Neural Comput & Applic (2001)10:184–192

Neural
Computing
& Applications

Financial Forecasting Using Support Vector Machines

Lijuan Cao and Francis E.H. Tay

Department of Mechanical and Production Engineering, National University of Singapore, Singapore

The use of Support Vector Machines (SVMs) is studied in financial forecasting by comparing it with a multi-layer perceptron trained by the Back Propagation (BP) algorithm. SVMs forecast better than BP based on the criteria of Normalised Mean Square Error (NMSE), Mean Absolute Error (MAE), Directional Symmetry (DS), Correct Up (CP) trend and Correct Down (CD) trend. S&P 500 daily price index is used as the data set. Since there is no structured way to choose the free parameters of SVMs, the generalisation error with respect to the free parameters of SVMs is investigated in this experiment. As illustrated in the experiment, they have little impact on the solution. Analysis of the experimental results demonstrates that it is advantageous to apply SVMs to forecast the financial time series.

Keywords: Back propagation algorithm; Financial time series forecasting; Generalisation; Multi-layer perceptron; Support vector machines

1. Introduction

Financial time series forecasting is one of the most challenging applications of modern time series forecasting. As explained by Deboeck and Yaser [1,2], financial time series are inherently noisy, non-stationary and deterministically chaotic. These characteristics suggest that there is no complete information that could be obtained from the past behaviour of financial markets to fully capture the dependency between the future price and that of the past.

Correspondence and offprint requests to: L. Cao, Department of Mechanical & Production Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore.

There are two main categories in financial time series forecasting: univariate analysis and multivariate analysis. In multivariate analysis, any indicator, whether it is related to the output directly or not, can be incorporated as the input variable, while in univariate analysis, the input variables are restricted to the time series being forecasted. A general univariate model that is commonly used is based on the AutoRegressive Integrated Moving Average (ARIMA) method. Compared to other multivariate models, the performance of ARIMA is not satisfactory because this model is parametric, and additionally, it is developed on the assumption that the time series being forecasted are linear and stationary. These constraints are not consistent with the characteristics of financial time series. Therefore, Artificial Neural Network (ANN) assisted multivariate analysis has become a dominant and popular tool in recent years. The prediction performance is greatly improved by the use of a neural network both in terms of prediction metrics and trading metrics [3–6]. It can be explained both in terms of the indicator aspect, and the characteristics of the neural network. Multivariate models can rely on greater information, where not only the lagged time series being forecast, but also technical indicators, fundamental indicators or inter-market indicators, are combined to act as predicators. Moreover, a neural network is more effective in describing the dynamics of non-stationary time series due to its unique non-parametric, non-assumable, noise-tolerant and adaptive properties. Neural networks are universal function approximators that can map any nonlinear function without *a priori* assumptions about the data.

However, a critical issue concerning neural networks is the over-fitting problem. It can be attributed to the fact that a neural network captures not only useful information contained in the given data, but also unwanted noise. This usually leads to a poor

level of generalisation. The performance of neural networks in terms of generalisation for the out-of-sample data – the data that are not used in training the network – is always inferior to that of the training data. Therefore, the development of neural networks and the tasks related to architecture selection, learning parameter estimation and training, require substantial care in order to achieve the desired level of generalisation. The significance of good generalisation is critical when using neural networks for financial time series forecasting.

The issue of generalisation has long been a concern to researchers, who have explored a variety of procedures for enhancing the generalisation ability of neural networks. A typical approach uses cross-validation, where the data given is divided into three sub-samples. The first is for training, the second for testing, while the third is for validating. This approach involves a substantial amount of computation that is often referred to as a weakness in theory. Nonparametric probability density estimation is another statistical tool used to improve generalisation [7], but its underlying assumption – that the distribution for both patterns and noise remains the same over the model estimation and forecasting period – could often not be satisfied in financial time series. Other techniques of enhancing the generalisation ability of neural networks include modifying training algorithms [8], pruning the connections or hidden nodes of the network [9], using adaptive learning parameters, and selecting significant variables [10,11].

Recently, SVMs developed by Vapnik [12] have provided another novel approach to improve the generalisation property of neural networks. Originally, SVMs were developed for pattern recognition problems. Recently, with the introduction of ϵ-insensitive loss function, SVMs have been extended to solve non-linear regression problems. Unlike most of the traditional learning machines that adopt the *Empirical Risk Minimisation Principle*, SVMs implement the *Structural Risk Minimisation Principle*, which seeks to minimise an upper bound of the generalisation error rather than minimise the training error. This will result in better generalisation than conventional techniques.

