INFORMATICA, 2002, Vol. 13, No. 4, 465–484

Research of Neural Network Methods for Compound Stock Exchange Indices Analysis

Darius PLIKYNAS
Department of Theoretical Economics, Vilnius University
Saulėtekio 9, 2040 Vilnius, Lithuania
e-mail: d.plikynas@delfi.lt

Leonas SIMANAUSKAS
Department of Economical Informatics, Vilnius University
Saulėtekio 9, 2040 Vilnius, Lithuania
e-mail: leonas.simanauskas@ef.vu.lt

Sigitas BŪDA
Institute of Mathematics and Informatics
Akademijos 4, LT-2021 Vilnius, Lithuania
e-mail: s.buda@it.lt

Received: August 2002

Abstract. The presented article is about a research using artificial neural network (ANN) methods for compound (technical and fundamental) analysis and prognosis of Lithuania's National Stock Exchange (LNSE) indices LITIN, LITIN-A and LITIN-VVP. We employed initial pre-processing (analysis for entropy and correlation) for filtering out model input variables (LNSE indices, macroeconomic indicators, Stock Exchange indices of other countries such as the USA – Dow Jones and S&P, EU – Eurex, Russia – RTS). Investigations for the best approximation and forecasting capabilities were performed using different backpropagation ANN learning algorithms, configurations, iteration numbers, data form-factors, etc. A wide spectrum of different results has shown a high sensitivity to ANN parameters. ANN autoregressive, autoregressive causative and causative trend model performances were compared in the approximation and forecasting by a linear discriminant analysis.

Key words: neural networks, artificial intelligence, forecasting, time series.

1. Introduction

In financial capital markets the profit crucially depends on capabilities to foresee market behaviour. SE indices are used like additive indicators for market movements. There exist two main approaches for SE indices analysis and prognosis: technical-historical and fundamental (Kancerovycius, 1999). The main goal of this article is construction of compound models that could include not only a historical data, but also macroeconomic data for the analysis and forecasting of SE indices movement. These movements of LNSE price indices LITIN, LITIN-A and LITIN-VVP[1] are investigated and autoregressive, autoregressive-causative and causative (Martisius, 2000) models designed. We used artificial intelligence systems (Trippi and Lee, 1996) for model construction. Nonlinear neural network methods were chosen, due to existing nonlinearities in the SE market, unclear SE value formation mechanism and due to the existence of unique (seldom occurring) events.

Beginning with the mid-eighties, a lot of articles was published about artificial intelligence systems for financial analysis: knowledge based expert systems, semantic nets, genetic algorithms, fuzzy logic, fractal theory, neural nets, etc. We found neural networks especially promising because of their non-linearity, flexibility, easy for use, and perspective former research results by other authors (Trippi and Turban, 1993).

C. Klimasauskas (1991) describes the rules for neural net [NN] optimisation in financial analysis. D.D. Hawley, J.D. Johnson, D. Raina describe NN possibilities for solving financial problems. D. Barker investigates integral models of NN and expert systems. P.K. Coats and L.F. Fant have made a diagnosis of corporate distress. W. Raghupathi, L.L. Schkade, B.S. Raju and R. L. Wilson, R. Sharda have developed a neural net approach for bankruptcy prediction. S. Dutta, S. Shekhar, A.J. Surkan, J.C. Singleton, H.L. Jenson have prepared debt risk assessment.

A number of authors are interested in stock exchange market analysis and prognosis, using the methods of neural nets. H. White carries out IBM daily stock prices prediction using neural networks. NN are distinguished by extremely dynamic behaviour and a good prediction performance. Y. Yoon and G. Swales compared the multidimensional discriminant analysis and the neural net performance. They declare that NN methods could significantly increase the SE price prediction accuracy. T. Kimoto, K. Asakawa, and M. Takeoka made a complex analysis using technical and fundamental methods. The authors used a system of several neural nets. The model performance with real data gave much higher profit yields than using the traditional methods.

The previously mentioned and many other works determined our choice for neural net methods in SE indices analysis and forecasting.

2. Research of Methods

A review of related literature suggests that backpropagation neural network method would better fit for our research because of the conditions mentioned above. The neural net method (back-propagation) refers to the input data $p(p_1 \ldots p_k)$, weights W (weight coefficients on the neuron junctions) vector, summation (n function) and transformation (a function) in a single NN element – so called artificial neuron (Trippi and Turban, 1993). Parallel-like neuron structures with couple of layers are capable to approximate any function with the limited number of breaks.

[1]The National Stock Exchange of Lithuania calculates three equity indices, namely, capitalization-weighted indices LITIN and LITIN-VVP and a market price index LITIN-10.

Back-propagation neural nets are learning by improving output performance $d(t)$ at every iteration. In the learning stage, NN output is compared to the known output results $d^T(t)$. The difference between $d(t)$ and $d^T(t)$ gives errors as arguments for weight adaptation rules. The weights are adapted to minimise differences between the NN output $d(t)$ and the output $d^T(t)$ (Trippi and Lee, 1996) which is known in advance. Default values for initial conditions are zeros. We used batch NN training: weight matrix was updated each time when all input data were presented to the net.

Various transformation functions and learning algorithms were used. The greatest part of the present research was executed by the perceptron and backpropagation NN methods (Hecht–Nielsen, 1990). The number of iterations used, net configurations, momenta, etc are presented in detail in the next chapters.

In Fig. 1, we described the overall research scheme which consists of a few basic stages: input data pre-processing and filtering, search for the best neural net method, neural net optimisation, and a compound analysis of stock exchange indices.

