Soft Computing 6 (2002) 406 – 413 © Springer-Verlag 2002
DOI 10.1007/s00500-002-0193-7

Granular neural web agents for stock prediction

Y.-Q. Zhang, S. Akkaladevi, G. Vachtsevanos, T. Y. Lin

Abstract A granular neural Web-based stock prediction agent is developed using the granular neural network (GNN) that can discover fuzzy rules. Stock data sets are downloaded from www.yahoo.com website. These data sets are inserted into the database tables using a java program. Then, the GNN is trained using sample data for any stock. After learning from the past stock data, the GNN is able to use discover fuzzy rules to make future predictions. After doing simulations with six different stocks (msft, orcl, dow, csco, ibm, km), it is conclusive that the granular neural stock prediction agent is giving less average errors with large amount of past training data and high average errors in case of fewer amounts of past training data. Java Servlets, Java Script and jdbc are used. SQL is used as the back-end database. The performance of the GNN algorithm is compared with the performance of the BP algorithm by training the same set of data and predicting the future stock values. The average error of the GNN is less than that of BP algorithm.

Keywords Intelligent Web Agents, Granular Computing, Fuzzy Logic, Neural Networks, Stock Prediction, Data Mining

1 Introduction

Individuals, companies and governments spend vast quantities of resources attempting to predict financial markets. Careful thought will tell you that there can be no "common knowledge" system that can predict financial markets. Assuming there is such a model, then everybody would know the best time to sell. Ideally everybody would be selling at this time. If everybody is selling, then who is buying? Therefore, there cannot be any publicly available system for predicting markets. On the other hand, a prediction software would be very useful to assist individual in reaching a final decision. Then (assuming that it is possible to predict markets), this paper is developed which uses granular neural networks technique to predict the future stock values.

The lagged values of $y(t)$, i.e. $y(t), y(t-1), y(t-2), \ldots,$ can be used to construct a feature vector, which is input to the neural network. This can be viewed as a nonlinear generalization of the AR method described in [3]. The motivations for this nonlinear generalization is the following. First, Takens theorem [14] states that for a wide class of deterministic systems, there exists a one-to-one mapping between past values $y(t-1), y(t-2) \ldots,$ $y(t-d)$ of the time series and the state of the system at time t. This further implies that there exists a function g so that $y(t) = g[y(t-1), y(t-2) \ldots, y(t-d)]$. Second, it has been shown that a neural network is an universal approximator, capable of approximating any continuous function [1]. These statements together give the basic motivation for the use of artificial neural networks in time series prediction. Overall, time series forecasting provides reasonable accuracy over short periods of time, but the accuracy of time series forecasting diminishes sharply as the length of prediction increases.

Recurrent neural networks have an architecture where the outputs from intermediate neurons are fed back to the input layer. In this way the network will contain memory of previous values of the input data. A single value $y(t-1)$ is used as input and a single value $O(t)$ is produced as output from the neural network. The temporal dependencies are modeled in the weights in the feedback loops in the network. Applications of recurrent networks can be found in Ellman [2]. Since a moving average is a past estimate, a technical trader often misses a lot of the potential in the stock movement before the appropriate trading signal is generated. Thus, although technical analysis may yield insights into the market, its highly subjective nature and inherent time delay does not make it ideal for the fast, dynamic trading markets of today.

This sophisticated method replaces the connections between nodes in a feed-forward neural network with FIR filters. The number of necessary weights can sometimes be significantly reduced by assuming an implicit symmetry in the optimal network structure [15].

When compared to these techniques, the granular neural network (GNN) is able to discover fuzzy rules from

Y.-Q. Zhang (✉), S. Akkaladevi
Department of Computer Science,
Georgia State University,
Atlanta, GA 30303, USA
e-mail: yzhang@cs.gsu.edu

G. Vachtsevanos
The School of Electrical and Computer Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332–0250, USA

T. Y. Lin
Department of Mathematics and Computer Science,
San Jose State University,
San Jose, CA 95192, USA

The authors would like to thank the reviewers, Dr. Saeid Belkasim and Dr. Raj Sunderraman very much for their comments.

training data and then the trained GNN can predict future values for new inputs. The GNN is more useful and more effective to process, because it uses fuzzy neural networks based on the knowledge discovery. The basic strategy is to produce a system that predicts the price or the value of a property or index in the future.

