A DEA-CASCOR MODEL FOR HIGH – FREQUENCY STOCK TRADING: COMPUTATIONAL EXPERIMENTS IN THE U.S. STOCK MARKET

Alexander Vaninsky[1]

[1]Mathematics Department, Hostos Community College of The City University of New York, Bronx, NY, USA

avaninsky@hostos.cuny.edu

ABSTRACT

The paper presents results of computer-assisted portfolio management simulation based on using a DEA-Cascor mathematical model. The model uses the Data Envelopment Analysis (DEA) ratio as a neuron with memory and combines it with Cascade Correlation Neural Network (Cascor) to forecast stock prices. The model is designed for using in high-frequency stock trading. It utilizes ability of DEA to concentrate multi-faceted information in one indicator scaled to the interval [0,1], a DEA efficiency index, and is aimed to compress market information. Cascor combines data of several consecutive periods using its flexible structure and generates a buy - sell strategy. The paper presents results of the simulation of a 50-stock portfolio during a period of 60 consecutive trade days chosen during one of the most problematic period of the U.S. stock market operation. Obtained results allow for optimism regarding its practical use for high-frequency stock trading provided availability of a convenient computer – based support.

KEYWORDS

Computer modelling, Neural Networks, High – frequency stock trading, Data Envelopment Analysis

1. INTRODUCTION

The paper presents a DEA-Cascor mathematical model and results of simulation of computer-assisted high – frequency stock trading based on this model. A portfolio of stocks of shares is formed including the stocks that are expected to be of common interest for variety of institutional and individual, both professional and non-professional, investors. Investment decisions are based on publicly available information only. The suggested model is aimed at simulation of the way of investors' thinking and making investment decisions ahead of the market movement caused by their trades.

High-frequency stock trading is a subject of numerous publications due to rich theory underlying the models and practical interest of institutional and individual investors, [1-10]. In contrary to the most existing approaches to high – frequency trading that operate with hundreds or even thousand stocks, this paper claims that a very moderate number of stocks is sufficient. It uses only a portfolio of 50 stocks selected without any specific guidance. The paper claims also that successful strategy should necessarily be zero – beta that is independent from the market movement. To achieve this goal, the model combines buying and selling stocks short. An additional advantage of such approach is the ability of using financial resources from selling stocks short for financing of purchase other stocks in the portfolio. The paper presents a DEA – Cascor mathematical model that uses the DEA algorithm as a neuron – a signal processing unit, and combines DEA - neurons into a Cascor neural network.

For experiments described in this paper we have chosen one of the most difficult periods in the stock market history: the last quarter of 2008 when the U.S. stock market has actually collapsed.

DOI : 10.5121/ijngn.2010.2301

The trades were conducted daily to simulate high – frequency operations. The results of simulation show that even in this difficult period suggested model achieved 4% gain, while the selected portfolio dropped 13%. It is hypothesized that using this model for really high – frequency trading including hundreds and even thousands of trades per day will allow obtaining even better results. On the other hand, if the high – frequency stock trading is prohibited, as some authorities suggest now to avoid financial crashes, this model may provide a reasonable alternative by switching to the permitted trading frequency.

The model suggested in this paper is different from described in the literature in two ways. First, it uses a DEA efficiency ratio as a neuron with memory. Second, it combines the DEA – neuron with a Cascade Correlation Neural Network (Cascor). By doing so, it utilizes ability of DEA to concentrate multi-faceted market information in one indicator: DEA efficiency index - scaled to the interval [0,1], and adaptive properties of Cascor neural network provided by its flexible structure. Thus, the model uses DEA for compression of the market information, and Cascor for combining and analyzing data of several consecutive periods and making the buy - sell decision.

