Expert Systems with Applications 36 (2009) 7313–7317

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Short-term stock price prediction based on echo state networks

Xiaowei Lin *, Zehong Yang, Yixu Song

State Key Laboratory of Intelligent Technology and System, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, FIT 1-509, Beijing 100084, China

ARTICLE INFO

Keywords:
Echo state network
Neural networks
Short-term price prediction
Principle component analysis

ABSTRACT

Neural network has been popular in time series prediction in financial areas because of their advantages in handling nonlinear systems. This paper presents a study of using a novel recurrent neural network–echo state network (ESN) to predict the next closing price in stock markets. The Hurst exponent is applied to adaptively determine initial transient and choose sub-series with greatest predictability during training. The experiment results on nearly all stocks of S&P 500 demonstrate that ESN outperforms other conventional neural networks in most cases. Experiments also indicate that if we include principle component analysis (PCA) to filter noise in data pretreatment and choose appropriate parameters, we can effectively prevent coarse prediction performance. But in most cases PCA improves the prediction accuracy only a little.

1. Introduction

Financial time series prediction in stock market has received focus from investors, speculators and researchers for a long time due to its potential profits. Advances in both analytical and computational methods have led to a number of interesting new approaches to financial time series mining, based on nonlinear and non-stationary models (Armono, Marchesi, & Murru, 2005). Among various methods, artificial neural networks (ANNs) are the most widely used approach to analyze the stochastic nonlinear system and show good performance in many cases. Applications of ANNs on stock markets range from stock price or index forecast, trend prediction to classification. ANNs are considered suitable for financial time series forecast because they can approximate any function (Kim & Shin, 2007), have effective learning algorithms, handle noisy data and use inputs of different kinds (Armono et al., 2005).

Back-propagation neural network (BPNN), time delay neural network (TDNN) and recurrent neural network (RNN) are three popular networks implemented in stock markets. Oh and Kim (2002) propose a stock trading model whose core component includes two phases of BPNNs. The best RMSE it achieves is 0.0782. Saad, Prokhorov, and Wunsch (1998) exploit time delay and recurrent neural network to predict stock trends. They can achieve the false alarm rate as low as zero but miss most of profit opportunities. Kim and Shin (2007) investigate the adaptive time delay neural networks (ATNNs) and the time delay neural networks (TDNNs), with the GAs in detecting temporal patterns for stock market prediction tasks. The MSE of GA-ATNN is 3.38 and that of GA-TDNN is 3.49. Both of them improve the prediction accuracy of standard TDNN. Lee and Liu (2001) propose a neural oscillatory-based recurrent network for finance prediction to provide both long-term and short-term stock price forecast. The recognition rate of overall pattern is 91%. The best average percentage error for long-term prediction is 1.073% and that for short-term prediction is 1.473%.

However, all the three networks have their own limitations, which may influence their prediction performance when applied in stock market. BPNN can learn only an input–output mapping of static (or spatial) patterns that is independent of time (Kim & Shin, 2007). TDNN seems possibly the simplest choice for representing a wide range of mappings between past and present values (Sitte & Sitte, 2000). But the fixed time delays in a TDNN remain constant throughout training after initialization, which takes a risk of a mismatch between the choice of time delay values and temporal location of important information in the input patterns (Kim & Shin, 2007). RNN offers some benefits over BPNN because their "memory feature" can be used to extract time dependencies in the data (Lawrence, 1997), but because traditional RNN algorithms based on gradient descent approach are notorious for slow convergence and high computational cost (Jaeger, 2002), RNN is difficult to develop in application.

In addition, past studies in stock data mining have another common limitation that the sample size in their experiments is too small. Few studies show the generalization performance of their models in a whole dataset, such as all 500 stocks in S&P 500. It makes the experiment results biased to some degree because good results in several stocks of a specific period hardly guarantee that

* Corresponding author. Tel.: +86 10 62777703.
E-mail addresses: linxw05@mails.tsinghua.edu.cn (X. Lin), yangzehong@sina.com (Z. Yang), songyixu@sohu.com (Y. Song).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.09.049

the model is reliable. Moreover, it also makes the comparison between different models little sense. Comparison between several stocks is not sufficient to judge different models.