The rest of this paper is organized as follows. Section 2 introduces what a market is, and how to predict markets. Section 3 gives different analytical models that can be used to predict stock markets. Section 4 discusses the GNN and the data mining algorithm stock prediction. Section 5 describes details on system implementation. Section 6 analyzes stock prediction performance of the fuzzy neural Web agent system. Section 7 summarizes conclusions and future work.

2 Theory about markets

Markets are filled with a certain energy and excitement. This excitement and emotion is brought about by the prospect of making a "buck", by buying and selling in the hope of getting rich. A market notorious for this behavior is the stock market. Many people buy and sell shares in a rush to make huge fortunes (usually huge losses).

Only through knowing future information about a particular market that nobody else knows can you hope to be able to make a definite profit. (A good example of this is insider trading – people with non-public information use this information to gain an unfair advantage.) As mentioned before, there are billions of dollars being spent on understanding capital markets. With this intelligence it hoped to be able to gain that competitive advantage. To make a capital gain through trading on the stock market, obviously requires knowing what the price will be in the future. The aim is purely to predict the value of the shares in the future.

You can bet this very question has been asked many times. If the future price movements of a particular market can be determined from using the related information, then markets are predictable. One theory that describes this dilemma is the Efficient Market Theory [16]. This theory states that the price of a share already includes all known information and any new information (the arrival of this information being random) is quickly incorporated. Therefore, because the arrival of this information is random then share price changes affected by this information will also be completely random.

The method is simple in principle. Look at all possible data relating to the particular market being studied. This adds up to a very large database. Producing a model for this would be near impossible. Another way is to use indexes or indicators. The reason for using indexes is to reduce the amount of raw data required for analysis. It is a form of pre-processing. The only problem is that some of these indicators are an average of what the market is doing and thus can give false impressions. How effective these indices are depends on what percentage of the market they represent. For example, the all ordinaries index is a relatively good indicator. It is calculated by dividing the aggregate value of companies listed on the stock exchange today by the aggregate value of the day before.

Today's charting packages have up to 160 predefined indices plus room for you to define your own in analyzing stock markets. Given the number of possible choices, what index/s do you use? Your choice depends on what you are attempting to model. If you are after daily changes in the stock market, then use daily figures of the all ordinaries index. If you are after long-term trends then use long-term indexes like the ten-year bond yield. Even if you use an indicator that is a good representation of the total market, it is still no guarantee of producing a successful result. The success depends heavily on the tool or model that use.

Today, there are many different models or tools available to help make investment decisions. These range from charting programs that analyze the trends in market value to financial reporting that determine the net value of company. Their success is highly debatable.

The charting programs are quite sophisticated. They are more like a package of statistical analysis software with everything from ninety day moving average to turtles analysis [10].

Company reports are a good tool for determining whether or not the company is over or under valued compared to its market value [6].

Another tool that is becoming fashionable is the use of neural networks. With their non-linear structures and the ability to generalize over a large number of inputs to determine patterns, makes them ideal for financial markets.

3 Analytical methods

Statistics, technical analysis, fundamental analysis, and linear regression are all used to attempt to predict and benefit from the market's direction. None of these techniques have proven to be a consistent correct prediction tool that is desired, and many analysts argue about the usefulness of many of the approaches. However, these methods are presented as they are commonly used in practice and represent a base-level standard for which neural networks should outperform. Also, many of these techniques are used to preprocess raw data inputs, and their results are fed into neural networks as input.

3.1 Technical analysis

The idea behind technical analysis is that, share prices move in trends dictated by the constantly changing attitudes of investors in response to different forces. Using price, volume, and open interest statistics, the technical analyst uses charts to predict future stock movements. Technical analysis rests on the assumption that history repeats itself and that future market direction can be determined by examining past prices. Thus, technical analysis is controversial and contradicts the efficient market hypopaper. However, it is used by approximately 90% of the major stock traders [11]. Despite its widespread use, technical analysis is criticized because it is highly subjective. Different individuals can interpret charts in different manners.

An example of a technical indicator is the moving average. The moving average averages stock prices over a given length of time allowing trends to be more visible.

Several trading rules have been developed which pertain to the moving average. For example, "when a closing price moves above a moving average a buy signal is generated."[11]. Unfortunately, these indicators often give false signals and lag the market. That is, since a moving average is a past estimate, a technical trader often misses a lot of the potential in the stock movement before the appropriate trading signal is generated. Thus, although technical analysis may yield insights into the market, its highly subjective nature and inherent time delay does not make it ideal for the fast, dynamic trading markets of today.

