forecast about the price movement for a given stock.

We consider Yahoo finance to be data source. Yahoo finance provides an API to download free stock quote. By mining stock data into MySQL database, a prediction model becomes deployable.

Client side is an application provide user a graphic interface, by logging to the system with a valid user id and password, the whole functions of the system are available to the user. We will implement web services to delivery information on web pages. Stock prices, trading volumes, and time stamps are main features for forecasting.

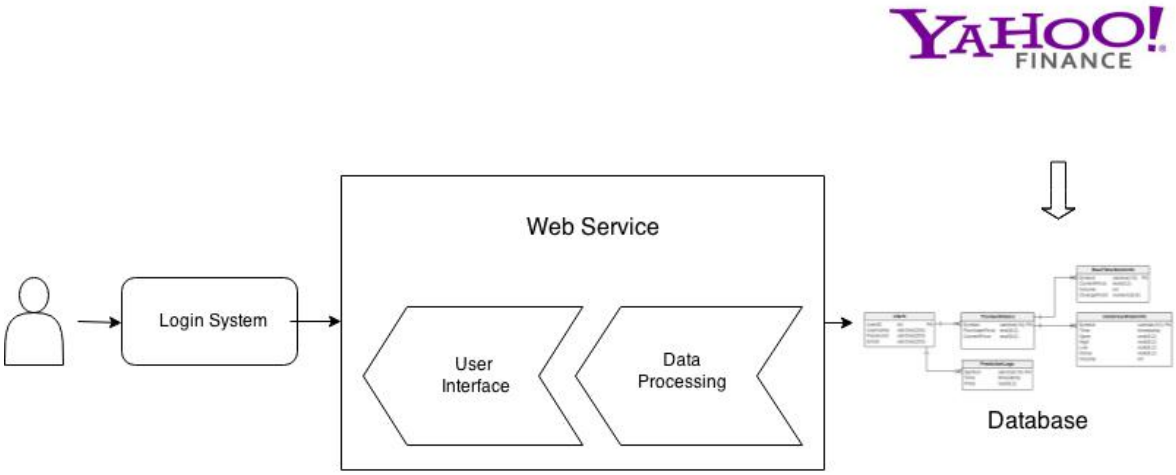


Figure 1.4 Outline of our system

1.4.3 Design Flow

i. Data Collection

This step stock data were extracted from Yahoo Finance.

Choose Stocks Information

The user will be allowed to get current information for selected stocks, like open close high, low, volume.

Get Real Time Stock Performance

Data mining application based on python will run continuously and provide user the real time performance.

Inquiry for Historical Price

Inquiry the historical price table and shows stock information in a certain time range.

ii. Model Build and Data Analysis

This step both Bayesian Curve Fitting and ANN model were build to evaluate the data just collected and make prediction

Obtain Predicted Price

Provide a stock price prediction with Bayes curve fitting.

Evaluation and Suggestion

Use historical data in database to simulate the future stocks value and give investment suggestion.

iii. User Interface and Interaction

This step shows the user the prediction result, also the webpage prints results for user to observe the data.

Login System

Make database to store all the user information.

Stocks Figure Page

Obtain the data from the database and use the charts API to draw the professional stock analysis figure for users.

Latest news

Get the headlines and summary of latest financial news to show in the homepage, give the link to users for the whole news pages in other website.

About us/Contact us

This page has developers' bios and their contact info so that users can get to know us and reach us by email.

1.4.4 System Interaction Block Diagram

This is a fundamental system of our web service. It allows user to enter a certain stock code or stock name, system will show user with basic information and data of this stock. Only successfully logged-in user can using this system .

In our dynamic stock chart, user has a lot of selections, they can select stock, select period, compare results, etc. Most of these operations can be handled by JavaScript chart alone. Some operations like prediction, selecting different stocks need accessing the database.

In our server, there are several stock panel and stock indicator calculator. Data will be extracted from database and used to calculate predictive results. Prediction results will be saved back to database. In the front end, stock interface get user's request "view long term prediction" or "view short term prediction". Chart data will be upgraded according to corresponding request.

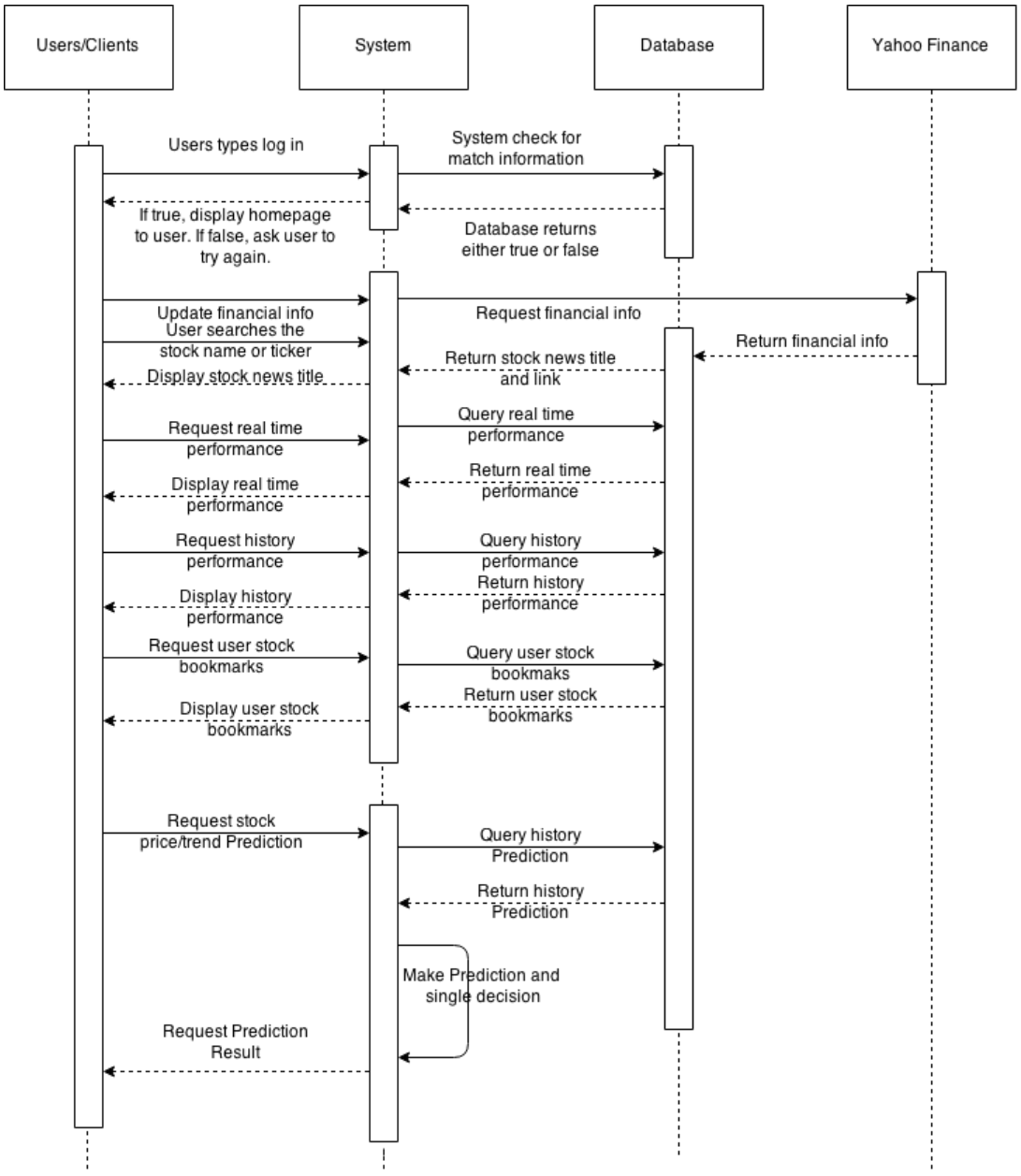


Figure 1.5 Basic system interaction diagram

1.5 Functional Requirements and Use Case

1.5.1 Stakeholders

Our system could get the information of the real time stock market, and make prediction of the price trend of the stock. So there are many people that would have interest in our system.

- Users/ Investors
- Administrators
- Sponsors

1.5.2 Actors and Goals

Actor	Goal
User	The user's goal is to access the system to get the information about the stock quickly as well as precisely.
Database	Records all stock information like price, time, date, high & low, open & close, volume, username, password and profile of users.
Yahoo! Finance API	The service provider for all the data in the database and allow user to get the stock result.
Administrator	The administrator is a special user and has top priority in access and change to the database and all other user information.

1.5.3 Use Case

Use Case	Name	Description
UC 1	Signup	Allow a visitor to create a new profile.
UC 2	Login	Allow a user to access the system.
UC 3	Check Homepage	Present links of all functions, like checking stock prices,view stock news etc.
UC 4	Database Update	Allow the administrator to update the database of stocks information collected.
UC 5	Choose Stock	Allow a user to choose a certain stock code or stock name to find the stock which he/she wants to know about its general, history and predicting information.
UC 6	Check Stock Overview	Allow a user to Check the general information of all stocks (open price, high price, low price, current price, volume, etc.)and compare with other stocks.
UC 7	Check Stock Chart	Allow a user to check dynamic charts of a certain stock.
UC 8	Get Short Term Prediction	Allow a user to get the short term prediction of a certain stock.
UC 9	Get Long Term Prediction	Allow a user to get the long term prediction of a certain stock.
UC 10	View Recommendations	Based on the result of the prediction, giving users the system's recommendation of a certain stock –what is the best action at present: buy, sell, or hold.
UC 11	FindAboutUs	Allow a user to find information about service provider.

Part II Algorithms

2 Prediction algorithms

We use two different methods to make predictions. One is bayesian curve fitting, which is for the short term prediction, and the other is Neural network for long term prediction. Both of them are trained by the former stock prices and follow the pattern to make some predictions about the future prices.

2.1 Bayesian Curve Fitting

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

$$m(x) = \beta\boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)t_n \qquad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\boldsymbol{\phi}(x)$$

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^{\mathrm{T}} \qquad \boldsymbol{\phi}(x_n) = \left(x_n^0,\ldots,x_n^M\right)^{\mathrm{T}}$$

Figure 2.1 The formula of Bayesian Curve fitting

This method is utilized for the short term prediction. Generally speaking, in short range, the data is more like in a sequence of some kind of function like quadratic or cubic or in some higher order, which is determined by the parameter M. What's the advantage of bayesian curve fitting is that it could provide us a theoretically continue smooth function (of course in this specific situation, that is prices prediction, it is discrete) which best fits on the former data.