Day trade has been selected as a widely used high – frequency investment technique providing an opportunity of testing the suggested investment strategies in real time. The paper presents results of simulation of a portfolio management based on use of DEA-Cascor model. The portfolio contains 50 stocks traded on the NYSE. It was investigated for a period of 60 consecutive trade days during one of the most difficult periods of the U.S. stock market history: the last quarter of 2008 when the stock market actually collapsed. Four different strategies were investigated. Obtained results allowed us pointing out a single strategy that has revealed positive and statistically significant outcome. This strategy paves the way to development of a commercial computer system of high – frequency stock trading including variety of stocks traded worldwide.

From financial perspective, obtained results may be interpreted as follows. The U.S. Stock market proved to be two-day efficient, while one-day inefficient. One of the suggested strategies was able to catch up and utilize this one-day market inefficiency. This result corresponds to the evidence presented in financial literature. Thus, publication [11] discusses practicality of the efficient market hypothesis vs. the "noisy market hypothesis" and states as follows. "Efficient-market believers still dominate the field of financial research… <but> there is now a new paradigm for understanding how markets work… It argues that prices can be influenced by speculators and momentum traders, as well as by insiders and institutions that often buy and sell stocks for reasons unrelated to fundamental value, such as for diversification, liquidity and taxes….Prices of securities are subject to temporary shocks,… <or> "noise" that obscures their true values. These temporary shocks may last for days or for years." In terms of information processing the suggested DEA-Cascor model reveals the ability of filtering a useful signal from the market noise and to transform it into investment strategy.

The paper is organized as follows. In section 2, three basic elements of the DEA – Cascor model are outlined: DEA, DEA-Neuron, and Cascor neural network. Section 3 contains data, procedures, and obtained results. The results are further discussed in section 4. Section 5 provides conclusions and perspectives of practical use of the suggested approach.

2. METHODOLOGY

The suggested mathematical model comprises the following elements:

- Data Envelopment Analysis (DEA), a method of vector optimization based on linear programming, [12];

- DEA-Neuron signal processor, providing the opportunity of inclusion of DEA algorithms and efficiency ratio into neural networks, [13];
- Cascade Correlation Neural Network (Cascor), [14], and evidence of its successful application in finance [15].

DEA was developed in [16, 17]. It deals with a group of objects referred to as Decision Making Units (DMU's), that consume inputs X_{kj}, $j = 1,...,s$, to produce outputs Y_{ki}, $i = 1,..,.r$, where $k = 1,\dots,n$ stands for an ordinal number of a DMU in a group. Each DMU seeks improving performance by increasing outputs or decreasing inputs. A measure of success to achieve this goal is called a DMU efficiency index. Analytically, the efficiency index is a ratio of weighted sum of outcomes (outputs) to weighted sum of expenses or investments (inputs):

$$E = \frac{\sum_{i=1}^{r} u_i Y_i}{\sum_{j=1}^{s} v_j X_j}, \; 0 \le E \le 1, \qquad (1)$$

where vector parameters $\boldsymbol{u}=(u_1,u_2,\dots,u_r)$ and $\boldsymbol{v}=(v_1,v_2,\dots,v_s)$ stand for the weights of outputs and inputs of a particular DMU, respectively. DEA is based on an assumption that each DMU assigns its own weights to inputs and outputs in a way that maximizes its own efficiency measure, that is makes it as close as possible to one. In addition, it is required that efficiencies of all DMU's, estimated with these weight coefficients, were less than or equal to one as well. The latter proposition allows DEA to assign values to the weight coefficients $\boldsymbol{u}$ and $\boldsymbol{v}$ objectively, as solutions to series of linear program problems, see [12] for details and discussion.

Neural network (NN) is a system of interconnected signal processors referred to as neurons that receive a number of input signals and transform them into an output signal scaled to the interval [0,1] or [-1,1], see [18] for detail. The transformation is performed in two steps. At the first step, a neuron produces an internal signal equal to a weighted sum of input signals plus a bias. At the second step, the internal signal is converted to an output one by means of an *S*-shaped ("sigmoid") function. Neurons receiving signals from the outside are called input neurons of a neural network. Their output signals serve as input ones to inner (hidden) neurons. The inner neurons transfer their output signals inside the neural network from one layer to another one until they eventually reach the outside the network, thus forming its output. Parameters of a neural network: weight coefficients, biases and steepness parameters of sigmoid functions - are determined during a training process, aimed to match actual output signals their expected values as well as possible.

