Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Neuro-evolutionary approach to stock market prediction

Jacek Mańdziuk and Marcin Jaruszewicz

Abstract— **A neuro-evolutionary method for a short-term stock index prediction is presented. The data is gathered from the German Stock Exchange (the target market) and two other markets (Tokyo Stock Exchange and New York Stock Exchange) together with EUR/USD and USD/JPY exchange rates. Neural networks supported by genetic algorithm (GA) are used as the prediction engine. The GA is used to find suboptimal set of input variables for a one day prediction. Due to high volatility of mutual relations between input variables, a particular choice of input variables found by the GA is valid only for a short period of time and a new set of inputs is generated every 5 days.**

The method of selecting input variables works efficiently. Variables which are no longer useful are exchanged with the new ones. On the other hand some particularly useful variables are consequently utilized by the GA in subsequent independent steps.

Simulation results of the proposed neuro-evolutionary system applied to prediction of the percentage change of closing value of DAX index are very promising and competitive to the ones obtained by the three other heuristical models implemented and tested for comparison.

I. Introduction

The problem of stock index prediction is one of the most popular targets for various prediction methods in the area of finance and economics. In the past many Computational Intelligence techniques have been applied to this task including neural networks [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], fuzzy and hybrid models [11], [12] or genetically developed prediction rules [13], [14], [15], [16], [17]. Despite enormous previous efforts and a wide range of CI, statistical and other methods applied to this problem, efficient stock market prediction remains a difficult task mainly due to complex and varying in time dependencies between factors affecting the price [18].

The approach presented in this paper relies on technical analysis of the stock market. The goal is to predict the change of closing value of German Stock Exchange (GSE) index DAX for the next day. Assuming mutual dependencies between international stock markets, except for the target market the data from two other markets namely American New York Stock Exchange (NYSE) with DJIA index and Tokyo Stock Exchange (TSE) with NIKKEI 225 (NIKKEI) index is also considered, together with exchange rates of EUR/USD and USD/JPY.

The most suitable set of input variables is chosen by the GA and validated by a simplified neural network training and testing procedure. Due to high volatility of dependencies between input variables the set chosen by the GA is only used for 5 days. After this period a new set of variables is chosen for the next 5 days, and so on. The system works without human intervention choosing desired input variables autonomously from a large number of possibilities. Variables are selected in independent steps of experiments with noticeable sense. The existence of repeated patterns of variables within subsequent 5-day windows can be observed.

The results obtained in a series of computer simulation experiments are very promising. The profit achieved by a proposed model exceeds the results of three other heuristic comparative models tested in the paper.

The reminder of this paper can be described as follows. The next section contains a description of the applied methods of data transformation. An analysis of dependencies between variables is presented. In Section III components of the proposed system (neural networks and GA) are described in detail. Extensions to the standard GA implementations intended for this specific research are introduced. Section IV presents definitions of five different stock market trading models. One of the models is based on a proposed neuro-genetic prediction system. Three other ones use simple heuristics. The last model, called prophetic, assumes the knowledge of the future movement in the stock market. This model, by its definition, achieves the highest possible profit under the predefined trading rules. Sections V and VI provide description and results of experiments carried out. The paper ends with conclusions.

II. Source and preprocessed data

The initial pool of source data considered in this work was composed of two exchange rates and the opening, highest, lowest and closing values of the indices of the three above mentioned stock markets in the past days. This data served as the basis for further transformations, e.g. the percentage change of the opening/closing value through the last 5, 10 and 20 days or averages of the opening/closing value through the last 5, 10 and 20 days. The above (and similar) variables are intuitively useful for prediction. They provide basic information of the past values in a compressed way. Especially moving averages allow the algorithm to omit some sudden local changes when looking for prediction rules.

On the above basis the well known oscillators used in technical analysis i.e. MACD, Williams, Two Averages, Impet, RSI, ROC, SO and FSO were calculated. Except for raw values, the buy/sell signals generated by the oscillators were also considered.