According to this method, we could derive the predicted mean function m(x) and the sigma function at any time point either in the past or in the future. But the trade-off is that the further prediction makes the worse precision, and this is reasonable considering that never a stock price curve is a perfect polynomial function. So now, we have the predicted mean function and sigma function for the any specific time, what's the last step is fitting them into the gaussian distribution function and randomly getting a result based on the function, which is the final predicted price that we need.

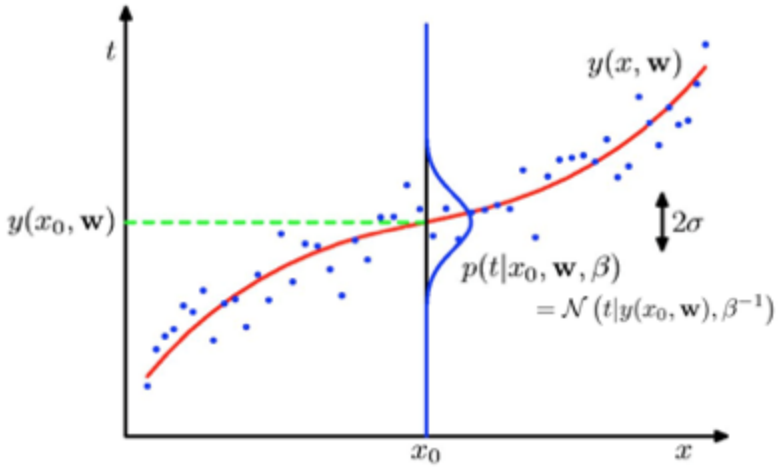


Figure 2.2 Bayesian Model Figure

In a fully Bayesian approach, we should consistently apply the sum and product rules of probability, which requires, as we shall see shortly, that we integrate over all values of w. Such marginalizations lie at the heart of Bayesian methods for pattern recognition.

```
# ================== Start prediction ============================ #
# ----------- Normalization ------------ #
# normalizing by subtracting the mean and dividing by the standard deviation
NormalTestPrices = []
for i in range(0, len(TestPrices)):
    middle = (TestPrices[i]-np.mean(TestPrices))/np.std(TestPrices)
    NormalTestPrices.append(middle)
# print NormalTestPrices

# ------------- Output the predicted values with de-normalization ----------- #
mu = np.mat(np.zeros([len(PredictTimeStamp), 1]))           # !
sigma = np.mat(np.zeros([len(PredictTimeStamp), 1]))        # !
x = np.mat(np.zeros([len(PredictTimeStamp), 1]))            # !
output = np.mat(np.zeros([len(PredictTimeStamp), 1]))       # !

for i in range(0, len(PredictTimeStamp)):                   # !
    mu[i] = m_x(TimeStamp, NormalTestPrices, i+1)
    sigma[i] = np.power(sigma2_x(TimeStamp, i+1), 0.5)
    x[i] = sigma[i] * np.random.randn() + mu[i]
    mu[i] = mu[i]*np.std(TestPrices)+np.mean(TestPrices)
    output[i] = x[i]*np.std(TestPrices)+np.mean(TestPrices)
# print mu
# print sigma
# print x
# print output


# ------------- Evaluation ----------------- #
abs_error = []
relative_error = []
for i in range(0, len(TestPrices)):
    abs_error.append(np.absolute(output[i]-TestPrices[i]))
    relative_error.append(abs_error[i]/TestPrices[i])
abs_error_mean = np.mean(abs_error)
relative_average_error = np.mean(relative_error)
s1 = 'Absolute error mean is: ' + repr(abs_error_mean)
s2 = 'Relative average error is: ' + repr(relative_average_error*100) + '%'
print s1
print s2


# ------------- Draw plots ---------------- #
yerr = np.squeeze(np.asarray(abs_error))
fig = plt.figure()
x1 = np.asarray(TimeStamp)
x2 = np.asarray(PredictTimeStamp)           # !
```

Figure 2.3 The training and predicting process of bayesian process

2.2 Artificial Neural Networking

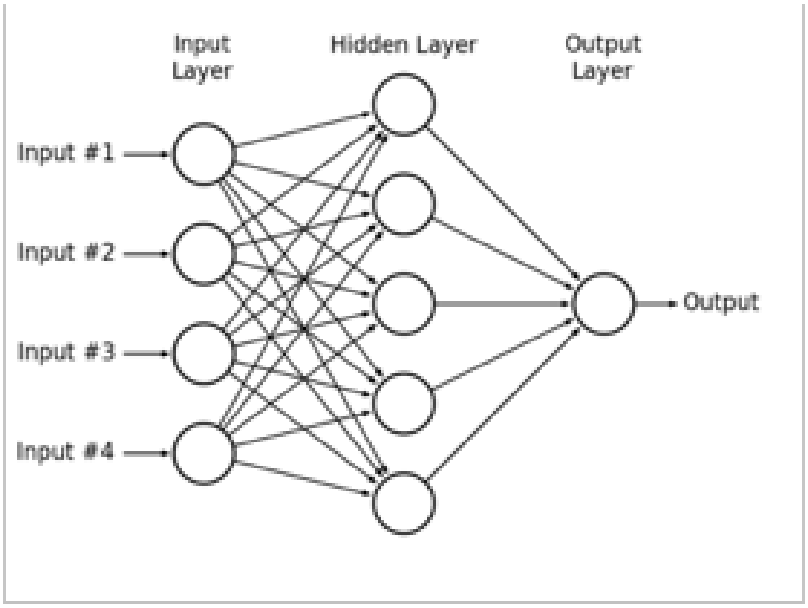


Figure 2.4 The concept map of ANN network

To achieve a good performance long term prediction, Bayesian curve fitting is too weak for that because it makes solid assertion that the curve must fit in a polynomial function which we all known unreasonable for long term. Thus we need a more sophisticated and more robust method to find a pattern that the prices trend could fit in. And here we propose using neural network to achieve that.

To make this clearer, let's clarify what the input and output are.

Input X is a vector consisted of different orders of the timestamp, here for example, the first feature is the 0 order of timestamp "5" that is 1, the second feature is the 1st order of timestamp "5" that is 5, the third is 2nd order of timestamp "5" that is 25 and so on so forth. The dimension of input is the order of prediction parameter M plus one (for the order 0).

Output is the stock price corresponding to the input timestamp. Training period is 30. When doing the prediction: 1) sequentially input the pair of timestamp and price; 2) in each period use gradient descendent to rectify the weights W; 3) after 30 period, we have the corrected weights and use this weights to predict the stock price of next timestamp; 4) this corrected weights W is the starting weights of next prediction.

So, we have the neural network for training the prices and after the training we have weights stored in the net which best fitting the former prices. and once new input timestamp is provided, we could transform the input to M+1 feature vector and use the trained network to

make prediction of input. In this way, the model bias of Bayesian curve fitting is eliminated and we will have better regression performance.

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.

For example, a neural network for handwriting recognition is defined by a set of neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (determined by the network's designer), the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read.

Like other machine learning methods - systems that learn from data - neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including computer vision and speech recognition.

```
y_full = y                                                  # save the original y
y_last = y[-1]                                              # save the true last y to compare prediction
y = y[0:-1, :]

# --------- Construct Neural Network ------------

net1 = NeuralNet(
    layers=[  # three layers: one hidden layer
        ('input', layers.InputLayer),
        ('hidden', layers.DenseLayer),
        ('output', layers.DenseLayer),
        ],
    # layer parameters:
    input_shape=(None, M+1),  # 96x96 input pixels per batch
    hidden_num_units=100,  # number of units in hidden layer
    output_nonlinearity=None,  # output layer uses identity function
    output_num_units=1,  # 30 target values

    # optimization method:
    update=nesterov_momentum,
    update_learning_rate=0.01,
    update_momentum=0.9,

    regression=True,  # flag to indicate we're dealing with regression problem
    max_epochs=40,  # we want to train this many epochs
    verbose=1,
    )

net1.fit(X, y)
y_predict = net1.predict(X)
row, col = y_predict.shape
y_predict = y_predict*Distance+Average
for i in range(0, row):
    y_predict[i] += np.random.normal(0, market_vibration)  # add price vibration into consideration
abs_error = np.absolute(y_predict-np.mat(ClosePrice[0:-1]).T)
relative_error = abs_error/np.mean(ClosePrice[0:-1])
print np.mean(abs_error)
print np.mean(relative_error)
print y_predict
print ClosePrice[0:-1]
print abs_error.shape
# -------------- plot ------------
plt.close("all")
fig = plt.figure()
err = np.squeeze(np.array(abs_error))
x1 = np.asarray(range(1, len(ClosePrice)))  # 1- 250
y1 = np.asarray(ClosePrice[0:-1])           # true prices
y2 = np.asarray(y_predict)                  # predict prices
```

Figure 2.5 The training and predicting process of ANN network

Part III Database

3.1 Data Collection Process

We need to collect both the historical data and real time data. All the data come from Yahoo finance. The historical data can be downloaded from Yahoo finance directly, the format is .csv. Then we store the historical data in database. As shown in Figure 3.1. The historical data includes date, open price, high price, low price, close price and volume information. All the historical data are saved in one database, and different stock has a different table in the database.