Echo state network (ESN) is a novel RNN recently proposed by Jaeger and Haas (2004). Its basic idea is to use a large "reservoir" RNN as a supplier of interesting dynamics from which the desired output is combined (Jaeger, 2002). The "reservoir" can remember the past history input and the memory will decay when time moves on. This feature coincides with our observation of stock markets. ESN was first applied to predict chaotic time series and obtained the best result on the benchmark task of Mackey–Glass series prediction. Until now, ESN has been implemented in wireless communication (Jaeger & Haas, 2004), robot control (Ishii, van der Zant, Becanovic, & Ploger, 2004), "figure 8" generation task (Jaeger, Lukosevicius, & Popovici, 2007), speech recognition (Jaeger et al., 2007; Skowronski & Harris, 2007), etc. Unfortunately, there is still no study concerning the application of ESN in financial area.

In this paper, we investigate the effectiveness of ESN to predict the future stock prices in a short-term. The Hurst exponent is utilized to choose a persistent sub-series with the greatest predictability for training from the original training set. A stock prediction system is built to forecast the closing price of the next trading day according to the history prices and technical indicators. We test nearly all the stocks in S&P 500 in a specific period of time and compare the results of ESN with some conventional neural networks, such as BPNN, Elman neural network and radial basis-function neural network (RBFNN). The experiments present that ESN outperforms other neural networks in most cases. To solve the problem that ESN nearly fails to predict the future price of some stocks, we also applied PCA to filter noise and extract reliable representation of raw data. The experiment results indicate that the combination of PCA and adjustment of parameters effectively avoid exceptional prediction performance.

The rest of this paper is organized as follows: Section 2 introduces ESN and presents our prediction system. Section 3 describes our experiments based on different neural networks, shows the results on daily prices prediction and gives our preliminary analysis. Finally Section 4 makes a conclusion and proposes some advice for future research.

2. Echo state network and stock prediction system

Echo state network was first proposed by Jaeger and Haas (2004) to learn nonlinear systems and predict chaotic time series. Comparing with traditional neural networks, ESN has the following advantages. First, the state of ESN contains information about the past input history in a way which reflects the recent history well and decays with the delay time. This feature of short-term memory solves the problem to add memory and determine the range of time delay values that many previous neural networks meet. Second, the training of ESN is very simple and it can get the global optimal parameters. Therefore, ESN does not need to worry about local convergence that conventional neural networks often confront with. Third, ESN performs much better than past neural network in stochastic time series prediction. In the benchmark problem of Mackey–Glass series prediction, ESN obtained a root mean square error as small as 10^{-42} (Jaeger, 2001; Jaeger & Haas, 2004) while other approaches often claim a prediction error between $10^{-0.52}$ and $10^{-2.3}$ (Martinetz, Berkovich, & Schulten, 1993; Zhu, Ren, Zhang, & Deng, 2002).

2.1. Architecture and training algorithm

A standard ESN consists of three layers-K units in the input layer, N nodes in the internal (hidden) layer and L units in the output layer (see Fig. 1). All K input nodes are connected to all N internal units and all N internal units are connected to the L nodes in the output layer. Also the output neurons can be fed back to the internal layer. Compared with other conventional neural networks, ESN often has a large number of neurons (on the order of 50–1000; Jaeger & Haas, 2004) in the internal layer, which are sparsely interconnected. Note that connections directly from the input to the output units and connections between output units are allowed (Jaeger, 2001). In order to have echo states, the magnitude of the largest eigenvalue of the internal connection weight matrix must satisfy $|\lambda_{\max}| < 1$. The activation of internal state $x(n+1)$ at time step $n+1$ is updated according to

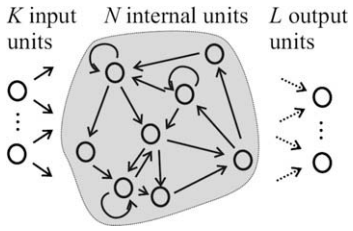

Fig. 1. The architecture of ESN (Jaeger, 2002).

