The Multi-Agent System for Prediction of Financial Time Series

Šarūnas Raudys[1] and Indre Zliobaite[2]

[1] Institute of Mathematics and Informatics
Dept. of Informatics, MIF, Vilnius University,
[2] Naugarduko st. 24, Vilnius 03225, Lithuania
raudys@ktl.mii.lt, indre.zliobaite@mif.vu.lt

Abstract. To take into account different character of distinct segments of non-stationary financial time series the multi-agent system based forecasting algorithm is suggested. The primary goal of present paper is to introduce methodological findings that could help to reduce one step ahead forecasting error. In contrast to previous investigation [6], instead of single prediction rule we use a system of several adaptive forecasting agents. The agents evolve, compete among themselves. Final decision is made by a collective of the most successive agents and present time moment. New multi-agent forecasting system allows utilizing shorter training sequences and results in more accurate forecasts than employing single prediction algorithm.

Keywords: Classification, Forecasting, Sliding window, Training, Dimensionality.

1 Introduction

The Problem. Today much attention is drawn to analyzing processes, which are changing rapidly over the time. The changes could be of different nature, they can be *temporary* or *permanent*. Financial time series are examples of a dynamic system. Financial time series here are defined as the price of an asset over time, including stocks, commodities, bonds, price indices or exchange rates. In the financial forecasting task, the algorithms ought to include means to reflect the changes, to be able to adapt to sudden situational changes [1]. Financial time series might be affected by utilization of *diverse forecasting algorithms* employed in real market [2]. Thus, the foremost important properties of financial forecasting algorithm should have, are: quick *adaptability* to changing environments and utilization of *short* historical information. The primary goal of present paper is to introduce methodological findings which could help to reduce forecasting error for one step ahead financial time series forecasting task.

Research in the field. There are a lot of research papers in the financial time series forecasting field; however, the disclosure is limited due to profit opportunities involved. The repeatability of such experiments is limited, first of all, due to the same profit opportunity reason, but moreover, due to constant environmental changes. Financial time series forecasting algorithms should be readapting

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 653–662, 2006.

to environmental (data) changes dynamically, otherwise they are not repeatable and they cannot be valuable. Many artificial neural networks (ANN) based approaches for solving financial time series forecasting tasks have been proposed. We would distinguish two main categories: *experimental design* orientated, which mostly assume neural networks as “black boxes” producing outputs from given inputs, and *methodological approaches*, analyzing what is inside the ANN, which usually do not have “plug-and-play” design to be used in real financial markets directly. We attribute our work to the second category.

Many existing financial time series forecasting algorithms employing ANN differ in formation of forecasting target (index), and in measuring the testing error. In this literature review we skip the experimental design oriented financial time series forecasting algorithms (“black box” approaches), and focus on the papers dealing with methodological issues, which to our mind have much more value added in the long term perspective.

One of the pioneering reviews was presented by Moody [3], where poor signal to noise ratio, non-stationarity and non-linearity of financial time series were addressed. The author pruned unnecessary MLP nodes, regularized MLP and used sliding window. He obtained minimum testing error at ~ 10 years training history. He used 1-12 months forecasting horizon (a number of steps ahead). Moody used rates of return as prediction index and market data ranging from 1950 to 1980 as inputs. It can be argued that market characteristics were different thirty years ago, at least due to an absence of powerful forecasting tools.

The reader is referred to an excellent review of foreign exchange rates forecasting by Huang *et al.* [4], where a number of forecasting methods using ANN were compared, to a large extent applicable to various financial time series. The authors aligned 11 papers comparing ANN forecasting with traditional financial time series forecasting methods, such as ARMA, ARIMA, GARCH and random walk. MLP was used in 10 out of the 11 papers. Although these papers highly differ in performance measures including several types of absolute and percentage errors as well as average relative variance, in 7 out of 11 papers, better results were achieved using the ANNs as compared to traditional statistical methods. In the remaining 4 papers mixed results were obtained.

Lendasse *et al.* [5] used radial basis function ANN for forecasting. They used stock market index data and obtained 42,8% testing error with 2100 days testing set. They reformulated the task into classification of increases and decreases of financial variables over time and used 500 days sliding window for training. The proposed approach to solve non stationarity problem was to disregard $t+1$ period forecasting if wrong predictions exceeded right in the last 5 days counting back from day t, obtaining 34,7% testing error in 1583 testing days.

