Improving the Accuracy of Financial Time Series Prediction Using Ensemble Networks and High Order Statistics

Roy Schwaerzel*
Division of Computer Science
University of Texas at San Antonio
6900 North Loop 1604 West
San Antonio, Texas 78249-0667
rschwaer@dragon.cs.utsa.edu.

Bruce Rosen
Division of Computer Science
University of Texas at San Antonio
6900 North Loop 1604 West
San Antonio, Texas 78249-0667
rosen@dragon.cs.utsa.edu.

Abstract

We apply neural network ensembles to the task of forecasting financial time series and explore the use of high order statistical information as part of network inputs. We show that the prediction accuracy of the time series can be significanlty improved utilizing this methodology. Since prediction accuracy is only an estimate for the profitability on the financial market, we report good and profitable results using a profit/loss metric based on market simulations. Our simulations show an improvement of between 1.3 to 12.4% over a simple buy and hold trading strategy, and an improvement of between 6.5 to 20.9% over trading strategy using linear autoregressive models.

Keywords: *Time Series Analysis, Ensemble Networks, Back-Propagation, High Order Statistics*

1. Introduction

Forecasting financial time series relies on the discovery of strong empirical regularities in observations of the system and has been widely discussed [2][4][5]. Because these regularities are often masked by noise and often have nonlinear and nonstationary behavior, it has been suggested that some financial time series are not predictable. When applied to daily rates on the financial market, the random walk hypothesis (RWH) states that a financial time series is defined by constant expected price changes between successive trading days and zero correlation between the price change for any pair of different trading days. If the RWH applies to financial series, it is unnecessary to attempt to develop a good forecasting system. All prediction systems based on the previous behavior of the time series are then useless, because the best prediction of tomorrow's price would be today's price plus some constant price change. However, the RWH does not appear valid for many financial time series. For example, using different trend statistics and tests, Taylor [6] concludes many financial time series have non-random behavior. Thus, the financial market is not always completely efficient and correlation and trends can sometimes be found within data. We investigated that the predictability of a time series could be improved using high order statistical features.

*This research was supported in part by the Friedrich Naumann Foundation (Germany) and by the Small Grants Program at UTSA.

2. Ensemble Neural Network Design

Although single neural network systems (NNS) have been used for the prediction of financial time series, neural network approaches combining the results of several individual neural networks often show improved performance [1][3]. Ensemble networks consist of several independently trained neural networks combined as inputs to a single master network. The master network is trained to find an optimal weight combination producing the minimum of the mean square error (MSE) between the desired and the master network's output with respect to the distribution of the training data. This design is based on the idea that an ensemble of neural networks will perform better than any individual neural network.

The Generalized Ensemble Method (GEM), as proposed by Perrone [3], generates a regression estimate which is as low or lower than the simple averaging estimator. The GEM is the linear combination of the estimators based on the empirical MSE. The GEM regression function, $f_{GEM}(x)$, is defined by

$$f_{GEM}(x) \equiv \sum_{i=1}^{N} \alpha_i f_i(x) \ , \qquad (1)$$

where the α_is are weighting parameters that satisfy the constraint $\sum \alpha_i = 1$. Each α_i is defined by

$$\alpha_i = \frac{\sum_j C_{ij}^{-1}}{\sum_k \sum_j C_{kj}^{-1}} \ , \qquad (2)$$

where C_{ij} are the elements of the covariance matrix of the errors from the function estimators f_i and f_j. We implemented the GEM model using high order statistical features (HOSF) on several major currency exchange rates.

3. Experimental Design

Daily currency exchange rates of four major currencies (British Pound, German Mark (DM), Japanese Yen (JY), and Swiss Franc) against the US Dollar (US$) were available from 62 contiguous months. We have chosen to predict the DM and the JY. The DM has the lowest Hurst coefficient (0.5091) among all four currency exchange rates, which indicates there is little persistency and predictability in the time series. Alternatively, the JY exhibits the highest Hurst coefficient (0.5332) which indicates a higher level of persistency in the time series, and hence more predictability. Several network configurations each consisting of multiple single networks and a master network were applied to predict the next day's and the next week's return for the DM and the JY. Using daily returns instead of the original exchange rates transformed the original nonstationary time series into a stationary one.