Cascor invented in [14] is a particular case of a neural network. Its specific feature is the determination of both parameters and the neural network topology together in a single training process. Cascor's training begins with a linear neural network with hidden neurons added in turn one by one. At each stage (epoch of training), input signals of a current additional hidden neuron are formed as Cascor input plus output signals of all previous hidden neurons. A training process stops when either acceptable accuracy is achieved or addition of the next hidden neuron does not increase the accuracy significantly. A training algorithm for the Cascor neural network is given in Figure 1 and is as follows; see [15] for detail:
- Step A. Begin with a linear network, with network inputs fully connected to each input neuron, and with parameters initialized randomly. Train the network to minimize output error.
- Step B. If the error is acceptably small then stop the process. The network is trained.

- Step C. If the error is not small enough, add a hidden neuron. Inputs to this neuron are all network inputs taken together plus the outputs of all previously added hidden neurons. Train the new hidden neuron to achieve maximal covariance of its output signal with the output error.

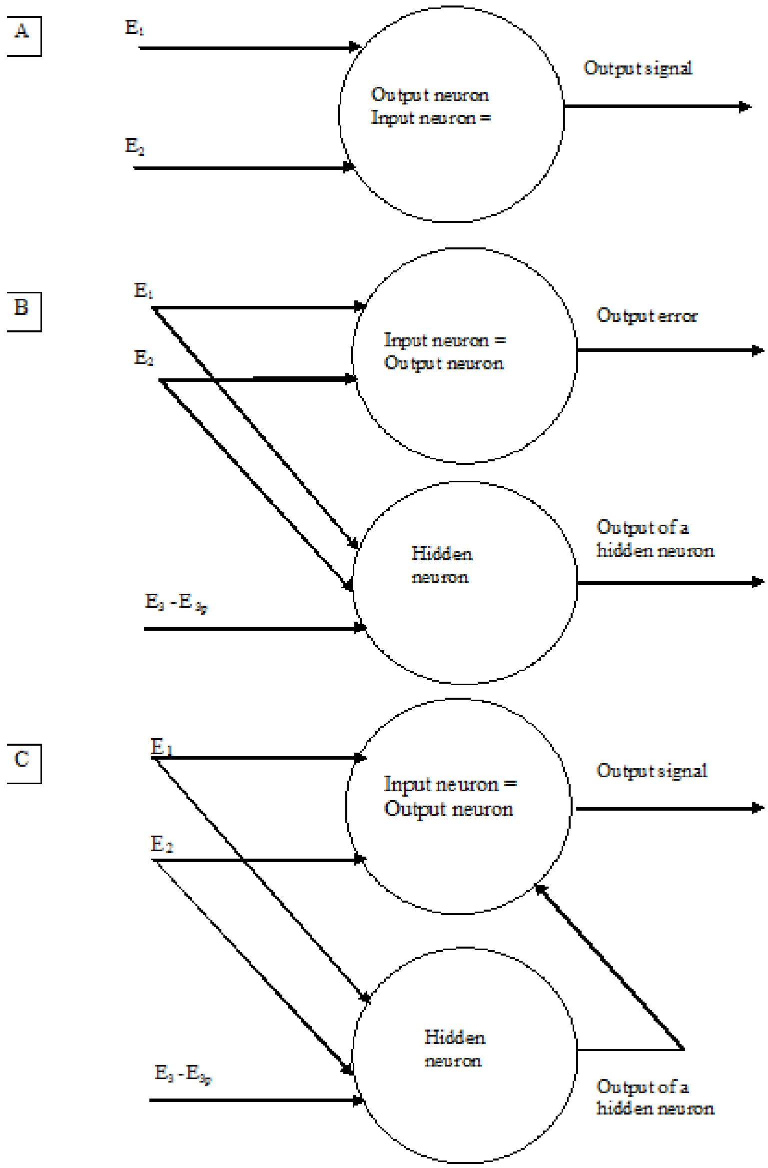

Figure 1. Construction and training of a Cascor model.

