Monika Miśkiewicz-Nawrocka

The application of random noise reduction by nearest neighbor method to forecasting of economic time series

Folia Oeconomica Stetinensia 13(21)/2, 96-108

2013

Folia Oeconomica Stetinensia
DOI: 10.2478/foli-2013-0020

Wydział Nauk Ekonomicznych i Zarządzania
Uniwersytetu Szczecińskiego

THE APPLICATION OF RANDOM NOISE REDUCTION BY NEAREST NEIGHBOR METHOD TO FORECASTING OF ECONOMIC TIME SERIES

Monika Miśkiewicz-Nawrocka, Ph.D.

University of Economics in Katowice,
Faculty of Management, Department of Mathematics,
1 Maja 50, 40-287 Katowice, Poland
e-mail: monika.miskiewicz@ue.katowice.pl

Received 20 October 2013, Accepted 17 January 2014

Abstract

Since the deterministic chaos appeared in the literature, we have observed a huge increase in interest in nonlinear dynamic systems theory among researchers, which has led to the creation of new methods of time series prediction, e.g. the largest Lyapunov exponent method and the nearest neighbor method. Real time series are usually disturbed by random noise, which can complicate the problem of forecasting of time series. Since the presence of noise in the data can significantly affect the quality of forecasts, the aim of the paper will be to evaluate the accuracy of predicting the time series filtered using the nearest neighbor method. The test will be conducted on the basis of selected financial time series.

Keywords: random noise reduction, nearest neighbor method, largest Lyapunov exponent, financial time series forecasting.

JEL classification: C01, C53, G17.

Introduction

The nearest neighbor method originated from the theory of nonlinear dynamical systems and was developed to predict the future values of a time series, but it can also be used to reduce random noise in the time series. The real time series (s_t) consist of the deterministic part (y_t) and the stochastic part (ε_t), which describes the level of random noise in the time series. The reduction of random noise allows to determine the properties of a time series (y_t) based on the analysis of a series of observations (s_t). Literature offers a number of methods used to reduce the level of random noise in dynamical systems and the main benefit of using these methods seems to be the improvement in time series forecasting capabilities.

In this article the hypothesis that the time series which are filtered by the nearest neighbor method give more accurate predictions than the unfiltered time series was verified. The aim of the paper was to assess the effect of random noise reduction using the method of nearest neighbors on the accuracy of the predictions obtained by the application of the method of the largest Lyapunov exponent and the nearest neighbor. The empirical research was based on the actual data of economic nature – the financial time series set up with the logarithms of daily returns on closing prices of selected stock exchange indices, equity prices, foreign exchange rates and commodity prices. The data cover the period from 3.01.2000 to 8.26.2013. To carry out the necessary calculations the author wrote programs in the Delphi programming language and an Excel spreadsheet.

1. The random noise reduction by the nearest neighbor method

The real time series can be described as dynamical systems (X, f) with the following equations[1]:

$$x_{t+1} = f(x_t + \eta_t) \tag{1}$$

$$s_{t+1} = h(x_{t+1}) + \xi_t \text{ , } t = 0, 1, 2, \ldots \tag{2}$$

where:

$X \subset R^m$, X – state space,

$f : X \to X$ – function describing the real dynamics of the system,

$h : X \to R$ – measuring function generating time series observations s_t of the dynamical system,

$x_t, x_{t+1} \in X$ – state of the unknown original multidimensional system at the moments t and $t + 1$ respectively,

s_{t+1} – an observation of the time series at the moment $t + 1$,

η_t – dynamic noise inside the system,

ξ_t – measurement noise.

In short, the real time series can be written in an additive form:

$$s_t = y_t + \varepsilon_t \tag{3}$$

where:

s_t – an observation of the time series at the moment t,

z_t – the deterministic part of the time series,

ε_t – the stochastic part of the time series (random noise consists of observation noise, system noise or their combination).

