Application of Support Vector Machine to Forex Monitoring

Joarder Kamruzzaman∗ Non-member
Ruhul A Sarker** Non-member

Previous studies have demonstrated superior performance of artificial neural network (ANN) based forecasting models over traditional regression models. This paper applies support vector machines to build a forecasting model from the historical data using six simple technical indicators and presents a comparison with an ANN based model trained by scaled conjugate gradient (SCG) learning algorithm. The models are evaluated and compared on the basis of five commonly used performance metrics that measure closeness of prediction as well as correctness in directional change. Forecasting results of six different currencies against Australian dollar reveal superior performance of SVM model using simple linear kernel over ANN-SCG model in terms of all the evaluation metrics. The effect of SVM parameter selection on prediction performance is also investigated and analyzed.

Keywords: Forecasting, foreign exchange, neural network, support vector machine.

1. Introduction

The foreign exchange rates play an important role in the international trade and business. As a result, the appropriate prediction of exchange rate is a crucial factor for the success of many businesses and fund managers. It is generally known that the currency exchange market is highly volatile and fluctuating with time. Therefore, it is not possible to exploit or predict its behaviour. A common myth of traders, however, is that there is certain predictability in the currency market dynamics. There are several recent works in the field of prediction and forecasting which indicate that these beliefs by traders are real and the market in some aspect shows predictable dynamics (2) (16)–(18).

Prediction of exchange rates is one of the challenging applications of modern time series forecasting. The rates are inherently noisy, non-stationary and deterministically chaotic (9) (33). These characteristics suggest that there is no complete information that could be obtained from the past behaviour of such markets to fully capture the dependency between the future rates and that of the past. One general assumption is made in such situations is that the historical data incorporate all those behaviour. That means the historical data is the major player in the prediction process. Under this condition, the obvious question is: how good are those predictions?

For the last two decades, Box and Jenkins’ Auto-Regressive Integrated Moving Average (ARIMA) technique (1) has been widely used for time series forecasting. Because of its popularity, the ARIMA model has been used as a benchmark to evaluate some new modelling approaches (14). However, ARIMA is a general univariate model which is developed based on the assumption that the time series being forecasted are linear and stationary (2).

Artificial Neural Network (ANN), the well-known function approximator for prediction and system modelling, has proved itself as a great tool for many applicability in time-series analysis and forecasting (31)–(34). ANNs are universal function approximators that can map any nonlinear function without a priori assumptions about the data (2). In several applications, Tang and Fishwich (27), Jhee and Lee (15), Wang and Leu (29), Hill et al. (13), and Shazly et al. (26), Leung et al. (22), and many other researchers have shown that ANNs perform better than ARIMA models, specifically, for more irregular series and for multiple-period-ahead forecasting. Kaastra and Boyd (16) provided a general introduction of how a neural network model should be developed to model financial and economic time series. Zhang and Hu (23) analysed backpropagation neural networks’ ability to forecast an exchange rate. Wang (29) cautioned against the dangers of one-shot analysis since the inherent nature of data could vary. Klein and Rossin (19) proved that the quality of the data also affects the predictive accuracy of a model. More recently, Yao et al. (31) evaluated the capability of a backpropagation neural network model as an option price forecasting tool. They have recognised the fact that neural network models are context sensitive and when studies of this type are conducted, it should be as comprehensive as possible for different markets and different network models. Kamruzzaman and Sarker (17) (18) investigated three ANN based forecasting models using Standard Backpropagation, Scaled Conjugate Gradient and Backpropagation with Baysian Regularization with six technical indicators to predict six major currencies against Australian dollar. Performance measured in terms of five metrics showed that the best result is achieved with
Support Vector Machine Regression Models

2. Support Vector Machine Regression Model

Support vector machine introduced by Vapnik (26) has attracted many research attention in recent years due to its demonstrated improved generalization performance over other techniques in many real world applications. It has been used in classification as well as regression tasks. The main difference between this technique and many other conventional regression techniques including neural network is that it minimizes the structural risk instead of the empirical risk. The principal is based on the fact that minimizing an upper bound on the generalization error rather than minimizing the training error is expected to perform better. The generalization error is bounded by the sum of training error rate and a term that depends on Vapnik-Chervonenkis (VC) dimension (12). VC dimension is a measure of complexity of the dimension space. Support vector machines find a balance between the empirical error and the VC confidence interval. SVMs perform by nonlinearly mapping the input data into a high dimensional feature space by means of a kernel function and then do the linear regression in the transformed space. The whole process results in nonlinear regression in the low-dimensional space.