The objectives of this paper are, first, to examine the feasibility of applying SVMs in financial forecasting, and secondly, to investigate the functional characteristics of the SVMs in financial forecasting. The functional characteristics are obtained through the selection of the free parameters of the SVMs. Since there is no structured way to choose the free parameters of SVMs, the variability in performance with respect to the free parameters of SVMs is examined, and the results are discussed.

Section 2 provides a brief introduction to SVMs, and Section 3 contains the experimental data. The techniques for data preprocessing and statistical performance metrics are presented in the same section. Section 4 describes the experimental results, and the comparison with BP. The last section concludes the work.

2. Theory of SVMs in Regression

Regression approximation addresses the problem of estimating a function based on a given set of data $\boldsymbol{G} = \{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_i^l$ ($\boldsymbol{x}_i$ is the input vector, $\boldsymbol{d}_i$ is the desired value), which is produced from the unknown function. SVMs approximate the function in the following form:

$$y = \sum_{i=1}^{l} w_i \phi_i(x) + b \tag{1}$$

where $\{\phi_i(\boldsymbol{x})\}_{i=1}^l$ are the features of inputs and $\{\boldsymbol{w}_i\}_{i=1}^l$, $\boldsymbol{b}$ are coefficients. They are estimated by minimising the regularised risk function (2):

$$\boldsymbol{R}(\boldsymbol{C}) = \boldsymbol{C}\frac{1}{\boldsymbol{N}}\sum_{i=1}^{N}\boldsymbol{L}_{\epsilon}(\boldsymbol{d}_i, \boldsymbol{y}_i) + \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{2}$$

$$\boldsymbol{L}_{\epsilon}(\boldsymbol{d}, \boldsymbol{y}) = \begin{cases} |\boldsymbol{d} - \boldsymbol{y}| - \boldsymbol{\epsilon} & |\boldsymbol{d} - \boldsymbol{y}| \geq \boldsymbol{\epsilon} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\boldsymbol{\epsilon}$ is a prescribed parameter.

The first term $\boldsymbol{L}_{\epsilon}(d, y)$ is the so-called ϵ-insensitive loss function. This self-explanatory function indicates the fact that it does not penalise errors below $\boldsymbol{\epsilon}$. The second term, $\frac{1}{2}\|\boldsymbol{w}\|^2$, is used as a measure of function flatness. C is a regularised constant determining the trade-off between the training error and model flatness. Introduction of the slack variables ζ, ζ^* leads Eq. (2) to the following constrained function:

Minimise:

$$\boldsymbol{R}(\boldsymbol{w}, \zeta^{(*)}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \boldsymbol{C}^* \sum_{i=1}^{n}(\zeta_i + \zeta_i^*)$$

Subjected to: (4)

$$w\phi(x_i) + b - d_i \leq \boldsymbol{\epsilon} + \zeta_i$$

$$d_i - w\phi(x_i) - b_i \leq \boldsymbol{\epsilon} + \zeta_i^*$$

$$\zeta^{(*)} \geq 0$$

Thus, Eq. (1) becomes the following explicit form:

$$f(x, a_i, a_i') = \sum_{i=1}^{l} (a_i - a_i') \boldsymbol{K}(x, x_i) + b \qquad (5)$$

Lagrange Multipliers. In function (5), a_i, a_i' are the Lagrange multipliers introduced. They satisfy the equality $a_i * a_i' = 0$, $a_i \geq 0$, $a_i' \geq 0$, $i = 1, \ldots, l$, and are obtained by maximising the dual form of function (4), which has the following form:

$$\phi(a_i, a_i') = \sum_{i=1}^{l} d_i(a_i - a_i') - \epsilon \sum_{i=1}^{l} (a_i + a_i') - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (a_i - a_i')(a_j - a_j') \boldsymbol{K}(x_i, x_j) \qquad (6)$$

with the following constraints:

$$\sum_{i=1}^{l} (a_i - a_i') = 0$$

$$0 \leq a_i \leq \boldsymbol{C}, \; i = 1, 2, \ldots, l$$

$$0 \leq a_j' \leq \boldsymbol{C}, \; i = 1, 2, \ldots, l$$

Based on the nature of quadratic programming, only a number of coefficients a_i, a_i' will be assumed as nonzero, and the data points associated with them could be referred to as support vectors.