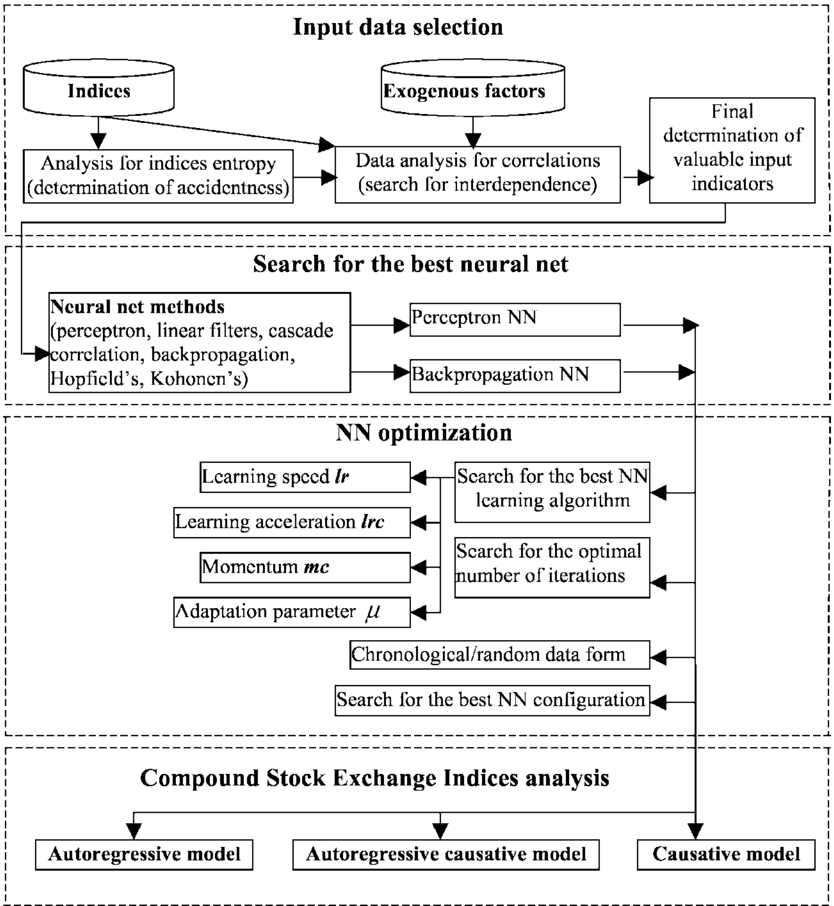

Fig. 1. Overall research scheme.

The authors used the stock data collected in LNSE databases, where price index estimation is based on International Finance Corporation [IFC] methods (see methods for indexes calculation, 2001) and indices are accounted according to the mean weighted session price and trading day rate of exchange (NSE, 2001).

3. Input Data Selection

In this stage, we estimated the overall stock exchange market environment. We were looking for indicators that may influence LITIN, LITIN-A and LITIN-VVP indice fluctuations. The investigation period includes data from 1997.01.01 to 2001.01.17 (see the curves in Fig. 2). We assumed that indices could be influenced not only by NSE market endogenous factors, but also by different exogenous factors, such as macroeconomic indicators.

Before the research, the authors had no idea as to what extent SE index movements are influenced by exogenous factors. Therefore, we were looking for the measure that might estimate accidentness of index movements. If accidentness is high, then chaos prevails and none of exogenous factors would help to make a better analysis and forecasting. We chose entropy (see Formula 1) as a relative measure which indicates the level of SE indices accidentness. The formula for entropy calculation is

$$H = -\int_{t1}^{t2} p(t) * \ln p(t) * \mathrm{d}t. \tag{1}$$

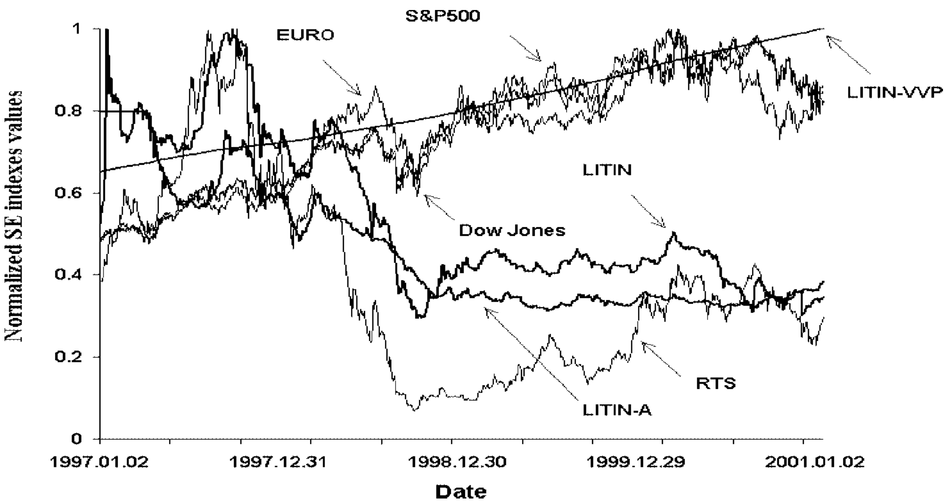

Fig. 2. Stock exchange indices movements in the period of investigation (Source data: Lithuania National Stock Exchange, 2001).

Table 1

Research data for SE indices entropy

Entropy	LITIN	LITIN-A	LITIN-VVP	RTS	S&P500
$H_{\max}$	4,5	5,08	4,20	3,99	4,39
$H_{\min}$	0	0	0	0	0
H	3,9	4,28	4,18	3,63	4,21
H (% of $H_{\max}$)	87	84	99	91	96

Entropy reaches the maximum, when probabilities for separate values are equal to $p_1 = p_2 = \ldots = p_n$ (values are accidental). Entropy reaches the minimum, when probability for one separate value is 1 and for the others values is zero (values are not accidental). Maximum and minimum limits for entropy in our data case are $H_{\max}$ and $H_{\min}$ (see Table 1).

The research shows a relatively low level of accidentness for indices LITIN and LITIN-A, which suggests the existence of trends and a possibly high influence of exogenous factors. The next stage of investigation concerns search for economic indicators able to influence index movements. This is of special concern for financial market participants, who might considerably increase profits by making more accurate predictions (Trippi and Lee, 1996).

We were looking only for those market indicators which might mostly influence the LNSE index movements. For this purpose, we employed the correlation analysis

$$\rho\,(X,Y) = \mathrm{cov}\,(X,Y)\,/\sqrt{\mathrm{D}X \cdot \mathrm{D}Y}, \tag{2}$$

where X and Y denote different variables, and $\mathrm{D}X, \mathrm{D}Y$ are the corresponding dispersions. Though this is not the proper method for nonlinear relations (Shumway, 1988), we were interested only in general interdependence and, therefore a simple correlation analysis helped us to reveal the most correlated variables such as:

a) macroeconomic indicators: GNP, unemployment, inflation, interbank exchange rate, interest rate for terminal deposits, deposits, investment portfolio, budget income, money M2,

b) SE price indices: US – Dow Jones and S&P500, EU – EUREX, Russia – RTS.

Correlations between indices are shown in Table 2. Correlations between indices and macroeconomic indicators are shown in Table 3.

From the tables above we made obvious assumptions on the correlations between indices and between indices and macroeconomic indicators. Therefore, in order to analyse and predict NSE indices LITIN, LITIN-A and LITIN-VVP movements, we had to include the most correlated fundamental macroeconomic indicators. In the next stages, we constructed compound models for the analysis and forecasting of SE indices.