- Step A. Initial state. Only one neuron is present serving as both input and output neuron. Inputs E_1 and E_2 are the output signals of DEA-neurons (not shown).

Step B. Intermediate state. Hidden neuron is added with inputs E_1, E_2, and (E_3 - E_{3p}), an additional input. The hidden neuron is trained to achieve maximum covariance of its output with the output error.

- Step C. Final state for the current epoch. Output of the hidden neuron is added to the inputs to the output neuron. Cascor is retrained with the frozen parameters of the recently added hidden neuron.

- Step D. The parameters of the hidden neuron are frozen. Its output is a new input to all output neurons. The rest of the network is trained to achieve minimal output error. Go to Step B if the error is acceptably small, and to Step C, if not.

In contrary to publications [14, 15], it was found in our experiments that outputs of hidden neurons were not significantly correlated with the output error, so that Step C, realized as suggested in these publications, has not led to the improvement of a current neural network. To continue training, we added a new input signal to each additional hidden neuron. Thus, each additional hidden neuron not only enriched the topology of Cascor but added new information as well. Speaking formally, the new inputs may be considered as existing in a pool with zero weights until the correspondent hidden neuron is added.

DEA-algorithm may be considered as a neuron with an internal memory that comprises DEA inputs/outputs of all DMU's in a group. Based on this interpretation, DEA is referred to below as a DEA-neuron with memory, [13]. This neuron transforms DEA inputs/outputs into DEA efficiency index (output signal) by means of DEA optimization algorithm, see figure 2. Training of a DEA-neuron is just "charging" its memory with inputs and outputs of all DMUs in a group and applying the corresponding linear programming procedure. Note that the content of the DEA-neuron memory may be different in training, testing, and applications.

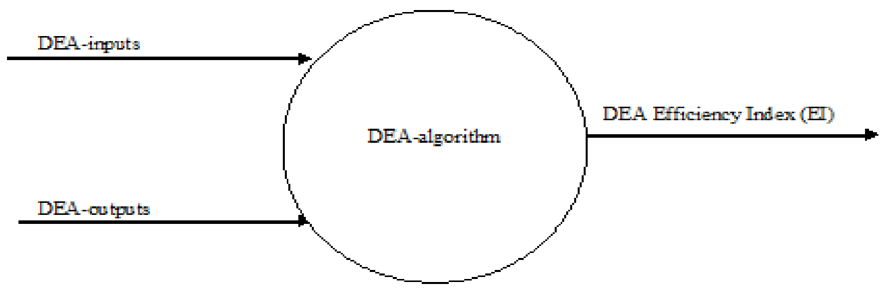

Figure 2. DEA-algorithm as a neuron.
DEA inputs and outputs serve as neuron inputs, DEA efficiency index, is an output signal.

A neural network containing DEA-neurons is referred to below as a DEA neural network, DNN. It consists of two components. One component contains DEA-neurons only, while another one consists of conventional neurons. The latter component is referred to below as Conventional Neural Network (CNN), so that DNN = DEA + CNN, see figure. 3. A DNN is a particular case of a modular neural network proposed in [19]. Inputs to a DNN contain DEA inputs/outputs and optional additional inputs. Output signals of DEA-neurons, DEA efficiency indexes serve as input signals to the CNN.

A process of the DNN training comprises two stages. At the first stage, only DEA-neurons are trained, at the second, only the CNN component. A combination of a DNN and CNN components is referred to below in this paper as a DEA-Cascor neural network. The process of DEA - Cascor training comprises two stages:
- Stage 1. Train DEA-neurons using DEA inputs/outputs and DEA linear programming algorithm.
- Stage 2. Train CNN Cascor as described above with DEA efficiency indexes as inputs and additional inputs.