Our previous research. Beforehand, we developed a method to increase the accuracy in situations when environments are changing permanently [6]. Training of forecasting rule was based on short data sequences what results to more accurate predictions as using lengthy historical data. Optimal training set size was determined both theoretically and experimentally. To reduce generalization error, the data dimensionality was reduced by mapping input vectors into low dimensional space using MLP. Forecasting was performed by SLP based classifier. While training, the perceptron it was initialized with weight vector obtained after training with previous portion of the data sequence. To save useful information accumulated in financial time series data, the early stopping procedure was utilized. *For reader convenience let us call the former algorithm presented above* ***MSSE*** *(M-mapping, S-short history, S-save weights, E-early stopping).*

Thus, there is a common understanding in the field of financial time series forecasting, that in changing environments, it is necessary to reduce complexity, find ways to make use of the shortest possible historical data and the algorithm should have means to adapt to complex and continuously changing environments. Inspired by the nature, we claim that successive adaptation is possible employing a number of diverse competing agents. An entity of the agents allows creating distinct forecasting "styles" to learn rapidly different environmental changes.

In present paper, we expand our method proposed in [6] from the single forecast to multi-agent system (MAS). While the MSSE contributed towards solving a problem of *permanently* changing environment, the integration of MSSE algorithm to the MAS contributes to solving *temporary* changing environment problems. Forecasting algorithm ought to adapt to the changes quickly and start to predict accurately. Adaptation speed depends on accuracy of determination of initial weight vector. Since a character of multidimensional time series is changing often, for a single forecasting rule it is difficult to adapt rapidly. We suggest using several distinct forecasting algorithms (SLPs) capable to adapt to diverse changes. In our paper, at first we analyze a behavior of such system while dealing with artificial time series, then test our algorithm with real data. We developed a simplified prototype of such system where only a part of our ideas were realized.

2 Proposed Method

We formulate financial time series forecasting task as pattern classification problem [6], defining the classes as *increase*, *decrease* and *insignificant change* of a chosen financial variable at time $t + 1$ as compared to time t. We use forecasting/prediction terminology when referring to the problem itself and classification terminology, when referring to the proposed methodology. Forecasting procedure is based on *sliding window* approach, which is often used in financial time series forecasting domain: the system is trained on a particular segment of the multivariate time series historical data and the performance of the trained algorithm is tested on subsequent segment. After recording testing results, the testing set becomes a part of training set, the oldest training data are left over. Then we consider a new testing segment, etc.

The first stage of MSSE algorithm [6] was data preparation and dimensionality reduction, where data was mapped into low dimensional space by wrapper approach based neuro-linear dimensionality reduction [7]. *The second stage* was derivation of polynomial features and single SLP training - testing using the sliding window approach. Here we extend the final stage of the algorithm. In order to make forecasting system more robust to changes, we use MAS for final forecast. In the new approach, r distinct adaptive forecasting agents are represented by r diverse SLPs. Each time we select r_{best} most successful agents for final decision making, see Stage II b in Table 1. For reader's convenience now on we call MSSE algorithm MSSE-1, having in mind single agent forecasting. The new algorithm based on MAS will be called ***MAFS*** (Multi Agent Forecasting System) or MAFS-r, when specific number of agents r will be used.

Table 1. The steps of MAFS algorithm

Stage I	Step 1 Data preparation (training TR and testing TE data blocks)
	Step 2 Neuro-linear dimensionality reduction (TR, TE $\rightarrow$ TR_3,TE_3)
Stage II a	Step 3 Derivation of polynomial features (TR_3, TE_3 $\rightarrow$ TR_9, TE_9)
	Step 4 Generating r agents with differentiated initial weights, using TR_9
Stage II b	Step 5 Each agent retraining; testing on TE_9 using "sliding window"
	Step 6 r_{best} best* agents (SLPs) vote to make final forecast
	Step 7 Retraining weights are saved only for r_w best* agents

* "best" in terms of factual testing error estimated in previous testing "window"

Stage I: Data preparation and dimensionality reduction. This stage mainly deals with experimental design.

Step 1. After dividing the data into training (TR) and testing (TE) slots, we leave testing data aside and use training data slot for determination of model parameters. We divide training days into three non-overlapping classes: C_1 – index "ups" (25% of highest increases of forecasting index in two consecutive days), C_2 – index "downs" (25% of highest decreases) and C_{middle} – the "middle" class (the remaining 50% of training days) which we eliminate from the experiment to exclude insignificant changes and account for transaction costs.