Oscillators are widely used by stock market analysts and considered to be the main tool in decision making. Therefore this data is a strong part of the whole range of potential

Jacek Mańdziuk is with the Faculty of Mathematics and Information Science, Warsaw University of Technology, Plac Politechniki 1, 00-661 Warsaw, POLAND; e-mail: mandziuk@mini.pw.edu.pl. Marcin Jaruszewicz is an independent IT Consultant; e-mail: jaruszewicz@data.pl.

1-4244-1380-X/07/$25.00 ©2007 IEEE

input variables. Another useful method of a trend prediction is pattern analysis. Several shapes can be observed on the index value chart. Extraction of them is based on local minima and maxima. A specific algorithm for pattern extraction was developed and used in our experiment, which allowed detection of the following technical analysis patterns: head and shoulders, triangles, bottoms and apexes. Some of them forecast change of prediction, others confirm the actual trend.

The test data covered the period of 100 trading days from April 7, 2004 to August 26, 2004. A chart of DAX closing values during that period is presented in Fig. 1. Sudden changes in different directions are noticeable. General rises and falls of trend are also visible.

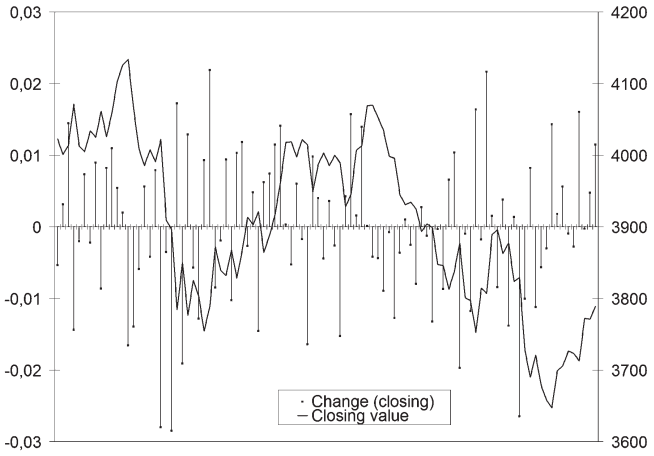

Fig. 1. Closing DAX value and its changes in the period 2004/04/07 to 2004/08/26.

As previously mentioned the situation on a given stock market usually strongly depends on other markets. Therefore the dependencies between variables from different stock markets were measured analytically by calculating the Pearson linear correlation coefficient (eq. (1)):

$$lc_{Pearson} = \frac{\sum_{i=0}^{n-1}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n-1}(x_i - \bar{x})^2 \sum_{i=0}^{n-1}(y_i - \bar{y})^2}} \quad (1)$$

where n is the number of available records. The bigger the value, the stronger the relation between x and y.

A chart presented in Fig. 2 shows relationships between the change of closing value of DAX and other variables from GSE, NYSE and TSE. Some characteristic correlations are visible in GSE and NYSE sets. Similar patterns are also present in TSE data, but with much smaller magnitude. This confirms the relationship between different stock markets, especially European and American ones. The importance of several variables is noticeable. Variables clearly correlated with predicted change of closing value are oscillators from GSE and NYSE and changes of closing values of DJIA and NIKKEI (the latter one with lesser magnitude).

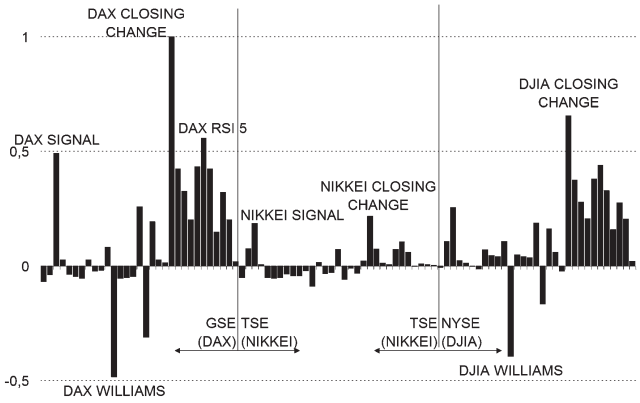

Fig. 2. Correlation between the closing value of DAX and other selected variables.