Table 2

Correlations between SE indices

	LITIN	LITIN-A	LITIN-VVP	Dow Jones	EURO	RTS	S&P500
LITIN	–	0.93	−0.76	−0.72	−0.70	0.88	−0.75
LITIN-A	0.93	–	−0.84	−0.87	−0.84	0.74	−0.89
LITIN-VVP	−0.76	−0.84	–	0.93	0.83	−0.52	0.93
Dow Jones	−0.72	−0.87	0.93	–	0.93	−0.53	0.99
EURO	−0.70	−0.84	0.83	0.93	–	−0.60	0.92
RTS	0.88	0.74	−0.52	−0.53	−0.60	–	−0.57
S&P500	−0.75	−0.89	0.93	0.99	0.92	−0.57	–

Table 3

Correlation between SE indices and Lithuania macroeconomic indicators

	GNP	TermDep	InvPortf	Budget	Money	CurrDepos	Unemployment
LITIN	−0.61	0.71	−0.18	−0.40	−0.81	−0.78	−0.58
LITIN-A	−0.59	0.78	−0.31	−0.40	−0.87	−0.84	−0.67
LITIN-VVP	0.67	−0.87	0.55	0.36	0.96	0.99	0.92
Dow Jones	0.63	−0.86	0.56	0.47	0.92	0.93	0.84
EURO	0.60	−0.79	0.46	0.52	0.85	0.83	0.72
RTS	−0.50	0.55	0.03	−0.46	−0.61	−0.56	−0.31

4. Neural Net Method Selection and Optimisation

Different neural net methods could be successfully applied in the same task resolution (Coats and Fant, 1993). For the autoregressive trend model, we used the most simple NN – perceptron method with only one neuron in the hidden layer which acts like a multidimensional linear regressor (Koutsougeras and Papachristou, 1988).

For autoregressive-causative and causative trend models, we used multilayer back-propagation perceptron methods, because they suit well for the nonlinearities, unique event recognition and prediction (Stornetta and Huberman, 1988). Neural net construction requires employment of optimisation techniques:

1) search for effective NN learning algorithms,
2) search for an optimal number of iterations,
3) search for an optimal data form factor,
4) search for the best fitting of NN configuration,
5) search for recurrence.

The text below will introduce each step. For further investigations we used NSE index LITIN movements.

Search for Effective NN Learning Algorithms

Neural net learning algorithms are the keystones of the NN method. They determine speed, accuracy, soundness and functionality of the NN learning stage (Caudill and Butler, 1992). We investigated the following basic learning algorithms:

- gradient descent algorithm (Fig. 3);
- batch gradient descent with momentum algorithm;
- variable learning rate algorithm (Fig. 4);
- conjugate gradient algorithm;
- Levenberg-Marquet algorithm.

The mean square error MSE (see Formula 3), the R^2-determination coefficient (see Formula 4) and learning duration are the main criteria for the mutual comparison of NN learning algorithms.

$$MSE = 1/N \sum_{t=1}^{N} \left(d\left(t\right) - d^T\left(t\right)\right)^2, \tag{3}$$

where $d(t)$ stands for the supposed values and $d^T(t)$ – for real testing data.

The statistics that is wide used to determine how well a regression fits the coefficient of determination R^2 or the multiple correlation coefficient (Anderson-Sprecher, 1994) represents the fraction of variability in y that can be explained by the variability in x. In other words, R^2 explains how much the variability in the y's can be explained by the fact, that they are related to x, i.e., how close the points are to the line. The equation for R^2 is

$$R^2 = (SSTotal - SS\mathrm{Re}s)/SSTotal = 1 - SS\mathrm{Re}s/SSTotal, \tag{4}$$

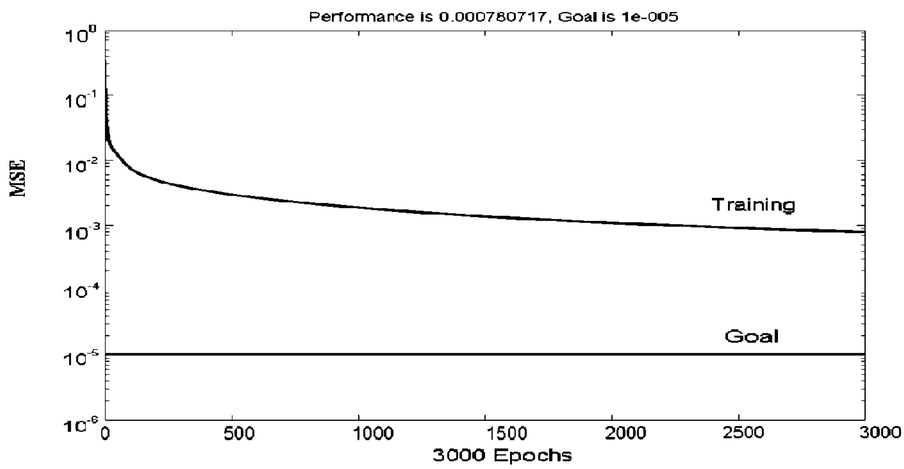

Fig. 3. NN learning function: MSE vs. the number of iterations (gradient descent learning algorithm, autoregressive causative model).

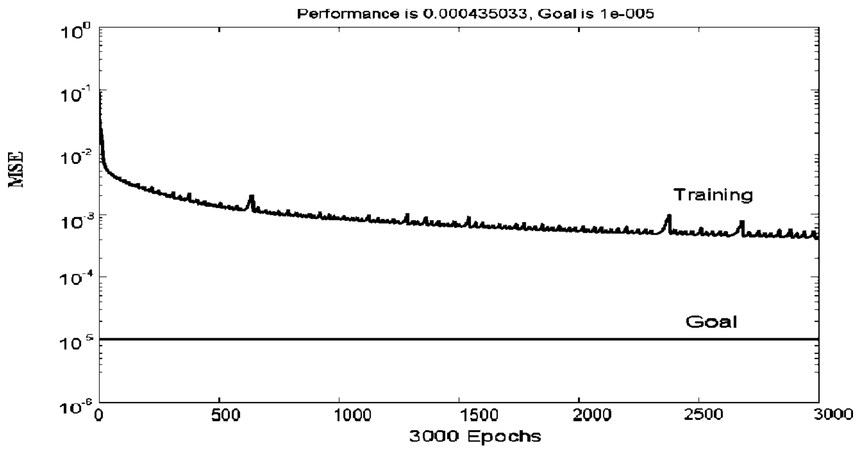

Fig. 4. NN learning function: MSE vs. the number of iterations (learning algorithm of the variable learning rate, autoregressive causative model).

where $SSTotal$ is the total sum of squares of the data and $SS\mathrm{Re}s$ is the sum of residual of the squares. In the case of a simple linear regression, R^2 is simply the square of the correlation coefficient.