We analyzed the autocorrelation coefficients of the DM and the JY currency exchange rates and found that future values are influenced not only by their immediate predecessor values but also by a series of preceding values. To capture this dynamic of time series, time windowing was used by transposing sequential segments of the original time series into spatial vectors [4]. The window, W_m^i, of fixed size m was used to partition the original data set into input pattern vectors. The output, W^o, was the future value of the time series d time steps in the future. It is assumed that the sequence in W_m^i is correlated to the value in W^o, and that the regularities of a time series can be recognized by moving both windows over the entire data set. Each pair of windows W_m^i and W^o, separated by the distance d, is taken as the input and output vector for our neural networks. The choice of the window size m and distance d was more critical, and often determines the quality of the prediction. The analysis of the autocorrelation coefficients indicates the existence of an underlying day-of-the-week effect, because the two major currencies have a positive autocorrelation coefficient for lags 5, 10, and 15. These lags correspond to daily returns exactly one, two, and three weeks previous to the current day[1]. Considering this effect, two window sizes, W_5^i and W_{10}^i, were chosen for our simulations. In the case of $m = 5$, time windowing generates a set of pairs, each consisting of an input vector of size 5 and an output vector of size 1. To predict the next day's return, d is set to 1. To predict the next week's return, d is set to 5.

A statistical description of the previous observations of a given time series often contains important information that is not easily detected by using the daily returns from the last five or ten trading days. We found that when predicting the daily or weekly return, x_{t+d}, the *moving average* $\bar{x}$, *standard deviation* s, *skewness* b, and *kurtosis* k, HOSF calculated from the previous five and twenty daily returns x_t provided significant additional information for predicting x_{t+d}. Further, the *exponential moving average*, e_α, with smoothing constants of $\alpha = 0.5$ and $\alpha = 0.8$ also significantly aided the prediction accuracy of our models

$$e_{t,\alpha} = \alpha * x_t + (1 - \alpha) * e_{t-1,\alpha} \ . \qquad (3)$$

Our experiments were heavily influenced by the values of the experiment's fixed parameters. The back-propagation algorithm with conjugate gradient learning was used to train the neural networks, and the network's weights were initialized randomly. There were 1,588 input and output pairs to predict the next day's return, and 1,584 pairs to predict the next week's return. The first 1,328 input-output pairs were used for the training and crossvalidation set, while the remaining data was used for testing. The test set contains all input-output pairs from the entire year 1995. The training set consisted of 70% of the randomly chosen pairs from November 1989 until December 1994, while the crossvalidation set contained the remaining 30% of the data. We used one layer of fully connected hidden units. For all our experiments, we trained ten networks which differed in the number of hidden units, starting from one hidden unit and increasing by two units for the next largest neural network. Crossvalidation determined when to stop training. Figures 1 and 2 show the MSE for the training, crossvalidation, and test sets plotted against the number of training cycles for typical predictions of the DM and the JY.

In both cases, neural networks with 15 input units, 9 hidden units, and 1 output units were trained to predict the next day's return. Both plots show that training should be stopped after a relative short training phase. Based on these preliminary findings, the number of training cycles was set at 1,000 cycles. Table 1 shows the three factors and levels that were chosen as factors for our full factorial design (currency type, input vector length and features type, and distance of the prediction).

We investigated three metrics to guage the performance of the networks. The *Mean Square Error (MSE)* was used directly by the back-propagation algorithm to determine the

[1] The data set for our currency exchange rates contains only trading days that are weekdays. Therefore, a time lag of five trading days represents the same day of the week.