III. System description

A large number of variables was available for prediction after the data preparation step described in Section II. The final selection of an input variable set was made by the GA. Each chromosome coded a simple neural network architecture and a set of input variables on which a neural network learnt on. A fitness function was correlated with an error made on a test set by a neural network coded by the chromosome. After the GA had finished its work the best fitted chromosome provided a suitable set of variables together with appropriate neural net architecture to be used in the final learning. This network was considered to be a valuable predictor for the current 5-day window.

A. Neural networks

Neural networks are a crucial component of the proposed prediction system. All networks considered in this paper are feed-forward architectures with 1 hidden layer. A standard back propagation algorithm with momentum is used for learning. Changes of network's weights are defined by the following equation:

$$\Delta w_k = \mu(-\frac{\partial Q_k}{\partial w_k}) + \alpha \Delta w_{k-1} \quad (2)$$

where Q is a mean square error of network's output, w a weight vector, k an iteration number.

B. Genetic algorithm

In order to select a small number of relevant variables for the neural network input the GA is used. Each chromosome is coding the list of input variables and the size of network's hidden layer. Therefore each chromosome describes specific network's architecture.

In order to calculate the fitness of a given chromosome three neural networks, each of an architecture coded by that chromosome, with random initial weights, undergo a limited training procedure. The fitness function is correlated with the average error on validation samples attained by these three networks. The smaller the error, the higher the fitness of a chromosome.

Selection of chromosomes for crossover is done by the rank method. Instead of classical crossover with one or more cutting points, the alternative crossover operator [19], which depends on the common part of two parent chromosomes and random selection of the rest of variables is used (see Fig. 3). This kind of crossover promotes variables that are repeated in the population. Parents are exchanged with assigned children only if the latter are better fitted.

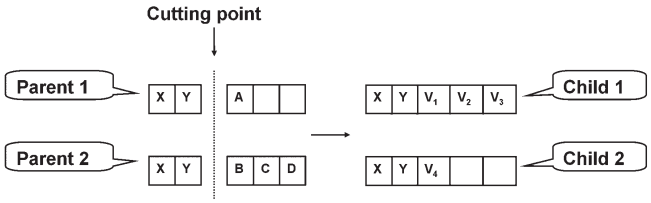

Fig. 3. The scheme of the combined crossover operation with random selection of variables. V_1, V_2, V_3, V_4 are randomly chosen from the remaining pool of variables (i.e. except for X and Y) with additional condition that they are pairwise different within each of the chromosomes, i.e. $V_2 \neq V_3, V_3 \neq V_4$ and $V_2 \neq V_4$.

During mutation variables coded by the chromosome are exchanged with variables chosen randomly from the remaining subset of all available ones. A single mutation can affect only one randomly selected variable. Selection of new variables for crossover and mutation is carried our from the current pool of all available variables composed of 74 elements (see Section V for details).

In every iteration of GA the best chromosome is appointed. The one of the three networks associated with the best chromosome, which achieved the smallest error in the simplified training procedure is saved and subsequently retrained and used for the final prediction.

C. Modifications of the basic genetic algorithm

Despite a new type of crossover operator introduced in previous subsection, several other extensions to the standard GA procedure are implemented. These extensions are described in the reminder of this subsection.

Firstly, the freedom of choosing variables for each chromosome is restricted by *forcing the presence of variable representing the change of the closing DAX value*. This forced variable is not affected by mutation and is obligatory for every chromosome.

Secondly, choosing the new variable for mutation or crossover is done with a specific probability. This selection probability depends on the history of variable's occurrence in the population. There are three types of variables: best chromosome, often selected, common. Variables which were present in any of the former best chromosomes have the type *"best chromosome"* variable. Variables which appear in the population more frequently than the average appearance are called *"often selected"* variables. The rest of variables are *"common"* ones. For each type of variables there is a different probability of selection during mutation or crossover. The probability for best chromosome variables is bigger than for often selected ones. The smallest probability of selection is for common variables. This diversity is applied to prefer variables treated by the algorithm as more important than the others. As the GA starts all the above probabilities are equal. After some predefined number of iterations these probabilities are redefined and equal to 1, 0.75, 0.5 for best chromosome, often selected and common variables, resp. Assigning the type to each variable is redone at the beginning of every iteration.

