A Novel Neural Network for Data Mining

Tony Kai Yun Chan*, Eng Chong Tan and Neeraj Haralalka
School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore.

(* Author to contact: Tony Kai Yun Chan, email: askychan@ntu.edu.sg)

Abstract

The algorithm proposed here for data mining deals with the standard Multilayer Perceptron using Temporal Backpropagation algorithm with the concept of Bollinger Band Crossover from the concept of Trading Systems added to it, and is referred to as the Bollinger Band Crossover Supervised Network (BBCSN). To visualize the performances of each of this algorithm, a portfolio management scheme was designed and translated into code using Visual C++. This enables a critical analysis of the algorithm, with respect to their financial performances. According to the results obtained in this project, the newly proposed and designed Bollinger Band Crossover gives much better results than generally obtained from a Multilayer Perceptron network.

1. Introduction

There is justifiable scepticism surrounding the idea that it is possible to make money by predicting price changes in a given market based only on its past behaviour and a number of publicly available indicators. Ignoring the cases involving insider information, this scepticism exists because of a variety of reasons; many of which are explained by the *efficient market hypothesis* [1]. The hypothesis states that the market follows a random walk that cannot be predicted by the past prices. Any chance of potential profits is snapped up immediately, removing the opportunity almost as soon as it is created, and certainly before the technical analyst has seen it in the data. The hypothesis relies on perfect knowledge, implying that perfect prediction technology would only serve to enforce conditions in which that technology is useless. As soon as a profit is predicted, it is snapped up and expected profits return to the level of risk free return plus a risk premium associated with a stock holding. In other words, chances of superior profits occur when the brokers set their prices incorrectly and the investors are able to spot the discrepancy before it is corrected.

However, there is a possible way around the efficient market hypothesis because it relies on the *public availability* of market information. If prices do not follow a *random walk*, but a *chaotic one*, then anybody who is able to model the price structure and make valid predictions using the model will have access to information that is not publicly available [2]. The efficient market hypothesis will not apply to that person until everyone gains access to the same technology and things even out once more. This would present the person with a well-designed and capable model with tremendous opportunities for profit. This forms a major attraction to develop good models, which can analyse the chaotic stock series, and construct a predictive model.

In short, the project aims at researching and developing a complete application, which provides the user with a suitable technique to perform data mining and prediction on the stock series of his choice, and presents him the results of the project in the form of a *visualization graph* for his own analysis. The data mining algorithm poses significant research opportunities as there are various approaches available and each of them has its own advantages and drawbacks.

Neural networks are probably the most common data mining technique, sometimes synonymous with data mining. They are simple models of neural interconnections in brains, adapted for use on digital computers. They learn from a training set and generalize patterns inside it for classification and prediction. They can be applied in both undirected data mining and in time series prediction. Back propagation neural networks are most successfully used for time series prediction and provide a reasonably good consistent performance.

The network proposed here utilises the Back-propagation algorithm, with modifications to include the temporal factor and the concept of Bollinger Band Crossover in it. This network is known as the Bollinger Band Crossover Supervised Network (BBCSN).

2. Related work

Neural Networks has also been used to perform temporal series predictions, but in general the backpropagation algorithm is used as the standard algorithm [4], [5], [6]. However, in this particular project, the supervised backpropagation algorithm is combined with the concept of Bollinger Band Crossover to give the BBCSN algorithm, which in general is expected to give a better performance than the supervised algorithm itself. On a general basis, Neural Networks have often been used to perform financial predictions but not in combination with the above-mentioned Trading System algorithms. Moreover, many Portfolio Management schemes have been formulated before, but this one has the exception of trading only if the return provided by investing almost assures the investor of a profit greater than what he would get if he invested the same principal in a bank or private limited company.

3. Algorithm design

Data Acquisition is the feeder module to the data-mining algorithm. It includes the collection of data from the specified data source (Web Server, Local Database etc) and pre-processing of the collected data before feeding into the Data Mining Algorithm (DMA). It has to provide the input data for both the training and prediction phase of the DMA.