$$x(n+1) = 1\Big/\Big(1+\exp\Big(-\alpha\times\Big(W^{\text{in}}u(n+1) + Wx(n) + W^{\text{back}}y(n)\Big)+\upsilon\Big)\Big) \quad (1)$$

where W^{in} represents a weight matrix from the input neurons to the internal ones; W represents the internal connection matrix; W^{back} represents connections that project back from the output to the internal units; $u(n+1)$ is the input fed into the network at time step $n+1$ and $y(n)$ is the output at time step n. α is a parameter and υ represents noise with a very small value. The output of the network is computed according to the following linear function

$$y(n+1) = W^{\text{out}}(u(n+1), x(n+1), y(n)) \quad (2)$$

where $(u(n+1), x(n+1), y(n))$ is the concatenation of the input, internal, and previous output activation vectors (Jaeger, 2001). W^{out} represents connections to the output units. One of the key features of ESN is that only the weights of the connections to the output units W^{out} should be modified during learning/training process. The topology of the hidden layer and other weight matrixes remain unchanged. Because there are no cyclic dependencies between the trained readout connections (Jaeger & Haas, 2004), any linear regression method is available. Therefore, we applied least-squares algorithm to learn the W^{out} during training.

2.2. Hurst exponent and R/S analysis

Before training an ESN, an initial transient should be dismissed first so that after the transient time the internal network state is determined by the preceding input history up to a negligible error (Ishii et al., 2004). However, choosing initial transient is still according to experience rather than reliable approaches. In our system, the Hurst exponent is applied to decide initial transient and extract sub-training samples from initial training set.

The Hurst exponent, proposed by Hurst (1951) for use in fractal analysis, provides a measure for the long-term memory and fractality of a time series (Qian & Rasheed, 2007). Since it is robust with few assumptions about the underlying systems, it has broad applicability for time series analyses (Qian & Rasheed, 2007). The value of Hurst exponent ranges between 0 and 1, based on which time series can be classified into three categories. (1) $H = 0.5$ indicates a random series. It means that the future is independent on the his-

tory. (2) $0 < H < 0.5$ indicates an anti-persistent series. It has a characteristic of "mean-reverting", which means an up value is more likely followed by a down value, and vice versa (Qian & Rasheed, 2007). If H is close to 0, the data changes totally irregularly and the time series is unpredictable. (3) $0.5 < H < 1$ indicates a persistent series. It means that the history data influences the future, that is, if the data has been increasing for a period, it is likely to continue increasing for another period, and vice versa. Furthermore, the closer H is to 1, the stronger the impact is. Therefore, before training an ESN, we can calculated the Hurst exponents in the training series and choose an initiate transient after which the value of Hurst exponent of the remaining series is not only beyond 0.5 but also the closest to 1.

Hurst exponent can be computed according to rescaled range analysis (R/S analysis). See a time series $X = (X_1, X_2, \ldots, X_n)$, in which each sub-series $X_i = (X_{(i-1)\times t+1}, X_{(i-1)\times t+2}, \ldots X_{i\times t})$ $(i = 1, 2, \ldots, k)$ has the same length of t $(t \times k = n)$ and shares no data with each other. For each sub-series, let

$$Y_r = X_r - \overline{X}, \quad r = 1, 2, \ldots t \tag{3}$$

$$Z_r = \sum_{i=1}^{r} Y_i \tag{4}$$

$$R = \max(Z_1, Z_2, \ldots, Z_t) - \min(Z_1, Z_2, \ldots, Z_t) \tag{5}$$

The Hurst exponent is defined as

$$(R/S)_n = \frac{1}{k}\sum_{j=1}^{k}\left[R_j(t)/S_j(t)\right] = cn^H \tag{6}$$

where H represents the Hurst exponent; c is a constant; $S_j(t)$ is the standard deviation of the sub-time series. Eq. (6) can also be described as

$$\log(R/S)_n = \log c + H \log n \tag{7}$$

Different values of n have their corresponding values of $(R/S)_n$, thus, the Hurst exponent H can be obtained through least-squares fitting method.