Thirdly, when the average error of the whole population gets close to its minimum error by less than 3%, the probability of mutation increases. This is done in order to increase the diversity of the population.

Fourthly, sizes of chromosomes in the population may change during the crossover when parents are exchanged with children since one parent is exchanged with that child which inherits the size from the other parent. Thus the mechanism of controlling sizes was implemented. If the average size of chromosomes is close to the current maximum or minimum size in the population the mutation of chromosome size is applied with some probability, considering the expected direction of change. If the average size of chromosomes is close to the maximum the mutation removes from the chromosome one randomly selected variable. Otherwise a randomly chosen variable is added to the chromosome. The probability of the size mutation p_{size} is defined by eq. (3):

$$p_{size} = p_{base} \frac{Max_{fitness} - chromosome_{fitness}}{Max_{fitness} - Min_{fitness}} \qquad (3)$$

with predefined base probability p_{base}. Probability of selecting a variable for adding or removing is defined according to its type (best chromosome, often selected, common).

D. Stopping conditions

The basic stopping condition for the GA and neural network learning is the number of iterations. For neural networks there is also the stopping condition dependant on validation samples. If the error on validation samples during learning rises, the process is interrupted.

There are two additional stopping conditions for the GA. The first one controls the diversity of the population according to the average fitness. If the majority of chromosomes is close to the average fitness the algorithm stops. The second stopping condition applies if the number of iterations without finding a new best chromosome exceeds the predefined value.

IV. Comparative models

In order to examine the effectiveness and usefulness of proposed system of prediction five models of stock market trading were implemented (one of them basing on the proposed system).

All models generate buy/sell signals to make the maximum profit of transactions on index value. There are two general assumptions: 1) there is no handling charge; 2) the index has the price equal to its current value. In the initial state there is a fixed budget for transactions. At the end of the day the signal

for the next day is generated. Transaction is done with the next day opening value. If the signal is "buy" as many indices as possible with the current budget are bought. Remaining money (the amount smaller than the cost of one index) becomes the budget for the next day. If the "sell" signal is provided all possessed indices are sold. The total budget is available for future transactions. The final result is the difference between the total budget after selling all indices at the last day and the initial budget.

Definitions of determination of signals for all implemented models are placed below.

Model 1 - *"buy and hold"* strategy: the first day signal is *"buy"*, the last day one is *"sell"*, and there are no other buy/sell signals generated. As a result the profit equals the change of an index value in a given time period.

Model 2 - signals are generated assuming that the change of an index value of the next day will be the same as the last change.

Model 3 - signals are calculated using the next day prediction of a neural network - (our model).

Model 4 - signals are calculated using the MACD oscillator. Signals are generated at the crossing of the oscillator lines.

Model 5 - signals are generated using the knowledge of the actual next day's index value. This model assumes that the exact knowledge of future index values is available beforehand and as such is not applicable in practice. It defines the upper limit of a profit that can be achieved under given trading conditions.

V. Experiment description

A single experiment is divided into 20 steps. In each step the GA is run in order to select the best neural network architecture together with assigned set of input variables. The assessment of a particular chromosome is carried out based on the result of training on 290 samples and 5 validation records. A neural network selected in the current step is tested on the subsequent 5 records (not used for training). In the next step of an experiment the process is repeated using data samples shifted by 5 records (see Fig. 4). Large part of data is shared between steps since in subsequent steps the time-window is shifted by 5 days forward (test samples from immediately previous step become validation ones and validation samples become the last part of the training days). In effect, in each experiment the consistent set of 100 (20×5) samples was considered as the test data (covering the period from 2004/04/07 to 2004/08/26 - cf. Fig. 1).