For the real time stock price, python script is used to get all the information we want. To get the data from website, we need to read the website information, and then get the markers of the data that we are interested in. Therefore, we read the stock's website information and get all the stock information that we want. Figure 3.2 shows the real time price stock information. It includes stock price, stock time, volume and change points. All the real time data are saved in one database, and different stock has different table in the database.

```
mysql> SELECT * FROM BABA;
+------------+----------+----------+----------+----------+----------+----------+
| DATE       | OPEN     | HIGH     | LOW      | CLOSE    | VOLUME   | ADJCLOSE |
+------------+----------+----------+----------+----------+----------+----------+
| 2015-04-24 | 82.88000 | 84.99000 | 82.80000 | 84.57000 | 21007900 | 84.57000 |
| 2015-04-23 | 81.88000 | 82.75000 | 81.62000 | 82.28000 | 11417500 | 82.28000 |
| 2015-04-22 | 82.78000 | 82.84000 | 81.52000 | 82.05000 | 13257800 | 82.05000 |
| 2015-04-21 | 82.86000 | 82.92000 | 82.14000 | 82.42000 |  9844100 | 82.42000 |
| 2015-04-20 | 82.25000 | 82.69000 | 81.52000 | 82.20000 | 13439400 | 82.20000 |
| 2015-04-17 | 83.08000 | 83.25000 | 81.61000 | 81.90000 | 22653200 | 81.90000 |
| 2015-04-16 | 85.15000 | 85.40000 | 83.89000 | 84.06000 | 11975100 | 84.06000 |
| 2015-04-15 | 85.08000 | 85.36000 | 84.16000 | 84.67000 | 12196000 | 84.67000 |
| 2015-04-14 | 84.63000 | 85.13000 | 83.53000 | 85.00000 | 11819000 | 85.00000 |
| 2015-04-13 | 85.16000 | 85.71000 | 83.82000 | 84.28000 | 15797100 | 84.28000 |
| 2015-04-10 | 86.68000 | 86.69000 | 84.15000 | 84.58000 | 18300300 | 84.58000 |
| 2015-04-09 | 87.03000 | 87.69000 | 85.20000 | 86.14000 | 27800400 | 86.14000 |
| 2015-04-08 | 83.30000 | 85.54000 | 83.06600 | 85.39000 | 26161500 | 85.39000 |
| 2015-04-07 | 81.94000 | 82.95000 | 81.88000 | 82.21000 |  9443000 | 82.21000 |
| 2015-04-06 | 82.05000 | 82.59000 | 81.61400 | 81.82000 | 12810500 | 81.82000 |
| 2015-04-02 | 82.88000 | 83.00000 | 81.25000 | 82.28000 | 19868200 | 82.28000 |
| 2015-04-01 | 83.37000 | 83.72000 | 82.18000 | 82.36000 | 14896400 | 82.36000 |
| 2015-03-31 | 83.64000 | 84.45000 | 83.20000 | 83.24000 | 11812900 | 83.24000 |
| 2015-03-30 | 85.03000 | 85.15000 | 83.75000 | 83.90000 | 10020500 | 83.90000 |
| 2015-03-27 | 84.78000 | 85.15000 | 83.34000 | 84.58000 |  9703800 | 84.58000 |
| 2015-03-26 | 83.20000 | 84.99500 | 82.96000 | 84.17000 | 12208000 | 84.17000 |
| 2015-03-25 | 83.94000 | 84.48000 | 82.67000 | 83.75000 | 13507200 | 83.75000 |
| 2015-03-24 | 84.35000 | 84.50000 | 82.39000 | 83.63000 | 14545100 | 83.63000 |
| 2015-03-23 | 85.25000 | 85.53000 | 84.21000 | 84.25000 | 11184000 | 84.25000 |
| 2015-03-20 | 86.33000 | 86.80000 | 85.09000 | 85.20000 | 21181500 | 85.20000 |
| 2015-03-19 | 85.11000 | 87.04000 | 85.00000 | 85.74000 | 30454700 | 85.74000 |
| 2015-03-18 | 83.87000 | 85.95000 | 83.30000 | 84.59000 | 35663000 | 84.59000 |
| 2015-03-17 | 84.01000 | 85.10000 | 83.51000 | 84.50000 | 17639000 | 84.50000 |
| 2015-03-16 | 82.01000 | 85.20000 | 81.94000 | 84.00000 | 16905700 | 84.00000 |
| 2015-03-13 | 81.80000 | 81.92000 | 80.77000 | 81.86000 | 12663300 | 81.86000 |
| 2015-03-12 | 82.10000 | 82.90000 | 81.53000 | 81.92000 | 11232000 | 81.92000 |
| 2015-03-11 | 83.00000 | 83.38000 | 81.19000 | 81.99000 | 12768700 | 81.99000 |
| 2015-03-10 | 81.09000 | 83.15000 | 80.65000 | 82.97000 | 13817000 | 82.97000 |
| 2015-03-09 | 84.35000 | 84.35000 | 81.48000 | 82.53000 | 17616500 | 82.53000 |
| 2015-03-06 | 85.74000 | 86.00000 | 84.05000 | 84.40000 | 10673700 | 84.40000 |
| 2015-03-05 | 85.75000 | 86.27000 | 84.01000 | 86.10000 | 18514800 | 86.10000 |
| 2015-03-04 | 80.27000 | 85.82600 | 80.17000 | 85.49000 | 36607200 | 85.49000 |
| 2015-03-03 | 82.95000 | 83.25000 | 80.03000 | 81.58000 | 39246100 | 81.58000 |
| 2015-03-02 | 85.00000 | 85.02000 | 83.75000 | 84.00000 | 11213400 | 84.00000 |
| 2015-02-27 | 85.95000 | 86.56000 | 85.00000 | 85.12000 |  8377200 | 85.12000 |
| 2015-02-26 | 86.92000 | 87.16000 | 85.20000 | 85.37000 |  8763300 | 85.37000 |
| 2015-02-25 | 84.38000 | 86.83000 | 84.36000 | 86.19000 | 13645700 | 86.19000 |
| 2015-02-24 | 85.53000 | 85.53000 | 83.88000 | 84.69000 | 15846400 | 84.69000 |
| 2015-02-23 | 86.51000 | 86.68000 | 85.25000 | 85.47000 |  9367900 | 85.47000 |
| 2015-02-20 | 87.25000 | 87.29000 | 86.38000 | 86.64000 |  7821700 | 86.64000 |
| 2015-02-19 | 86.81000 | 87.87000 | 86.71000 | 86.89000 |  7604100 | 86.89000 |
| 2015-02-18 | 87.10000 | 87.43000 | 86.50000 | 86.74000 |  7422300 | 86.74000 |
| 2015-02-17 | 88.78000 | 88.99000 | 86.70000 | 86.85000 | 15191700 | 86.85000 |
| 2015-02-13 | 88.20000 | 89.30000 | 87.65000 | 89.05000 | 14665200 | 89.05000 |
| 2015-02-12 | 85.60000 | 88.30000 | 85.55000 | 87.10000 | 15177800 | 87.10000 |
| 2015-02-11 | 87.58000 | 87.70000 | 85.82000 | 86.00000 | 12391900 | 86.00000 |
| 2015-02-10 | 87.01000 | 87.47000 | 86.52000 | 87.26000 | 12043300 | 87.26000 |
| 2015-02-09 | 85.83000 | 86.75000 | 85.47000 | 86.00000 | 12109500 | 86.00000 |
| 2015-02-06 | 87.11000 | 87.40000 | 85.42000 | 85.68000 | 17587600 | 85.68000 |
```

Figure 3.1 The storage of historical data

STOCK_PRICE	STOCK_TIME	STOCK_VOLUME	change_point
81.88	12:34PM EDT	5,383,125	81.88
81.78	12:35PM EDT	5,390,175	-0.0012212
81.78	12:36PM EDT	5,409,883	0
81.76	12:37PM EDT	5,429,826	-0.0002445
81.79	12:38PM EDT	5,441,059	0.00036692
81.77	12:39PM EDT	5,455,386	-0.0002445
81.83	12:40PM EDT	5,465,029	0.00073376
81.82	12:41PM EDT	5,478,250	-0.0001222
81.77	12:42PM EDT	5,502,773	-0.0006110
81.79	12:43PM EDT	5,515,241	0.00024458
81.81	12:44PM EDT	5,533,564	0.00024452
81.76	12:45PM EDT	5,542,760	-0.0006111
81.71	12:46PM EDT	5,567,067	-0.0006115
81.70	12:47PM EDT	5,583,373	-0.0001223
81.69	12:48PM EDT	5,616,106	-0.0001223
81.72	12:49PM EDT	5,635,078	0.00036724
81.69	12:50PM EDT	5,659,398	-0.0003671
81.69	12:51PM EDT	5,669,501	0
81.65	12:52PM EDT	5,709,926	-0.0004896
81.64	12:53PM EDT	5,750,669	-0.0001224
81.65	12:54PM EDT	5,777,326	0.00012248
81.64	12:55PM EDT	5,792,967	-0.0001224
81.70	12:56PM EDT	5,803,222	0.00073493
81.65	12:57PM EDT	5,814,627	-0.0006119
81.64	12:58PM EDT	5,841,470	-0.0001224
81.66	12:59PM EDT	5,861,839	0.00024497
81.65	1:00PM EDT	5,884,971	-0.0001224
81.65	1:01PM EDT	5,915,860	0
81.65	1:02PM EDT	5,943,355	0
81.63	1:03PM EDT	5,973,049	-0.0002449
81.66	1:04PM EDT	5,994,630	0.00036751
81.65	1:05PM EDT	6,003,198	-0.0001224
81.64	1:06PM EDT	6,017,958	-0.0001224
81.67	1:07PM EDT	6,090,821	0.00036746
81.67	1:08PM EDT	6,128,645	0
81.66	1:09PM EDT	6,170,437	-0.0001224
81.66	1:10PM EDT	6,185,892	0
81.65	1:11PM EDT	6,200,551	-0.0001224
81.64	1:12PM EDT	6,220,179	-0.0001224
81.63	1:14PM EDT	6,248,988	-0.0001224
81.66	1:14PM EDT	6,268,697	0.00036751
81.64	1:15PM EDT	6,280,578	-0.0002449
81.63	1:16PM EDT	6,290,050	-0.0001224
81.57	1:17PM EDT	6,327,699	-0.0007350
81.60	1:19PM EDT	6,348,810	0.00036778
81.61	1:19PM EDT	6,382,568	0.00012254
81.59	1:20PM EDT	6,401,518	-0.0002450
81.57	1:22PM EDT	6,425,019	-0.0002451
81.57	1:23PM EDT	6,432,595	0
81.57	1:24PM EDT	6,446,849	0
81.59	1:25PM EDT	6,458,479	0.00024518
81.60	1:26PM EDT	6,466,459	0.00012256
81.56	1:27PM EDT	6,485,297	-0.0004901
81.57	1:28PM EDT	6,493,756	0.00012260
81.62	1:29PM EDT	6,506,686	0.00061297