Kernel Function. $\boldsymbol{K}(x_i, x_j)$ is named as the kernel function. The value is equal to the inner product of two vectors $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in the feature space $\phi(x_i)$ and $\phi(x_j)$. That is, $\boldsymbol{K}(x_i, x_j) = \phi(x_i) * \phi(x_j)$. Any function that satisfies Mercer's condition [12] can be used as the kernel function. There are many choices of the kernel function: common examples are the polynomial kernel $K(x,y) = (xy + 1)^d$ and the Gaussian kernel $\boldsymbol{K}(x,y) = \exp(-\delta^2 (x - y)^2)$.

Finally, it should be pointed out that training SVMs is equivalent to optimising the Lagrange multipliers a_i, a_i' with constraints based on function (6) [13,14].

3. Experimental Settings

3.1. Data Set

The S&P 500 Daily Index in the Chicago Mercantile is selected for the experiment. The training data points cover the time period from 01/04/1993 up to the end of December 1994, while the data points starting from 01/03/1995 up to the end of December 1995 are used as the test data.

The original data is transformed into a five-day Relative Difference in Percentage (RDP) of the S&P 500 index. As interpreted by Thomason [15,16], there are four advantages in applying this transformation. The most prominent is that the distribution of transformed data will become more symmetrical and closer to normal, as illustrated in Fig. 1. This modification in the data distribution trend will improve the predictive power of the neural network.

Two groups of variables are chosen as the inputs to the neural network. The first group (G-1) is constructed from four lagged RDP values based on 5-day periods, and the transformed closing price that is obtained by subtracting a 15-day exponential moving average. The subtraction is performed to eliminate the trend in price. The second group (G-2) is constructed by adding another three technical indicators: the Moving Average Convergence Divergence (MACD), On Balance Volume (OBV) and volatility. The MACD is defined as the difference between two exponential moving averages, and it is commonly used to predict market trends in financial markets. The volatility denotes the range of the highest and lowest prices in one day, and it is often used as a measure of the market risk. OBV moves in the same direction as price, i.e. as price increases, the OBV will gain in magnitude. As OBV can relate the volume into price, it is also used here as input. The purpose of using these two groups of variables is to observe whether multivariate analysis can improve prediction accuracy while outsourcing with more information. The calculation for all indicators is listed in Table 1. There is a total of 500 data patterns in the training set, and 200 data patterns in the test set.

The long left tail in Fig. 1(b) indicates that there are outliers in the data set. Since outliers may make it difficult or time-consuming to arrive at an effective solution for a neural network, RDP values beyond the limits of $\pm$ 2 standard deviations are selected as outliers. They are replaced with the closest marginal values. Finally, all the data points are scaled into the range of $[-0.9, 0.9]$ as the data points include both positive and negative values. Similar preprocessing procedures are applied to the dependent variable RDP+5. The only difference is that RDP+5 is obtained by smoothening the closing price with a 3-day exponential moving average.

3.2. Performance Criteria

The prediction performance is evaluated using the following statistical metrics: Normalised Mean Squared Error (NMSE), Mean Absolute Error

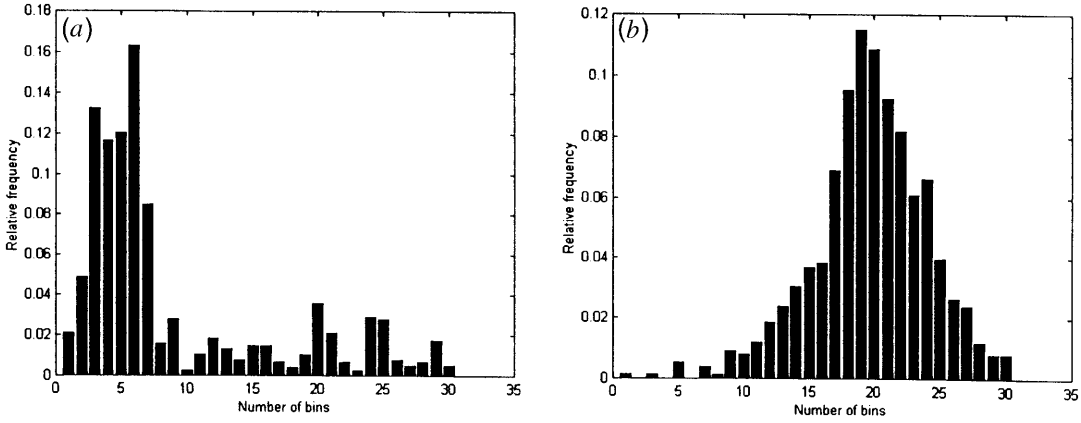

Fig. 1. (a) Histogram of S&P 500 daily closing price; (b) Histogram of RDP+5. RDP+5 values have a more symmetrical and normal distribution.