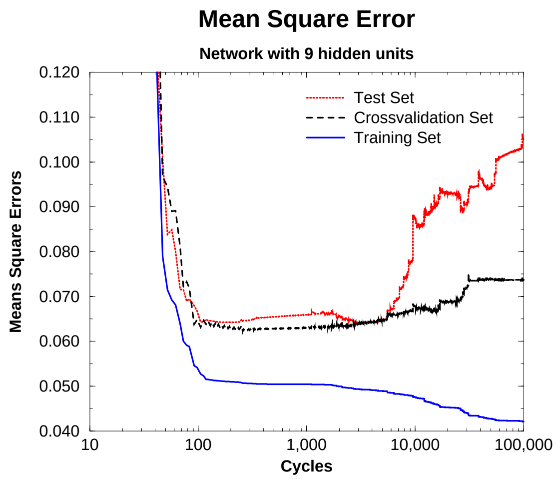

Figure 1. Plot of the MSE for a slightly oversized neural network (15 input units, 9 hidden units, and 1 output unit) against the number of training cycles for the next day's return predictions of the DM. The minimum crossvalidation error was in epoch 257.

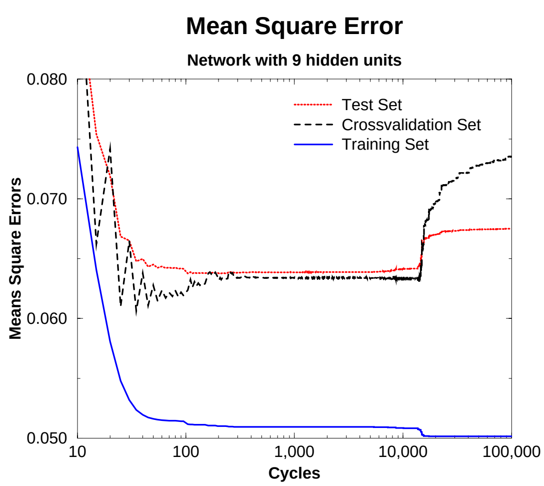

Figure 2. Plot of the MSE for a slightly oversized neural network (15 input units, 9 hidden units, and 1 output unit) against the number of training cycles for the next day's return predictions of the JY. The minimum crossvalidation error was in epoch 35.

Factor	Levels	Comment
Currency Currency	German Mark (DM)	lowest Hurst coefficient
	Japanese Yen (JY)	highest Hurst coefficient
Input Vector	Last five trading days	last week information
	Last ten trading days	last two weeks information
	10 HOSF	statistical indices of last 4 weeks
	Last 5 days and 10 HOSF	last week information, stats of last 4 weeks
	Last 5 days from 4 major currencies	multivariate analysis, last week information
	Last 10 days from 4 major currencies	multivariate analysis, last two weeks information
Output Vector	Next day's return	short term predictions short term predictions
	Next week's return	utilization of the day-of-the-week effect

Table 1. Definition of the factors and the levels for each factor for the full factorial experimental design.

neural network weight updates and the stopping point for training. The *predicted direction metric (PD)* is the number of right predictions of the direction for the next daily or weekly return to the total number of predictions and gives an initial estimate of the profitability of the system. If a system can predict the direction of the next trading day's exchange rate with 60% accuracy, this system may be able to produce a profit. The *profit/loss analysis* (market simulation) simulates the real cash flow of investments in the financial market. In this method, a foreign currency is bought or held whenever a positive return is predicted, and is sold, when the predicted return is negative. Simulations including trading costs were used for the performance evaluation of our NNS.

4. Simulation Results

Ensemble networks with a different number of subnetworks were used to predict the next day's or next week's return for the DM and the JY. A full factorial design of 24 experiments was performed. Each experiment consisted of training 10 single neural networks with different numbers of hidden units (1, 3, ..., 19). Each single network was trained for 1,000 cycles, and the weight set with the best crossval-

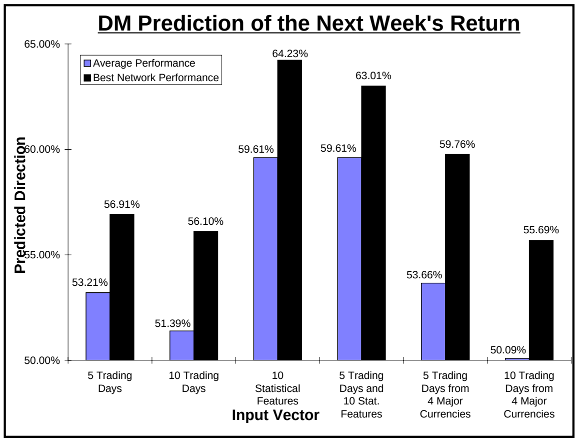

Figure 3. Comparison of PD index of the next week's return using different input vectors for trading the DM.

