Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

An entropy-LVQ system for S&P500 downward shifts forecasting

Salim Lahmiri*

Department of Computer Science, University of Quebec at Montreal, Montreal, Canada

ARTICLEINFO

Article history:
Received July 20, 2011
Accepted 7 October 2011
Available online
8 October 2011

Keywords:
Stock market
Neural networks
Loss limit
Forecasting

ABSTRACT

The purpose of this paper is to predict the S&P500 down moves with technical analysis indicators using learning vector quantization (LVQ) neural networks and probabilistic neural networks (PNN). In addition, entropy-based input selection technique is employed to improve the prediction accuracies. The out-of-sample simulations show that LVQ outperforms PNN. In addition, the Entropy-LVQ system achieved higher accuracy in comparison with the literature.

1. Introduction

The stock market is an attractive investment to make high profits. However, forecasting stock market movements has been a major challenge for investors and scholars since it is complex, noisy, and non-stationary. Moreover, stock market follows a random walk and is time varying (Chang & Liu, 2008). However, in recent years soft computing techniques have been widely used to model and predict stock market prices because of their abilities to handle uncertainty and noise in the stock market (Atsalakis & Valavanis, 2009; Yao & Herbert, 2009). In addition, many papers in the literature have employed technical analysis indicators to predict stock market movements because of their simplicity. For instance, this technique analyzes the market history in order to forecast its actual and future behavior. In particular, the price movement is predictable since stock market trends are observable and can be easily detected (Gorgulho et al., 2011).

Indeed, technical indicators were successfully employed to model international stock market movements with artificial neural networks (Chavarnakul & Enke, 2008, Chang et al., 2009; Majhi et al., 2009), fuzzy logic (Chang & Liu, 2008), genetic programming (Chen et al., 2009), support vector machines (Wen et al., 2010), genetic algorithms (Gorgulho et al., 2011), and probabilistic neural

* Corresponding author. Tel: +1-514-649-7771.
E-mail addresses: lahmiri.salim@courrier.uqam.ca (S. Lahmiri)

doi: 10.5267/j.msl.2011.10.006

networks (Schierholt & Dagli, 1996; Prokhorov & Wunsch, 1995, 1998; Mehrara et al., 2010). The purpose of our work is to predict the S&P500 price index downs by 0.1%, 0.2%, 0.3%, and 0.4% using technical indicators as inputs to learning vector quantization (LVQ) neural networks. Indeed, investors fear losses below a certain limit (March & Shapira, 1982). Therefore, we examine the effect of loss limit on the prediction accuracy of the LVQ. The LVQ is a supervised competitive neural network introduced by Kohonen (1995).

It operates by direct definition of the class borders according to the nearest neighbor rule. For instance, the classes that the competitive layer defines depend only on the distance between input vectors. The LVQ was chosen for its good performance for complex classification problems and fast learning (Kohonen, 1995; El-Banna et al., 2008), and for its capability to process large input data with a small computational burden (Kohonen, 1995; Tosaka et al., 2001). These features have also made LVQ popular in engineering classification problems (Tosaka et al., 2001; Dieterle et al., 2003; Lee et al., 2004; Mala et al., 2006; El-Banna et al., 2008).

The LVQ have also been employed in business and financial modeling applications; including prediction of financial distress (Brokett et al., 2006), exchange rates (Stahlbock, 2008), financial failure of banks (Boyacioglu et al., 2009), bankruptcy (Neves & Vieira, 2009), and credit rating (Chen et al., 2011). The performance of the LVQ will be compared to the performance of the probabilistic neural network (PNN) due to its effectiveness in financial classification problems (Schierholt & Dagli, 1996; Prokhorov & Wunsch, 1995, 1998; Bensic et al., 2005; Mehrara et al., 2010). However, to the best of our knowledge the LVQ neural network has not been used for stock market modelling and forecasting. Therefore, the main goal of this paper is to examine its effectiveness in the prediction of S&P500 movements. In addition, unlike previous works, a filter-based technique is employed to identify the relevant inputs; namely technical indicators. For instance, entropy is employed to perform this task.

The remaining of this paper is organized as follows. Section 2 introduces our methodology and data. In Section 3, the results are presented and discussed. Finally, we conclude the paper along with lines of future research in Section 4.