In every step the following sequence of activities was performed: **A:** creating initial pool of variables, **B:** finding the best chromosome by the GA, and **C:** performing the final training with neural network architecture and input variables coded by that chromosome.

Parameters of the experiments are described below grouped according to the above activities.

A: The initial pool of variables consists of the following variables: all variables from the target stock market (GSE) for the current day (30 variables), moving averages and changes

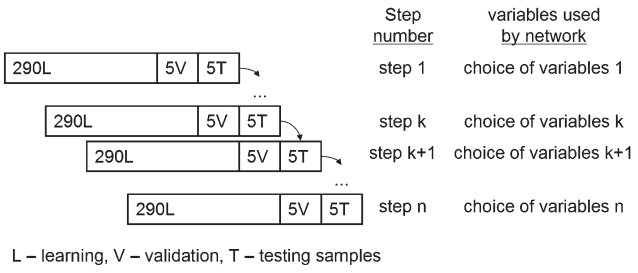

Fig. 4. Schema of learning, validation and testing samples.

of all index values from the target stock market for past days (40 variables), changes of closing index value in two other stock markets and two exchange rates (4 variables).

B: The GA runs on the following parameters:

- population size = 48 chromosomes,
- maximum number of iterations (generations) = 500,
- initial chromosomes sizes between 3 and 10 (plus one forced variable),
- number of parallel neural networks in each chromosome for fitness calculation = 3,
- number of iterations during neural network learning for fitness calculation in GA = 200,
- the probability of mutation if average fitness is close to the minimum fitness = 0.2.

C: The final neural network learning depends on the following parameters:

- number of iterations during learning = 2000,
- the learning coefficient is between 0.1 and 0.8,
- the value of learning coefficient is renewed before each iteration,
- the momentum coefficient = 0.2.

After the final training the prediction of the next day's closing DAX value is carried out.

VI. Results

A. Numerical efficacy of prediction

Results of performed experiments are summarized in Table I. In each experiment the amount of initial money was equal to 100, 000 units.

Model/Experiment	Profit/Loss [%]
Model 1	-5.46
Model 2	-7.54
Model 3	9.31
Model 4	-3.59
Model 5	39.56

TABLE I
PROFITS ATTAINED BY THE CONSIDERED MODELS.

A comparison between Models 1, 2, 4 and Model 3 is presented in Fig. 5. Model 1 (long term buy and hold) and Model 2 (short term no change) got very similar result except for a few days when Model 1 was slightly better than Model 2.

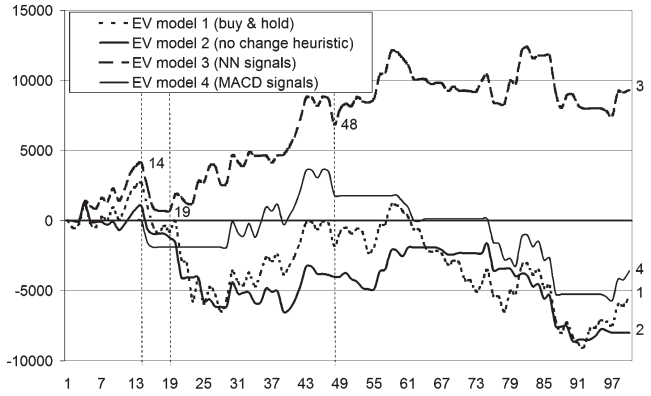

Fig. 5. Comparison of profit of proposed system and comparative models (except the prophetic Model 5).

Model 4 based on MACD oscillator was finally 2% more profitable than Model 1 and exceeded the profit of Model 2 by 4%.

Our model based on nero-genetic prediction (Model 3) outperformed all comparative models (except the prophetic Model 5 not presented in the figure). In the first 4 days all models yielded similar results. In subsequent days Model 3 was more profitable than Models 1, 2 and 4. After the 19th day accurate decisions allowed Model 3 to gradually increase its advantage.