3. DATA, PROCEDURES, AND RESULTS

For experimental calculations, we formed a virtual portfolio of 50 stocks traded on the U.S. stock market. The stocks were selected based on analytical reports and interviews with

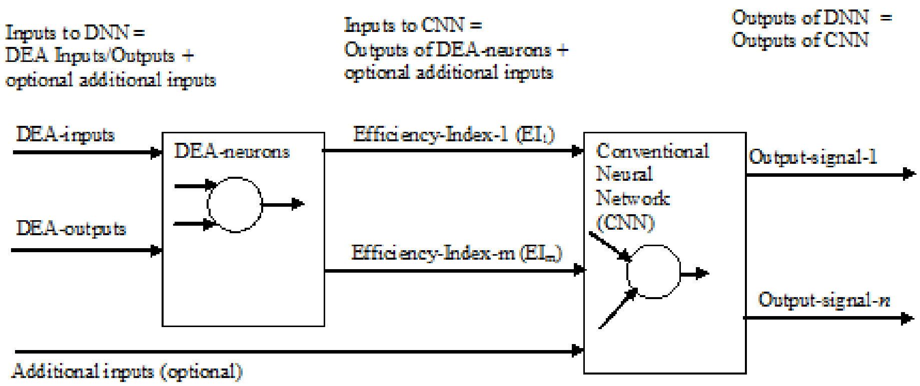

Figure 3. DEA-neural network (DNN).

A DEA-neural network consists of two modules: a DEA-module comprising DEA-neurons, and a CNN-module (Conventional Neuron Network), containing conventional neurons. The DEA-module receives input signals from outside the network (DEA inputs and outputs) and transforms them into intermediate signals, DEA efficiency indexes (EI's). The EI's together with additional inputs serve as inputs to the CNN-module. Output signals of the CNN-module are the outputs of the DNN.

professional and non-professional traders. Some stocks were selected randomly. Our goal was to form a portfolio that may be expected to be of interest for a wide group of investing public and institutions. Information used in the model was collected on the Internet. Data was processed for the period of 60 consecutive trade days. Financial indicators included in the model were as follows: close price, maximal and minimal daily prices, daily trade volume, and daily market capitalization. These indicators were used to form the following DEA inputs/outputs: daily gain or loss with regards to the portfolio index, market turnover, and daily volatility measured as the difference between maximum and minimum stock prices. The first indicator was used as DEA output, two last, as DEA inputs. For comparability, all indicators were expressed in relative units. As standard DEA cannot operate with negative inputs or outputs, daily gains and losses were expressed as percentage of the closing price of a previous trade day: 100% meant no change in price. The whole time period was separated into moving sub-periods of 5 trade days, with each consecutive forecast using the information of the preceding five trade days. As shown in [13], day traders usually do not use longer history.

In a conventional neuron, we used a hyperbolic tangent sigmoid function that produced an output signal as

$$z = tanh(L(\mathbf{E},\mathbf{v})) = tanh(\sum_{i=1}^{n} v_i E_i + b)\,, \qquad (2)$$

where z is an output signal, $\boldsymbol{E}=(E_i,\ i=1,\dots,n)$, DEA efficiency indexes of previous periods, vector $\boldsymbol{v} = (v_1,\dots,v_n;b)$ represents the coefficients of a linear function L. The output signal was automatically scaled to the interval (-1,1), the range of the hyperbolic tangent function.

In the DEA component of the model, we interpreted a stock daily gain or loss in excess of that of the portfolio as an indicator of stock profitability; indicators of turnover and max - min spread were used as the characteristics of liquidity and risk (volatility), correspondingly. We used an output maximization, variable returns to scale DEA model. The reason for the choice of this DEA model was this. Day traders, in contrast to the long-term investors, are risk takers and pursue profit maximization, rather than to minimize risk. This way, the output maximization DEA model has been chosen. The virtual portfolio was formed by inclusion of all stocks in equal dollar amounts. With this approach, using multiples of particular stocks in the virtual DMU was prohibited. This observation has led us to the variable returns to scale DEA model, see [12] for detail.