Consider a data set consisting \( G = \{(x_i, y_i)\}_{i=1}^N \) of \( N \) data points where each input \( x_i \) is mapped into the corresponding output \( y_i \). Given that the data set realizes some unknown function \( g(x) \), we need to determine a function \( f \) that approximates \( g(x) \), based on the knowledge of data set \( G \). In SVM, \( x_i \) is first mapped into a higher dimensional space \( F \) via a nonlinear mapping, and perform linear regression in this space. The SVM approximates the function as

\[
f(x) = \sum_{i=1}^{D} \omega_i \Phi(x_i) + b \quad \text{with} \quad \Phi : \mathbb{R}^n \rightarrow F, \omega \in F,
\]

(1)

where \( \omega_i \) are the coefficients and \( b \) is a threshold value. This approximation can be considered as a hyperplane in the \( D \)-dimensional feature space \( F \) defined by the functions \( \Phi(x) \) where the dimensionality can be very high, possibly infinite. Since \( \Phi \) is fixed, \( \omega_i \) can be determined from the data by minimizing the sum of empirical risk and a complexity term defined in the following risk function.

\[
R = C \frac{1}{N} \sum_{i=1}^{N} \epsilon_i + \frac{1}{2} \| \omega \|^2,
\]

(2)

where \( \epsilon \) is a parameter to be set a priori and an error below \( \epsilon \) is not penalized according to the following error function.

\[
\epsilon_i = \begin{cases} 
0 & \text{if} |y_i - f(x_i)| < \epsilon \\
|y_i - f(x_i)| & \text{otherwise}.
\end{cases}
\]

(3)

The first term in Eq. (2) describes the \( \epsilon \)-insensitive loss function and the second term is a measure of function flatness. Thus SVMs perform linear regression in high dimensional feature space using \( \epsilon \)-insensitive loss and at the same time tries to reduce model complexity by minimizing \( \| \omega \|^2 \). The constant \( C > 0 \) is a regularization constant determining the trade-off between the training error and model flatness. Introducing the slack variables \( \xi \) and \( \xi^* \), SVM regression is formulated as minimization of the following optimization problem:

\[
\text{minimize} \quad \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*)
\]

(4)
subject to \[
\begin{aligned}
&f(x_i) + b - y_i \leq \epsilon + \xi_i \\
&y_i - f(x_i) - b \leq \epsilon + \xi_i^* \\
&\xi_i, \xi_i^* \geq 0
\end{aligned}
\]

The above optimization with constraint is a quadratic programming problem that can be solved by constructing a Lagrangian and transforming into the dual problem, and its solution is given as follows\(^{(28)}\).

\[
f(x, \alpha, \alpha^*) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i)K(x_i, x_i) + b, \cdots \cdots (5)
\]

where Lagrange multipliers \(\alpha_i\) and \(\alpha_i^*\) are associated with each data point \(x_i\), and subject to the constraints \(0 \leq \alpha_i^* \leq C\) and \(\sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0\). Training points with nonzero Lagrange multipliers are called support vectors. The smaller the fraction of support vectors, the more general the solution is but large support vectors do not necessarily mean an overtrained solution. The predicted value \(X_{t+1}\) is the following week’s exchange rate.

### 3.2 Evaluation of the Model

The forecasting performance of the above model is evaluated against five widely used statistical metrics, namely, Normalized Mean Square Error (NMSE), Mean Absolute Error (MAE), Directional Symmetry (DS), Correct Up trend (CP) and Correct Down trend (CD). These criteria are defined in Table 1. NMSE and MAE measure the deviation between the actual and forecasted values. Smaller values of these metrics indicate higher accuracy in forecasting. DS measures correctness in predicted directions. CP and CD measure the correctness of predicted up and down trend, respectively. In general, it is desirable for a forecasting model to attain low NMSE and MAE but high DS, CP and CD. In practice, we may have a model that yields superior NMSE and MAE but inferior DS, CP and CD than other models, or vice versa. Several authors have argued that directional change metrics may be a better standard for determining the quality of forecasting. However, which metric should be given more importance depends on trading strategies and how the prediction is used for trading advantages\(^{(31)}\).

### 4. Simulation Results and Discussion

The foreign exchange rates of six different currencies against Australian dollar are used to build the model. The data showing the exchange rate of US dollar, British Pound, Japanese Yen, Singapore dollar, New Zealand dollar and Swiss Franc against Australian dollar during January 1991 to July 2000 is made available by the Reserve Bank of Australia. In total 565 weekly data were considered of which the first 500 weekly data were used as the training set to build the SVM model. The remaining 65 weekly data were used to validate the model.