2. Data and methodology

This section describes our data and the artificial neural networks (ANN) used for forecasting the stock market ups and downs; namely the learning vector quantization (LVQ) and the probabilistic neural network (PNN). For the models examined, we let $\ell = 0.1\%,\ 0.2\%,\ 0.3\%, 0.4\%$ to be the limit[1] of loss, $X=\left(x_1, x_2, \ldots, x_i\right)$ represent the technical indicators available at time t which are inputs into the predictive systems (LVQ and PNN) used to classify actual inputs into one of two groups m_1 and m_2 (up and below the predetermined limit or floor ℓ respectively) at some future time $t + 1$. In particular, the output (group) takes the value -1 for a downtrend below the floor (limit) and +1 for an uptrend above the floor (limit). In this paper, we use time series data from the S&P500 price index for the period of October 21st, 2003 to January 29th, 2008.

Fig. 1 exhibits the S&P500 price index, and Table 1 lists the technical indicators used in our study. In order to select relevant indicators to be fed to the classifiers, the indicators are ranked based on their entropy statistics. For instance, the smaller a technical indicator's entropy is, the more discriminatory it is. The entropy is a measure of the uncertainty of a random variable, and it is considered in our study because of its effectiveness in removing both irrelevant and redundant inputs (Sanchez-Marono et al., 2007) and because it overcomes the issue of data sparseness (Zhu et al., 2010). The entropy of the variable (technical indicator) x after observing the output y (group or class) is defined as:

[1] The values of loss limit are chosen arbitrarily.

$$H(x \mid y) = -\sum_j P(y_i) \sum_i P(x_i \mid y_i) \log(P(x_i \mid y_i)), \quad 1$$

where $P(x_i)$ is the prior probabilities for all values of X, and $P(x_i|y_i)$ is the posterior probabilities of X given the values of y the output or the class. We select indicators having an entropy value less than 0.1. Otherwise, the four indicators with the lowest entropy values are selected as inputs. For instance, the values of entropy are sorted in an ascending order to consider those technical indicators with lowest entropy values.

According to the entropy-based ranking shown in Table 2, STOD, X4, DIS, and EMV are selected for -0.10% limit, STOD, DIS, X4, and EMV for -0.20% limit, STOD, DIS, EMV, and BR for -0.30% limit, and STOD, X2, EMV, and BR for -0.4% limit. Finally, all the input data are normalized in the interval [-1,1] to obtain accurate classification results using the classifiers. The normalization is performed as follows:

$$x' = \frac{(2 * x - (Max[x] + Min[x]))}{Max[x] - Min[x]} \quad 2$$

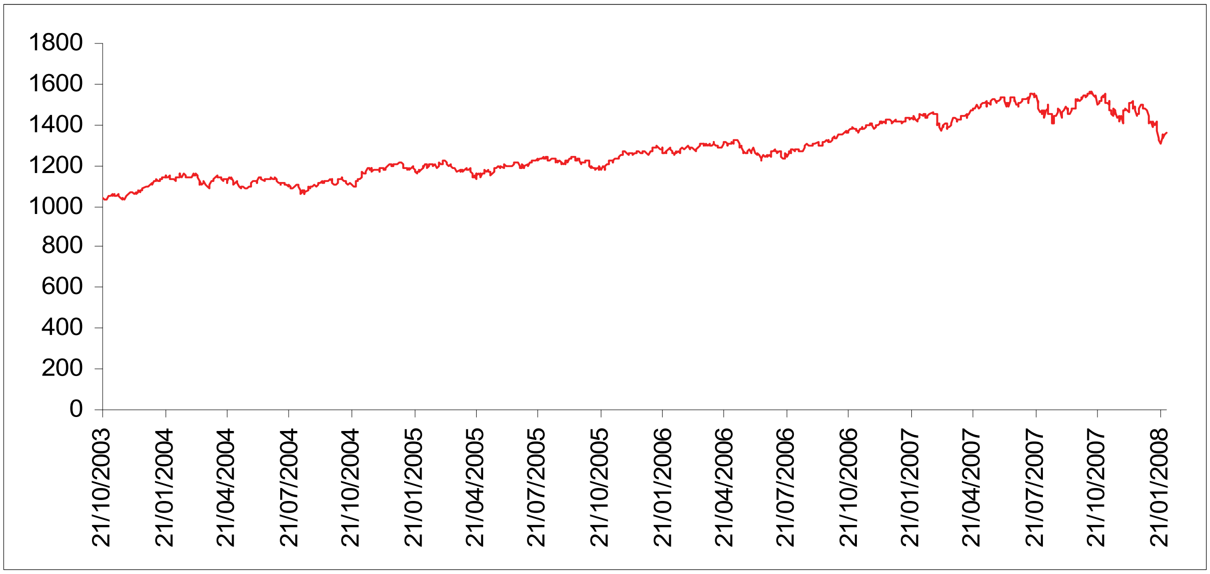

Fig. 1. The S&P500 price index

Table 1
List of technical indicators

Indicators	Description
MA(t)	moving average
X1(t)	[X(t) - lowest price(t)] / [(highest price(t) - lowest price(t)]
X2(t)	[X(t) - MA(t)] / MA(t)
X3(t)	X(t) - MIN[X(t),…,X(t-5)]
X4(t)	X(t) - MAX[X(t),…,X(t-5)]
MPM(t)	[(highest price(t)-lowest price(t))-(highest price(t-2)-lowest price(t-2))]/2
BR(t)	volume(t)/[(highest price(t) - lowest price(t))/2]
EMV(t)	MPM(t)/BR(t)
MOM(t)	closing price (t) - closing price (t-6)
ROC(t)	closing price (t) / closing price (t-6)
DIS(t)	[closing price (t) / MA(t-6)]*100
STOD(t)	[closing price(t) - low price(t,t-6)]/ [high price(t, t-6) – low price(t,t-6)]

Table 2
Entropy statistics

Limit	Feature	Entropy	Feature	Entropy	Feature	Entropy	Feature	Entropy
Rank	10%	10%	20%	20%	30%	30%	40%	40%
1	STOD	0.001	STOD	0.0054	STOD	0.0092	STOD	0.0787
2	X4	0.6658	DIS	0.9266	DIS	1.3347	X2	1.9130
3	DIS	0.8189	X4	1.0892	EMV	1.3644	EMV	1.9182
4	EMV	0.8845	EMV	1.1205	BR	1.4043	BR	2.0734
5	MOM	1.1590	ROC	1.6723	ROC	1.6715	ROC	2.2355
6	MA	1.2204	MA	1.7841	MA	2.1985	MA	3.1988
7	ROC	1.2865	X2	2.5451	X2	2.9411	MOM	3.9008
8	BR	1.9705	X3	2.5451	MOM	2.9411	DIS	3.9008
9	MPM	1.9794	MOM	2.5985	MPM	3.0827	MPM	4.6641
10	X3	1.9794	MPM	2.6860	X3	5.2400	X3	8.0463
11	X2	2.6266	BR	4.3674	X4	6.8902	X4	11.6524
12	X1	6.4411	X1	10.4108	X1	18.2745	X1	15.7890

2.1 The learning vector quantization

The LVQ is a competitive neural classification network introduced by Kohonen (1995). It is composed of three layers. The first layer is an input layer with one neuron per input variable. The neurons in the input layer receive data. The second layer is a hidden layer with neurons that learn patterns and perform the classification. The third layer is an output layer with one node for each class to be recognized and classified. The output neurons are associated with input neurons by prototypes. For instance, the learning is based on the neurons representing prototype vectors and the nearest neighbour approach for classifying data. In particular, the vector of the weights of the connections between all input neurons and one hidden neuron is called a codebook vector which is a prototype representing a labelled class. During the training, the prototypes are updated to optimize the class boundary. In particular, the codebook vector is set close to the presented input vector for correct classifications. Otherwise, it is set in the opposite direction for incorrect classifications. The algorithm of the LVQ is briefly described as follows:

a) A map is initialized with a grid of prototypes and class labels.

b) An instance x_i is selected as input.

c) The Euclidean distance is calculated between x_i and each prototype m_i.