Inside a 5-day period, we combined daily information in pairs to calculate DEA inputs/outputs as follows: days 1 and 2, 2 and 3, 3 and 4, and 4 and 5. At the first stage, the indicators of the first four days were used to forecast stock prices for the fifth day. Comparison of the forecasted stock prices with actual ones for the fifth day was used as a test of the ability of the model to predict the change in stock prices. At the second stage, stock prices were forecasted beyond the 5-day period and used for investment decision-making for the sixth day. A decision rule was this. If the forecast for the fifth day obtained at stage 1 for a specific stock was correct with respect to the direction of change in stock price up or down, then the forecast for the sixth day for this stock was considered as consistent one. Only consistent forecasts were used to form the investment strategy to buy or to sell a specific stock. Stocks with inconsistent forecasts were excluded for a current period. The model recommendations were this: to buy stocks with prices expected to rise and to sell short stocks with prices expected to decline. It was assumed that all stocks were sold short in equal dollar amounts, and that the proceedings from the selling were used to finance buying the recommended stocks. The model recommendations during the period of investigation were used to test the following four strategies. To buy recommended stocks or to sell short one subset of stocks and to buy another one, with each strategy applied for the next trade day and for the second trade day. There were days when the model recommended to buy or to sell only.

In may be mentioned that "buy only" strategy is preferable for non-professional and some professional investors because it requires less commission payments. At the same time, such strategy strongly depends on the market behaviour, and with the market moving down, results in non-conclusive output. Also, such strategy requires more money for investments because all purchases of stocks should be paid in full. Contrary to that, the "sell short and buy" strategy is able to provide zero beta, that means be independent from the market movement. Also, it is usually less fund demanding, because purchase of stocks may be partially paid off with the proceedings from the short sell. At the same time, the "sell short and buy" strategy is more risky than "buy only" one, and may result in greater losses.

Results of the computational experiment are presented in table 1. In calculations, an assumption was made that the investment capital remains unchanged for the period of research. This means that daily gains were separated from the money aimed at investment, while daily losses were compensated by either part of previous gains or by additional capital invested into the portfolio. These assumptions allowed us to operate with simple interest in the calculations.

The following two parameters were chosen for the evaluation of each strategy: average gain for the period and percentage of correct predictions. In the latter case, a hypothesis H_0 was tested stating that the predictions were made randomly, so that the model was useless. We used the binomial distribution to test this hypothesis. As known, see e.g. [20], the binomial distribution deals the number of successes in a sequence of N independent experiments, each of which yields success with probability p. It has a mean value of $\mu = p \times N$, and a standard deviation $\sigma = \sqrt{Npq}$, where $q = 1 - p$, the probability of failure. The binomial distribution converges to the normal distribution as N approaches infinity. To test a hypothesis H_0 that is the assumption that the results obtained with the suggested model are no more than just random guesses, we assumed the worst case of equal probabilities of success and failure: $p = q = 0.5$ leading to the widest confidence interval. With 60 observations we get $N = 60$, so that the mathematical expectation of the correct results is $\mu = 60\times 0.5 = 30$, and the standard deviation $\sigma = \sqrt{60\times 0.5\times 0.5} = 3.87$. Assuming the confidence level of 99%, we arrive at the critical value of $z = 2.58$. This value corresponds to the right boundary of the confidence interval equal to

Table 1. Summary of the results

Strategy	**Horizon of forecast, trade days**	**Average gain/loss for the period[1], %**	**Percentage of correct predictions [2]**
Buy	1	1.20 (0.78)	66.1
	2	-2.41 (1.24)	71.2
Buy + sell short	1	29.88 (1.61)	71.2
	2	95.99 (11.78)	55.9

Notes.
[1] Standard deviation is given in parentheses.
[2] Mean value is 30, standard deviation, 3.87; the 99% right end point of the confidence interval for 60 trials is 40 correct predictions or 66.7%.