In Table 4, we present the learning results with different learning algorithms in the learning and testing stage. The neural net performance is compared to the multidimensional discriminant analysis, for which

1) in the learning stage $MSE = 0.000088 =$const and $R^2 = 0.9987 =$const;

2) in the testing stage $MSE = 0.0098 =$const and $R^2 = 0.9883 =$const.

Table 4

Approximation performance of different NN learning algorithms in the learning and testing stages

Learning algorithm (MatLab function)	**Learning stage**		**Testing stage**		**NN learning time, min. (learning stage)**
	MSE	R^2 determin. coeff.	MSE	R^2 determin. coeff.	
Gradient descent (traingd, $lr = 1.2$)	0,00078	0.9599	0.0144	0.8127	29
Batch gradient descent with momentum (traingdm, $lr = 1.2$, $mc = 0.35$)	0.00054	0.9641	0.0139	0.8248	31
Variable learning rate (traingda, $lr = 0.9$, $lr\_inc = 1.05$, $lr\_dec = 0.7$)	0.00043	0.9862	0.0129	0.8381	28
Conjugate gradient (traincgf, function srchcha)	0.000060	0.9984	0.0096	0.8891	63
Levenberg-Marquet (trainlm, 30 iterations)	0.000056	0.9981	0.0095	0.8957	23

Experimentation was conducted using 23 different input variables (indices and macroeconomic data) presented in the chronological order. Time scale as a linear measure makes sense if the "business" day approach is employed. It means, that on the time scale we have not calendar date, but stock exchange session dates. Beginning from 1997.01.01 to 2000.12.01 about 997 relevant (successive) data rows were collected for NN learning and 28 rows for NN testing. For NN learning purposes we used 3000 iterations. Neural net stopped learning when minimal MSE, marginal minimal value of gradient descent or maximal iteration number were reached. NN configuration: 23 input neurons, 15 neurons in hidden layer and 1 neuron in output layer. We used log-sigmoid transformation function. We chose an optimal neural net configuration following the NN optimization rules formulated by C. Klimasauskas (1991).

Therefore, from the research above we got the best performance for conjugate gradient and Levenberg-Marquet learning algorithms (both algorithms employ quasi Newton methods). These algorithms are capable of accurate and rapid approximation and prediction.

Search for the Optimal Neural Net Learning Iterations Number

Theoretically NN could learn forever, i.e., it could endlessly converge to $\min(MSE)$, but there are two main reasons which stop such a process: time limit and over-training. More powerful computers could coup with the processing time problem, but it will not help with overtraining: at the end of long convergence the NN will be too zealous in the approximation, but too bad for prediction (see Table 5). Therefore, an optimal number of iterations were found using trial and error method. In Table 5, we used the Levenberg-Marquet (trainlm) learning algorithm and the NN configuration 23:15:1. For the com-

Table 5

Search for the optimal NN learning iteration number (chronological data order)

Number of iterations	Neural net		Learning time, min
	MSE	R^2 deter. coeff.	
	Learning stage		
2	0.0014	0.8958	2
	Testing stage		
	0.0168	0.7286	X
	Learning stage		
10	0.000097	0.9991	12
	Testing stage		
	0.0313	0.7054	X
	Learning stage		
30	0.000056	0.9981	23
	Testing stage		
	0.0545	0.6851	X

parison multidimensional linear regression gives $MSE = 0.000088$, $R^2 = 0.9987$ in learning stage and $MSE = 0.0098$, $R^2 = 1.2117$ in testing stage.

The use of universal rules without any justification is not recommended. This could substantially decrease an optimal NN performance. Usually special software is used for NN optimisation (Trippi and Turban, 1993).

Search for the Optimal Data form Factor

The data for learning are usually presented to NN in random order. This is justified if many trends are present in SE index movements, because in this case NN could perform approximation better. If just several basic trends persist, then it is better to present the data for NN in a chronological order, because NN learns trends within its learning stage.

In Table 6, we present NN behaviour if the data are presented in a random order. We used Levenberg-Marquet (trainlm) learning algorithm; NN configuration 23:15:1. The comparison of Tables 6 and 5 clearly show (in both cases other conditions were the same) a better approximation and prediction if the data are presented in a chronological order. It proves that NSE index LITIN has just several basic trends.

Search for the Best Fitting NN Configuration

Neural net performance extremely depends on NN configuration: the number of layers, connections topology and neuron numbers in the layers (Trippi and Turban, 1993). A common practice reveals: 1) if the net has too little neurons or connections, then bad approximation results occur, 2) if the net has too much neurons or connections, then bad prediction occurs. Besides, the net performance strongly depends on the number of variables and interdependencies between the variables and the data amount (Klimasauskas, 1991). In the Table 7, we present the search for the best NN configuration (for autoregres-

Table 6

Search for the optimal data form factor (random order)

Number of iterations	**Neural net**		**Learning time, min**
	MSE	R^2 deter. coeff.	
	Learning stage		
2	0.0082	0.9551	2
	Testing stage		
	0.0930	0.6452	–
	Learning stage		
10	0.000068	0.9972	12
	Testing stage		
	0.0716	0.7455	–
	Learning stage		
30	0.000061	0.9984	23
	Testing stage		
	0.1026	0.6289	X

Table 7
Search for the best NN configuration (topology)

Number of neurons in the layers	Learning stage			Testing stage	
	MSE	R^2 determin. coeff.	Learning time, min.	MSE	R^2 determin. coeff.
10:5:1	0.000087	0.9963	2	0.0441	0.5487
10:15:1	0.00012	0.9906	3	0.0166	0.7288
15:10:1	0.00011	0.9911	3	0.0095	0.8857
15:10:1 (20 iter.)	0.000057	0.9987	6	0.0308	0.6451
15:15:1	0.000074	0.9954	5	0.0348	0.6214
23:5:1	0.00013	0.9916	4	0.0398	0.6149
23:10:1	0.00010	0.9937	7	0.0240	0.6841
23:15:1	0.000097	0.9944	12	0.0313	0.6499
23:20:1	0.000086	0.9950	13	0.0188	0.7301
23:5:10:1	0.000095	0.9948	12	0.0574	0.4951
23:10:5:1	0.0012	0.9056	12	0.0146	0.7352
Multid. linear regr.	0.000088	0.9987	X	0.0098	0.8822

sive causative and causative models, see below). We used Levenberg-Marquet learning algorithm, number of iterations by default was 10.

Relative to multidimensional linear regression the NN configuration 15:10:1 gave us better prediction, but worse approximation results. Meanwhile, configuration 23:20:1 gave us better approximation, but worse prediction results. In the case of greater number of layers or neurons, we experienced overtraining (Hecht–Nielsen, 1990) and more computer processing time. The research shows complex and ambiguous NN configuration influence to the index approximation and prediction results.