d) The winner m_i is one with the smallest distance is selected as the best matching pattern according to :

$$c = \arg\min_{k} \left\| x_i - m_i \right\| \qquad 3$$

2.2 The probabilistic neural networks

The PNN was proposed by Specht (1990). The PNN employs Bayesian decision-making theory based on an estimate of the probability density of the data. Then, PNN can identify nonlinear decision boundaries that approach the Bayes optimal. The PNN employs an exponential activation function and the basic network topology consists of four layers. The first layer is the inputs layer. In the second layer, the probability density function (PDF) of each group of patterns is directly estimated from the set of training samples. The third layer performs the summation of all PDFs. Finally, the Bayesian decision is made in the fourth layer. The PDF is assumed to follow a Gaussian distribution. Then, the PDF for a feature vector X to be of a certain category A is given by:

$$f_A(X)=\left(1/(2\pi)^{0.5p}\sigma^p\right)m^{-1}\sum_{i=1}^{m}\exp\left(\left(X-X_{Ai}\right)'\left(X-X_{Ai}\right)/2\sigma^2\right) \qquad 4$$

where, p is the number of patterns in X, m is the number of the training patterns of category A, i is the pattern number, and σ is the smoothing factor of the Gaussian curves used to construct the PDF.

Finally, the performance of the neural networks classifiers is measures by the correct classification rate (Hit ratio) defined as:

$$correct\ classification\ \ rate=\frac{correctly\ \ classified\ \ samples}{classified\ \ samples} \qquad 5$$

To conclude this chapter, the overall methodology is presented in Fig. 2.

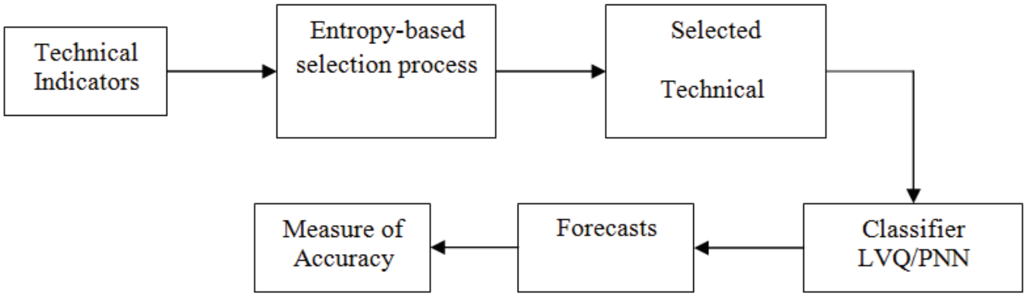

Fig. 2. The overall methodology

3. Results

In this study, 50% of the data is used for learning and 50% is used for testing (out-of-sample prediction). Indeed, the testing data is used to assess the classification/prediction accuracy of the LVQ and PNN after training. The accuracy of each classifier is given in Figure 3. The simulation results show that the LVQ outperform the PNN in the prediction of the S&P500 decreases by less than 0.4%, 0.3% and 0.1%. For instance, the correct detection (classification) rate achieved by the LVQ is 81.4%, 78.6% and 67.91% for 0.4%, 0.3% and 0.1% loss limit respectively.

On the other hand, the correct detection (classification) rate achieved by the PNN is 18.6%, 75.81% and 32.09% for 0.4%, 0.3% and 0.1% loss limit respectively. However, LVQ underperforms PNN in terms of the forecasting of downward movements by less than 0.2%. For instance, LVQ and PNN obtained respectively 71.63% and 77.21%. In sum, the simulation results show that LVQ is in general more accurate than PNN in the prediction of S&P500 down moves.

In this study, the reason that LVQ outperforms the PNN is that PNN is sensitive to noisy data such as financial time series. Indeed, PNN models the probability density function (PDF) of each group of patterns by assuming that they follow a Gaussian distribution. On the other hand, instead of modeling the class densities, LVQ models the discrimination function defined by codebook vectors and the nearest neighbourhood search between the codebook vectors and data. In addition, LVQ employs a gradient descent style to minimize the error of the vector quantization approximation. As a result, LVQ is less sensitive to noise. In sum, the accuracy of the Entropy-LVQ system was 67.91%, 71.63%, 78.60%, and 81.4% depending on the loss limit. The overall results are interesting since some previous studies reported that stock prices are approximately close to the random walk process and; consequently; an accuracy of 56% in the predictions is a satisfying result for stock forecasting (Walczak, 2001; Qian & Rasheed, 2007).