$(30+2.58\times 3.87) = 39.98$ correct predictions. Assuming 40 correct predictions as the right boundary, we get $40/60\times 100 = 66.7\%$ as a critical point. Above this level, we reject the hypothesis of random guesses and consider the model consistent.

4. DISCUSSION OF THE OBTAINED RESULTS

The "buy only" strategy for one - day forward forecast turned out to be insignificant with regards to percentage of correct predictions. Its level has not exceeded the confidence level. The "buy only" strategy for two-day forward forecast resulted in average daily loss, though the percentage of correct predictions did exceed the confidence level. The “sell short and buy” strategy for two-day forecast proved to be statistically insignificant with regards to percentage of correct estimations, though has resulted in the highest average gain. This is namely the case that the model may be assumed to give a good result by chance. The only strategy that turned out to be statistically significant and gave positive average gain for the period was the “sell short and buy” strategy for one - day forward forecast. It resulted in the 29.88% average gain with standard deviation of just 1.61%, and in 71.2% of correct estimations, exceeding the right end point of the confidence level.

From financial perspective, obtained results may be interpreted as follows. One of the four strategies, the “sell short and buy” one, for the one - day forward forecast revealed the ability to catch up the stock market inefficiency, thus providing an opportunity to use it practically as the base of investment strategy. When evaluating practical possibilities provided by this strategy, it may be noted that daily losses may be restricted by continuous computer - based monitoring of stock prices during a trade day and closing long or short positions respectively if some stocks are moving in the undesired directions. For example, a portfolio may be sold back in case of 1 -

2% loss in its market value, thus restricting theoretical loss close to this amount. Contrary to that, in case of gain in the portfolio market value, investors can keep it to the end of a trade day or up to the moment when the decline in the portfolio market value begins. By doing so, about 80 - 90% of theoretically possible gain may be preserved.

5. CONCLUSIONS.

The paper presents a DEA-Cascor mathematical model for high – frequency stock trading and the results of computer - assisted portfolio management simulation. The model combines Data Envelopment Analysis (DEA) with Cascade Correlation Neural Network (Cascor). DEA is used as a neuron with memory. The model utilizes the ability of DEA to concentrate multi-faceted market information in one indicator, a DEA efficiency index. The Cascor neuron network utilizes its adaptive structure and combines efficiencies of several time-intervals together to obtain investment decisions.

The main assumption underlying the DEA - Cascor model is daily market inefficiency. The latter states that computer-based scientific-intense information processing allows obtaining profit from day trade while dealing with publicly available information only. Other assumptions represent mathematical formalization of a hypothesized way of thinking and market behaviour of typical day traders. The model is free from unlawful use of insider information, analysts' fraud, or after - time trade.

The model uses a relatively small portfolio of stocks and allows for wide flexibility in the choice of specific indicators. These properties of the model provide an opportunity of inclusion of investors' or managers' personal preferences into the model. Relatively small number of stocks and indicators used in the model makes it unlikely that many model users will choose the same stocks and the same strategy to buy or to sell short.

In our investigation the forecast of stock prices was based on the information for five previous days. The portfolio contained 50 stocks, and its performance was simulated for a period of 60 consecutive trade days. Several strategies of portfolio management were tested, distinct in the horizons of forecast and availability of short sell. Only one - day forward "sell short and buy" strategy resulted in positive and statistically significant outcomes. This strategy provides, at least theoretically, zero beta and thus allows obtaining gain with market movement in any direction. Obtained outcome may be further improved by continuous computer - based monitoring of stock prices and closing long or short positions appropriately.

Obtained results may serve as a basis for the development of a computer system of high – frequency stock trading operating with a large number of stocks traded worldwide.

REFERENCES

[1] Avellaneda, M. and S. Stoikov, (2008) "High-frequency trading in a limit order book", *Quantitative Finance*, 8 (3), 217-224.

[2] Bradley, R., Brabazon, A. and M. O'Neill, (2009) "Dynamic high frequency trading: A neuro-evolutionary approach", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 5484 LNCS, 233-242.