In order to examine the properties of Model 3 in more detail two specific periods were cut out from the test days range: the first one from day 28th to day 59th represents an upward trend and the other one between days 14 and 28 - a downward trend. An examination of the two above periods shows that prediction based on neuro-genetic model works well both in uptrend and downtrend. When index value goes down our model is able to gain some extra money on local changes of the trend. At the same time Model 4 based on MACD signals is comparable, but slightly worse. MACD generated signal "buy" on the first day and signal "sell" on the next day in that period. No more signals were generated, so the model was not active during the downtrend. During the increasing trend our algorithm is still able to utilize local changes of an index rate but thanks to consequently rising index value the final result only slightly outperforms the one of Model 1 (which makes profit by definition in case of upward trend). Model 4 places the worst in uptrend contrary to downtrend situation.

In summary, proposed neuro-evolutionary prediction model outperformed all tested models, except the prophetic one, during the long period of time when different situations on the market were present. Comparable results were also obtained for another test set composed of 100 days from the year 2001 (not presented in the paper). Closer examination of result shows that proposed model is effective in both downward and upward trend cases.

B. "Rationality" of GA-based variable selection

In addition to numerical tests the analysis of variables chosen in subsequent steps was carried out. The hypothesis to be verified is that the occurrence of variables repeats in different steps as their importance grows in the current situation. Thus variables are not randomly chosen but discovered with some logic by the algorithm.

Actually in the 20 steps of the experiment the majority of chosen variables came from the target market (GSE), but aside from these variables there were also variables from the other sources (NYSE, TSE and exchange rates). The frequency of choosing variables from different sources is presented in Table II. These figures should be compared with the overall availability of variables from particular sources. Table III presents a comparison of percentage use of variables from particular sources across all 20 steps and the percentage availability of them in the pool of 74 variables used by the GA. These two distributions are clearly different. Except for the data from GSE, which covers the majority of available variables each of the remaining four sources provides one variable, hence having the same probability of occurrence $(1, 35\%)$. The GA visibly prefers the data from American NYSE (DJIA index closing value) over the one from Japanese TSE (NIKKEI index closing value). Such preference is in line with investors' knowledge, which considers the US stock market as being more indicative for the changes on European stock markets than the Japanese one.

Another conclusion from Table III is that stock related variables are generally more important than exchange rates. This observation is again in line with the knowledge possessed by investors and stock analysts.

Origin of variable	Occurrence in steps
GSE	20 steps
NYSE	10 steps
TSE	6 steps
USD/JPY	5 steps
EUR/USD	3 steps

TABLE II

FREQUENCY OF CHOOSING VARIABLES DURING THE EXPERIMENT.

Origin	Occurrence [%]	Availability for GA [%]
GSE	84.11%	94.60%
NYSE	6.62%	1.35%
TSE	3.97%	1.35%
USD/JPY	3.31%	1.35%
EUR/USD	1.99%	1.35%

TABLE III

PERCENTAGE USE OF VARIABLES.

It is worth to underline that any variable chosen by GA is present in at least a few (more than one) steps. Some of the variables are being chosen (survive) in a few steps in a row. Both the above observations suggest that the selection of input variables carried out by the neuro-genetic system in not

accidental and some variables are visibly preferred over the other ones.

An interesting observation is related to frequent appearance of oscillators among chosen input variables. The oscillators are usually preferred for one or at most two consecutive time steps and then replaced by another ones. This is in accordance with stock market analysts' judgement who use a particular oscillator only for a limited period of time, since changes of stock market's situation require adequate adaptation of the instruments. The most popular oscillators are Impet (chosen in 8 steps), Stochastic and MACD (both selected in 5 steps). The above figures prove the usefulness of the information represented by oscillators in stock market index prediction.

VII. Conclusions

A new hybrid neuro-evolutionary prediction method with application to prediction of the closing value of DAX index is presented in the paper. The system works autonomously, without human intervention or prompting.

Assuming the flow of information between different stock markets and their mutual relations the proposed system uses data from five different sources: three stock markets and two exchange rates.