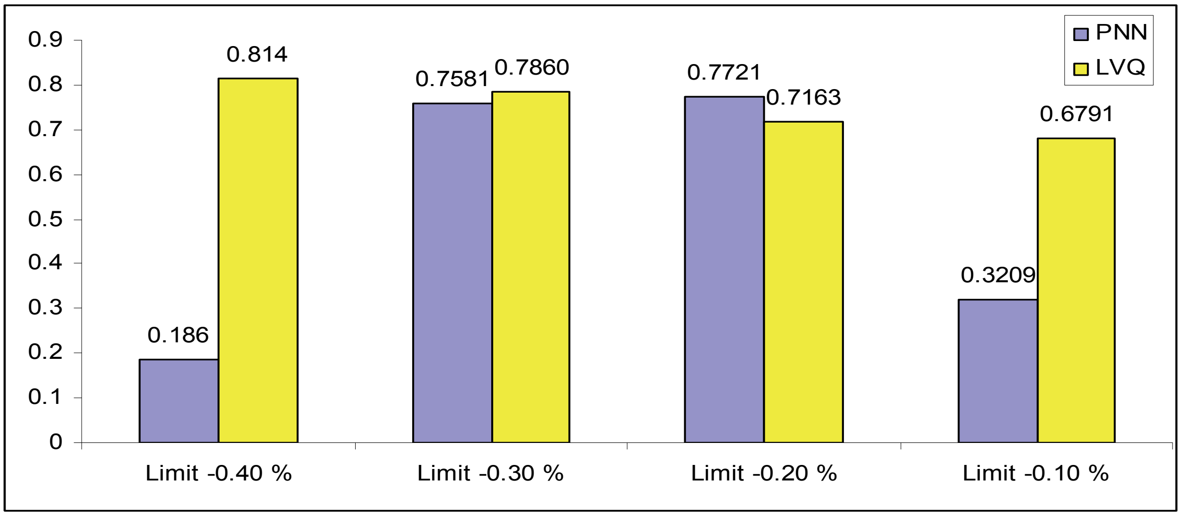

Fig. 3. Simulation results

Although previous works have employed different technical indicators, classifiers and data sets, their results are given for comparison purpose. For instance, Chavarnakul and Enke (2008) obtained 64.68% to predict the S&P500 using generalized regression neural networks, Chang et al., (2009) achieved 65.2% in the forecasting of the Taiwan stock exchange using an intelligent system that hybridizes back propagation neural network and case based reasoning system, and Yao and Herbert (2009) obtained 62.91% to predict the New Zealand stock exchange using rough sets. Therefore, our simulation results are encouraging as mentioned before. Thus, the Entropy-LVQ system is promising.

4. Conclusion

The main purpose of this paper is to model and predict S&P500 moves below predetermined loss limits using technical indicators and learning vector quantization (LVQ) neural network. The entropy is employed to reduce the initial technical indicators set from twelve to four relevant and non-redundant predictive inputs. In addition, probabilistic neural networks (PNN) are also employed for comparison purpose. Finally, the accuracy is measured with the correct classification rate. The obtained simulation results show evidence of the superiority of LVQ over PNN in the prediction of S&P500 downward moves by less than 0.4%, 0.3% and 0.1%. In addition, the Entropy-LVQ system achieved higher accuracy than previous works found in the literature. For future researches, different loss limits, inputs selection techniques, and international stock markets will be considered.

References

Atsalakis, G.S., & Valavanis, K.P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications*, 36, 5932-5941.

Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *Intelligent Systems in Accounting, Finance and Management*, 13, 133-150.

Boyacioglu, M.A., Kara, Y., & Baykan, O.K. (2009). Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: a comparative analysis in the sample of savings deposit insurance fund (SDIF) transferred banks in Turkey. *Expert Systems with Applications*, 36, 3355-3366.

Brockett, P.L., Golden, L.L., Jang, J., & Yang. C. (2006). A comparison of neural networks, statistical methods, and variable choice for life insurers' financial distress prediction. *The Journal of Risk and Insurance*, 2006, 73 (3), 397-419.

Chang, P.C., & Liu, C.H. (2008). A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications*, 34 (1), 135-144.

Chang, P.C., Liu, C.H., Lin, J.L., Fan, C.Y., & Ng, C.S.P. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, 36 (3), 6889-6898.

Chavarnakul, T., & Enke, D. (2008). Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Systems with Applications*, 34 (2), 1004-1017.

Chen, N., Vieira, A., Ribeiro, B., Duarte, J., & Neves, J. (2011). A stable credit rating model based on learning vector quantization. *Intelligent Data Analysis*, 15, 237-250.