To build an SVM model the user needs to select the kernel type and the value of the parameters \(C\) and \(\epsilon\). In this study, the model was built using the simple linear kernel defined as \(K(x_i, x_j) = <x_i, x_j> \) where \(< \cdot >\) is the inner product operator. The regularization parameter \(C\) was varied from 0.1 to \(10^5\) and \(\epsilon\) parameter was varied from 0.0005 to 0.006 to investigate their effects on prediction and discussed later in this section. The forecasted result over the test data was compared with the Scaled Conjugate Gradient (SCG) neural network model studied by Kamruzzaman and Sarker\(^{(17)}\)(18). In that study, it was concluded that the scaled conjugate gradient based model was the best among the three neu-
Support Vector based Forex Forecasting

crnetial network models tested for predicting those currency exchange rates under consideration. The performance metrics were calculated over 35 weeks and 65 weeks forecasted data as well as over the training data set. Table 2 shows the performance metrics over the test and training data set for six currencies against Australian dollar and presents a comparison with the neural net model. The table presents the best result achieved in each case by SVM model with C varied within the range specified above and keeping C fixed at 0.001.

A comparison of results with neural network based model illustrates that SVM based forecasting model can predict the exchange rate better, in terms of both accuracy (NMSE, MAE) and trend prediction (DS, CP, CD). SVM based model produces lower NMSE and MAE and higher DS, CP and CD than those of NN based model. Especially in the first 35 weeks, prediction by SVM is much better than those achieved by the neural network

### Table 1. Performance metrics used in this experiment

<table>
<thead>
<tr>
<th>Metric</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMSE</td>
<td>$NMSE = \frac{1}{N} \sum (x_k - \hat{x}_k)^2$</td>
</tr>
<tr>
<td>MAE</td>
<td>$MAE = \frac{1}{N} \sum</td>
</tr>
<tr>
<td>DS</td>
<td>$DS = \frac{100}{N} \sum d_k,$</td>
</tr>
<tr>
<td>CP</td>
<td>$CP = \frac{100}{N} \sum d_k,$</td>
</tr>
<tr>
<td>CD</td>
<td>$CD = \frac{100}{N} \sum d_k,$</td>
</tr>
</tbody>
</table>

### Table 2. Performance metrics calculated over the training and test data for six currencies against Australian dollar

<table>
<thead>
<tr>
<th>Period</th>
<th>Criteria</th>
<th>GB Pound</th>
<th>US Dollar</th>
<th>Japanese Yen</th>
<th>Singapore Dollar</th>
<th>NZ Dollar</th>
<th>Swiss Franc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NN</td>
<td>SVM</td>
<td>NN</td>
<td>SVM</td>
<td>NN</td>
<td>SVM</td>
</tr>
<tr>
<td>35 week forecast</td>
<td>NMSE</td>
<td>0.1578</td>
<td>0.1276</td>
<td>0.2624</td>
<td>0.2198</td>
<td>0.1264</td>
<td>0.108</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.0030</td>
<td>0.0027</td>
<td>0.0065</td>
<td>0.0031</td>
<td>0.0243</td>
<td>0.0569</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>77.145</td>
<td>80.00</td>
<td>80.00</td>
<td>80.00</td>
<td>80.00</td>
<td>82.857</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>81.25</td>
<td>87.50</td>
<td>82.353</td>
<td>82.353</td>
<td>81.818</td>
<td>81.818</td>
</tr>
<tr>
<td></td>
<td>CD</td>
<td>73.684</td>
<td>73.684</td>
<td>82.353</td>
<td>82.353</td>
<td>76.923</td>
<td>76.923</td>
</tr>
<tr>
<td>65 week forecast</td>
<td>NMSE</td>
<td>0.0729</td>
<td>0.0573</td>
<td>0.0418</td>
<td>0.0380</td>
<td>0.0411</td>
<td>0.0350</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.0023</td>
<td>0.0020</td>
<td>0.0029</td>
<td>0.0028</td>
<td>0.5188</td>
<td>0.4789</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>84.615</td>
<td>84.615</td>
<td>81.538</td>
<td>81.538</td>
<td>81.538</td>
<td>81.538</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>87.879</td>
<td>87.879</td>
<td>78.947</td>
<td>78.947</td>
<td>83.784</td>
<td>83.784</td>
</tr>
<tr>
<td></td>
<td>CD</td>
<td>83.871</td>
<td>83.871</td>
<td>88.461</td>
<td>88.461</td>
<td>78.571</td>
<td>78.571</td>
</tr>
<tr>
<td>Training Data</td>
<td>NMSE</td>
<td>0.0052</td>
<td>0.0052</td>
<td>0.0023</td>
<td>0.0024</td>
<td>0.0035</td>
<td>0.0033</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.0025</td>
<td>0.0025</td>
<td>0.0027</td>
<td>0.0027</td>
<td>0.5454</td>
<td>0.5145</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>76.800</td>
<td>76.800</td>
<td>78.800</td>
<td>78.800</td>
<td>79.200</td>
<td>75.200</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>76.623</td>
<td>76.190</td>
<td>77.381</td>
<td>77.381</td>
<td>77.291</td>
<td>76.892</td>
</tr>
<tr>
<td></td>
<td>CD</td>
<td>77.528</td>
<td>77.903</td>
<td>80.894</td>
<td>81.807</td>
<td>73.684</td>
<td>72.874</td>
</tr>
</tbody>
</table>
model. This suggests that further improvement in forecasting may be obtained by rebuilding the model probably after every 30 weeks. Better prediction performance by SVM was consistently observed in all currencies. For example, in forecasting Singapore dollar, NMSE produced by neural network over 35 weeks is 0.2321 and that by SVM is 0.1512 which is large improvement over neural network. Significant improvement by SVM is also shown in case of US dollar and British Pound, however, improvement is little in New Zealand dollar. This suggests that the amount of improvement depends on other factors including the pattern of the data. An analysis of the historical data of all the currencies shows that Japanese Yen experienced the most fluctuation within the period we considered in the training phase. On the other hand New Zealand dollar is the least fluctuating one. This suggests that, with much fluctuation in data, neural network is unable to find appropriate set of weight that can accurately forecast the exchange rate. However, with low fluctuation, it is easier for neural network to learn the training data. This makes SVM is more effective for predicting exchange rate when the training data is much fluctuating. Figure 1(a)–(f) shows the actual and forecasted time series by VSM and scaled conjugate gradient based neural network. For example, in predicting exchange rate of British Pound between weeks 18~24 where a sudden sharp fall is followed by a steep rise (Fig.1(a)), SVM model could closely follow the actual rate whereas NN model suffered from gross deviation. Similar trend is also evident in other currencies.