3.2 Fundamental analysis

Fundamental analysis involves the in-depth analysis of a company's performance and profitability to determine its share price. By studying the overall economic conditions, the company's competition, and other factors, it is possible to determine expected returns and the intrinsic value of shares. This type of analysis assumes that a share's current (and future) price depends on its intrinsic value and anticipated return on investment. As new information is released pertaining to the company's status, the expected return on the company's shares will change, which affects the stock price.

The advantages of fundamental analysis are its systematic approach and its ability to predict changes before they show up on the charts. Companies are compared with one another, and their growth prospects are related to the current economic environment. This allows the investor to become more familiar with the company. Unfortunately, it becomes harder to formalize all this knowledge for purposes of automation (with a neural network for example), and interpretation of this knowledge may be subjective. Also, it is hard to time the market using fundamental analysis. Although the outstanding information may warrant stock movement, the actual movement may be delayed due to unknown factors or until the rest of the market interprets the information in the same way. However, fundamental analysis is a superior method for long-term stability and growth.

3.3 Traditional time series forecasting

Time series forecasting analyzes past data and projects estimates of future data values. Basically, this method attempts to model a nonlinear function by a recurrence relation derived from past values. The recurrence relation can then be used to predict new values in the time series, which hopefully will be good approximations of the actual values. A detailed analysis and description of these models is beyond the scope of this paper. However, a short overview is presented as the results from these models are often compared with neural network performance.

There are two basic types of time series forecasting: univariate and multivariate. Univariate models, like Box–Jenkins, contain only one variable in the recurrence equation. Box–Jenkins is a complicated process of fitting data to appropriate model parameters. The equations used in the model contain past values of moving averages and prices. Box–Jenkins is good for short-term forecasting but requires a lot of data, and it is a complicated process to determine the appropriate model equations and parameters.

Multivariate models are univariate models expanded to "discover casual factors that affect the behavior of the data."[11]. As the name suggests, these models contain more than one variable in their equations. Regression analysis is a multivariate model, which has been frequently compared with neural networks. Overall, time series forecasting provides reasonable accuracy over short periods of time, but the accuracy of time series forecasting diminishes sharply as the length of prediction increases.

3.4 The efficient market hypopaper

The efficient market hypopaper (EMH) states that at any time, the price of a share fully captures all known information about the share. Since all known information is used optimally by market participants, price variations are random, as new information occurs randomly. Thus, share prices perform a "random walk", and it is not possible for an investor to beat the market.

The EMH is important because it contradicts all other forms of analysis. If it is impossible to beat the market, then technical, fundamental, or time series analysis should lead to no better performance than random guessing. The fact that many market participants can consistently beat the market is an indication that the EMH may not be true in practice. The EMH may be true in the ideal world with equal information distribution, but markets contain several privileged players who can outperform markets by using inside information or other means.

3.5 Other techniques

Many other computer-based techniques have been employed to forecast the stock market. They range from charting programs to sophisticated expert systems. Fuzzy logic has also been used. Expert systems process knowledge sequentially and formulate it into rules. They can be used to formulate trading rules based on technical indicators. In this capacity, expert systems can be used in conjunction with neural networks to predict the market. In such a combined system, the neural network can perform its prediction, while the expert system could validate the prediction based on its well-known trading rules. The advantage of expert systems is that they can explain how they derive their results. With neural networks, it is difficult to analyze the importance of input data and how the network derived its results. However, neural networks are faster because they execute in parallel and are more fault tolerant.

The major problem with applying expert systems to the stock market is the difficultly in formulating knowledge of the markets because we ourselves do not completely understand them. Neural networks have an advantage over expert systems because they can extract rules without having them explicitly formalized. In a highly chaotic and only partially understood environment, such as the stock market, this is an important factor. It is hard to extract information from experts and formalize it in a way usable by expert systems. Expert systems are only good within their domain of knowledge and do not work well when there is missing or incomplete information. Neural networks handle dynamic data better and can generalize and make "educated guesses." Thus, neural networks are more suited to the stock market environment than expert systems.

3.6 Comparing the various models

In the wide variety of different modeling techniques presented so far, every technique has its own set of supporters and detractors and vastly differing benefits and shortcomings. The common goal in all the methods is predicting future market movements from past information. The assumptions made by each method dictate its performance and its application to the markets.