Search for recurrence

We investigated recurrence of our results (Charalambous, 1992). It showed the possibility to get the same results in other set of experimentation (research conditions remained the same). In the Table 8, we present an investigation for recurrence. We used Levenberg-

Table 8
Search for the recurrence

Number of neurons in the layers	Learning stage			Testing stage	
	MSE	R^2 determin. coeff.	Learning time, min.	MSE	R^2 determin. coeff.
1	0.0022	0.9098	2	0.1087	1.8546
2	0.0032	0.8983	2	0.1085	1.5174
3	0.0011	0.9400	2	0.1045	1.2823
4	0.00035	0.9908	2	0.0849	1.1094
5	0.0026	0.8486	2	0.1321	0.9702
Multidimensional linear regression	0.0023	0.9658		0.3583	5.4228

Marquet learning algorithm, number of iterations by default was 10.

In the learning stage MSE changes $\pm 40\%$ and the determination coefficient changes $\pm 9\%$ from the corresponding mid values. In the testing stage MSE changes $\pm 25\%$ and determination coefficient changes $\pm 40\%$. It shows, that NN learning and testing results considerably depend on the initial weights and bias (in our search of recurrence were just 10 iterations). We have noticed that recurrence significantly increased with the higher number of iterations (MSE and determination coefficient converged to the stable values).

5. Applications of NN Methods for Research of SE Indices Trend Models

So far, we have done a selection of the fitting input variables, (see Tables 2 and 3), optimisation of NN methods and their parameters (see Figs. 1, 3 and 4; Tables 4–8). Therefore, we prepared the means for the compound analysis that embrace not only technical, but also fundamental data. Afterwoods we were ready for the final research of autoregressive, autoregressive causative and causative trend models. Below we briefly describe the research results for every model.

Autoregressive trend model

This model embraces just technical historical analysis. Authors investigated capabilities of NN methods for approximation of NSE indices LITIN, LITIN-A and LITIN-VVP movements. We used the simplest NN model – perceptron with three input values: 1, 3 and 5 days old historical values of the corresponding index. For the perceptron learning were 1028 data rows collected in the period from 1997.01.01 to 2001.01.17. Perceptron learning executed until 1000 iterations. For the research, we have used MATLAB 6.0 v. Neural Network Toolbox software.

Table 9 shows, that even in the most simple (perceptron) case when we have just two neurons, the linear regression method (see Formula 5) slightly gives up to the NN perceptron method.

$$y = \sum_i \alpha_i \cdot x_i + \alpha_0, \tag{5}$$

where x_1 – 1 day, x_2 – 3 days and x_3 – 5 days old index LITIN values; $\alpha_1 = 0.55$, $\alpha_2 = 0.3$, $\alpha_3 = 0.15$.

Table 9

NN perceptron method performance vs. linear regression

Index	MSE	
	Perceptron	**Linear regression**
LITIN	71,9	72,1
LITIN-A	612,3	635,9
LITIN-VVP	0,16	0,17

Autoregressive Causative Trend Model

This is a compound model aimed not only for the technical, but also for the fundamental (exogenous factors like macroeconomic data) analysis as well. We employed the backpropagation multilayer perceptron method for the investigation of NSE index LITIN. NN input layer consisted of 23 neurons (each neuron for each variable):

1) autoregression part consisted of
   - one, three and five days old LITIN values,
   - current and 1, 3 and 5 days old LITIN-A, LITIN-VVP values,
2) causative part consisted of
   - other countries SE indices (US – Dow Jones and S&P, ES – EUREX/DAX, Russia– RTS),
   - macroeconomic indicators (unemployment, inflation, interbank exchange rates, deposit's rates, deposits amount, investment portfolio, budget income, money supply M2).

The analysis of LITIN movements, because of the model's complexity, gives us (if to compare to autoregressive trend model) one level higher approximation and prediction accuracy. All measurements in the previous chapter were made by use of autoregressive causative model. Tables 4–8 could well represent this model.

NSE index LITIN movements, multidimensional linear regression and NN approximation results are presented in the Figs. 5 and 6; prediction results in the Figs. 7, 8 and 9.

Causative Trend Model

In this case, NSE index LITIN movements are analysed just by the fundamental macroeconomic indicators and other countries SE indices, but not by LITIN values themselves.

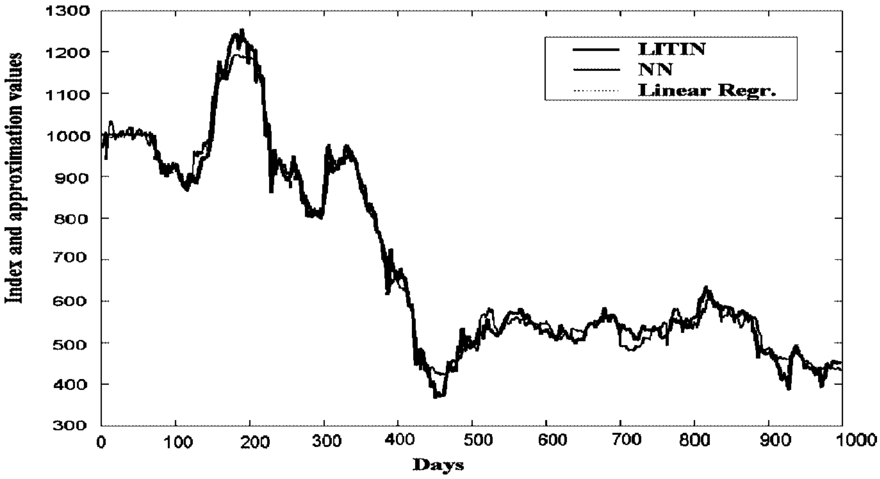

Fig. 5. NSE index LITIN movement approximation with the linear regression and NN methods (3000 iterations, batch gradient descent learning algorithm, configuration 23:15:1).