Input data vectors X_t are formed using *four* consecutive days price history (from day $t-3$ to day t) of each of the *five* considered financial variables. In this way, we generate 20-dimensional vectors. The length of input has been determined experimentally and is a subject of user's choice. We apply the C_1 and C_2 input vectors transformation towards zero mean and unit variance.

Step 2. We apply the MLP classifier for dimensionality reduction from 20 down to 3 new features [7]. This simple feature extraction (FE) method performs linear FE with nonlinear performance criterion. The l *new features*: $Z = (z_1, z_2, \ldots, z_l)$, are linearly weighted sums, $z_s = \sum_{j=1}^{p} w_{sj} x_j$, $(s = 1, 2, \ldots, l)$ of p inputs $(l < p)$ calculated in l hidden neurons. The new extracted feature space depends on minimization criterion used in training, i.e. on complexity of decision boundary. In spite of simplicity, the neuro-linear FE method is very powerful tool, using information contained in all input features.

The parameters determined in Steps 1 and 2 using TE were applied to TR data set. For classification into 3 pattern classes we have chosen thresholds: $X_t \in C_1$, if $Y_t > Y_{\max}$, $X_t \in C_2$, if $Y_t < Y_{\min}$, where Y_t is the value of forecasted index at day t and $Y_{\max} = \max Y | Y \in C_1$, TR; $Y_{\min} = \min Y | Y \in C_2$, TR.

We did not recalculate Stage I after each training window, since the gain was negligible.

Stage II a: The MAFS algorithm architecture

Step 3. Financial data is complex. Thus, non linear decision boundary is needed. The MLP based classifier can be easily trapped into bad local minima. Therefore, we have chosen SLP classifier to work in the 2^{nd} order polynomial feature space derived from TR_3 and TE_3.

$Z = (z_1, z_2, z_3) \rightarrow (z_1, z_2, z_3, (z_1)^2, (z_2)^2, (z_3)^2, z_1z_2, z_1z_3, z_2z_3) = Q$, here $Q = (q_1, q_2, \ldots, q_9)$ leads to 9-dimensional input feature space (TR_9 and TE_9) [6].

Step 4. We generate r agents (SLPs) with identical architecture. In order to obtain diversity among the predictors, the perceptrons differ in their initial weight vectors $\mathbf{w}_{\mathrm{start}(i)}$, $j = 1, 2, \ldots, r$. At first, we divided training set TR_9 into $(r-5)/2$ non-intersecting segments. We utilized 60% of each segment for training and remaining 40% of vectors we used for validation. We had run trainings on each of those segments in order to determine starting weights of $(r-5)/2$ agents. The starting weights of other $(r-5)/2$ agents were determined by adding uniform distribution random components ξ, $\xi \in (-0,5; 0,5)$, to $(r-5)/2$ previously obtained weights. Remaining four agents got absolutely random weights $5{*}\xi$. The very last agent got zero initial weights (see Table 2 for the agent selection).

Table 2. Initial agent selection for final decision making

Method for the agent selection	Number of agents
1 Initial weights obtained via training agents on the training block	$(r-5)/2$
2 Initial weights obtained via training agents on the training block + uniformly distributed random variable $\zeta \in (-0,5; 0,5)$	$(r-5)/2$
3 Uniformly distributed random weights $\mathrm{w}_{\mathrm{start}} \in (-2,5; 2,5)$	4 agents
4 Zero component initial weights	1 agent

We believe that for each segment of the data or short period of time we have different "styles" of financial data fluctuations, since non-stationarity assumption holds. Thus, *for each fluctuation style we do need different classification (forecasting) algorithms.* We train and test all r agents with the same portion of data, however, start training from the agent's individual weight vector.

Stage II b: Forecasting as the Multi-agent system. This stage gets into feedback loop, which steps through all the testing set TE_9.

Step 5. We retrain the system on subset $\mathrm{TR}'_9 = (Q_{t-k}, \ldots, Q_t)$ and test the performance on subset $\mathrm{TE}'_9 = (Q_{t+1}, \ldots, Q_{t+m})$, where k stands for training window size and m stands for system moving step, t stands for the time (today).

Step 6. The final forecasting decision for subset TE'_9 is made by r_{best} agents' majority voting procedure. The r_{best} agents are selected according to forecasting performance on subset $\mathrm{TE}_9^{-1} = (Q_{t-m+1}, \ldots, Q_t)$.

Step 7. If training of the jth agent was successful (it falls into a pool of the r_{w} best agents characterized by the smallest testing errors on TE'_9), its final weight vector was used as a starting weight in subsequent data segment training.