Chen, Y., Mabu, S., Shimada, K., & Hirasawa, K. (2009). A genetic network programming with learning approach for enhanced stock trading model. *Expert Systems with Applications*, 36 (10), 12537-12546.

Dieterle, F., Muller-Hagedorn, S., Liebich, H.M., & Gauglitz, G. (2003). Urinary nucleosides as potential tumor markers evaluated by learning vector quantization. *Artificial Intelligence in Medicine*, 28, 265-279.

El-Banna, M., Filev, D., & Chinnam, R.B. (2008). Online qualitative nugget classification by using a linear vector quantization neural network for resistance spot welding. *The International Journal of Advanced Manufacturing*, 36, 237-249.

Gorgulho, A., Neves, R., & Horta, N. (2011). Applying a GA kernel on optimizing technical analysis rules for stock picking and portfolio composition. *Expert Systems with Applications*, 38, 14072-14085.

Kohonen, T. (1995). *Self-organizing Maps*. Springer, Berlin, 1995.

Kosaka, T., Omatu, S., & Fujinaka, T. (2001). Bill Classification by using The LVQ Method. *IEEE International Conference on Systems, Man, and Cybernetics*, 3, 1430-1435.

Lee, W.-L., Hsieh K.-S., & Chen Y.-C. (2004). A study of ultrasonic liver images classification with artificial neural networks based on fractal geometry and multiresolution analysis. *Biomedical Engineering Applications, Basis, and Communications*, 16, 59-67.

Majhi, R., Panda, G., & Sahoo, G. (2009). Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications*, 36 (3 Part 2), 6800-6808.

Mala, K., Sadasivam, V., & Alagappan, S. (2006). Neural network based texture analysis of liver tumor from computed tomography images. *International Journal of Biological and Life Sciences*, 2 (1), 33-40.

March, J.G., & Shapira, Z. (1982). Behavioral decision theory and organizational decision theory. In G. Ungson and D. Braunstein (Eds.), *New Directions in Decision Making*, Boston, Mass.: Kent Publishing Co. Reprinted in M. Zey (1992) *Decision Making: Alternatives to rational choice models*, Newbury Park, CA: Sage Publications.

Mehrara, M., , Moeini A., Ahrari, M., & Ghafari A. (2010). Using technical analysis with neural network for prediction stock price index in Tehran stock exchange. *Middle Eastern Finance and Economics*, Euro Journals Publishing, Inc.

Neves, J.C., & Vieira, A. (2006). Improving bankruptcy prediction with hidden layer learning vector quantization. *European Accounting Review* ,15 (2), 253-271.

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

Saad, E.W., Prokhorov, D.V., & Wunsch, D.C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions of Neural Network*, 9(6), 1456-1470.

Sanchez-Marono, N., Alonso-Betanzos, A., & Tombilla-Sanroman, M. (2007). Filter methods for feature selection – A comparative Study. *Lecture Notes in Computer Science*, 4881, 178-187.

Schierholt K., & Dagli, C.H. (19960. Stock market prediction using different neural network classification architectures. *Proceedings of the IEEE Computational Intelligence for Financial Engineering*, 72 -78.

Specht, D.F. (1990). Probabilistic neural networks, *Neural Networks*, 3, 109-118.

Stahlbock, R. (2008). Neural classification approach for short term forecast of exchange rate movement with point and figure charts. *IEEE International Joint Conference on Neural Networks*, 2841-2848.

Tan, H., Prokhorov, D.V., & Wunsch, D.C. (1995). Conservative thirty calendar day stock prediction using a probabilistic neural network. *In Proceedings of computational intelligence for financial engineering*, 113-117.

Walczak, S. (2001). An empirical analysis of data requirements for financial forecasting with neural networks. *Journal of Management Information Systems*, 17, 203-222.

Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and SVM algorithm. *Expert Systems with Applications*, 37 (2), 1015-1022.

Yao, J., & Herbert, J.P. (2009). Financial time-series analysis with rough sets. *Applied Soft Computing*, 9 (3), 1000-1007.

Zhu, S., Wang D., Yu, K., Li, T., & Gong, Y. (2010). Feature selection for gene expression using model-based entropy. *IEEE Transactions on Computational Biology and Bioinofrmatics*, 7, 25-36.