Data is central to any data-mining project. It is needed for the proper training and initialization of the Data Mining Algorithm, which is the essence of the project. The data from the metastock database is transformed to text file representation for every stock ticker. The text file has the Price (High, Low, and Close) and the Volume of the stock ticker with the corresponding date listed in a tabular fashion. The length of the time period for which the data is available is different for different tickers, some ranging for as long as past seven years while others as less as just the past three years. Owing to the huge data repository needs of the program, popular stocks with databases of over five years are selected for training, testing and performance evaluation. The database offers four different attributes, each of which can be selected for the mining and prediction algorithm. These attributes are volume, high price, low price and close price.

Each of these attributes appears to be quite independent of each other after a superficial view. However, a deeper view reveals that these attributes are rather closely interrelated and follow each other. For example, each of these three price attributes defines different financial status for the same ticker, within the same day. Hence, there is not going to be a wide variation in general (barring the cases of extreme volatility such as a market crash), and any one of these three attributes can be chosen to represent the ticker's price. The volume information too is related to the price information in general, with a greater number of units being traded on days of high market volatility, indicated by larger intra-day price fluctuations. The Intra-Day High price of a ticker is chosen to represent the price of a ticker. Hence, the time series fed in to the DMA comprises of the intra-day high price of the ticker sorted by time.

It has to be definitely understood and agreed upon the fact that the nature of the stock data is generally a chaotic one. There are no limits to the ticker price, which can shoot up to great heights or dip down to zero level, all within a very short period, depending on the market volatility and company performance. This poses a great difficulty to the data-mining algorithm, which can get disturbed by the large fluctuations in the price, to loose the essence of the series. Furthermore, the activation function used in the data mining algorithms is a *bounded function* and will not be capable of exceeding its upper bound, which the stock data can easily overshoot. This can cause inconsistencies in both the training and prediction phase, and confuse the learning mechanism.

To avoid this pitfall, all the data in the input series is *normalised to a value between zero and one*, which allows the data mining algorithm to comprehend the data more intelligently and make valid deductions from the input series.The algorithm involves a well-known and tested temporal back-propagation algorithm using Input Delayed Neural Networks (IDNN). To improve performance of this network, the concept of Bollinger Band Crossover is used in combination with this supervised network and this network is called the Bollinger Band Crossover Supervised Network (BBCSN). Bollinger Band Crossover Supervised Network (BBCSN) is an adaptation of the popular Multi-layer Feed Forward Networks which are commonly used as benchmarks for determining the performance of any other network. A multi-layer feed forward network typically comprises of a set of sensory neurons (source nodes) that constitute the *input layer*, one or more *hidden layers* of computation nodes, and a set of motor neurons that constitute the *output layer*. The input signal propagates through the network in a forward direction, on a layer by layer basis. These networks are also known as Multi-Layer Perceptrons (MLP) and are a generalisation of single layer perceptrons. Multi-layer perceptrons are trained in a supervised manner with the *error back-propagation algorithm*. This algorithm is based on the *error correction learning rule*. It consists of two passes through the different layers of the network: a forward pass and a backward pass. In the *forward* pass an activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual responses of the network. During the *backward* pass, the weights of the network are adjusted in accordance with an error correction rule.

The relations of the back propagation algorithm can be summarized as follows:

$$\begin{pmatrix} Weight \\ correction \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} Learning \\ rate \\ \eta \end{pmatrix} \begin{pmatrix} Local \\ gradient \\ \delta_j(n) \end{pmatrix} \begin{pmatrix} Input \\ signal \\ y_j(n) \end{pmatrix}$$

The local gradient $\delta_j(n)$ depends upon whether the neuron *j* is an output node or a hidden node:

(i) If neuron *j* is an output node, $\delta_j(n)$equals the product of the derivative $\varphi_j'(v_j(n))$and the error signal $e_j(n)$, both of which are associated with neuron *j*.