Table 1. Independent and dependent variables.

Indictor	Calculation
SP-EMAI5	$\boldsymbol{P(i)} - \overline{\boldsymbol{EMA}_{15}(i)}$
RDP-5	$(p(i) - p(i-5))/p(i-5) * 100$
RDP-10	$(p(i) - p(i-10))/p(i-10) * 100$
RDP-15	$(p(i) - p(i-15))/p(i-15) * 100$
RDP-20	$(p(i) - p(i-20))/p(i-20) * 100$
MACD	$\overline{EMA_{10}(i)} - \overline{EMA_{20}(i)}$
OBV	$\begin{cases} p(i) \geq p(i-1)\ obv+ = volume(i) \\ p(i) \leq p(i-1)\ obv- = volume(i) \end{cases}$
Volatility	$k * sqrt\left(1/n * \sum_{i=1}^{n} \log^2(h(i)/l(i))\right)\ (k = 80)$
RPD+5	$\overline{(p(i+5)} - \overline{p(i)})/\overline{p(i)} * 100$
	$\overline{p(i)} = \overline{EMA_3(i)}$

EMA_n (i) is the n-day exponential moving average of the ith day; $p(i)$, $h(i)$, $l(i)$ are the closing, highest and lowest price of the ith day.

(MAE), Directional Symmetry (DS), Correct Up trend (CP) and Correct Down trend (CD). The definitions of these criteria are illustrated in Table 2. NMSE and MAE are measures of the deviation between actual and predicted values. The smaller the values of NMSE and MAE, the closer are the predicted time series values to that of the actual value. DS provides the correctness of the predicted direction of RDP+5 in terms of percentage. CP and CD provide the correctness of the predicted up trend and predicted down trend of RDP+5 in terms of percentage.

4. Experimental Results

4.1. Results of SVMs and BP

In this investigation, the Gaussian function is used as the kernel function of the SVMs. Our experiments show that a width value of the Gaussian function of 0.015 is found to produce the best possible results. C and ϵ are arbitrarily chosen to be 10 and 10^{-3}, respectively. The Sequential Minimal Optimisation algorithm for solving regression problem extended by Scholkopf [17–19] is implemented in this experiment. The program is constructed using the VC^+ language.

A standard three-layer multi-layer perceptron trained using the back propagation (BP) algorithm is used as a benchmark [15,16]. There are five input nodes for G-1 and eight input nodes for G-2, which are equal to the number of indicators. The output node is equal to 1, whereas the number of hidden nodes is determined by using the formula W<=M/5, where W is the number of interconnection weights that satisfies the following equality:

$$W=(I+O)*H$$

where M = the number of training examples, I = the number of input nodes, O = the number of output nodes, and H = the number of hidden nodes.

The size of the network is controlled by ensuring that the ratio of the number of weights to the number of training samples is equal to or smaller than 0.2. So, there are approximately 15 hidden nodes in the network for both data sets. Selection of the learning rate as 0.005 and the momentum term as 0.9 can be attributed to the fact that the BP, with these settings of the learning parameters,

Table 2. Performance metrics and their calculations.

Metrics	Calculation
NMSE	$NMSE = 1/(\delta^2 n) * \sum_{i=1}^{n} (a_i - p_i)^2$
	$\delta^2 = 1/(n-1) * \sum_{i=1}^{n} (a_i - \bar{a})^2$
MAE	$MAE = 1/n * \sum_{i=1}^{n} \lvert a_i - p_i \rvert$
DS	$DS = 100/n * \sum_{i=1}^{n} d_i$
	$d_i = \begin{cases} 1 & (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$
CP	$CP = \frac{100}{n} \sum_{i=1}^{n} d_i$
	$d_i = \begin{cases} 1 & (p_i - p_{i-1}) > 0, \ (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$
CD	$CD = \frac{100}{n} \sum_{i=1}^{n} d_i$
	$d_i = \begin{cases} 1 & (p_i - p_{i-1}) < 0, \ (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$

can converge at the fastest speed without oscillation, which will occur because of the use of a large learning rate [20].

The behaviour of the NMSE (Normalised Mean Squared Error) is illustrated in Fig. 2. It is evident that the NMSE on the training set is monotonically decreasing during the entire training period in both BP and SVM networks. On the contrary, the NMSE on the test set in BP is decreasing in the first few thousands of epochs, like that of training set, but it increases in the remaining epochs. This shows that over-fitting has occurred in BP. In SVMs, however, the NMSE on the test set fluctuates during the initial training period, but at a later stage it gradually converges to a constant value. All the above results are found to be consistent with the statistical learning theory.