The EMH assumes that fully disseminated information results in an unpredictable random market. Thus, no analysis technique can consistently beat the market as others will use it, and its gains will be nullified. I believe that the EMH has some merit theoretically, but in real-world applications, it is painfully obvious that there is a uneven playing field. Some market participants have more information or tools, which allow them to beat the market or even, manipulate it. Thus, stock market prices are not simply a random walk, but are derived from a dynamic system with complexities to vast to be fully accounted for.

If an investor does not believe in the EMH, the other models offer a variety of possibilities. Technical analysis assumes history repeats itself and noticeable patterns can be discerned in investor behavior by examining charts. Fundamental analysis helps the long-term investor measure intrinsic value of shares and their future direction by assuming investors make rational investment decisions.

In conclusion, these methods work best when employed together. The major benefit of using a neural network then is for the network to learn how to use these methods in combination effectively, and hopefully learn how the market behaves as a factor of our collective consciousness.

4 Granular neural networks

The granular neural networks (GNN) is capable of processing various granular data (granules) [18, 19]. Granules could be a class of numbers, a cluster of images, a set of concepts, a group of objects, a category of data, etc. These granules are inputs and outputs of GNNs as multimedia data are inputs and outputs of biological neural networks in the human brain.

To train the above GNN, the heuristic-knowledge-based learning algorithm (HLA) [19] is used. The simplified HLA with a constant $\gamma (\gamma \in [0,1])$ is described below:

Suppose: Given n-dimensional input data vectors $x^p (i.e., x^p = (x_1^p, x_2^p, \ldots, x_n^p))$ and one-dimensional output data vector y^p for $p = 1, 2, \ldots, N$. The energy function is defined by

$$E^p = \frac{1}{2}[f(x_1^p, \ldots, x_n^p) - y^p]^2 \tag{1}$$

For simplicity, let E and f^p denote E^p and $f(x_1^p, \ldots, x_n^p)$, respectively.

Step 1: Begin.
Step 2: Heuristic-knowledge-based initialization of parameters.
Step 3: Gradient descending learning from data.

For simplicity, we define

$$\gamma = 1 - \gamma^k + \frac{\gamma^k}{n} \tag{2}$$

$$\phi = \frac{1}{L}\sum_{i=1}^{n} w_k^i \frac{(x_i - a_i^k)}{\sigma_i^k} \tag{3}$$

$$\theta = \frac{(f^p - y^p)z_k^y}{\sum_{k=1}^{m} z_k^y} \tag{4}$$

$$z_k = \frac{1}{L}\prod_{i=1}^{n} \mu_{A_i^k}(x_i^k) \tag{5}$$

and w_i^k are weights. Then we can get the following learning algorithms for $i = 1, 2, \ldots\ldots, n, k = 1, 2, \ldots, m$, $p = 1, 2, \ldots, N, t = 0, 1, 2, \ldots,$ and learning rate $\lambda > 0$.

Step 3.1: Train b^k

$$\left.\frac{\partial E}{\partial b^k}\right|_t = \theta\big|_t \tag{6}$$

$$b^k(t+1) = b^k(t) - \lambda \left.\frac{\partial E}{\partial b^k}\right|_t \tag{7}$$

Step 3.2: Train η^k

$$\left.\frac{\partial E}{\partial \eta^k}\right|_t = \theta\phi \tag{8}$$

$$\eta^k(t+1) = \eta^k(t) - \lambda \left.\frac{\partial E}{\partial \eta^k}\right|_t \tag{9}$$

Step 3.3: Train a_i^k

$$\left.\frac{\partial E}{\partial a_i^k}\right|_t = \frac{\theta}{n(\sigma_i^k)^2}\left[2n\gamma(a - \eta^k\sigma_i^k w_i^k)\right] \tag{10}$$

$$a = 2n\gamma(x_i^p - a_i^k)(b^k - f^p + \eta^k\phi) \tag{11}$$

$$a_i^k(t+1) = a_i^k(t) - \lambda \left.\frac{\partial E}{\partial a_i^k}\right|_t \tag{12}$$

Step 3.4: Train σ_i^k

$$\left.\frac{\partial E}{\partial \sigma_i^k}\right|_t = \frac{\partial E}{\partial a_i^k}\frac{(x_i^p - a_i^k)}{\sigma_i^k} \tag{13}$$

$$\sigma_i^k(t+1) = \sigma_i^k(t) - \lambda \left.\frac{\partial E}{\partial \sigma_i^k}\right|_t \tag{14}$$