Figure 3.2 The storage of real time data

3.2 Database Schema

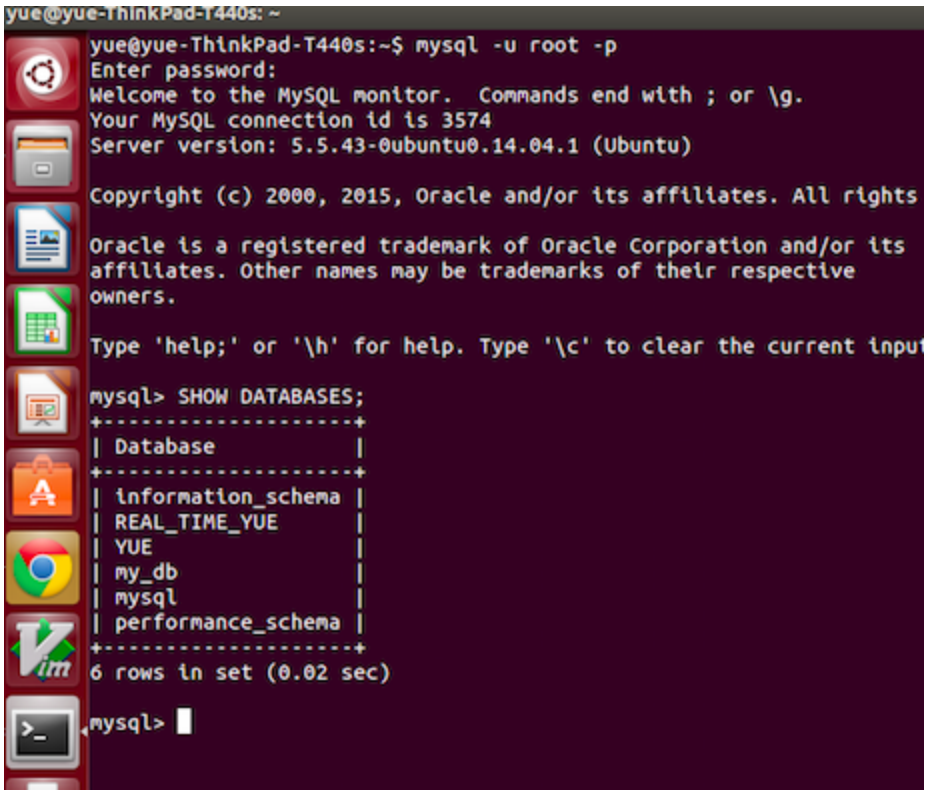


Figure 3.3 Show the databases

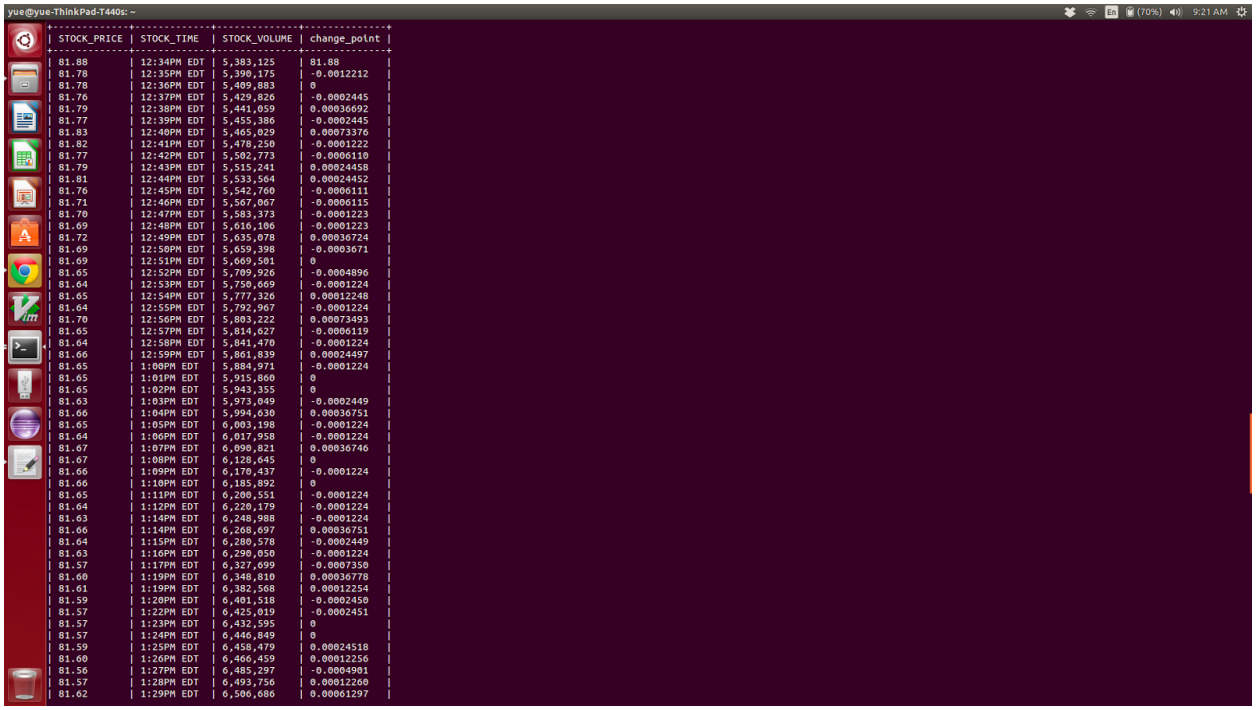
We have two databases to store all the stock information, as shown in Figure 3.3. The database which name is “YUE” , stores all the historical data information and the database which name is “REAL_TIME_YUE” stores all the real time stock information. We can use command “SHOW DATABASES” to show all the database information.

To see the specific information of a stock, first we need to select a database. For example, we use the command “USE REAL_TIME_YUE” to select the database that stores all the historical data. The use command “SHOW TABLES” to show all the tables in the database. Each table stores the historical data of each stock. As shown in Figure 3.4, there are 13 stocks’ historical data and real time data saved in the database. To change a another database, we just need to type in the “USE YUE” to see the stock information in database YUE.

Each stock’s information can be shown from the database. We can select a database, and then select a table in the database, finally we can show the stock information. From Figure 3.5, we can see that after select a table, which include the information of a specific stock, the stock information can be shown in from the database.

Besides the databases that store the the store information, we also have a database to store the customers' information. For each customer, we need to record the customer's information in the database so the customer can register from our website and login to the system next time. There is only one table is the database that stores customers' information. The table stores the information of memberID, username, password, email, active, resetToken and resetComplete. So for the registered customer, we can search the database next time and find the customer's information and let the customer login into the system. The new customer can register from our website and the information of the new customer will be stored in the database.

Figure 3.4 The storage of historical data and real time data



STOCK_PRICE	STOCK_TIME	STOCK_VOLUME	change_point
81.88	12:34PM EDT	5,383,125	81.88
81.78	12:35PM EDT	5,390,175	-0.0012212
81.78	12:36PM EDT	5,409,883	0
81.76	12:37PM EDT	5,429,826	-0.0002445
81.79	12:38PM EDT	5,441,059	0.00036692
81.77	12:39PM EDT	5,455,386	-0.0002445
81.83	12:40PM EDT	5,465,029	0.00073376
81.82	12:41PM EDT	5,478,250	-0.0001222
81.77	12:42PM EDT	5,502,773	-0.0006110
81.79	12:43PM EDT	5,515,241	0.00024458
81.81	12:44PM EDT	5,533,564	0.00024452
81.76	12:45PM EDT	5,542,760	-0.0006111
81.71	12:46PM EDT	5,567,067	-0.0006115
81.70	12:47PM EDT	5,583,373	-0.0001223
81.69	12:48PM EDT	5,616,106	-0.0001223
81.72	12:49PM EDT	5,635,078	0.00036724
81.69	12:50PM EDT	5,659,398	-0.0003671
81.69	12:51PM EDT	5,669,501	0
81.65	12:52PM EDT	5,709,926	-0.0004896
81.64	12:53PM EDT	5,750,669	-0.0001224
81.65	12:54PM EDT	5,777,326	0.00012248
81.64	12:55PM EDT	5,792,967	-0.0001224
81.70	12:56PM EDT	5,803,222	0.00073493
81.65	12:57PM EDT	5,814,627	-0.0006119
81.64	12:58PM EDT	5,841,470	-0.0001224
81.66	12:59PM EDT	5,861,839	0.00024497
81.65	1:00PM EDT	5,884,971	-0.0001224
81.65	1:01PM EDT	5,915,860	0
81.65	1:02PM EDT	5,943,355	0
81.63	1:03PM EDT	5,973,049	-0.0002449
81.66	1:04PM EDT	5,994,630	0.00036751
81.65	1:05PM EDT	6,003,198	-0.0001224
81.64	1:06PM EDT	6,017,958	-0.0001224
81.67	1:07PM EDT	6,090,821	0.00036746
81.67	1:08PM EDT	6,128,645	0
81.66	1:09PM EDT	6,170,437	-0.0001224
81.66	1:10PM EDT	6,185,892	0
81.65	1:11PM EDT	6,200,551	-0.0001224
81.64	1:12PM EDT	6,220,179	-0.0001224
81.63	1:14PM EDT	6,248,988	-0.0001224
81.66	1:14PM EDT	6,268,697	0.00036751
81.64	1:15PM EDT	6,280,578	-0.0002449
81.63	1:16PM EDT	6,290,050	-0.0001224
81.57	1:17PM EDT	6,327,699	-0.0007350
81.60	1:19PM EDT	6,348,810	0.00036778
81.61	1:19PM EDT	6,382,568	0.00012254
81.59	1:20PM EDT	6,401,518	-0.0002450
81.57	1:22PM EDT	6,425,019	-0.0002451
81.57	1:23PM EDT	6,432,595	0
81.57	1:24PM EDT	6,446,849	0
81.59	1:25PM EDT	6,458,479	0.00024518
81.60	1:26PM EDT	6,466,459	0.00012256
81.56	1:27PM EDT	6,485,297	-0.0004901
81.57	1:28PM EDT	6,493,756	0.00012260
81.62	1:29PM EDT	6,506,686	0.00061297

Figure 3.5 The real time data includes stock price, time, volume and change point.