![Fig. 1. Forecasting of different currencies by SVM and SCG based ANN model over 65 weeks.](image-url)
The ability of SVM model to closely follow the actual rate in such a situation will make a significant difference in the outcome of financial gain in actual exchange trading.

In order to investigate the effect of regularization parameter $C$ on prediction performance, we have experimented with different values of $C$ ranging from 0.1 to $10^5$. In general when $C$ is too large, emphasis is placed on minimizing the empirical risk which may result in over-fitting of training data and lead to a deterioration of prediction performance. Too small value of $C$ means the model will not learn from the training data properly resulting in under-fitting and leading to inferior performance. Figure 2(a)–(c) illustrates the prediction error of Japanese Yen versus $C$. Figure 2(a) shows that NMSE over 65 weeks forecasting result drops from 0.06 to 0.035 when $C$ is increased from 0.5 to 50. For any value larger than 50, NMSE remains the same. MAE also exhibits the similar behaviour. Figure 2(c) shows that DS varies from 75.4% to 83.0% when $C$ is increased from 0.5 to 7.0, later DS remains 81.54% as $C$ increases further. This indicates little impact on the variation of prediction error with respect to regularization parameter at this stage. This kind of behaviour, i.e., negligible effect of $C$ on performance when $C$ is larger than a particular threshold is also supported in a recent study by Cherkassky et al. (4).

The performance of SVM also depends on the parameter $\epsilon$ which controls the width of $\epsilon$-insensitive tube used to fit the training data. A larger value means support vectors will allow more margin of error while a smaller value will tightly fit the training data set but generalize poorly on the unseen data. It can also affect the number of support vectors that constructs the regression function. Larger $\epsilon$-value results in fewer SVs and more ‘flat’ (less complex) regression estimates (4). In order to investigate the effect of $\epsilon$ on prediction performance in this study, we experimented with 5 different values within the range 0.0005 to 0.006. Usual practice is to keep the value of $\epsilon$ small, however, the most appropriate value would depend on the particular data set. Figure 3 shows the effect of $\epsilon$ on NSME, MAE and DS. It displays that the best result is achieved when $\epsilon=0.001$. All the three metrics deteriorate for any smaller or larger value of $\epsilon$. However, even for relatively large value (0.006) SVM

![Graphs showing prediction performance versus regularization parameter C for predicting Japanese Yen exchange rate.](a) ![Graphs showing prediction performance versus regularization parameter C for predicting Japanese Yen exchange rate.](b) ![Graphs showing prediction performance versus regularization parameter C for predicting Japanese Yen exchange rate.](c)

Fig. 2. Prediction performance versus regularization parameter $C$ for predicting Japanese Yen exchange rate.