Step 3.5: Train $wleft_i^k$ and $wright_i^k$

IF $x_i^p \leq a_i^k$
THEN

$$\left.\frac{\partial E}{\partial wleft_i^k}\right|_t = \frac{\theta\eta^k(x_i^p - a_i^k)}{n\sigma_i^k} \tag{15}$$

$$wleft_i^k(t+1) = wleft_i^k(t) - \lambda \frac{\partial E}{\partial wleft_i^k}\bigg|_t \tag{16}$$

ELSE

$$\frac{\partial E}{\partial wright_i^k}\bigg|_t = \frac{\theta \eta^k (x_i^p - a_i^k)}{n\sigma_i^k} \tag{17}$$

$$wright_i^k(t+1) = wright_i^k(t) - \lambda \frac{\partial E}{\partial wright_i^k}\bigg|_t \tag{18}$$

Step 4: Discovering Fuzzy Knowledge (Fuzzy IF-THEN Rules).
Once the learning procedure has been completed, all parameters for a FNNKD have been adjusted and optimized. As a result, all *m* fuzzy rules have been discovered from training data. Finally, the trained FNNKD can generate new values for new given input data.

Step 5: End.

The network design consists of 3 inputs and 1 output. For stock prediction the 3 inputs are open, high, low values of past-historical data of a particular stock company and 1 output is the close amount for each day in the historical data.

5 System implementation

To implement this stock prediction system, Java Servlets, Java Script and Jdbc are used. SQL is used as the back-end database.

5.1 Concepts of Input Data

The system can predict future for any stock or market index. For example to predict Intel stock values for some days ahead, historical data (past) is required. You can change the prediction to one-day, one-week, or one-year ahead by substituting the historical data.

Data for neural networks is probably the most important aspect for training. How well the network performs is very dependent on the quality of the input data.

A data set is a collection of variables used to make a prediction. Examples of the variables are various stocks and indexes, and each stocks date, open, low, high, close and volume data. The output of the Predictor is a distribution chart or values from which we can easily understand the future prospects of that particular stock. Database tables have been generated using stock-historical data from www.vahoo.com site.

Without user intervention a particular stock data in the form of an Excel spreadsheet is obtained. Preprocessing operation is needed as preparatory step for next stage. As an example of preprocessing, the downloadable data is in DD-MM-YY format. The format should be changed to DD-MM-YYYY so that it can be properly inserted into the database tables. The data is saved in ASCII format. A Java program does the process of inserting this data into the database. This program coverts the whole file line by line into values and inserts them into the table. The same name is used for the text file as the stock symbol name. So the program uses the text file name to create a table with the same name.

Each line of the data set must contain 5 values date, open, high, low, close values of the stock. Each value is separated by space (see Table 1).

The names of the stocks used and their corresponding symbols are given in Table 2.

Here filer is the file reading java program and orcl is the input text file, after the execution of the program, it creates a table called orcl with all the past historical data.

5.2 Software parameters

The prediction algorithm takes all these parameters as input. This algorithm is called when we click predict future from the system after entering the stock symbol. This algorithm makes the neural network learn. The algorithm returns the future values as output witch are eventually stored in results table for each stock. We will keep track of these results until we click on the average error for all simulations for this particular stock.simulationresults.java program displays all the results of predictions for any particular stock at a time on the web page and clears the contents of the table. This means that now the table is ready to store any new predictions on that stock symbol.

5.3 Overview of implementation

A full run of the program implementation will be described, going through all the main features of the program:

- Download the historical data from the Internet.
- Copy the whole data into a text and run the program, which inserts the data into the database.
- A program is written through which users can buy and sell the stock shares, and the corresponding data is stored in the database.
- A program is written through which each user can see his/her own transactions.

Table 1. Meaning of data parameters

dd/mm/yyyy	Date	Date of the day
Double	Open	Price at the beginning
Double	Low	Lowest traded price
Double	High	Highest traded price
Double	Close	Price of the last trade
Double	Volume	Total number of stocks

Table 2. Stock symbols

Stock name	Stock symbol
Oracle	orcl
Cisco	csco
Dow Chemical Co	dow
Microsoft	msft
IBM	ibm
K Mart Corp	km

- Algorithm, which trains the granular neural networks using the mean square error as stop criterion for learning, while never exceeding the maximum number of cycles which can take testing data from the initial date to the user entered date, and predicts the future stock closing values.
- A program is written, which compares the predicted values with real values.