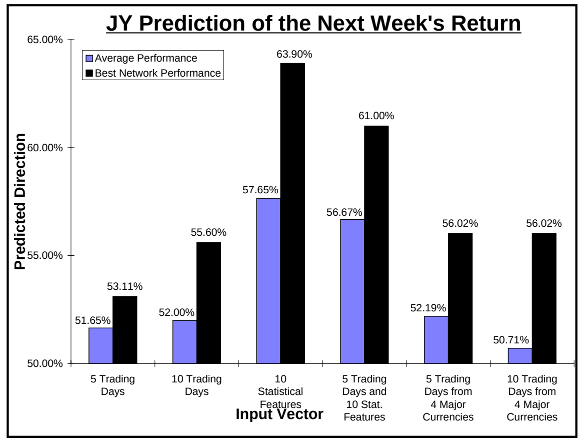

Figure 4. Comparison of PD index of the next week's return using different input vectors for trading the JY.

idation error found during that time was used for testing. These networks were successively combined to form 9 different ensemble networks starting from 2 single networks. Each successive ensemble network contained one additional network. Although the MSE was the performance metric for the training and crossvalidation phase, the PD index was used to evaluate the prediction accuracy.

In general, neural networks utilizing our HOSF ($\bar{x}$, s, b, k from two input windows of length 5 and 20 trading days, and $e_{0.5}$ and $e_{0.8}$)performed significantly better than networks without the HOSF. For example, predictions for the next day's return of the DM based on the last 5 trading days and the 10 HOSF gave better results than predictions based on other input vectors, e.g. the previous 10 trading days or even the last 10 trading days from four major currencies. The best results for the next week's predictions of the DM and JY were achieved using the input vector containing the ten HOSF only (Figures 3 and 4).

Ensemble neural networks were able to lower the mean and the variance of the prediction error by averaging the forecasts from their individual neural networks. For example, the analysis of the next week's prediction of the German Mark shows the mean of the ensemble networks (60.21%) is higher than the mean of all single neural networks (59.07%). Further, the worst performance of a single neural network is 52.85% PD accuracy, whereas the worst performance for a master network is 58.54%. Similar results were achieved for predicting the JY, where the average PD accuracy for the master networks is 58.41% compared with 56.97% for their individual neural networks.

5. Performance Evaluation

The *buy-and-hold strategy (BHS)* was used as a benchmark for comparisons with a linear regression trading model and our NNS. Trading costs of 0.05% per round-trip (one buy and one sell) transaction were used for our final trading simulations. The *linear autoregression model (LAM)* was applied to predict the daily returns of the JY and the DM. This model used the same data that was used to train the neural networks, but used only the returns from the previous 10 trading days. The predictions were incorporated into the following trading strategies: (1) Buy or hold a currency whenever a positive return is expected, and (2) sell a currency when a negative return is expected.

The market simulation of these trading strategies based on predictions of the LAM initiated 74 round-trip (buy and sell) transactions for the DM and 79 round-trip transactions for the JY. In the case of the German investor trading US$ (Figure 6), the regression model reduced the loss for the BHS from 7.44% to only 1.36% for the regression model. But this is the only case where the simple trading strategy based on predictions of the LAM performed better than the BHS. Trading between the JY and the US$ based on the regression model produced a negative return for both directions, as shown in Figures 7 and 8.