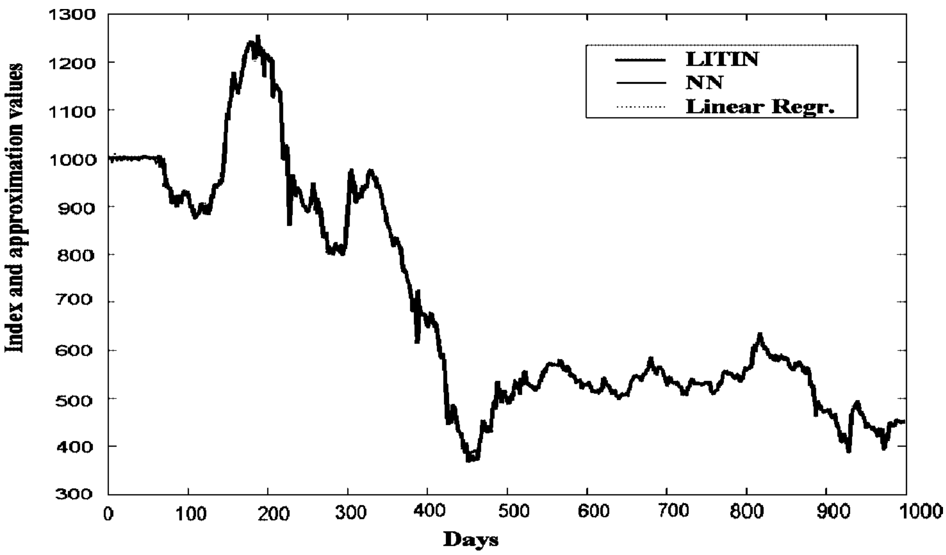

Fig. 6. NSE index LITIN movement approximation with the linear regression and NN methods (30 iterations, Levenberg-Marquet learning algorithm, configuration 23:15:1).

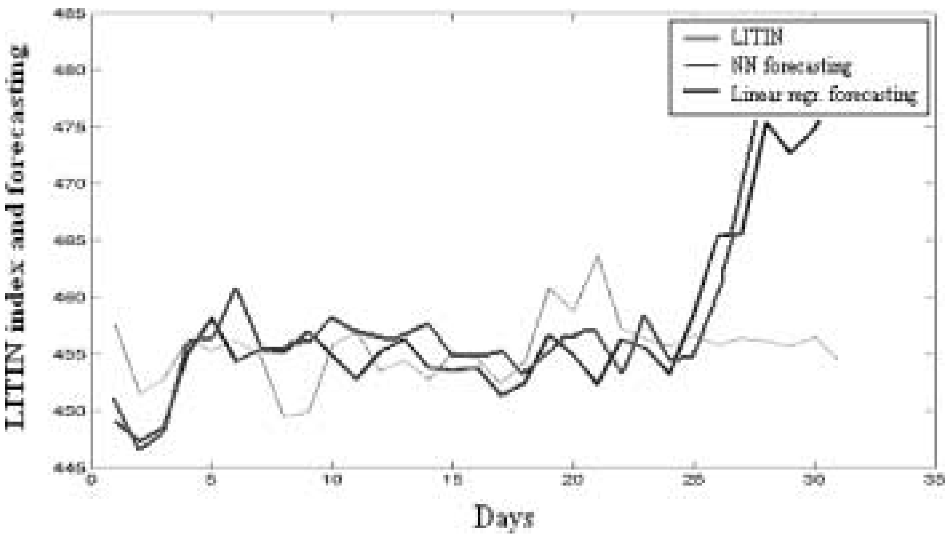

Fig. 7. NSE index LITIN movement prediction (from 1 till 32 days) with the linear regression and NN (3000 iterations, batch gradient descent learning algorithm, configuration 23:15:1) methods.

Twenty input variables were presented to the NN: LITIN-A and LITIN-VVP values (current and 1, 3 and 5 days old values), other countries SE indices (US – Dow Jones and S&P, ES – EUREX/DAX, Russia – RTS), macroeconomic indicators (unemployment, inflation, interbank exchange rates, deposits rates, citizens deposits amount, investment portfolio, budget income, money supply M2).

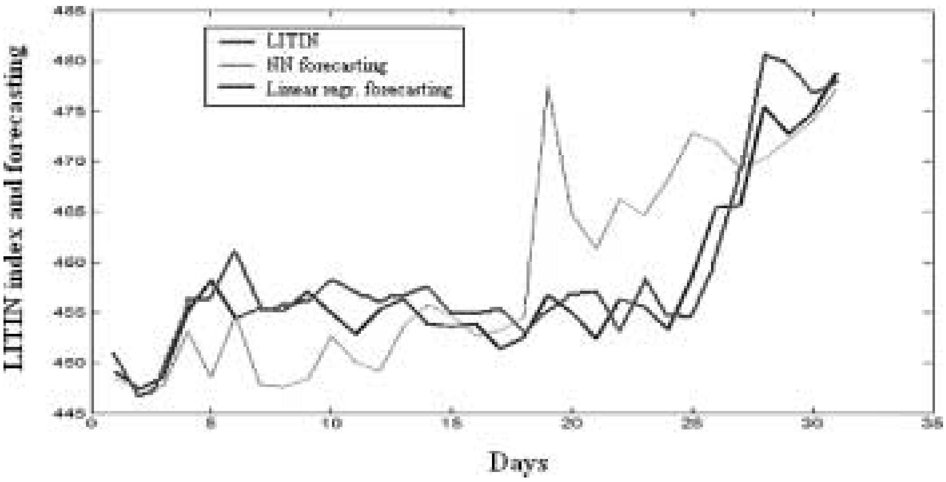

Fig. 8. NSE index LITIN movement prediction (from 1 till 32 days) with the linear regression and NN methods (30 iterations, Levenberg-Marquet learning algorithm, configuration 23:15:1).

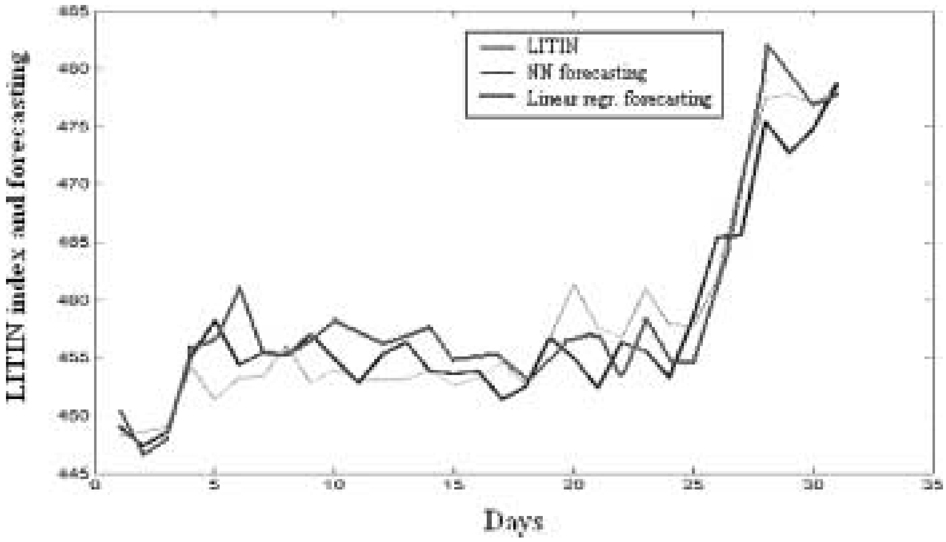

Fig. 9. NSE index LITIN movement prediction (from 1 till 32 days) with the linear regression and NN methods (10 iterations, Levenberg-Marquet learning algorithm, configuration 23:15:1).