The comparison of BP is based on the results of BP-1 and BP-2. BP-1 corresponds to the network with epochs equal to 800, based on the assumption that there is no over-fitting problem in BP, and the generalisation error will increase if the training of the network is continued. BP-2 corresponds to the network in which the number of epochs is arbitrarily chosen as 8000. All the results included in Tables 3 and 4 correspond to the best records obtained in this experiment.

Table 3 gives the results for G-1, and the results of G-2 are listed in Table 4. For G-1, the NMSE and MAE (Mean Absolute Error) of SVMs are 0.001, 0.0005 on the training set, and 1.4383, 0.3403 on the test set. It is evident that these values are much smaller than those of BP-1 and BP-2, indicating that there is a smaller deviation between the actual and predicted values in SVMs. Moreover, the DS (Directional Symmetry), CP (Correct Up trend) and CD (Correct Down trend) of SVMs are as high as 95.59, 100 and 91.63 on the training set. Although they are slightly lower than those of BP-1 and BP-2 on the test set, the DS, CP and CD of SVMs are still higher than those of the other two techniques when applied to G-2. These three criteria provide a good measure of the consistency in prediction of the price direction. Thus, it can be concluded that SVMs forecast far better than BP, even though BP does not suffer from the over-fitting problem in

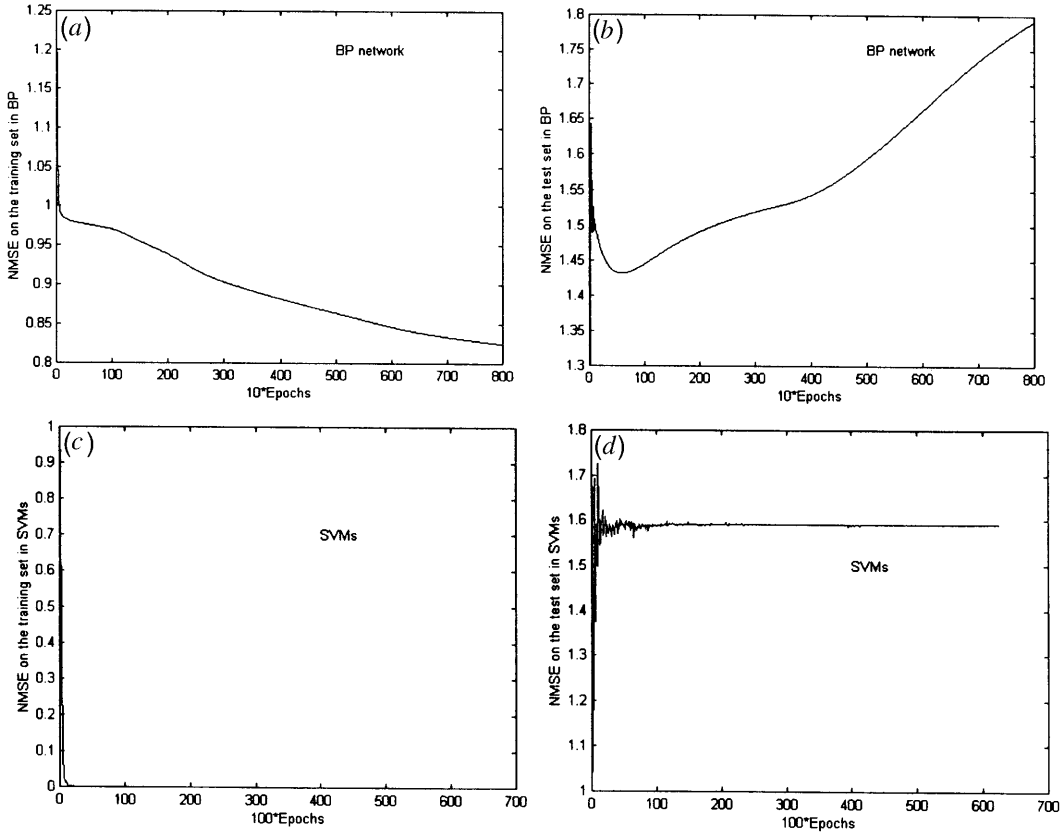

Fig. 2. (a&b) The behaviour of NMSE in BP. The NMSE on the training set (a) keeps on decreasing in the entire period, while the NMSE on the test set (b) is decreasing in the first 1000 epochs, but it changes to increase in the remaining epochs. This suggests that over fitting has occurred in BP; (c&d) the behaviour of NMSE in SVMs. As in BP, the NMSE on the training set (c) keeps on decreasing during the entire period, while the NMSE on the test set (d) fluctuates during the initial period, and gradually converges to a constant value.