(ii) If neuron j is a hidden node, $\delta_j(n)$ equals the product of the associated derivative $\varphi_j'(v_j(n))$ and the weighted sum of the δs computed for the neurons in the next hidden or output layer that are connected to the neuron j.

4. Architecture

A Bollinger Band Crossover Supervised Network (BBCSN) (variation of the Multilayer Feed Forward Network) has been designed to perform *data mining* on the stock database provided. The architecture of the network comprises of three layers, two hidden and one output layer as shown in Fig.1. The output layer comprises of a single neuron, the output of which is the prediction of the network for the next time instant. The hidden layers have twenty and fifteen neurons respectively. All the layers use unipolar sigmoidal function as the *transfer* function.

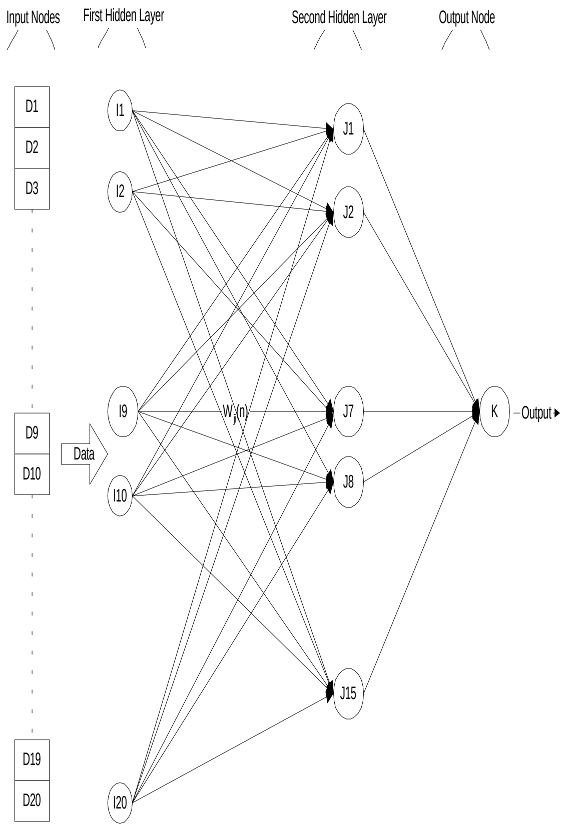

Fig.1 Architecture of the MFFN

5. Adding of Bollinger band crossover

It has been observed that if Trading Bands were constructed based on volatility of price, they would contract and expand according to market forces. A more precise match of price movement could then be achieved, helping to better identify overbought and oversold levels.

In the construction of Bollinger Bands, standard deviation of price (SD) is used as the measure of volatility. SD is determined in the following manner:

If, P = Number of periods considered, N_{sd} = Number of Standard Deviations used, and C = Closing price.

Then, MA = P-Period of the Exponential Moving Average of the current day's close price,

SUM = $(C_1 - MA)^2 + (C_2 - MA)^2 + \ldots + (C_p - MA)^2$, and SD = $(SUM^{1/2})/P$

For each current value, plot MA + (N_{sd} * SD) to derive the predicted value. This is the value plotted on the graph to depict the predicted value as got from the Bollinger Band Crossover Supervised Network (BBCSN).