It is known that if the weights of the well trained non-linear SLP based classifier become large, they start slowing down further training process [8]. To overcome this complicatedness, and to save possibly useful information contained in starting 10-dimensional weight vector $\mathbf{w}_{\text{start}}$, each new training session was started from a *scaled* weight vector $\kappa \times w_{end}$, where $\mathbf{w}_{\text{end}}$ stands for weight vector obtained from previous training. Optimal value of parameter κ was determined from the minimum of the cost function calculated from the testing subset TR'_9 after recording current test results. In contrast, if previous training of the jth agent was unsuccessful, the initial weight vector for this agent training remains unchanged. In present version of the forecasting algorithm, a number of training epochs, n_{train}, was fixed *a priori.* It is a subject of future investigations.

3 Experiment Design

We used artificial data set for setting architecture, global parameters of MAFS system and investigation of its primary characteristics. Real world data was used to check usefulness of the MAFS and compare it with previous research.

Artificial data. Long-lasting non-stationary multidimensional data was used in the first part of the experiments. The data was generated by excitable media model composed of 250×250 cells in a hexagonal grid as described in [9]. Each cell mimics single financial market participant and affects six neighboring cells. Sums of the excitations in five non-overlapping areas served as five features that mimic random fluctuations of five financial variables. Random changes in strength and directions of wave propagation, refractory period, threshold value (when the cell is excited), were manually introduced in order to have *severe unexpected environmental changes.* The waves of cell excitation were assumed to represent information reaching some market participants and not known to the other ones. Such financial market model was not used by other researchers yet.

Artificial data used in the experiment consist of 6884 "days", 1884 from which were used for training and the rest of them – 5000 "days" for testing. A couple of excerpts from generated artificial time series are shown in Fig. 1a. In addition, in Fig. 2b real time series used in the experiment are pictured for comparison.

Real market data. As in [6], the real data was taken form commodity exchanges during period 1993-06-08 – 2005-10-27. It consisted of 3211 observation days. The data array was comprised of 5 time series: x_1 – Crude Oil-WTI Spot Cushing U\$/BBL, x_2 – Cocoa-ICCO Daily Price US\$/MT, x_3 – Corn No.2 Yellow Cents/Bushel, x_4 – Gasoline, Unld. Reg. Non-Oxy, NY, C/Gal and x_5 – LME-Copper, Grade A Cash U\$/MT, the price index of latter security was predicted. We used 711 days for training and the remaining 2500 days for testing.

Experimental parameters. After several attempts we have chosen: $r = 41$, $r_w = 13$ and $r_{\text{best}} = 9$. The testing window size was chosen $|\text{TE}'_9| = m = 20$ days and TR'_9 was a variable in the range $|\text{TR}'_9| = k = 20...800$ days to come up to the optimal training window size with the smallest testing error. For *each* testing attempt with different training window length we repeatedly used

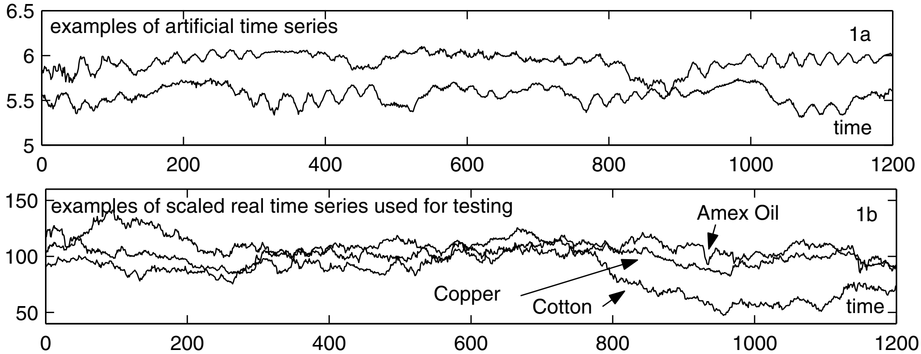

Fig. 1. (a) artificial time series; (b) real market time series (scaled to fit the grid)

$|\mathrm{TE}_9| = 5000$ testing "days" in the experiments with artificial data and $|\mathrm{TE}_9| = 2500$ testing days in the experiments with real data.

Index. We use rather simple measure of returns to construct less typical index, for labeling training and testing data. Our index is as follows:

$$\text{Index:} \quad Y_t = \log\big((B_{t+1} + B_t)/(B_t + B_{t-1})\big), \tag{1}$$

where B_t is the price vector of given security in time t.