Therefore, in this case we had just exogenous variables. This made the model accuracy dependant from the strength of correlation between exogenous variables and index of investigation (Steiner, 2001). As we saw, the index LITIN is highly correlated with the index LITIN-A ($k = 0,93$), Russian index RTS ($k = 0,88$), money supply M2 ($k = -0,81$), citizens deposits ($k = -0,78$) and etc (see Tables 2 and 3). Such a high correlation gives us good results for approximation and prediction. In the Chapter No. 4 "Neural net method selection and optimisation" Tables 7 and 8 are made using a causative

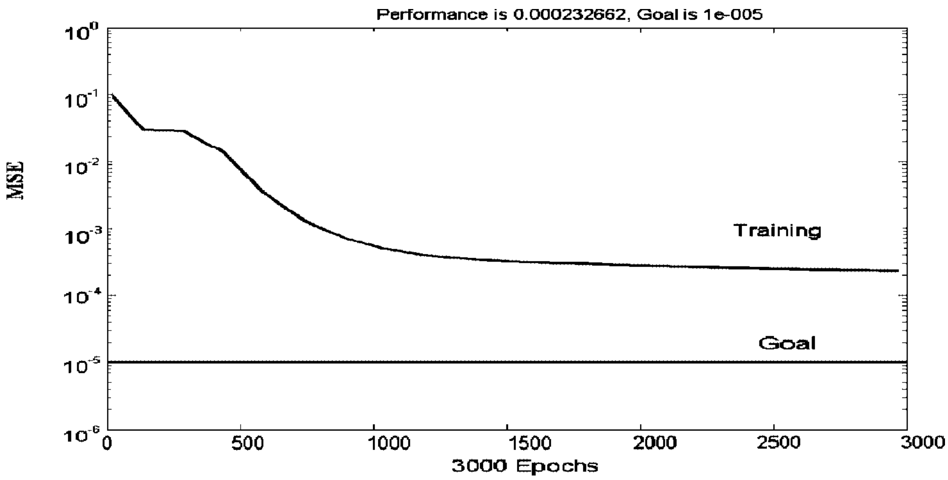

Fig. 10. NN learning function: MSE vs. number of iterations (Levenberg-Marquet learning algorithm, 20 iterations, configuration 15:5:1, causative trend model).

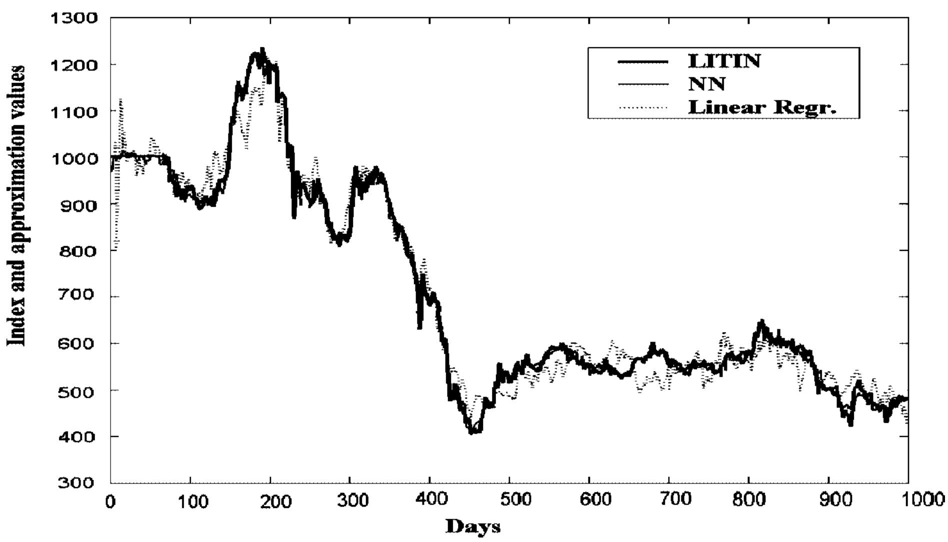

Fig. 11. NSE index LITIN movement approximation with the linear regression and NN (Levenberg-Marquet learning algorithm, 20 iterations, configuration 15:5:1, causative trend model).

model. Learning function is presented in Fig. 10, approximation results in Fig. 11 and prediction results in Fig. 12.

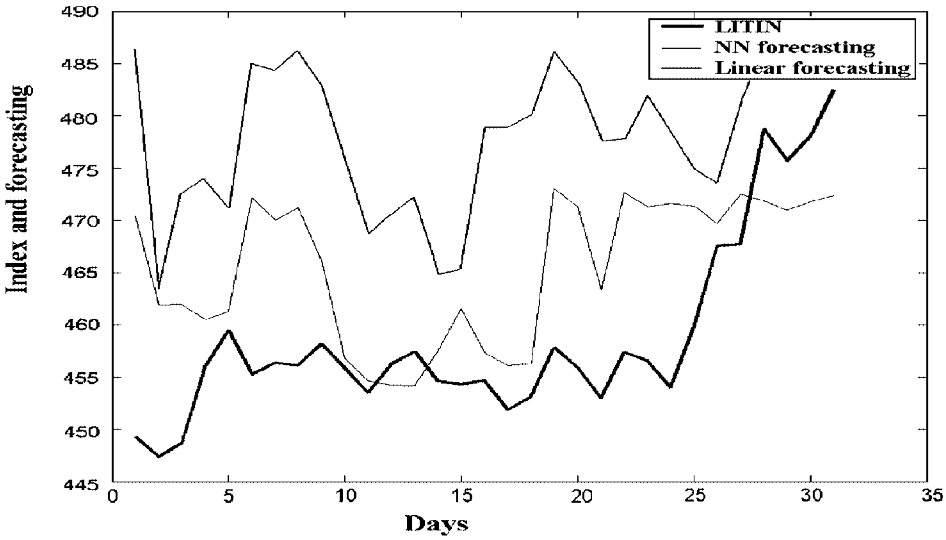

Fig. 12. NSE index LITIN movement prediction (from 1 till 32 days) with the linear regression and NN (Levenberg-Marquet learning algorithm, 20 iterations, configuration 15:5:1, causative trend model).