We extract a set of sub-series of closing price from the initial training samples. All of them end at the last point of the training set but begin from different points. Then we calculate the Hurst exponent for each sub-series. The sub-samples whose length is larger than the length of testing size and whose Hurst exponent is the closest to 1 are the final training set. To restrict the minimum length of the final training set is to guarantee more training samples than testing ones.

2.3. Data preparation and principle component analysis

Our system intends to predict the raw closing price (not the adjusted closing price) of the next trading day according to the history data. It is believed that the influence of the macroeconomic environment and a company's financial conditions is negligible in short-term prediction. Practical experience also shows that generally the next price tends to follow the current one. Therefore, the prices of the last trading day are very important features and we did not have to consider fundamental indicators in our system. Besides original prices, simple moving average, which emphasizes the direction of a trend and smoothes out the fluctuation of price, is frequently used as one of technical analysis tools. Short length moving averages, such as 5-day moving average, are more sensitive and can identify new trends earlier so that they are often used as one of a stock's features in financial prediction. We tried various kinds of indicators, including original prices and technical indicators, to find the optimal combination through experiments. All the input indicators fed into our network are listed as follows:

- *Intra-day high:* the maximum price of a stock ticker during the intra-day trading.
- *Intra-day low:* the minimum price of a stock ticker during the intra-day trading.
- *Open price:* the first price of a stock ticker during the intra-day trading.
- *Close price:* the final price of a stock ticker during the intra-day trading.
- *5-Day high:* the highest high price during the past five days.
- *5-Day close moving average:* the average of close price during the past five days.

Since the values of the input usually fluctuate a little in a specific period, the input data is linearly normalized as follows:

$$y_t = 2 \times \frac{x_t - m}{M - m} - 1 \tag{8}$$

where x_t is the original data; m is the minimum value of the training set; and M is the maximum value of the training data. After preprocessing, all the input data in the training set is mapped to $[-1, 1]$. The input in the testing set is also pretreated according to Eq. (8).

The combination of price data and technical indicators provide a perspective to extract underlying information to analyze price action. But those data may contain much noise and depend on each other. Here we also utilize PCA to summary the most important information and filter noise, and study its influence on stock price data.

PCA is one of the key tools in data analysis technique. It aims at finding n linearly transformed components, F, thus to explain as much amount of variance of the original p-dimension variables, x, as possible (Mok, Lam, & Ng, 2002). It is usually the first step of many data analysis procedures to reduce the dimension of data (Mok et al., 2002). Through PCA, the original data can be transformed to components according to the following equation:

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_p \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ & & & \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} \tag{9}$$

where X is the original data and F is the components. According to the variance of each component, it is easy to get the principle components which contribute most.

3. Experiments and results

In our experiments, the stock information between 6th December, 2001 and 25th November, 2005 (totally 1100 days) of 491 stocks in S&P 500 is adopted. The prices of the first 1000 days are for training and those of the last 100 days are for testing. The remaining nine stocks are dismissed because they lack part of data in this period. S&P 500 is considered as the benchmark for the United State's equity performance. It consists of 500 companies from a diverse range of industries and represents approximately 70% of the value of the US equity market. Hence, S&P 500 time series is a popular choice among researchers for illustrating prediction methods in financial time series. All the data come from Yahoo Finance.

Our stock prediction system is built based on a standard ESN with 600 internal units, only 5% of which are interconnected. Input connection weights W^{in} are set to small random values sampled from a uniform distribution between $[-0.1, 0.1]$; the reservoir weights W are randomly sampled from a uniform distribution between $[-1, 1]$; feedback connection weights W^{back} are randomly sampled from the uniform distribution between $[-5, 5]$. The reservoir matrix W is rescaled to a spectral radius of 0.1, thus ensuring

Table 1
Daily stock prediction performance (for next-day close price prediction)