Table 3. Results of BP and SVMs for G-1.

G-1 set	Training data			Test data		
	BP-1	BP-2	SVM	BP-1	BP-2	SVM
NMSE	0.9144	0.8273	0.0001	1.4446	1.6937	1.4383
MAE	0.3218	0.3097	0.0005	0.3496	0.3726	0.3403
DS	51.70	53.10	95.59	49.24	49.75	47.72
CP	52.96	55.51	100.0	51.08	50.00	46.21
CD	50.57	50.95	91.63	47.66	49.53	48.60

Table 4. Results of BP and SVMs for G-2.

G-2 set	Training data			Test data		
	BP-1	BP-2	SVM	BP-1	BP-2	SVM
NMSE	0.8577	0.4195	0.0004	2.1592	4.4260	1.6762
MAE	0.3124	0.2157	0.0011	0.4353	0.6347	0.3706
DS	55.51	59.12	95.19	41.23	39.20	46.23
CP	55.08	61.86	99.15	41.30	40.21	47.82
CD	55.89	56.65	91.63	40.47	38.31	44.86

the case of BP-1. The same conclusion can also be achieved by analysing Table 4. Furthermore, by comparing the results of the two tables, it can be seen that the multivariate analysis does not improve the predictive power of the networks. On the contrary, it worsens the prediction in all techniques in terms of the specified criteria on the test data.

The actual RDP+5 values and predicted values from both networks are illustrated in Fig. 3, where only G-1 is considered, since G-1 has better performance than G-2. It is obvious that the predicted values obtained from SVMs are closer to the actual values than those of BP.

4.2. Sensitivity of SVMs to Parameters

Since there is an absence of a structured method to select the free parameters of SVMs, the generalis-