6. Conclusions

1. LNSE indices LITIN, LITIN-A and LITIN-VVP are investigated by the compound models of technical and fundamental means of analysis which is aimed at a good approximation and prediction of the nonlinear data, sensitive reaction to unique and seldom occurring events, flexibility and easiness for use.

2. The employment of the initial data pre-processing (analysis for entropy and correlation) showed the exogenous factors existence (macroeconomic indicators, other countries indices), which influenced the formation of index movements.

3. The employment of artificial neural networks allowed us to implement the compound autoregressive, autoregressive-causative, and causative models. The backpropagation multilayer perceptron methods met all the research conditions best after the relevant optimisation techniques have been employed: the search for the effective NN learning algorithms, optimal number of iterations, optimal data form factor, the best fitting NN configuration, etc. MSE and the determination coefficient was used to measure accuracy.

4. The research of autoregressive trend model (just technical analysis) with the simplest NN perceptron method (two neurons) showed the same or even a little better approximation results, than the multidimensional linear regression method.

5. The research of the autoregressive-causative trend model by the NN multilayer perceptron method gave the approximation and prediction results higher by one or even two orders than the simple autoregressive trend model. The NN prediction performance crucially depends on NN parameters. If the relevant optimisation technique is employed, NN could outperform a multidimensional linear regression.

6. The research of the causative trend model by the NN multilayer perceptron method gave us surprisingly good approximation and prediction results. This happened because of a high correlation between the index LITIN and other exogenous variables.

References

Anderson-Sprecher, R. (1994). Model Comparisons and R^2. *America Statistics*, **48**, 113–117.

Caudill, M., C. Butler (1992). *Understanding Neural Networks: Computer Explorations*, Vols. 1 and 2. Cambridge, MA, the MIT Press.

Cekanavicius, V., G. Murauskas (2000). *Applications of Statistics* (I book). TEV, Vilnius.

Charalambous, C. (1992). Conjugate gradient algorithm for efficient training of artificial neural networks. *IEEE Proceedings*, **139**(3), 301–310.

Coats, P., L. Fant (1990). Recognizing financial distress patterns using a neural network tool. *Financial Management*, Nov., 142–55.

Geralavicius, V. (2000). Lithuania economy: lessons from the last years. *Ekonomika*, **50**.

Hecht– Nielsen (1990). *Neurocomputing*. Reading, MA, Addison– Wesley.

Kancerovycius, G. (1999). *Technical analysis*. Business Machine Company, Vilnius.

Klimasauskas, C. (1991). Applying neural networks. *PCAI*, Jan/Feb, 30–33; Mar/Apr, 27–34; May/Jun, 20–24.

Koutsougeras, C., G. Papachristou (1988). Training of a neural network for pattern classification based on an entropy measure. *Proceedings of IEEE ICNN*.

Lithuania National Stock Exchange [NSE] (2001). (Downloadable from website `http://www.nse.lt`).

Lithuania Statistics Department (2000). *Economic and Social Development of Lithuania*, Vilnius. (Downloadable from website `http://www.lsd.lt`).

Martisius, S. (2000). *Econometrics and Prognoses*. Vilnius, Vilnius University Press.

Methodic for indexes calculation (2001). (Downloadable from website `http://www.nse.lt/beta/indeksuskaiciavimas.htm`).

Shumway, R.H. (1988). *Applied Statistical Time Series Analysis*. Englewood Cliffs, NJ, Prentice-Hall.

Steiner, B. (2001). *Key Market Concepts*. Reuters, Pearson Education, London.

Stornetta, W.S., B.A. Huberman (1988). An improved three layered backprop algorithm. In M. Caudill and C. Butler (Eds.), *Proceedings of the IEEE First International Conference on Neural Networks*. San Diego, CA, SOS Printing.

Trippi, R.R., J.K. Lee (1996). *Artificial Intelligence in Finance&Investing: State-of-the-Art Technologies for Securities Selection and Portfolio Management*. Irwin, Chicago.

Trippi, R.R., E. Turban (1993). *Neural Networks in Finance and Investing*. Probus, Chicago.

D. Plikynas received his physics-engineer degree on communication and networks in 1997 and MBA degree in 1998. Currently he is a doctorate student and lecturer at Vilnius University (cathedra of theoretical economics). His research interests focus on artificial intelligence in capital markets.

L. Simanauskas professor, Department of Economical informatics in Vilnius University.

S. Būda received his master's degree (1998) in mathematics from the faculty of Mathematics and informatics in Vilnius University. Currently he is a doctorate student at the Institute of Mathematics and Informatics. His research interests focus on optimisation of neural networks and genetic algorithms.

Dirbtiniai neuroniniai tinklai kompleksinėje vertybinių popierių indeksų analizėje

Darius PLIKYNAS, Leonas SIMANAUSKAS, Sigitas BŪDA

Straipsnio tikslas – informuoti apie atliekamus tyrimus, aiškinantis dirbtinių neuroninių tinklų galimybes prognozuoti Lietuvos NVPB indeksų LITIN, LITIN-A ir LITIN-VVP vertes. Atlikta Lietuvos VP indeksų entropijos bei koreliacijos analizė leidžia daryti išvadas apie indeksų kitimo atsitiktinumo bei priklausomybės lygį ir atitinkamai apie galimybes surasti indeksų kitimą įtakojančius ekonominius veiksnius. Dėl dirbtinių neuroninių tinklų metodų lankstumo, gebėjimo atsižvelgti į retai pasikartojančius įvykius ir dėl gebėjimo analizuoti netiesinius sąryšius, jie pasirinkti kompleksinei (techninei ir fundamentinei) NVPB indeksų analizei. DNT apmokymas atliekamas remiantis praėjusių periodų atitinkamų nacionalinių indeksų vertėmis, šalies makroekonominių rodiklių (nedarbo lygio, infliacijos, tarpbankinių palūkanų normų, terminuotų indėlių normų, gyventojų indėlių, investicijų portfelio, biudžeto pajamų, pinigų masės M2) bei kitų šalių (JAV – S&P500, Dow Jones; ES – EUREX/DAX; Rusija – RTS) VP kainos indeksų vertėmis. Ištirtos dirbtinio neuroninio tinklo aproksimavimo ir prognozavimo galimybės, esant skirtingiems mokymo algoritmams, duomenų pateikimo būdams, neuroninio tinklo konfigūracijoms. Tyrimo rezultatai lyginami su multidimensine tiesine regresija (palyginimo kriterijai – VKN ir determinacijos koeficientas). Nustatytos optimalios dirbtinio neuroninio tinklo konfigūracijos, leidžiančios gauti geresnius NVPB kainų indeksų aproksimavimo ir prognozavimo rezultatus negu tiesinės regresijos metodu.