Stock	ESN (%)	BPNN (%)	Elman (%)	RBF (%)	ESN + PCA (%)
ACE	1.15	3.87	3.75	3.09	**1.11**
AHC	**1.69**	8.77	10.06	9.15	1.72
AMD	1.94	2.66	2.54	2.99	**1.93**
BBT	0.84	0.87	0.83	0.91	**0.80**
CIEN	3.81	3.55	3.68	2.48	**2.46**
CPN	845.86	6.20	4.58	4.15	*22.28*
FDO	16.34	3.90	2.24	4.40	9.15
GD	**0.65**	2.07	3.25	1.32	0.65
GTW	326.52	2.64	2.60	2.70	*31.59*
HRB	461.49	14.58	15.65	30.60	**2.72**
IR	587.12	1.97	2.15	5.49	2.38
JCP	1.58	2.04	2.01	2.26	**1.55**
KMG	1.51	5.18	6.37	8.25	**1.50**
LXK	106.44	1.63	2.14	2.27	1.70
NBR	1.78	6.55	6.11	4.27	**1.55**
NSC	1.12	2.66	2.31	2.48	**1.08**
PBI	1.14	0.68	0.69	0.68	0.81
PPL	33.46	2.74	2.40	36.04	**2.06**
PSA	**0.97**	2.31	2.70	1.75	0.98
RHI	**1.61**	6.41	6.28	9.99	**1.53**
SFA	**1.53**	3.47	3.22	3.21	1.65
SRE	1.17	3.21	2.95	1.92	**0.99**
THC	38.40	3.30	3.58	2.04	***1.36***
UIS	114.02	3.32	1.95	4.21	*1.99*
USB	0.79	0.75	0.74	0.79	**0.74**

The bold letter shows the best prediction accuracy given by ESN; the italic letter means that those results are obtained through including PCA during data pretreatment and adjusting parameters of ESN.

the echo state property. When calculating the internal state, the noise data υ is randomly sampled from $[-0.00001, 0.00001]$ and the parameter α is usually set to 5 (see Eq. (1)).

Because seldom past studies give their prediction results on the whole dataset, we also applied other three conventional neural networks – BPNN, Elman neural network and RBFNN (using Matlab Neural Network Toolbox) on the same task, and compare their predictive capabilities with that of ESN. The BPNN has three layers with sigmoid transfer function in the hidden layer and linear transfer function in the output layer. The hidden layer has 10 neurons. Its training algorithm combines adaptive learning rate with momentum training. The Elman network also has three layers with the same transfer functions as BPNN. The hidden layer has 12 neurons. In the RBFNN, we use 15 as the spread for the radial basis layer. Note that all the parameters of the three networks are chose by manual experiments to make sure that most stocks will obtain satisfying prediction results.

Average percent error (APE, See Eq. (10)) is the criteria to judge all networks in our experiments.

$$\mathrm{APE} = \frac{|y_t - \hat{y}_t|}{\hat{y}_t} \times 100\% \tag{10}$$

where y_t is the actual output and $\hat{y}_t$ is the desired one. We first compare ESN with other neural networks using only raw data. We find that ESN outperforms other three neural networks in most cases. It performs the best in 57.03% cases, especially much better than others in some cases, such as AHC, KMG, NBR and RHI in Table 1. BPNN performs the best in 14.87% cases; Elman neural network performs the best in 20.16% cases and RBFNN performs the best in 7.94% cases. Table 1 lists some results that are randomly chosen.

Although ESN does not give the best results in some cases, even the worst results, such as CIEN and USB in the above table, the gap between ESN and other three networks is very small in general. Statistically, in 138 cases out of other 211 stocks where ESN does not perform the best, ESN gives results very close to the optimal one given by other networks (the gap is smaller than 0.5%). Only in a few cases (16/491), ESN totally fails to give reliable prediction results, such as CPN, GTW and THC.

To solve the problem, we try to apply PCA during data pretreatment on those 16 stocks and attempt to adjust parameters such as the spectral radius of reservoir matrix W, the input connection matrix and the feedback connection matrix. Experiments show that if only including PCA in the model, the prediction performance of 10 stocks can be enhanced, even better than other neural networks (see HRB and PPL in Table 1). For other six stocks, PCA hardly improves prediction performance. However, if we also manually explore the appropriate parameters besides PCA in the data pretreatment for the other 10 stocks (see the italic letters in Table 1), some of them obtain the satisfying results, such as THC and UIS. Unfortunately, for the others, although the performance is improved a lot, it is still much worse than other neural networks (see CPN and GTW in Table 1). We also find that of totally 491 stocks, PCA helps to improve the prediction accuracy of 325 stocks, but the improvement is very small (see Table 1).