One of the most important factors here is to construct a neural network deciding on what the network will learn. A neural network must be trained on some input data. The two major problems in implementing the training are

- Defining the set of input to be used (the learning environment)
- Deciding on an algorithm to train the network

For the input data, I used past historical data and granular neural network method to train the network. The network is trained properly without over training.

When the program is used for prediction on new values, the network should be trained up to the date to be predicted, before making forecast. A new learning set has to be made, which contains the values up to the desired date. It is also possible to make an entirely new prediction (i.e. a prediction where the target is unknown). In this case we do not compare the predicted values with the real values because we do not have real values for the future.

6
Performance analysis
Using the developed system to predict the future stock values using granular neural networks we can do some simulations to know the performance of the algorithm.

6.1
Predicting a stock using complete data
By using the complete past historical data, if we predict stock values for future 30 days from the algorithm we are now able to compare the predicted values with the real values. The average error for this simulation is 1.582. In the graph the dotted curve shows the predicted values and the solid curve shows the real values in Fig. 1.

6.2
Predicting a stock using less data
In another sample simulation, we take the data from 1981 to 1994 and predict the stock values for some future days. The average error for this simulation is 2.11. The graph is shown in Fig. 2. The dotted curve shows the predicted values and the solid curve shows the real values.

The average error for this simulation is greater than the previous simulation. In this graph the results are not close to the real values compared to the previous chart. So based on average errors and the graphs we can conclude that, more the data we have the better training the neural network gets and gives more close results. This means that, more the available data for predicting financial markets, the greater the chances of an accurate forecast.

6.3
Predicting a stock with different errors
For the same test case if we decrease the maximum training error parameter from 0.0001 to 0.000015, we are getting more close results. For dow stock the curve is shown in Fig. 3. The dotted curve shows the predicted values with error 0.0001 and the solid dotted lines shows the predicted values with maximum training error 0.000015 and the solid curve is the real data.

The average error is only 1.05 when compared to 1.582 in case of high training error. By comparing average error and the graphs, we can conclude that, the resulting future stock values are closer than the future stock values with high training error.

Simulations are done on six stocks (msft, orcl, dow, csco, ibm, km). From the simulation results it is conclusive that, the average error for simulations using lot of data is small compared to the average error using less amount of data and the more data for training the neural network, the better prediction it gives.

6.4
Comparison between GNN and BP
The performance of the GNN algorithm is compared with the performance of the BP algorithm by training the same set of data and predicting the future stock values. If the