![Graphs showing variation of NMSE, MAE and DS with parameter $\epsilon$ for predicting Japanese Yen exchange rate.](a) ![Graphs showing variation of NMSE, MAE and DS with parameter $\epsilon$ for predicting Japanese Yen exchange rate.](b) ![Graphs showing variation of NMSE, MAE and DS with parameter $\epsilon$ for predicting Japanese Yen exchange rate.](c)

Fig. 3. Variation of NMSE, MAE and DS with parameter $\epsilon$ for predicting Japanese Yen exchange rate.
model did not perform inferiorly to ANN model.

In literature a number of techniques have been proposed to optimally determine the parameters $C$ and $\epsilon$. Mettger et al. suggested to set parameter $C$ equal to the range of output values (23). But its does not take into account the possible effect of outliers in training data. Hastie et al. proposed to estimate the optimal $C$ value by cross validation on the data set (31) but this approach is computationally very expensive and data intensive. Kwok proposed asymptotically optimal values which are proportional to noise variance (20). But this proposal does not reflect sample size while it is generally known that $\epsilon$ should be smaller for larger sample size. Hastie et al. proposed an approach where $\epsilon$ is optimally tuned for a particular noise density (11). As Cherkassky et al. pointed out there have been many conflicting strategies suggested for setting the value of $C$ and $\epsilon$ parameters (4). Very recently, they proposed a methodology to analytically determine the value of $C$ and $\epsilon$. The method is claimed to determine “practical” values of $C$ and $\epsilon$, however, whether this method can select the best or optimum value is yet to be verified in a wide range of real world problems.

5. Conclusion

Earlier studies (17) reported that an ANN based forex model trained with scaled conjugate gradient learning algorithm predicted currency exchange rate more accurately than standard backpropagation based learning algorithm, and both the models outperformed traditional ARIMA model. Superiority of ANN model over ARIMA has also been supported by other studies (32). For further improvement in prediction accuracy, we investigated the suitability of support vector machine in forecasting of exchange rates of six currencies against Australian dollar. An SVM based forecasting model was found to outperform the best ANN model measured on five commonly used evaluation metrics. Moreover, the performance of ANN model depends on a number of factors, e.g., the large number of degrees of freedom related to network architecture, tuning of learning parameters, complexity of the model etc. ANN training may stuck in local minima resulting in a very slow convergence and may require many trials before finding an optimal architecture that yields acceptably good performance.

In SVM, the only free parameters are $C$ and $\epsilon$. In this study, we observed that over a small range of value, $C$ affected generalization performance moderately, but its effect was negligible when $C$ exceeded certain threshold. This means that over a wide range of value, $C$ has very little impact on prediction performance. The value of $\epsilon$ slightly affected the prediction accuracy. However, it did not cause any drastic degradation in performance.

Future research direction will focus on achieving further improvement in prediction by considering other variations of support vector machines, use of complex or mixed kernels and automatic selection of parameters.  

(Manuscript received Dec. 8, 2003, revised June 16, 2004)

References

Support Vector based Forex Forecasting


Joarder kamruzzaman (Non-member) received a B.Sc and M.Sc. in Electrical Engineering from Bangladesh University of Engineering & Technology, Dhaka, Bangladesh in 1986 & 1989 respectively, and a PhD in Information System Engineering from Muroran Institute of Technology in 1993. Currently he is a faculty member in the Faculty of Information Technology, Monash University, Australia. His research interest includes neural networks, fuzzy system, genetic algorithm, computer networks and bioinformatics etc. He has published more than 60 refereed papers in international journals and conference proceedings. He is a program committee member of a number of international conferences.

Ruhul A. Sarker (Non-member) received his Ph.D. in 1991 from DalTech, Dalhousie University, Halifax, Canada, and is currently a senior faculty member in the School of Information Technology and Electrical Engineering, University of New South Wales, ADFA Campus, Canberra, Australia. His main research interests are Evolutionary Optimization, Neural Networks and Applied Operations Research. He has recently edited four books and has published more than 90 refereed papers in international journals and conference proceedings. Dr Sarker is actively involved with a number of national and international conferences & workshop organizations in the capacity of chair / co-chair or program committee member. He has recently served as a technical co-chair for IEEE-CEC2003. He is also a member of task force for promoting evolutionary multi-objective optimization operated by IEEE Neural Network Society.