We try to analyze the price movement of stocks in which ESN quite outperforms other neural networks and in which ESN fails to give better prediction accuracy than others. We find that if the second half part of the training samples generally share the same obvious trend with the testing samples, and the fluctuation of price action is small (see Fig. 2a), ESN tends to predict more accurately than other neural networks. If the price fluctuates a lot or does not have obvious trend in a quite long specific period or the trend of most part of training set differs from that of the testing set, the

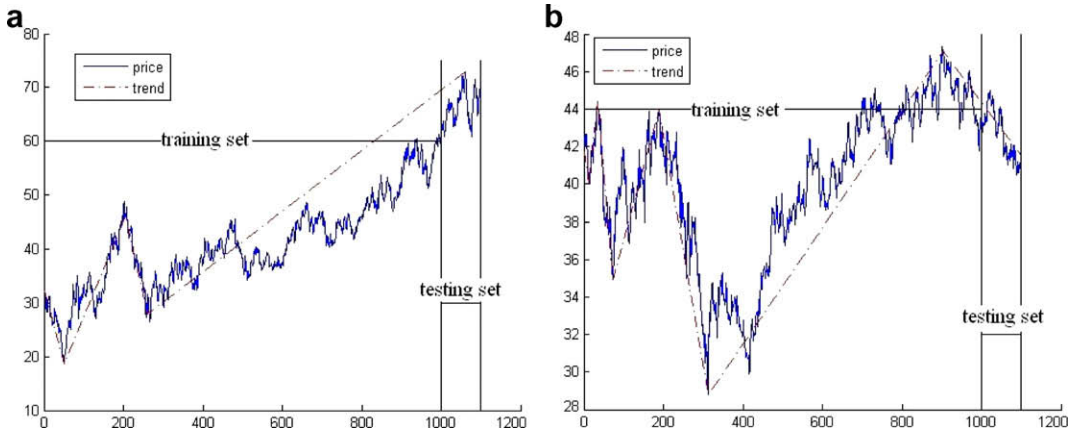

Fig. 2. Close price (6th December, 2001–25th November, 2005).

Table 2
Average performance of every neural network

ESN (with Hurst exponent) (s)	ESN (fixed transient) (s)	BPNN (s)	Elman (s)	RBF (s)
55.05	6.86	17.68	23.91	3.27

performance of ESN tends to be worse than other networks (see Fig. 2b). It coincides with the short-term memory ability of ESN declared by Jaeger that the reservoir has "remembered" the trend information and the memory fades when time moves on. However, it is just qualitative observation, which needs further study.

Table 2 lists the average elapse time of every neural network to train and predict one stocks on a Core 2 CPU T5500 1.0G computer. Obviously, ESN takes much more time than other neural networks to predict stock prices. However, if we choose a fixed transient instead of applying Hurst exponent value to select transition state, we can save most time. It indicates that the training algorithm of ESN is efficient. If we can find substitution method to select transient state of ESN, the computer cost of our prediction system can be decreased a lot.

4. Conclusions

In this paper, we build a short-term stock price prediction system based on ESN and compare its predictive accuracy in S&P 500 with other three traditional neural networks – BPNN, Elman network and RBFNN. In ESN, the Hurst exponent is used to guide transient selection before training since it is a reliable measure for chaotic series predictability. The experiments demonstrate that ESN is an effective model to predict time series with its ability of short-term memory and performs better than other conventional neural networks in most cases. The application of PCA effectively avoids exceptional prediction results of ESN but has only a little positive influence in other stocks in which ESN can perform well. The analysis of experiments also demonstrates that ESN is more suitable for stocks that have lasting obvious trend and fluctuate a little.