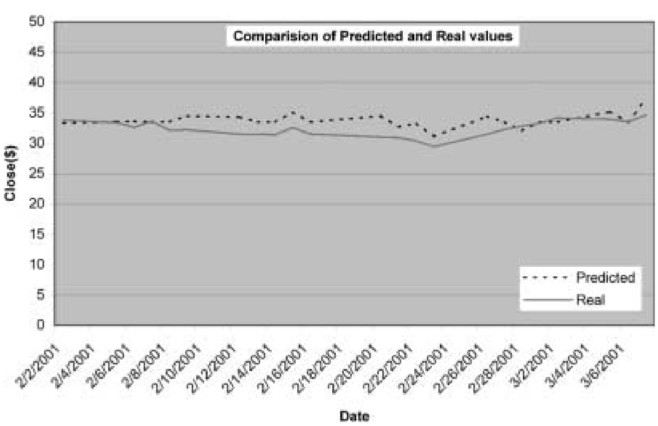

Fig. 1. Graph of predicted and real values for dow stock using complete data

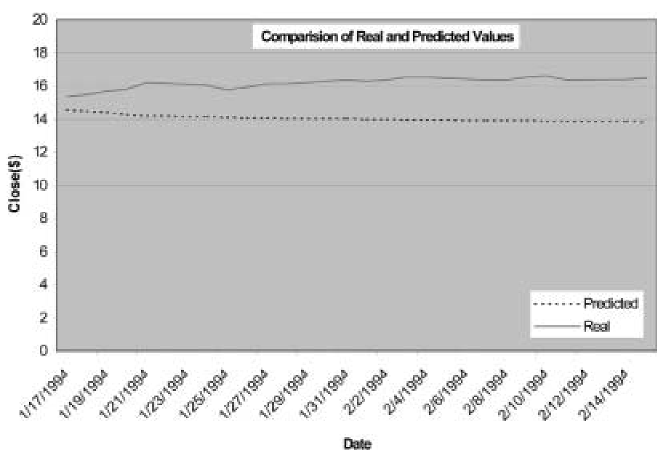

Fig. 2. Graph of predicted and real values for dow stock using less data

training error was set at 0.03 and the neural network was trained for dow stock data using both the algorithms. The GNN took 2 min 58 s to train the neural network where as BP took 2 hours and 55 minutes. The GNN's average error was 1.39 where as BP gave 3.38. The graph for this data with date on X-axis and the closing value in dollars on the Y-axis is as shown in Fig. 4. The average error for GNN is less compared to the average error for BP algorithm. From the average error and the graph it is conclusive that, GNN produced closer future stock values with the real stock values compared to the BP algorithm using less training error. If the training error was set at 0.07 and the neural network was trained for csco stock data using both the algorithms. The GNN took 2 min to train the neural network where as BP took 1 hour and 48 minutes. The GNN's average error was 6.09 where as BP gave 7.16. The graph for this data with date on X-axis and the closing value in dollars on the Y-axis is as shown in Fig. 5. The average error for GNN is less compared to the average error for BP algorithm. From the average error and the graph it is conclusive that, GNN produced closer future stock values

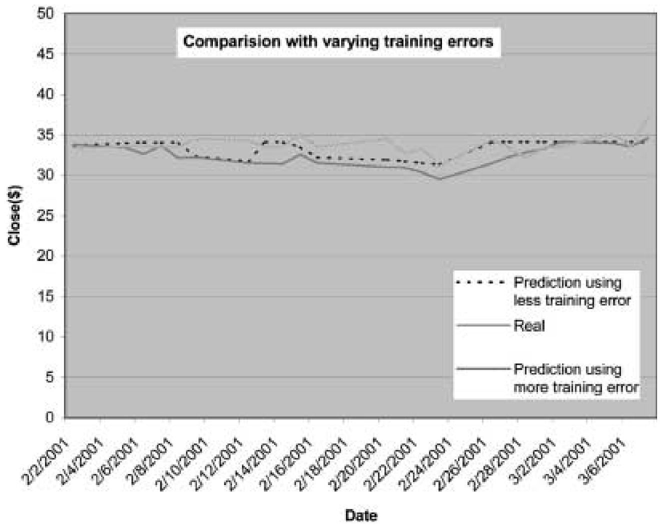

Fig. 3. Graph of predicted and real values for dow stock with varying training error

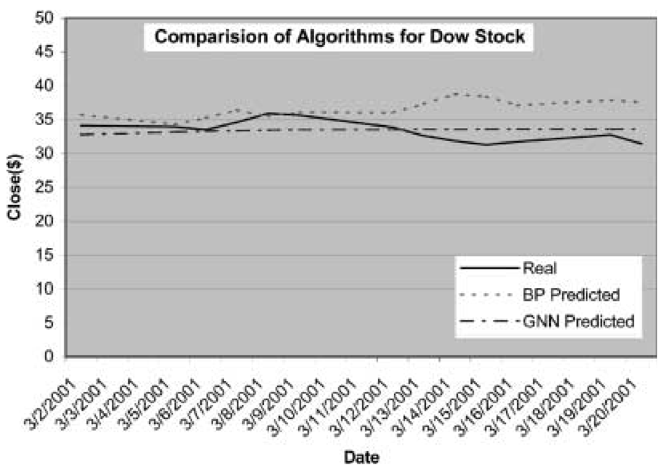

Fig. 4. Comparison of algorithms with low training error

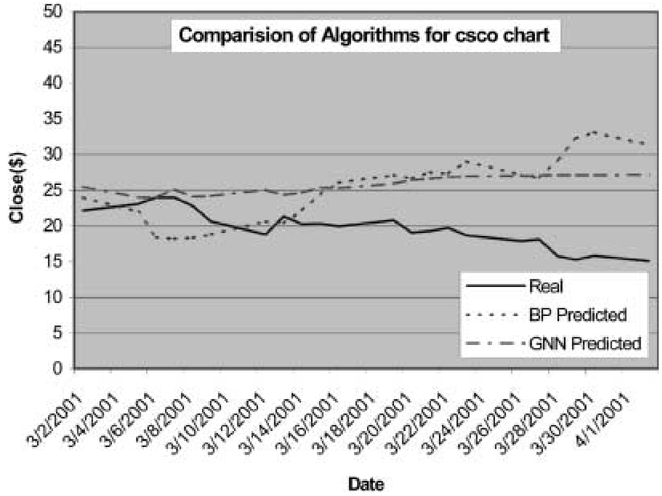

Fig. 5. Comparison of algorithms with high training error

with the real stock values compared to the BP algorithm using less training error. Based on the above two simulations, the overall performance with GNN technique is better than BP technique.