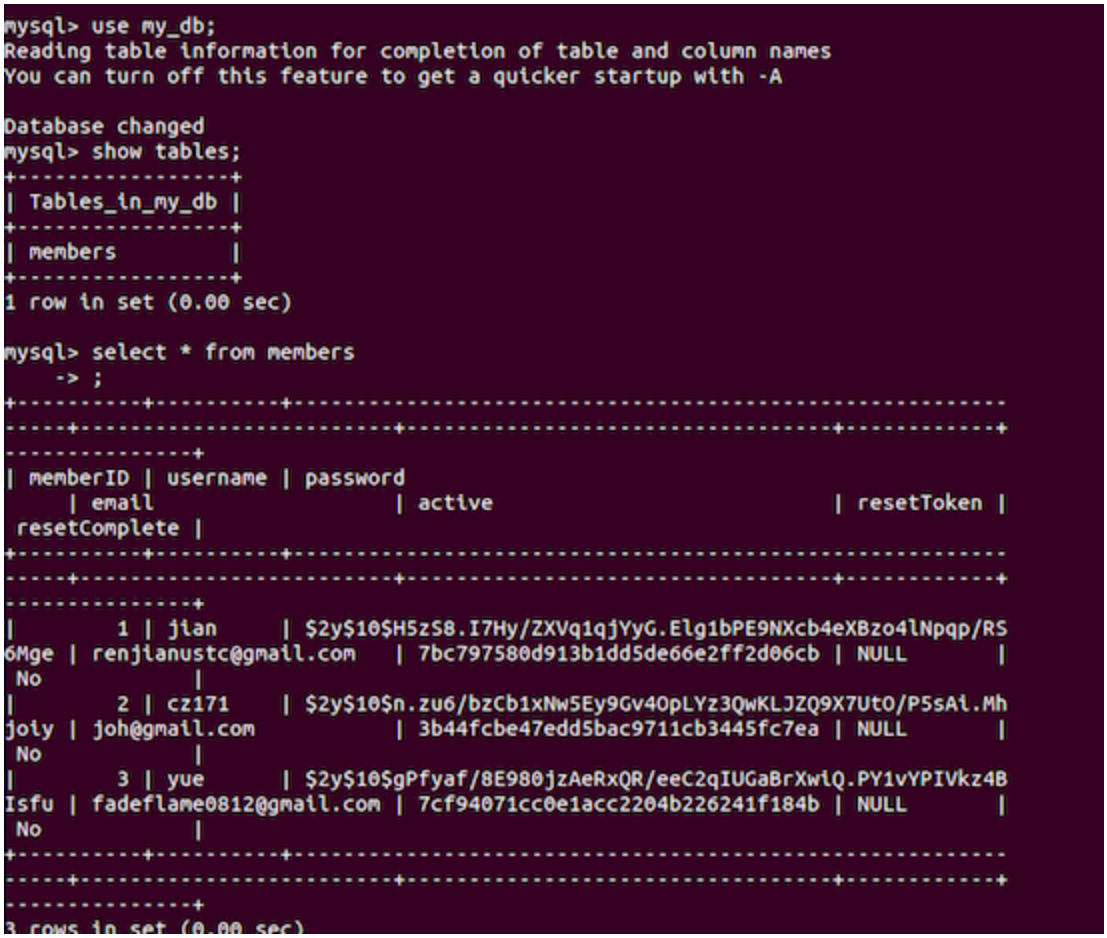


Figure 3.6 The login system database

Part IV

Web Service Design

The website we design can let new users to register and let the old users to login. Users are required to sign up an account before they can log into the page. Also, our website has many functions. For example, we can show the data of historical and real time, do the statistical analysis for the data, predict the price of the stock and so on.

4.1 Web Service

Our web service consists of four major functional blocks:
1. The framework to interact with the web client
2. The Prediction Algorithm
3. The module used to connect and query the database

4.2 List of files in Web Services

|-sign up
|----sign in
|-sign out
|-index.php
|-services.php
|----drawfigure.php
|-predict.php
|----tablelist.php
|--------StockNet.py
|----shortterm.php
|--------curve_fitting.py
|-contact.html

<==========FILE STRUCTURE==========>

[code]([] indicates folder) ---------- Source code and relative data of the project
	[classes] ---------- Encript password and store username into database
	[css] ---------- CSS for webpage
	[css_pirobox] ---------- CSS-support material
	[images] ---------- Images used in pages
	[includes] ---------- Database config for login system
	[js] ---------- Javascript and JQuery used in the project
	[layout] ---------- Header and footer in login pages
	[Prediction] ---------- Python script called to predict stock price
	about.html ---------- Introduction for the application
	contact.html ---------- Contact information of team members
	db.sql ---------- SQL command for creating user info database
	drawfigure.php ---------- Draw figure for historical data
	index.php ---------- Index page
	login.php ---------- Login to the system
	register.php ---------- Register page
	service.php ---------- Query from historical data
	shortterm.php ---------- Imported for short term prediction
	tablelist.php ---------- Imported for long term prediction

4.3 User Interface Design Implementation

4.3.1 Sign up

Firstly, all the customers are required to identify before logging into our system. Figure 4.1 shows the webpage for a user to register and login directly. The new customer needs to register first, the information of the new customer, such as user name, email and password will be save in our built database. While the old user can login our system by clicking login instead of registering.

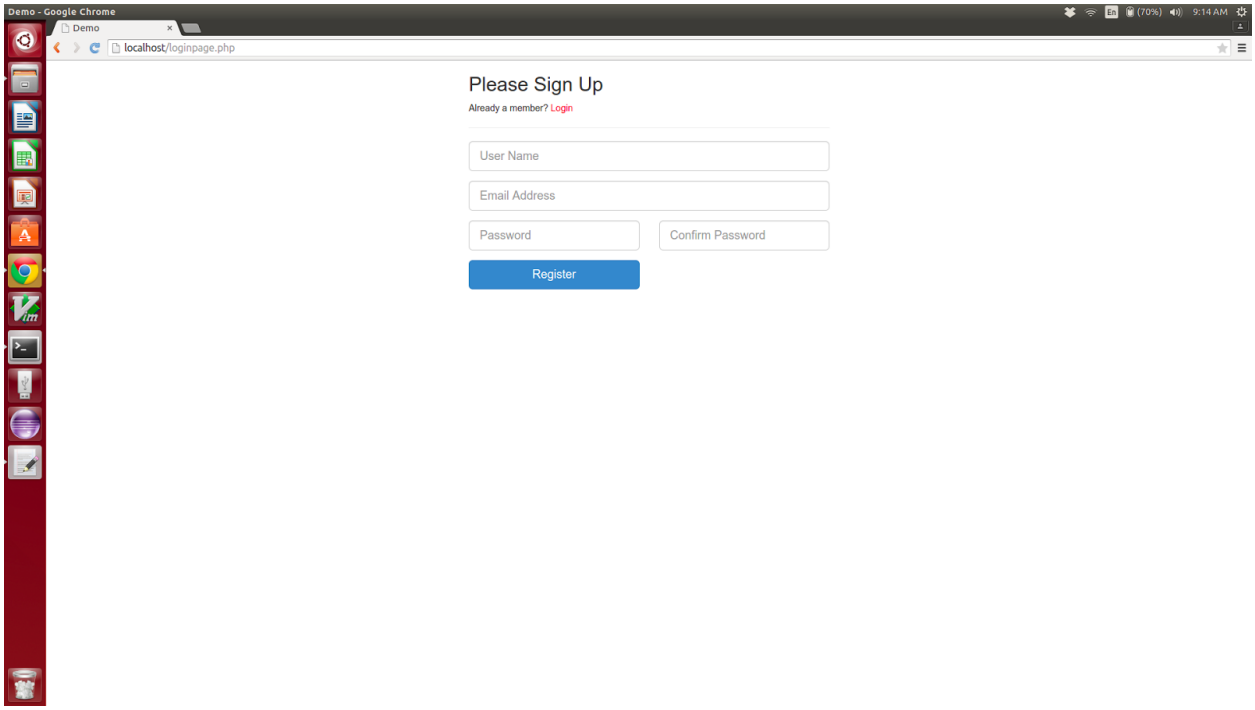


Figure 4.1 User signup and login webpage

4.3.2 Login

The older user can login our system without registering. Figure 4.2 shows the webpage that a user need to login before using our system. The older user need to input the user name and passord to enter the system.

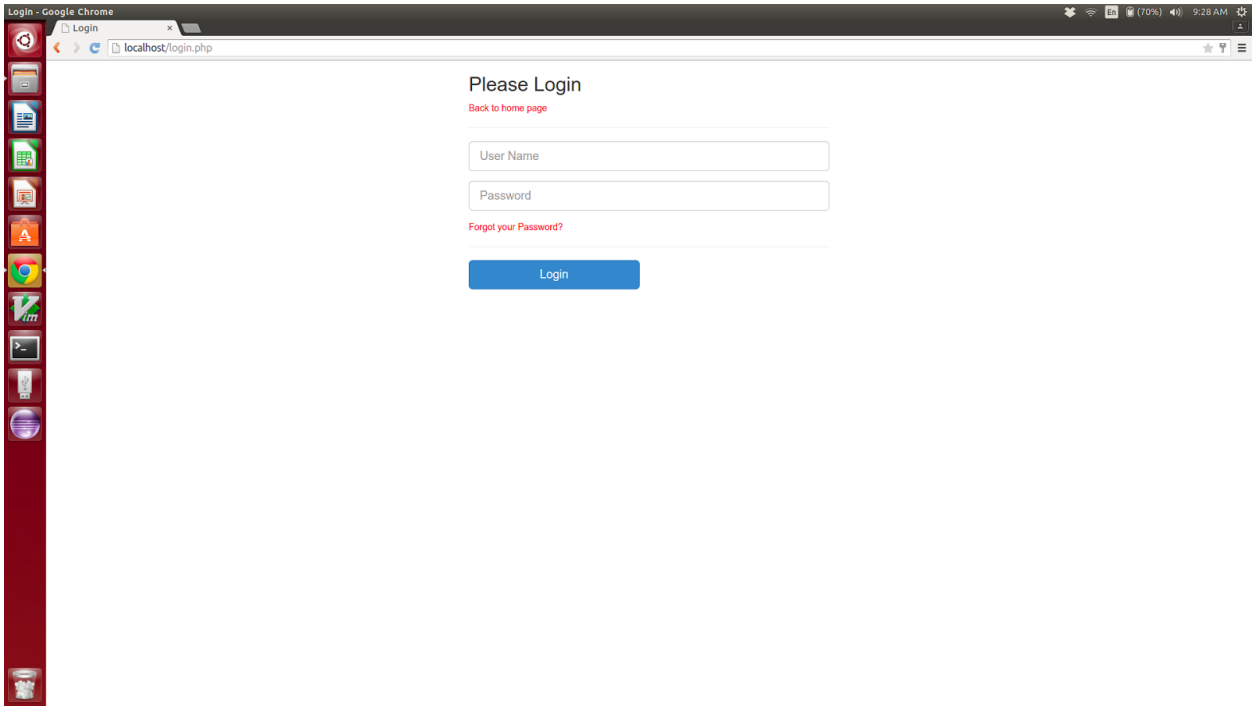


Figure 4.2 Older user can login our system directly.