The neural networks with the best PD index for the test data were used for our performance evaluation. The best NNS for the next day's return predictions of the DM (PD accuracy of 60.40%) was an ensemble network consisting of five subnetworks. The best NNS for the next day's return prediction of the JY (PD accuracy of 60.82%) was an ensemble network consisting of two subnetworks. The performance of these selected NNS was evaluated by utilizing

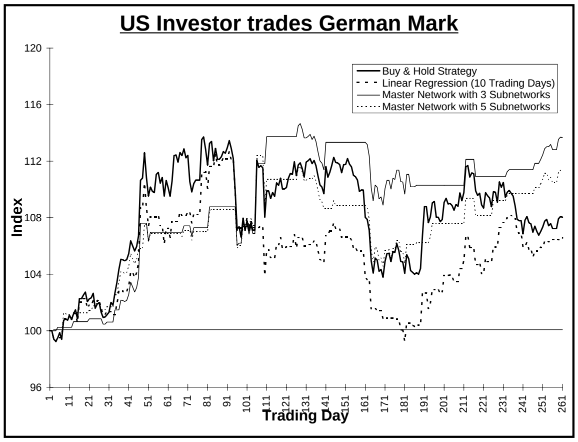

Figure 5. Profitability of the BHS, the LAM, and two NNS for trading DM against US$.

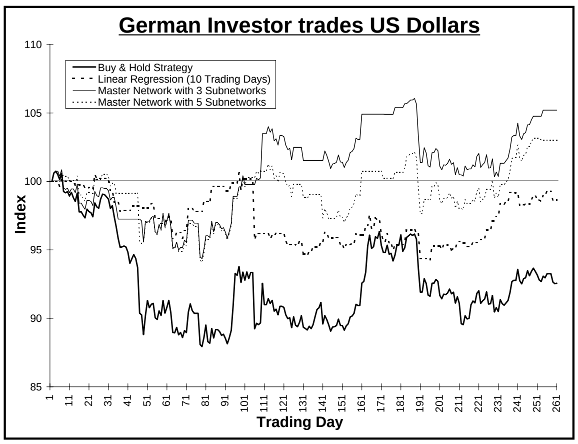

Figure 6. Profitability of the BHS, the LAM, and two NNS for trading US$ against DM.

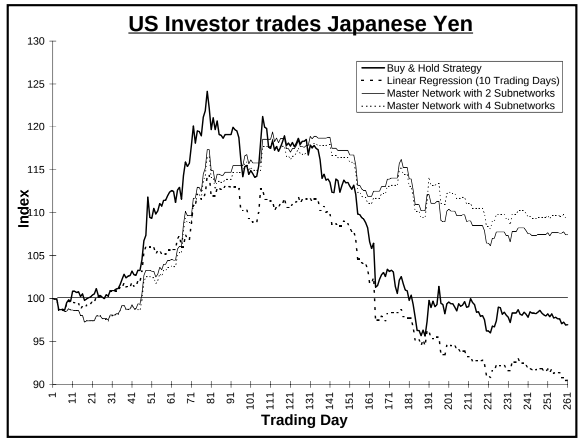

Figure 7. Profitability of the BHS, the LAM, and two NNS for trading JY against US$.

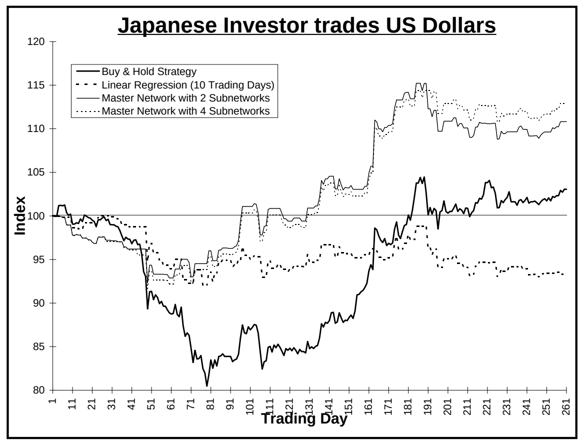

Figure 8. Profitability of the BHS, the LAM, and two NNS for tradingUS$ against JY.

the same simple trading strategy, previously described, and using the predictions of our neural networks. The profitability of the market simulation for 1995 for the BHS, the LAM, and the trading based on predictions of two NNS is summarized in Table 2. Our two selected neural networks for the DM predictions initiated 29 and 36 round-trip transactions, which is much less compared with 74 round-trip transactions for the LAM.