7 Conclusions

After completing several simulations for predicting several stocks based on the past historical data, it is conclusive that, the average error for simulations using lot of data is small compared to the average error using less amount of data. This means that, the more data for training the neural network, the better prediction it gives. If the training error is low, predicted stock values are close to the real stock values. The results are good when we use the GNN algorithm when compared to the BP algorithm.

One possibility for future work is to update the system, so that it can read the past stock data automatically from the web and store them in the database. In this way the system will become Internet ready for predicting any stock market and is ready at any time. Another possibility for future work is to update the system, which can allow to trade the stock, which means users can manage to buy and sell the stock after seeing the prediction values from this system. In this way the system can eventually become Internet ready to be used anywhere in the world at any time.

The system can be updated, so that it will consider the other stock information as inputs to train the neural network. Then the system would become more reliable in the real world. The system can also be updated for mutual fund applications, which is similar to the stock prediction application. The system can be updated so that the, stock prediction results from all simulations can be compared with the existing neural network techniques such as mat lab or which are existing on online on the internet.

It is also possible to make the fuzzy neural Web-based stock prediction agent system as a commercial application by updating such that it gives more user-friendly functionality and by giving more valuable information to the users.

References

1. **Cybenko G** (1998) Approximation by superpositions of a sigmoidal function, Technical Report, University of Illinois
2. **Ellman J** (1990) Finding structure in time, Cognitive Sci pp. 179–211
3. **HellStrom T, Hollmstrom K** (1998) Predicting the Stock Market, Malardalen University, Technical Report Screen IMa-TOM-1997–2007
4. **Hertz J, Anders K, Palmer G** (1991) Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Company
5. http://www.yahoo.com/
6. **Huntley I, ShareHolder** (1996) Top 500 Companies, Ian Huntley Pty Limited
7. **Karayiannis N, Venetsanopoulos A** (1993) Artificial neural networks. Learning Algorithms, Performance Evaluation and Applications, Kluwer Academic Publishers
8. **Klimasauskas C** (1993) Applying neural networks. In: Neural Networks in Finance and Investing, Probus Publishing Company, pp. 47–72
9. **Lawrence R** (1997) Using Neural Networks to Forecast Stock Market Prices. University of Manitoba
10. http://www.metastock.com
11. **Robert J, Eyden V** (1996) The Application of Neural Networks in the Forecasting of Share Prices, Finance and Technology Publishing
12. **Sing A** (1997) Application of neural networks for predicting financial markets. University of Queensland
13. **Steiner M, Wittkemper H-G** (1995) Neural networks as an alternative stock market model, John Wiley and Sons, pp. 137–148
14. **Takens F** (1981) Detecting strange attractors in fluid turbulence, Dynamical Systems and Turbulence, Springer
15. **Wan EA** (1994) Time Series prediction by using a connectionist network with internal delay lines, Time Series Prediction. Forecasting the Future and Understanding the Past, SFI Series in the Sciences of Complexity, ProcAddison-Wesley
16. **White H** (1988) Economic prediction using neural networks: The case of IBM daily stock returns, Proceedings of the IEEE International Conference on Neural Networks (ICNN'88), San Diego, California, USA
17. **Yoon Y, Swales G** (1993) Predicting stock price performance: a neural network approach. In: Neural Networks in Finance and Investing, Probus Publishing Company, pp. 329–342
18. **Zhang Y-Q, Fraser MD, Gagliano RA, Kandel A** (2000) Granular neural networks for numerical-linguistic data fusion and knowledge discovery, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, IEEE Trans Neural Networks **11**(3): 658–667
19. **Zhang Y-Q, Kandel A** (1998) Compensatory genetic fuzzy neural networks and their applications, Series in Machine Perception Artificial Intelligence, **30**, World Scientific