4.3.3 Home Page

If a user can login to our system successfully, the a homepage will be shown. The name of our system is Stock Forecaster. There are five main parts of our system, which are home, about system, stock data, predict and contact us, as shown in Figure 4.3.

- In the first table below the title, four latest news and pictures are shown, if users want to see the whole news, they can simply click the picture and go the news website. From Figure 4.4, we can see different news can be shown in the home page. And the user can view the news by clicking the figure.

- The table "Stock Price" shows the stock price, volume and time of six well-known company stock.

- In the “What we do” table, we show customers the brief function of our website and tell customers what our website can do for them.

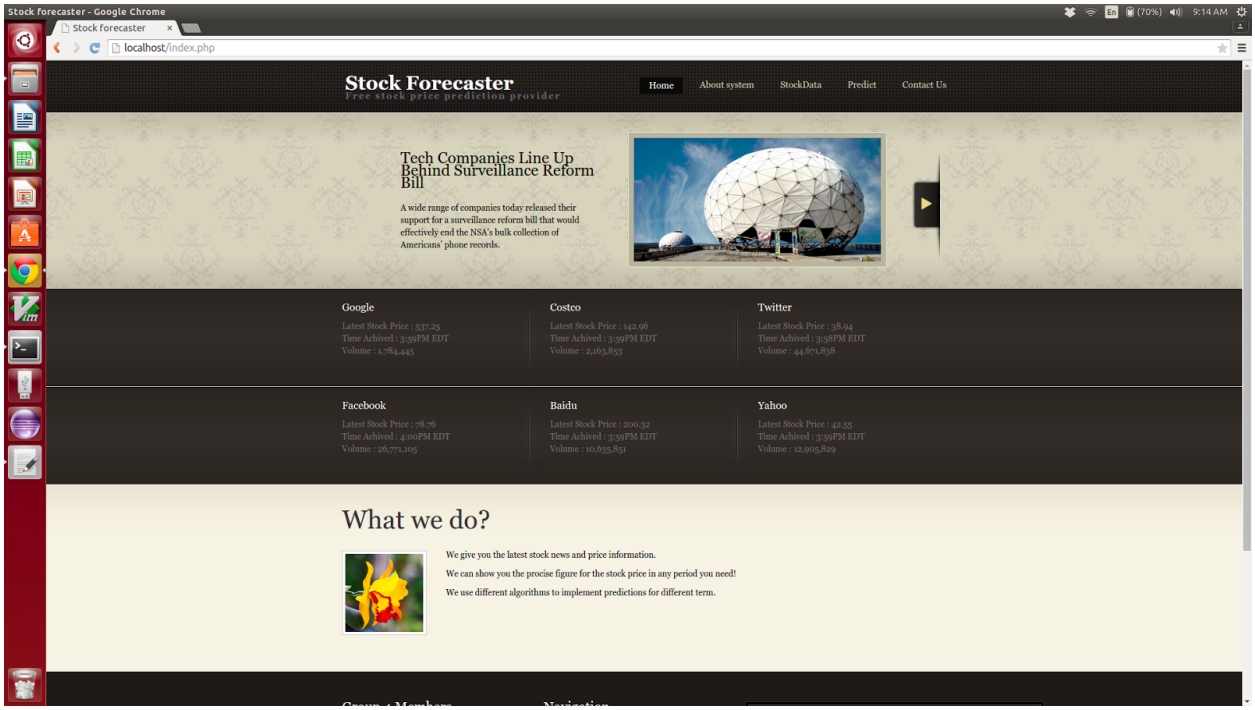


Figure 4.3 Home page of our system

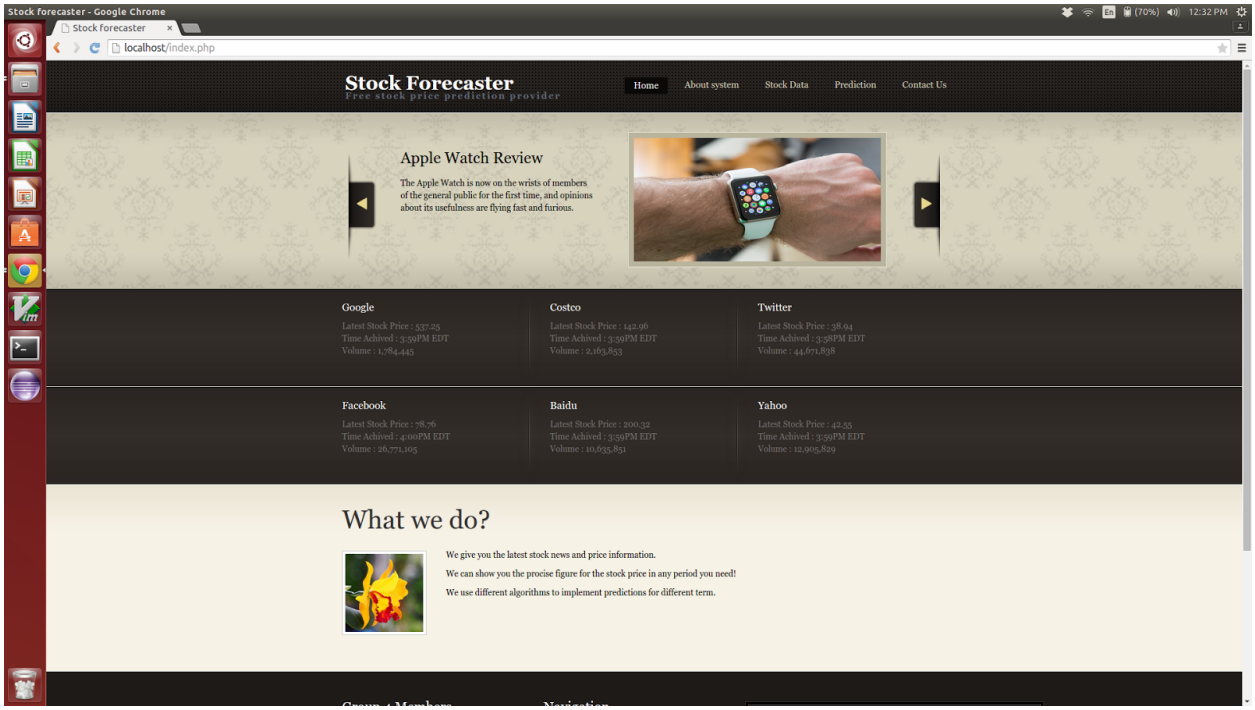


Figure 4.4 Several news will be shown in the homepage

4.3.4 Introduction of system

In this page, we introduce our prediction aim and algorithms, show our customers how we get the prediction and make investment decision for them. Also, user can click retrive stock info to search for achived stock data from historical database, click show chart to use online api to plot user's interested data and click prediction to predict long term or short term trendency.

If our customers want to read more about the algorithms, we also give the link of wikipedia for our two algorithms. User can click Wiki under Bayesian to see the details of Bayesian algorithm and click the wiki under artificial neural networking to see the detail of artificial neural networking.

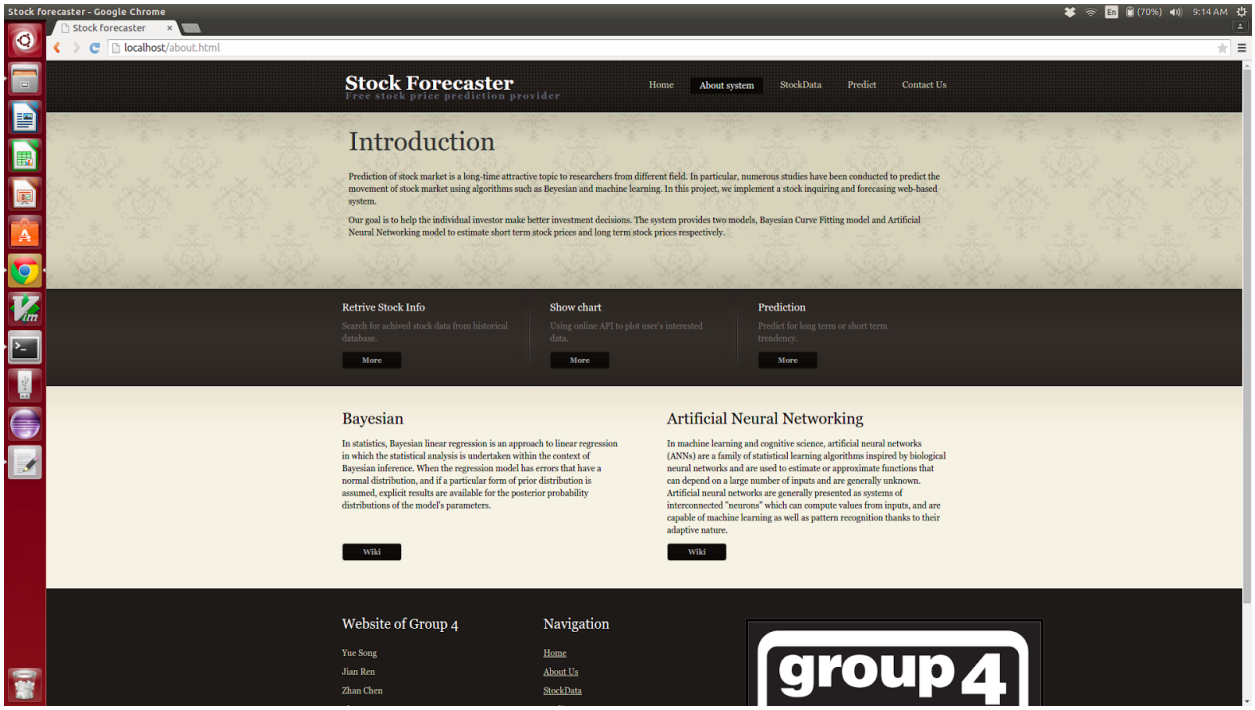


Figure 4.4 Webpage of about systems

4.3.5 Inquire for the historical stock data

In this webpage, user can view the stock data. User can select a stock from database, and the select the start date and end date. Then click submit, the figure will be shown below, as in Figure 4.5. The name of the selected stock will be shown under the figure along with the stock's price. Also, user can get the min stock price, max stock price, average stock price and

compare the price with other stocks by clicking the buttons. As shown in Figure 4.6, the select query will be shown on the webpage.