The neural network performances (112.03% and 109.32%) were higher than the performance of the BHS (107.98%) and the performance of the LAM (102.69%) when trading the DM against the US$ (Figure 5). When trading the US$ against the DM (Figure 6), the neural network approach still produced a positive return of 103.69% and 101.18% compared with a loss of 92.51% for the BHS and 95.05% for the LAM.

The market simulation for the JY is plotted in Figures 7 and 8. Although both neural network approaches initiated 71 and 72 round-trip transactions (compared with 79 for the LAM), they still performed much better than the BHS and LAM benchmark. The second best network, an ensemble network consisting of four subnetworks, performed best when trading the JY (109.42%) against the US$, and when trading the US$ (112.88%) against the JY.

The regression models performed poorly returning only 90.47% in the former case and 93.32% in the later case, while the BHS produced a small profit of 3.05% when trading the US$ against the JY and a loss of 3.06% when trading the JY against the US$. The results of our first NNS, an ensemble network with two subnetworks, were similar to our second NNS. The first neural network predictions produced a profit of 7.43% for the US investor and a gain of 10.82% for the Japanese investor.

The LAM initiated many buy and sell orders which increased the trading costs and lowered the return. As a result, LAM trading system did not show a better overall prediction performance when compared with the BHS. Trading of both currencies based on predictions of two selected neural networks showed improved performance in all cases if compared with the BHS and the trading based on the LAM. The NNS had fewer round-trip transactions, especially when trading the DM. The NNS produced the highest returns for all simulations and gave even a positive return even when the BHS produced a loss.

	BHS [%]	LAM [%]	Best NNS [%]	2nd Best NNS [%]
US Investor trades DM	107.98	102.69	**112.03**	109.32
DM Investor trades US$	92.51	95.05	**103.69**	101.18
US Investor trades JY	96.94	90.47	107.43	**109.42**
JY Investor trades US$	103.05	93.32	110.82	**112.88**

Table 2. Profits for the 1995 market simulation of the BHS, the LAM, and two Neural Network Approaches.

6. Conclusions

We have shown that the ensemble neural network design lowers the prediction error variance by averaging forecasts from multiple individual networks. Further, the average performance of these master networks was better than the average performance of the individual networks.

The neural networks utilizing our HOSF achieved higher predictive accuracy for all but the next day's return prediction of the JY. The best results for the next week's predictions of the DM and the JY were achieved using only the 10 HOSF. Besides defining the best indicators the presentation of this data to the neural network is a very important issue to improve the generalization of the neural networks.

Our neural network predictions outperformed trading based on the LAM and the BHS. These networks produced a positive return and performed better than any of our benchmarks. An interesting area of further research is the application of our HOSF preprocessing technique to other financial instruments such as currency futures or option market, or even time series predictions in general.

References

[1] S. Hashem. *Optimal Linear Combinations of Neural Networks*. PhD thesis, Purdue University, Dec. 1993.

[2] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka. Stock market prediction system with modular neural networks. In *Proceedings of the International Joint Conference on Neural Networks,* San Diego, CA, 1990.

[3] M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In *Neural Networks for Speech and Image processing*. Chapman & Hall, 1992.

[4] A. N. Refenes. Constructive learning and its application to currency exchange rate forecasting. In *Neural Network Applications in Investment and Finance Services*. Probus Publishing, 1991.

[5] A. N. Refenes, M. Azema-Barac, and S. A. Karoussos. Currency exchange rate forecasting by error backpropagation. In *Proceedings of the 25th Hawaii International Conference on System Sciences*, Kauai, Hawaii, Jan. 1992.

[6] S. Taylor. *Modelling Financial Time Series*. John Wiley & Sons, New York, NY, 1986.