Due to changing dependencies between variables describing financial markets (indices, oscillators, exchange rates, etc.) and assuming that the usefulness of any variable is limited in time, the proposed approach relies on applying the GA for *frequent* input data selection (among the predefined pool of variables from the above mentioned five sources) and also for choosing the appropriate neural network architecture to be trained based on that selected input data. The standard GA procedure is enhanced by adding a new type of crossover operator and a mechanism that controls the range of chromosomes' sizes.

During the test phase, the proposed method attained 9% of a profit compared to the loss of 5% or more yielded by the other tested methods (excluding the omnipotent, prophetic model). Moreover, the proposed system was able to keep up with the prophetic (locally optimal) model in several subperiods of testing days. The system works well in both upward and downward trend situations in a stock market.

Despite promising numerical results, the system proved its ability to choose the efficient set of input variables and neural net's architecture adequately representing the current situation in the predicted stock market. Various similarities between choices of input variables in consecutive steps were discovered. On the other hand, in each step a number of new variables is selected in place of the "used" and not adequate ones.

References

[1] R. Gencay and M. Qi, "Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 726–734, 2001.

[2] T. Chenoweth and Z. Obradović, "A multi-component nonlinear prediction system for the S&P 500 index," *Neurocomputing*, vol. 10, pp. 275–290, 1996.

[3] J. Yao and C. Tan, "A case study on using neural networks to perform technical forecasting of FOREX," *Neurocomputing*, vol. 34, pp. 79–98, 2000.

[4] M. Jaruszewicz and J. Mańdziuk, "One day prediction of NIKKEI index considering information from other stock markets," *L. Rutkowski et al. (Ed.), ICAISC, Lect. Notes in Art. Int.*, vol. 3070, pp. 1130–1135, 2004.

[5] ——, "Neuro-genetic system for DAX index prediction," *A. Cared and L. Rutkowski and R. Tadeusiewicz and J. Zurada (Ed.), Artificial Intelligence and Soft Computing*, pp. 42–49, 2006.

[6] E. Saad, D. Prokhorov, and D. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1456–1470, 1998.

[7] R. Lee, "iJADE Stock Advisor: An intelligent agent based stock prediction system using hybrid RBF recurrent network," *IEEE Transactions on Systems*, vol. 34, no. 3, pp. 421–428, 2004.

[8] P. Tino, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865–874, 2001.

[9] L. Cao and F. Tay, "Support Vector Machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[10] J. V. Hansen and R. D. Nelson, "Data mining of time series using stacked generalizers," *Neurocomputing*, vol. 43, pp. 173–184, 2002.

[11] V. Kodogiannis and A. Lolis, "Forecasting financial time series using neural network and fuzzy system-based techniques," *Neural Computing & Applications*, vol. 11, pp. 90–102, 2002.

[12] P. Lajbcygier, "Improving option pricing with the product constrained hybrid neural network," *IEEE Transactions on Neural Networks*, vol. 15, no. 2, pp. 465–476, 2004.

[13] M. Dempster, T. Payne, Y. Romahi, and G. Thompson, "Computational learning techniques for intraday FX trading using popular technical indicators," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 744–754, 2001.

[14] B. LeBaron, "Empirical regularities from interacting long- and short-memory investors in an agent-based stock market," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 5, pp. 442–455, 2001.

[15] K. Kim and W. Lee, "Stock market prediction using artificial neural networks with optimal feature transformation," *Neural Computing & Applications*, vol. 13, pp. 255–260, 2004.

[16] F.-L. Chung, T.-C. Fu, V. Ng, and R. Luk, "An evolutionary approach to pattern-based time series segmentation," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 471–489, 2004.

[17] S. Hayward, "The role of heterogeneous agents past and forward time horizons in formulating computational models," *Computational Economics*, vol. 25, pp. 25–40, 2005.

[18] S. Thawornwong and D. Enke, "The adaptive selection of financial and economic variables for use with artificial neural networks," *Neurocomputing*, vol. 56, pp. 205–232, 2004.

[19] J. Bródka, "Private communication," 2006.