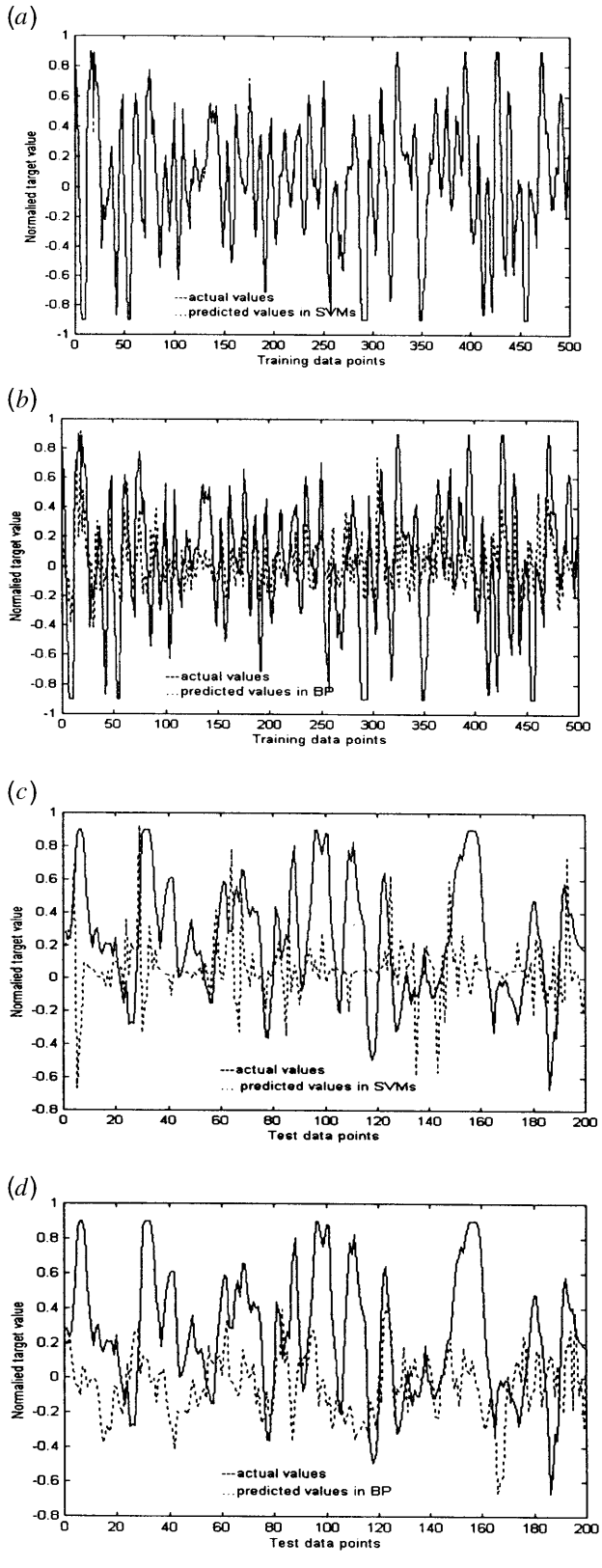

Fig. 3. (a) The predicted values of SVMs on the training data; (b) predicted values of BP-1 on the training data; (c) predicted values of SVMs on the test data; (d) predicted values of BP-1 on the test data. All the results indicate that the predicted values of SVMs are closer to the actual values than BP-1.

ation error and number of support vectors with respect to C and ϵ are studied. The data set used is G-1.

Figure 4(a) illustrates the generalisation error versus C. From Fig. 4, the generalisation error is not influenced greatly by C. The values of the generalisation errors range only between [1.25, 1.60] when C is increased from 0.0001 to 1000. Figure 4(b) shows that the number of support vectors increases rapidly when C rises from 0.1 to 1, and then it remains almost stable after C reaches 1.

Figure 5(a) illustrates the generalisation error versus ϵ; ϵ is also found to have little impact on the generalisation error, but the number of support vectors reduces to 0 with the increment of ϵ.

5. Conclusions

The use of SVMs in financial forecasting is studied in this paper. The study has concluded that SVMs provide a promising alternative to time series forecasting. They offer the following advantages:

1. There is a smaller number of free parameters compared to BP. C and ϵ are the only two free parameters of SVMs if the kernel function has

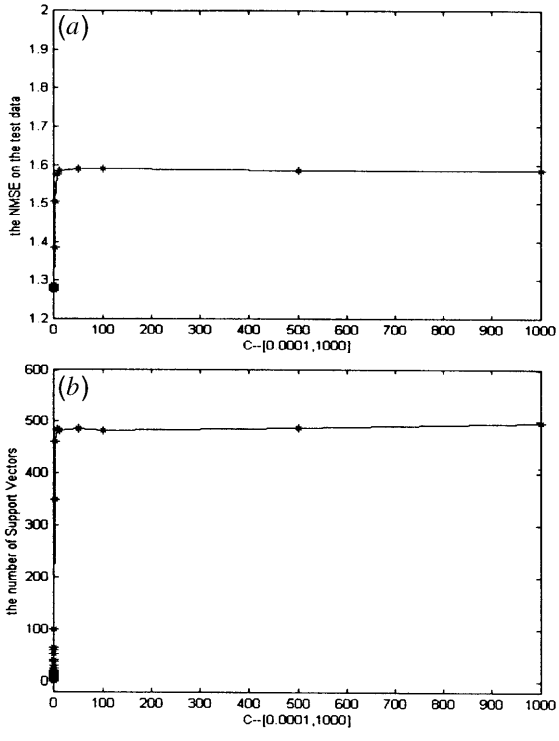

Fig. 4. (a) The generalisation error versus c. The generalisation error is not much influenced by c; (b) the number of support vectors versus c. The number of support vectors increases rapidly when c rises from 0.1 to 1, and then it remains almost stable after c reaches 1.

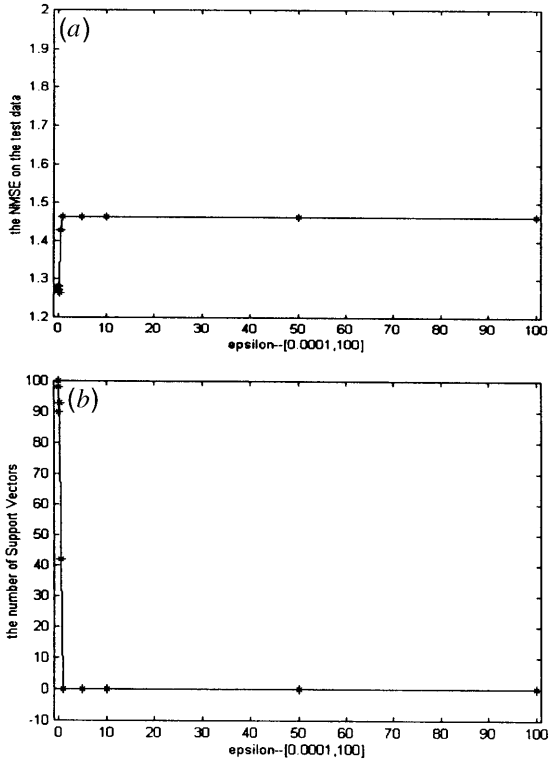

Fig. 5. (a) The generalisation error with ϵ. The generalisation error is not much influenced by ϵ; (b) the number of support vectors with ϵ. The number of support decreases to zero with the increment of ϵ.

been considered. As illustrated in the experiment, they also have little impact on the solution. However, for BP the size of the network, learning parameters and training all greatly affect the prediction performance.