This way we aim to forecast if the return tomorrow will be significantly higher (label 1) or significantly lower (label 0) than it was today. We know today's and historical prices. Therefore, we can calculate simplified returns. *It should be noted that the model is designed for testing of our methodology. The index is not a core aspect in this research. It might be changed by another one at user's convenience.*

4 Results

We compare the testing error results obtained using MAFS-41 with the results of forecast MSSE-1, proposed in previous paper. A number of differently generated artificial time series were used to develop new algorithm. In Fig. 2 we see forecasting examples where the previous algorithm and the new one are compared.

While considering artificial time series, we generated the series with discreet environmental changes of needed length. Our experiments show that the more non-stationary time series are, the more clear training window optimum we get and we have clearly convex pattern of experimental graph. Fig. 2a shows that MAFS is able to constantly outperform MSSE-1 during the *whole* training window spectrum. It is very positive and promising result. For the same set of model parameters, utilization of MLP gave almost constant 27% testing error rate, thus giving clear loose in accuracy as compared to both of our suggested strategies.

We validated the experiment with the real market data (Fig. 2b and Table 3). We observed notable gain in absolute testing error as well as we gained 170 days

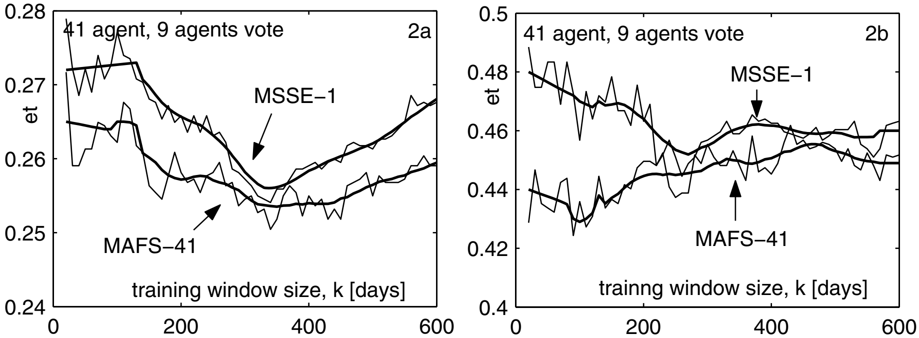

Fig. 2. Classification error e_t as a function of training window size k (in days): (a) artificial data, (b) real data. Bold lines – smoothed results, thin lines – original results.

in training window size. We achieved the gain *for all training window lengths, k.* (Fig. 2b). This aspect of the MAFS is essential as the optimal training window can hardly be determined in advance in applications for real time systems. In terms of testing error, we can argue that new MAS based forecasting algorithm is capable reacting to abrupt market changes more successfully.

Having the same time horizon and other experimental parameters we tested with other commodity prices, the results of three of which are provided in Table 3. We achieved gains in testing error and mixed results in terms of training history, gaining in accuracy in several experiments, but losing in the other ones.

5 Implementation and Limitations

A contribution of present paper is the suggestion to utilize adaptive multi-agent system for collective decision making in the final stage of MSSE algorithm. Such approach enables accumulation of prior information about different "styles" of the time series and makes adaptation to changing environments easier. Employing the MAFS for forecasting of financial variables allows creating diverse adaptation and forecasting "styles" that are good for distinct segments of time series. Use of MAFS improves the testing results of MSSE-1 and helps to reduce the length of training history. Moreover, while employing such forecasting system in practice, one does not know in advance, which agent is the best. Therefore, cooperation (voting) of the most successful agents also gives additional gain.

We suggest using several distinct forecasting algorithms (SLPs) capable to adapt to diverse changes rapidly. The algorithms should differ both in initial weight vectors and learning parameters (learning speed, the length of multidimensional segment of time series used for training, etc.). We set these parameters to be dissimilar to distinct predictors, organize them into adaptive MAFS, capable to change its structure and the global parameters interactively.

We let our MAFS work in real changing environments and then select these initial weight that have been successful at least in one time segment. Then we

Table 3. Results achieved using different real commodities time series starting with different initial conditions

	Testing error, e_t	Optimal "window", k	Testing error, e_t	Optimal "window", k
Index commodity	**MSSE-1 (1 agent)**		**MAFS-41 (MAS)**	
Artificially generated data	25,60%	340	25,35%	350
Copper	45,19%	270	40,29%	100
Amex Oil price index	39,73%	300	37,23%	190
Cotton	42,12%	170	41,57%	260

perform cluster analysis of a set of these initial weight vectors and use cluster centers as the initial weights for the agents of MAFS in subsequent work.