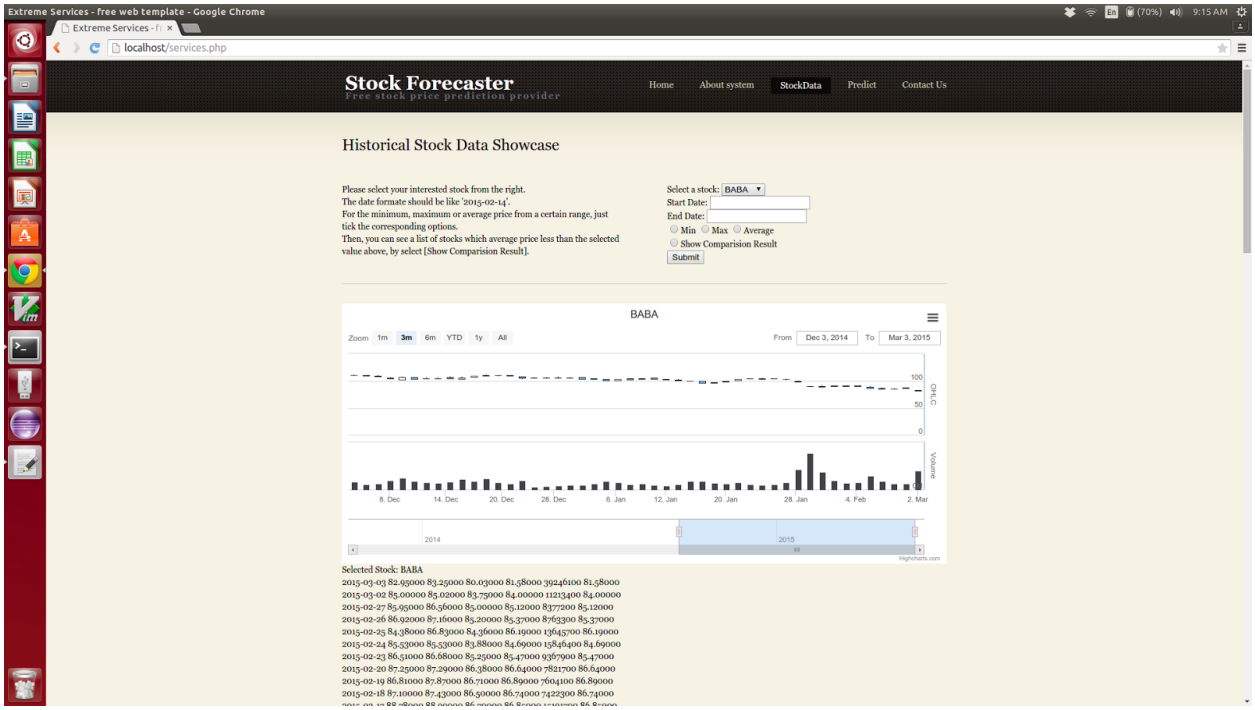


Figure 4.5 Select a stock and show the stock's information

Figure 4.6 Show the query of one stock

The time interval can be change by using different start date and end date. If you use different time interval, then the figure below will indicate the interval by using a blue box automatically. The price and the volume are both shown. Also, if the user put the mouse on the figure, the user can see the details of the stock at a specific time. For example, the user can view open price, close price, high price, low price and volume with the specific date. The name of the stock will be shown on the chart, as shown in Figure 4.7. Besides, the user can use zoom in buttons to view different time interval. There are 1m, 3m, 6m, YTD, 1y and all buttons to choose. For the figure that user may interested, user can download the figure with different formats, as shown in Figure 4.8.

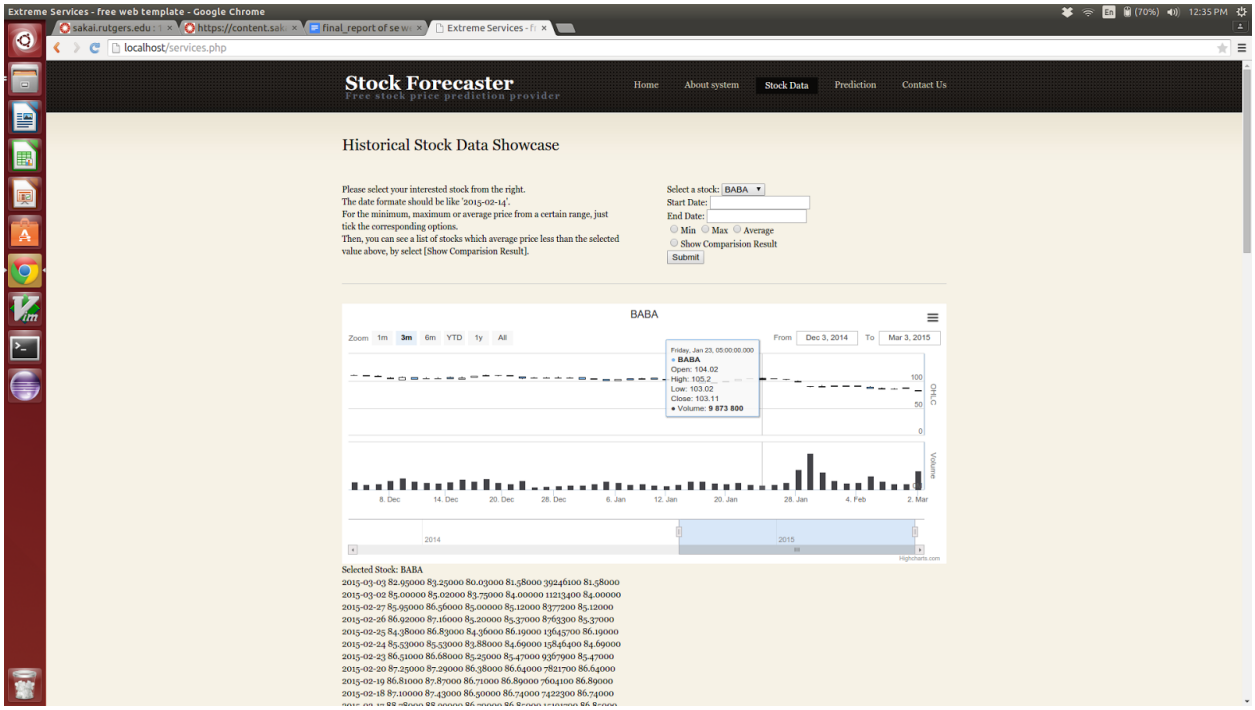


Figure 4.7 View the details of stock at each point

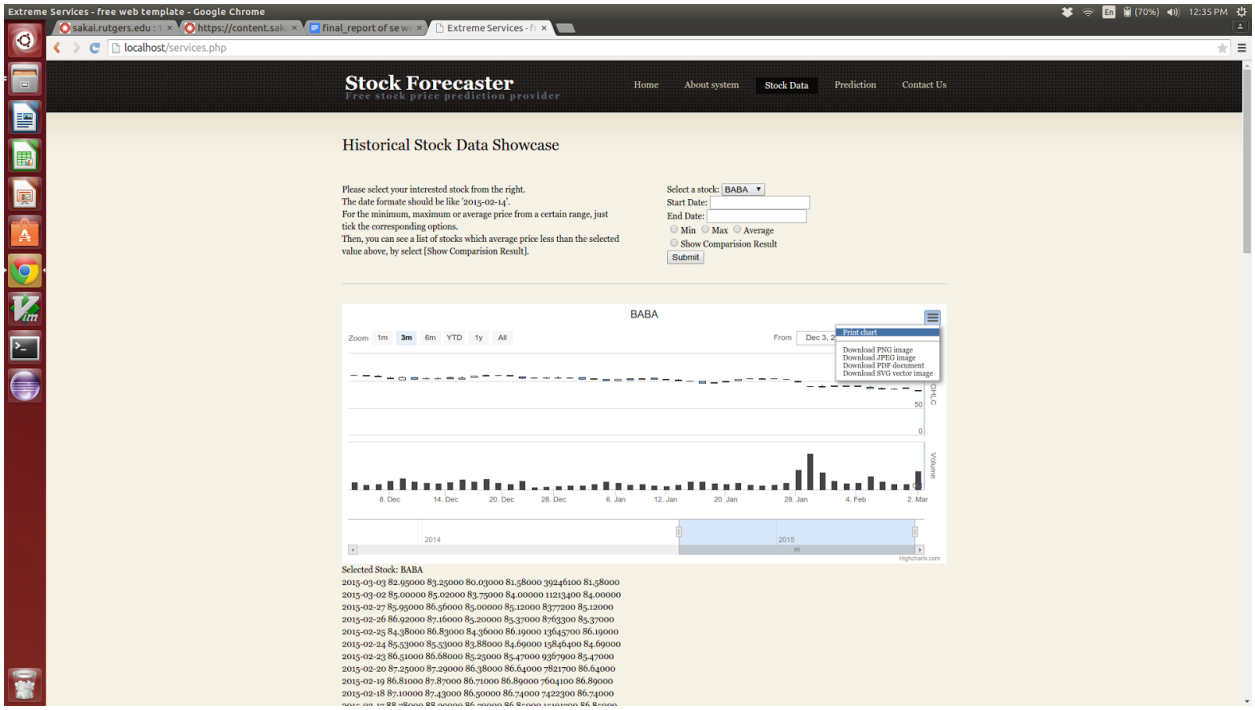


Figure 4.8 Save the figure as the interested format

4.3.6 Inquire for predicted price

In the page of predict, users can predict both the long term and short term strategy. For the short term strategy, we use Bayesian to predict the volume price. The training set is all the real time data price, and we predict 10 stock values. And we analyze the trend of previous data set, and calculate the the trend of predict values.

Compared with the statistical analysis of both previous value and predict value, we can give recommendation. We have four recommendation, which include strong buy, strong sell, negative hold and positive hold. For the example shown in Figure 4.9, the selected stock is baba and the recommendation for this stock is “strong buy” since it has a trend of increasing price.