6. Data mining process

The input is in the form of an ordered series, which has the intra-day high price of the stock ticker to be predicted. Each unit in the series consists of the intra-day high price of the ticker and the date identifying the temporal position of the unit in the series. The data mining algorithms discussed in the previous section do not process the full series at one go. They process the series in a sequence of patterns [7], the length of whom being determined by the *Window Size* of the time series. The window size determines the amount of time series visible to the data mining algorithms in one processing cycle. Training constitutes the first phase of the data mining algorithms. Each of these algorithms has to be trained over the series that they are going to predict, which leads us to a paradox. If we have the series before the prediction, then what is the need of going through all the algorithms to predict it? This is a valid question to be raised and attempts to reduce the whole scope of the project to trivia. However, the algorithms do perform *prediction* in the real sense of the word, as the full time series is not visible to the algorithm during the training phase. The input series is divided into training and testing series, with the first part of the series being used to train the network, which goes on to predict the remaining part, thus attempting to predict the portion of the series previously unknown to the network. The training series is determined by allowing the network to keep learning until the error is below a set level (convergence parameter).

Once the training phase of the algorithm is completed, it is ready to predict the series on which it is trained upon. The prediction process is very similar to the training cycle, except for the evaluation and parameter adjustment steps. Prediction can be *one-point* or *m-point* in time. *One-Point* prediction implies that the algorithm will predict the point just after the end of the sliding window that forms the input to the algorithm. *M-Point* prediction implies that the algorithm will attempt to predict m points in advance. However, the prediction mode is dependent on the training mode, which should be the same. Hence for *m-point* prediction, the training should have been carried out in the same mode. The algorithms are capable of performing both *one-point* and *m-point* predictions, although the performance varies. A detailed discussion of the algorithm performances and the strengths and drawbacks of each algorithm is given in the section on Performance Analysis. One-point prediction in general gives the best performances of each algorithm and the error factor increases geometrically with *m*, the degree of prediction attempted.

7. Experimental results

Fig. 2 shows the graph of the actual data and the predicted data against time (approximately 5 years), of the stock ticker JadeTech as predicted by the supervised BBCSN Network. As can be seen from the graph, the predicted values are very close to the actual values and thus this network is able to predict the values quite accurately. Even further reduction in the amount of training data did not have any negative impact on the prediction and the network was able to predict as accurately as before.

Fig. 3 shows the portfolio of JadeTech, got by using the supervised BBCSN Network. The portfolio starts with an initial value of 10,000 with 'Liquid Cash' at the beginning of the transactions being 10,000 and 'Portfolio Value' also being 10,000 since no stocks have been invested in yet. After that there is a lot of buying and selling done over approximately 1000 transactions resulting in the final Portfolio Value being approximately 21,000 thus resulting in a profit of 11,000. Liquid Cash at the end is approximately 12,000 thereby signifying that about 9,000 is still invested in stocks. It is to be noted that during the 1000 transactions, there was active buying and selling of stock almost all the time as can be seen in the figure.