Our experimental design might be questioned by a practitioner as we repeated the experiment several times and used averages from test data sequences to determine mean generalization error, e_t. From a point of view of theoretician, we behaved correctly, since we compared different algorithms in identical conditions and showed principal way how to predict prices in unavoidably changing environments, making use of: (1) short training sequences, (2) dimensionality reduction, (3) early stopping to save previous information (4) faster adaptation to sudden severe changes by utilizing MAFS.

The key goal of the research was to introduce the pool of diverse prediction agents which could help to reduce forecasting error in one step ahead of markedly non-stationary financial time series forecasting task. We did not aim to design ready made system for trading in real market. Our analysis demonstrated usefulness of application of MAFS and indicated that much wider experimentation is necessary in order to determine global parameters of the decision making algorithm, which here were determined from several empiric experiments.

Unfortunately, we cannot compare our method with many of the methods proposed in the field in quantitative terms before rearranging and repeating the experiment due to lack of common problem formulation standards and wide variety of experimental designs in the field. The repetition of published experiments is often impossible due to lack of details provided. However, our paper deals with several important problems often addressed in this field, the most important of which is complexity of influencers and changing environments. Therefore, qualitative results here we believe more important than quantitative gains.

6 Conclusion

We expanded MSSE algorithm from single forecast to the MAFS approach. The MSSE-1 contributed towards solving a problem of *permanently* changing environments. The MAFS additionally contributes to solving *temporary* changing environments problem. It is done through utilization of a great number of diverse prediction rules that learn and adapt to environmental changes differently.

We achieved gains in testing error, while changes in training window size were controversial: for artificial time series minimum smoothed testing error was reduced from 25,60% to 25,35% by 0,25% (MLP gave 26,97% error). Training window size, however, was slightly increased from 340 to 350 days. For the real market Copper price time series minimum smoothed testing error was reduced from 45,19% to 40,29% and training window size decreased from 270 to 100. At 100 days training window MSSE-1 gave 46,9%. Therefore, at that point the gain in accuracy was even larger.

Positive effect tendencies of MAFS as compared to MSSE-1 can be clearly seen in the graphs. Although absolute gain is not large, we got promising principal results in artificial, as well as real financial variables testing. The most important result of our gain is that the improvement was achieved over all lengths of the training window sizes. The integration of MSSE algorithm to MAFS in addition to contribution towards solving *permanently* changing environment problem contributes to solving *temporary* changing environment problem.

Acknowledgements

The authors thank Dr. Aistis Raudys and Simas Tamošiūnas for useful and challenging discussions, sharing the Matlab codes ant the data sets.

References

1. Raudys S.: Survival of intelligent agents in changing environments. Lecture Notes in Artificial Intelligence, Vol 3070. Springer-Verlag, Berlin Heidelberg (2004) 109–117.
2. Fama, E.F.: Efficient capital markets: A review of theory and empirical work. Journal of Finance, Vol. 25. Blackwell Publishing Malden USA (1970) 383–417.
3. Moody J.: Economic forecasting: challenges and neural network solutions. In Proceedings of the International Symposium on Artificial Neural Networks, Hsinchu, Taiwan, 1995.
4. Huang W., Lai K.K., Nakamori Y. and Wang S.: Forecasting foreign exchange rates with artificial neural networks: a review. International Journal of Information Technology & Decision Making, Vol. 3(1). World Scientific Publishing, (2004) 145–165.
5. Lendasse, A., De Bodt, E., Wertz, V., Verleysen, M.: European Journal of Economic and Social Systems, Vol. 14(1). EDP Sciences Les Ulis Cedex, France (2000) 81–92.
6. Raudys S., Zliobaite I.: Prediction of commodity prices in rapidly changing environments. Lecture Notes in Computer Science, Vol. 3686. Springer-Verlag, Berlin Heidelberg (2005) 154–163.
7. Raudys A., Long J.A.: MLP based linear feature extraction for nonlinearly separable data. Pattern Analysis and Applications, Vol. 4(4). Springer London (2001) 227–234.
8. Raudys S.: An adaptation model for simulation of aging process. International Journal of Modern Physics, C. Vol. 13(8).World Scientific (2002) 1075–1086.
9. Raudys S.: Information transmission concept based model of wave propagation in discrete excitable media. Nonlinear Analysis: Modeling and Control, Vol. 9(3). IMI Vilnius (2004) 271–289.