However, ESN is a young discrete model for chaotic time series mining that needs many further studies. Future work will focus on the following aspects. First, finding an optimal architecture and parameters for each stock should be investigated. As mentioned in the experiments, adjusting parameters helps enhance prediction accuracy especially in those stocks that ESN fails to give satisfying prediction results. To find a reasonable choice of parameters, we have attempted generic algorithms (GAs), which is one of the popular methods for adaptively adjusting neural networks' architecture. Unfortunately, the inclusion of GA hardly influences the predictive capability of ESN because an appropriate combination of parameters in the training set does not guarantee a satisfying result in the testing set. Therefore, how to decide suitable parameters is worth further exploration. Second, we can obviously see from the experiments that ESN achieves outstanding results in many stocks, performs similar to other networks in some cases and obtain very bad results in very few stocks. This produces an issue that which type of data ESN is most suitable for. If we can identify the common characteristics of those series, we can apply ESN on similar series to get desirable prediction quality. Because ESN is a complicated nonlinear dynamic system, the answer is still unknown. Our preliminary observation demonstrates that the price trend may influence prediction accuracy, but further investigation is required. One possible approach would be to perform more experiments over various price series and attempt to cluster them. Finally, whether ESN has the ability of long-term stock data mining is still of interest.

References

Armono, G., Marchesi, M., & Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences, 170*, 3–33.

Hurst, H. E. (1951). Long-term storage of reservoirs: An experimental study. *Transactions of the American Society of Civil Engineers, 116*, 770–799.

Ishii, K., van der Zant, T., Becanovic, V., Ploger, P. (2004). Optimization of parameters of echo state network and its application to underwater robot. In *SICE Annual Conference in Sapporo* (Vol. 3, pp. 2800–2805).

Jaeger, H. (2001). *The "echo state" approach to analyzing and training recurrent neural networks*. GMD-German National Research Institute for Computer Science, GMD Report 148.

Jaeger, H. (2002). *Tutorial on training recurrent neural networks covering BPPT, RTRL, EKF and the echo state network approach*. Technical Report, GMD Forschungszentrum Informationstechnic GmbH.

Jaeger, H. (2002). *Short term memory in echo state networks*. GMD-Report 152, GMD-German National Research Institute for Computer Science.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communications. *Science, 3*.

Jaeger, H., Lukosevicius, M., & Popovici, D. (2007). Optimization applications of echo state networks with leaky integrator neurons. *Neural Networks, 20*, 335–352.

Kim, H.-J., & Shin, K.-S. (2007). A hybrid approach based on neural networks genetic algorithms for detecting temporal patterns in stock markets. *Applied Soft Computing, 7*(2), 569–576.

Lawrence, R. (1997). *Using neural networks to forecast stock market prices*. Manitoba, BC, Canada: University of Manitoba. http://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf.

Lee, R. S. T., & Liu, J. N. K. (2001). NORN predictor-stock prediction using a neural oscillatory-based recurrent network. *Intelligent Journal of Computational Intelligence and Applications, 1*, 439–451.

Martinetz, T. M., Berkovich, S. G., & Schulten, K. J. (1993). "Neural-gas" network for vector quantization its application to time-series prediction. *IEEE Transactions on Neural Networks, 4*(4).

Mok, P. Y., Lam, K. P., Ng, H. S. (2002). An ICA design of intraday stock prediction models with automatic variable selection. In *IEEE International Joint Conference on Machine Learning and Cybernetics* (Vol. 2, pp. 4–5).

Oh, K. J., & Kim, K.-J. (2002). Analyzing stock market tick data using piecewise nonlinear model. *Expert Systems with Applications, 22*, 249–255.

Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence, 26*(1), 25–33.

Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay recurrent probabilistic neural networks. *IEEE Transactions of Neural Networks, 9*, 1456–1470.

Sitte, R., & Sitte, J. (2000). Analysis of predictive ability of time delay neural networks applied to the S&P 500 time series. *IEEE Trans on Systems, Man, and Cybernetics – Part C: Applications and Reviews, 30*(4).

Skowronski, M. D., & Harris, J. G. (2007). Automatic speech recognition using a predictive echo state network classifier. *Neural Networks, 20*, 414–423.

Zhu, J.-Y., Ren, B., Zhang, H.-X., & Deng, Z.-T. (2002). Time series prediction via new support vector machines. In *Proceedings of the First International Conference on Machine Learning and Cybernetics* (pp. 4–5).