2. SVMs forecast better than BP. As shown in our results, SVMs provide a smaller NMSE and MAE and a larger DS, CP and CD than those of BP. This is because SVMs adopt the Structural Risk Minimisation Principle, eventually leading to better generalisation than conventional techniques.
3. Training SVMs is faster than BP. The regression function in SVMs is only determined by the support vectors, and the number of support vectors is much smaller compared to the number of training samples. In BP, it often converges slowly, and may even converge to local minima because of its local gradient descent learning algorithm.

The study also indicates that the multivariate analysis with the addition of MACD, OBV and volatility indicators does not improve the predictive power of the neural networks.

Although there is little impact on the generalisation error with respect to the free parameters of SVMs, we believe that there is still much room for improvement in SVMs with respect to forecasting financial time series. Future work will focus on this aspect, and the selection of input variables for the multivariate analysis.

Acknowledgements. We would like to express our special thanks to Mr Manzoor UL.S. and Ms. Xiaobu Hong for their assistance in improving the paper. Further, we would like to thank the Man-Drapeau Research Company for providing the data.

References

1. Deboeck GJ. Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets. New York, Wiley, 1994
2. Yaser SAM, Atiya AF. Introduction to financial forecasting. Applied Intelligence 1996; 6: 205–213
3. Abecasis SM, Lapenta ES. Modeling multivariate time series with a neural network: comparison with regression analysis. Proceedings of INFONOR'96: IX International Symposium in Informatic Applications, Antofagasta, Chile, 1996, pp. 18–22
4. Cheng W, Wanger L, Lin CH. Forecasting the 30-year US treasury bond with a system of neural networks. J Computational Intelligence in Finance 1996; 4: 10–16
5. Sharda R, Patil RB. A connectionsist approach to time series prediction: an empirical test. Neural Networks in Finance and Investing, 1993; 451–464
6. Van E, Robert J. The Application of Neural Networks in the Forecasting of Share Prices. Finance & Technology Publishing, Haymarket, VA, USA, 1997
7. Pan ZH, Wang XD. Wavelet-based density estimator model for forecasting. J Computational Intelligence in Finance 1998; 6: 6–13
8. Kimoto T, Asakawa K, Yoda M, Takeoka M. Stock market prediction system with modular neural networks. Neural Networks in Finance and Investing 1993; 343–357
9. Dorsey R, Sexton R. The use of parsimonious neural networks for forecasting financial time series. J Computational Intelligence in Finance 1998; 6: 24–30
10. Abecasis SM, Lapenta SL. Modeling the MERVAL index with neural networks and the discrete wavelet transform. J Computational Intelligence in Finance 1997; 5: 15–19
11. VanAussem A, Campbell J, Murtagh F. Wavelet-based feature extraction and decomposition strategies for financial forecasting. J Computational Intelligence in Finance 1998; 6: 5–12
12. Vapnik VN. The Nature of Statistical Learning Theory. New York, Springer-Verlag, 1995
13. Mukherjee S, Osuna E, Girosi F. Nonlinear prediction of chaotic time series using support vector machines. Proc IEEE NNSP'97, Amelia Island, FL, 1997
14. Vapnik VN, Golowich SE, Smola AJ. Support vector method for function approximation, regression estimation, and signal processing. Advances in Neural Information Processing Systems 1996; 9: 281–287
15. Thomason M. The practitioner methods and tool. J Computational Intelligence in Finance 1999; 7: 36–45

16. Thomason M. The practitioner methods and tool. J Computational Intelligence in Finance 1999; 7: 35–45
17. Scholkopf B, Burges CJC, Smola AJ. Advances in Kernel Methods. The MIT Press, London, England, 1999
18. Smola AJ, Scholkopf B. A tutorial on support vector regression. NeuroCOLT Technical Report TR, Royal Holloway College, London, UK, 1998
19. Smola AJ. Learning with Kernels. PhD Thesis, GMD, Birlinghoven, Germany, 1998
20. Haykin, S. Neural Networks: a Comprehensive Foundation. Prentice Hall International, 1999