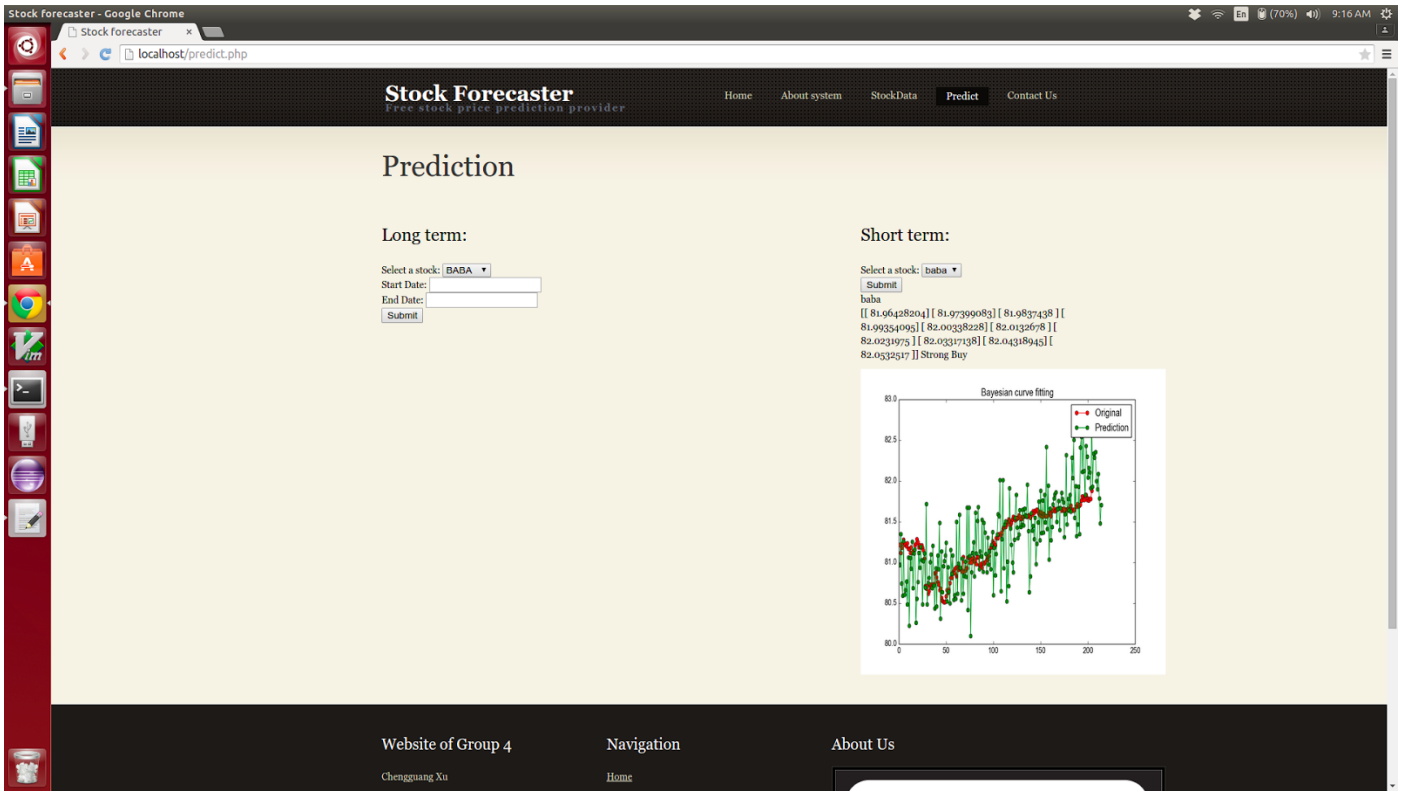


Figure 4.9 The prediction strategy of short term

For the long term prediction, we use ann to give recommendation. User need to select a stock and time interval, then click submit to give recommendation. We use ann to predict 30 prices and do the statistical analysis of the predict values and previous values and then give

recommendation. The predicted prices are shown in the webpage, along with stock name and recommendation.

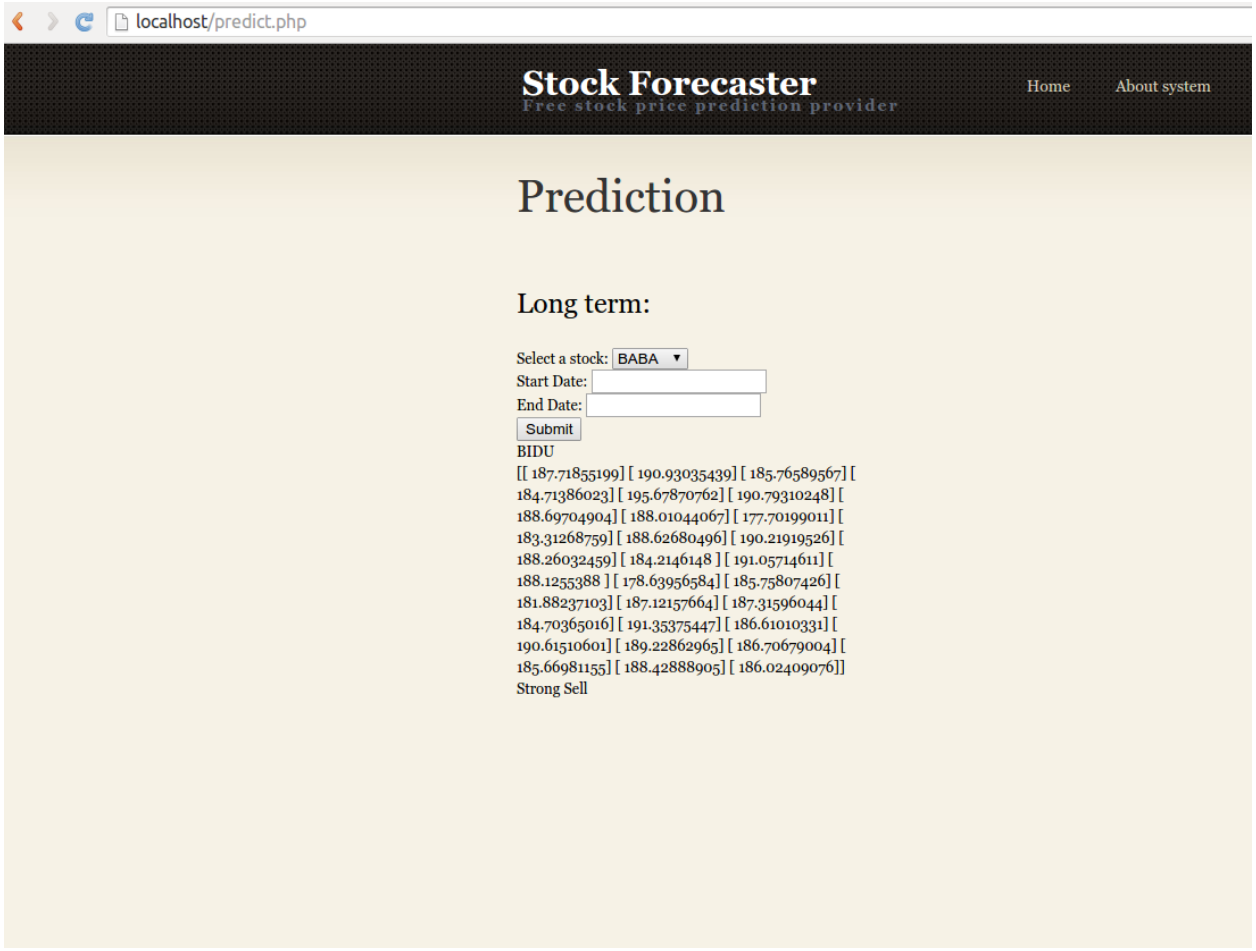


Figure 4.10 The prediction strategy of short term

4.3.7 Special features

The special features of our system includes friendly user interface, news recommendation, user login system, accuracy recommendation algorithm, price chart and so on.

Part V
Conclusion and future work

In this project, we propose a stock prediction system to predict stock values and give recommendation. It's a website based recommendation system with self-based databases. The user information and stock information are saved in database, and users can do the analysis via the website. To accelerate the recommendation accuracy and speed, more advanced computer hardwares are required to satisfy the computation requirements of ANN.

Part V
Reference

[1] Ivan Marsic “Software Engineering”
http://www.ece.rutgers.edu/~marsic/books/SE/ Sep 10, 2012.

[2] Web Based Stock Forecaster
https://content.sakai.rutgers.edu/access/content/attachment/cf199dad-2c83-4b7e-a84a-420a384c242e/Announcements/e14ea94d-69d6-427c-b0ce-2b180de4dbee/5.%20Sample%20Design%20Report.pdf March 12, 2014.

[3] Wikipedia:Stock Market Prediction
http://en.wikipedia.org/wiki/Stock_market_prediction

[4] Wikipedia:Bayes Estimator
http://en.wikipedia.org/wiki/Bayes_estimator

[5] Apache
http://httpd.apache.org/

[6] MySQL 5.6 Reference Manual
http://dev.mysql.com/doc/refman/5.6/en/index.html

[7] Curve Fitting & Multisensory Integration
http://www.cogsci.ucsd.edu/~ajyu/Teaching/Cogs202_sp14/Slides/lect3.pdf

[8] Highstock API Reference Book
http://api.highcharts.com/highstock

[9] PHP 5 Tutorial
http://www.w3schools.com/php/

[10] JavaScript reference

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

[11] PyBrain
http://pybrain.org/

[12] Prediction using neural networks: tutorial

http://www.obitko.com/tutorials/neural-network-prediction/

[13] Top 7 Technical Analysis Tools
http://www.investopedia.com/slide-show/tools-of-the-trade/

[14] Artificial neural network

http://en.wikipedia.org/wiki/Artificial_neural_network

[15] HTML(5) Tutorial

http://www.w3schools.com/html/

[16] Lippmann, R.P., “An introduction to computing with neural nets”, MIT Lincoln Lab., Lexington, MA, USA

[17] Lefteri H. Tsoukalas, Robert E. Uhrig, “Fuzzy and Neural Approaches in Engineering”, John Wiley & Sons, Inc, New York, NY, USA

[18] Peter Muller, Alaattin Erkanli and Mike West, “Bayesian curve fitting using multivariate normal mixtures