[3] Coates, J., Gurnell, M. and A. Rustichini, (2009) "Second-to-fourth digit ratio predicts success among high-frequency financial traders", *Proceedings of the National Academy of Sciences of the United States of America.* 106 (2), 623-628.

[4] Coates, J. (2009) "Reply to Millet: Digit ratios and high frequency trading", *Proceedings of the National Academy of Sciences of the United States of America,* 106 (11), E31.

[5] Eisler, Z., Kertész, J., Yook, S. and A. Barabási, (2005) "Multiscaling and non-universality in fluctuations of driven complex systems", *Europhysics Letters*, 69 (4), 664-670.

[6] Gao, W., Meng, X. and S. Zhang, (2008) "Research on stock liquidity based on trade size, order imbalance in shanghai A-share market", *Shanghai Jiaotong Daxue Xuebao/Journal of Shanghai Jiaotong University,* 42 (11), 1776-1779+1783.

[7] Hautsch, N. (2008) "Capturing common components in high-frequency financial time series: A multivariate stochastic multiplicative error model", *Journal of Economic Dynamics and Control,* 32 (12), 3978-4009.

[8] Kablan. A, (2009) "Adaptive neuro fuzzy inference systems for high frequency financial trading and forecasting", *3rd International Conference on Advanced Engineering Computing and Applications in Sciences*, ADVCOMP 2009 , #5359574, 105-110.

[9] Kasprzak, A., Kutner, R., Perelló, J. and J. Masoliver, (2010) "Higher-order phase transitions on financial markets", *European Physical Journal B*, 1 – 15. (In press.)

[10] Menkveld, A. (2008) "Splitting orders in overlapping markets: A study of cross-listed stocks", *Journal of Financial Intermediation*, 17 (2),145-174.

[11] Siegel, J. (2006) "The 'Noisy Market' Hypothesis", *The Wall Street Journal*, June 14, 2006, A14.

[12] Cooper, W.W., Seiford, L.M. and K.Tone, (2006) *Introduction to Data Envelopment Analysis and Its Uses*. Springer Publishers.

[13] Vaninsky, A. (2004) "Combining Data Envelopment Analysis with Neural Networks: Application to Analysis of Stock Prices", *Journal of Information and Optimization Sciences,* 25(3), 589-611.

[14] Fahlman, S.& C. Lebiere, (1990) "The Cascade-Correlation Learning Architecture", In *Advances in Neural Information Processing Systems*, 2, 524-532. San-Mateo, CA: Morgan Kaufmann.

[15] Lacher R., Coats P., Sharma S.C. and L. Fant, (1995) "A Neural Network for Classifying the Financial Health of a Firm", *European Journal of Operational Research*, 85(1), 53-65.

[16] Charnes, A., Cooper, W.W. and E. Rhodes, (1978) "Measuring the Efficiency of Decision Making Units", *European Journal of Operational Research*, 2(6), 429-444.

[17] Banker, R.D., Charnes A. and W.W. Cooper, (1984) "Some Models for Estimating Technical and Scale Inefficiency in Data Envelopment Analysis", *Management Science*, 30(9), 1078 - 1092.

[18] Haykin, S. (1998) *Neural Networks: A Comprehensive Foundation*. NJ: Prentice Hall.

[19] Altman E.I., Marco G. and F. Varetto, (1994) "Corporate Distress Diagnosis: Comparisons Using Linear Discriminant Analysis and Neural Networks*", Journal of Banking and Finance*, 18, 505 - 529.

[20] McClave J. and T. Sincich (2005) *Statistics*. 10th Ed., NJ: Prentice Hall.

Author

Dr. Alexander Vaninsky is an Associate Professor in the Mathematics Department of Hostos Community College of The City University of New York, USA. He received MS degree in Mathematics from Moscow State University, and Ph.D. and D. Sc. (Habilitation) Degrees from Moscow State Academy of Finance, Russia. His research interests include computer modelling of economic and financial systems.

.