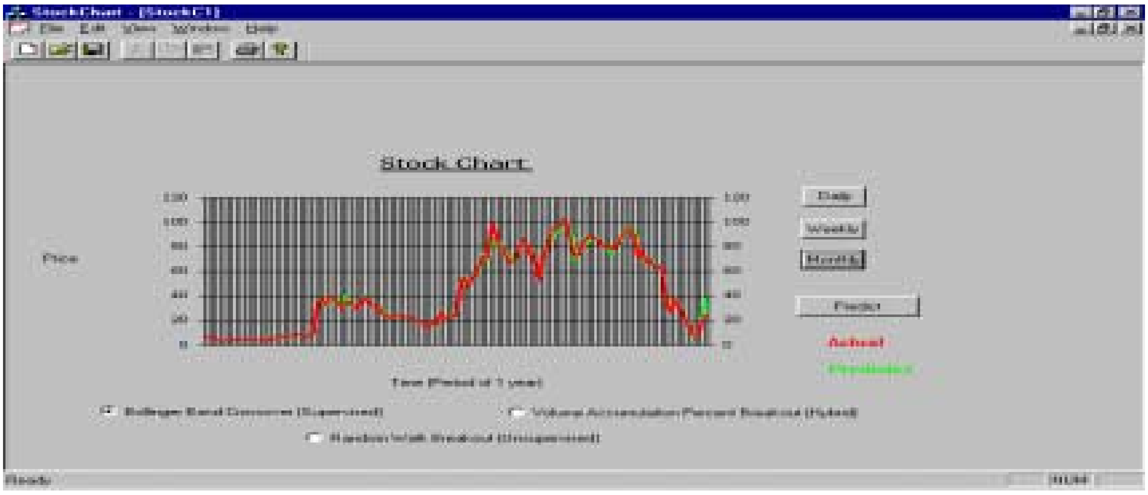

Fig. 2: Prediction of *JadeTech* by the BBCSN Network

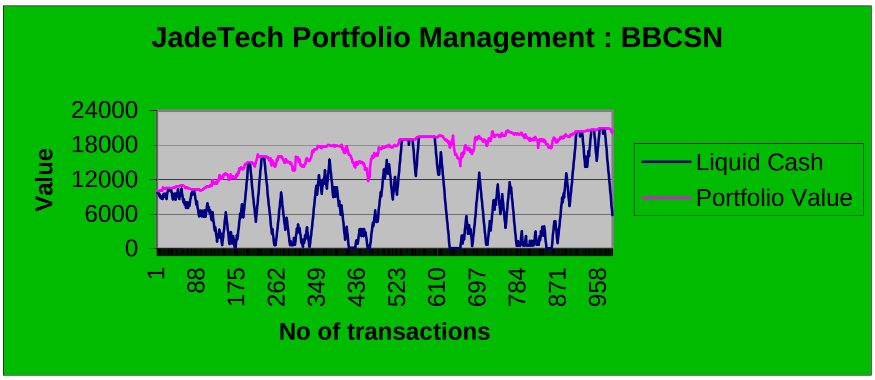

Fig. 3: Portfolio of *JadeTech* by the BBCSN Network

8. Conclusion

The Supervised BBCSN networks perform with reasonable amount of accuracy in terms of prediction. However, it can be thrown off balance if the input series is very chaotic in nature. Sometimes, though due to the introduction of the concept of Bollinger Band Crossover, it may perform even better than its original performance capability. This algorithm has a very good performance when it comes to predicting the temporal series and using intelligent investment strategies to benefit from the knowledge gained ahead of time. Portfolio analysis of this algorithm also provides interesting insights into its workings. The analysis involved trading via a particular number of transactions on the stock exchange and trying to extract the maximum amount of profit that this algorithms can provide us with. This algorithms enables the user to work around the *efficient market hypothesis* and provide him with knowledge that is not available to everyone. This causes opportunities of profit to be translated into real cash profit. Thus, it can be concluded that data mining for prediction of stocks does hold a very good promise for the future.

9. References

[1] Newbold P. and Bo T., Introductory Business and Economic Forecasting, 1994.

[2] Sharpe W. F., ‘Likely Gains from Market Timing’, *Financial Analysts Journal*, Vol. 31, No. 2, pp 60-69, 1975.

[3] Brock W. A., Hsieh D. A. and Baron L., ‘Nonlinear Dynamics, Chaos and Instability, *MIT Press*, 1991.

[4] Han J., Lu H. and Feng L., ‘Stock Movement Prediction and N-Dimensional Inter Transaction Association Rules’, 1998.

[5] Harp S., Samad T. and Guha A, ‘Toward The Genetic Synthesis of Neural Networks’, *Proceedings of 3rd International Conference on Genetic Algorithms* 1989.

[6] John G. H. and Miller P., ‘Building Long/Short Portfolios Using Rule Induction’, *IEEE Conference in Computational Intelligence in Financial Engineering*, 1996.

[7] Agarwal R. and Srikant R., ‘Mining Sequential Patterns’, *Proceedings of the 20th Conference on Very Large Databases*, pp 478-499, September 1994.

[8] Kwon Y. and Han I., ‘The Prediction of Industry Stock Index Using Artificial Neural Networks: Case of Construction Industry and Banking’, 1997.