The main causes of observation noise in the time series are measurement errors and rounding errors, while the causes of system noise are exogenous factors affecting the dynamics of the system, which are impossible to identify[2].

The basis of the nearest neighbor method which is used to reduce random noise is the reconstruction of the state space[3]. This reconstruction allows to restore the state space of the dynamical system based on the one-dimensional time series observations. The elements of the reconstructed state space are delays vectors, so-called d-stories, in the following form:

$$s_t^d = \left(s_t, s_{t-\tau}, \ldots, s_{t-(d-1)\tau}\right) \tag{4}$$

where:

s_t – an observation of time series at the moment t,

d – embedding dimension,

τ – delay time,

$(d - 1)\, \tau + 1 \leq t \leq N$.

The algorithm for determining the value y_n, $1 < n < N$ of the time series (s_1, s_2, s_N) using the nearest neighbor method is as follows:

1. For estimated embedding dimension d[4] and delay time $\tau = 1$ we create the delay vector in the following form:

$$s_t^d = \left(s_t, s_{t-1}, ..., s_{t-(d-1)}\right) \tag{5}$$

so that the filtered observation s_n is one of the central coordinates of the vector s_t^d .

2. We determine k nearest neighbors (in Euclidean distance sense) of the vector s_t^d in the following form:

$$s_{l(1)}^d, s_{l(2)}^d, ..., s_{l(k)}^d \tag{6}$$

3. Based on the designated nearest neighbors we estimate the value y_n as the arithmetic average of the first coordinates of the nearest neighbors:

$$y_n = \frac{1}{k}\sum_{i=1}^{k} s_{l(i)} \tag{7}$$

2. The time series forecasting

2.1. The nearest neighbor method NNM

The theoretical basis of the nearest neighbor method is the fact that the states of the deterministic system evolve over time in a similar manner. In the case of time series, if we do not know the function f describing the dynamics of the system and we have only a one-dimensional series of observations $\left(s_1, ..., s_N\right)$, we can use the state space reconstruction. If $s_{t_0}^d$ is the nearest neighbor of point s_N^d , then also $f_T\left(s_N^d\right) \approx f_T\left(s_{t_0}^d\right)$, and hence it indicates that $s_{N+T} \approx s_{t_0+T}$. Thus, the value of s_{t_0+T} may be taken as a forecast of observation s_{N+T} in the analyzed time series[5].

In the nearest neighbor method, the forecast for $(N + 1)$-th element $\hat{s}_{N+1}$ is estimated as a weighted average of observations s_{i+1}, where the vectors s_i^d are k nearest neighbors of vector s_N^d in the reconstructed d-dimensional state space:

$$\hat{s}_{N+1} = \sum_{i=1}^{k} w_i s_{i+1} \tag{8}$$

The weights are chosen so that the closer neighbors have a greater impact on the obtained forecast. Accordingly, the weight of the i-th neighbor is estimated by the formulas[6]:

$$w_i = \frac{1}{k-1}\left(1 - \frac{d_i}{\sum_{i=1}^{k} d_i}\right) \tag{9}$$

$$w_i = \frac{2(k+1-i)}{k(k+1)} \tag{10}$$

$$w_i = \frac{e^{-d_i}}{\sum_{i=1}^{k} e^{-d_i}} \tag{11}$$

where $d_i = \left\| s_N^d - s_i^d \right\|$ is the distance between vectors s_N^d and s_i^d, $i = 1, 2, ..., k$.

2.2. The largest Lyapunov exponent method LEM

Lyapunov exponents are defined as limits[7]:

$$\lambda_i(x_0) = \lim_{n\to\infty} \frac{1}{n} \ln \left|\mu_i(n, x_0)\right|, \qquad i = 1, ..., m, \text{ dla } m \geq 1, \tag{12}$$

where $\mu_i(n, x_0)$ are the eigenvalues of the Jacobi matrix of mapping f^n, f^n is an n-fold submission of function f, and f is the function that generates a dynamic system.

The Lyapunov exponents measure the rate of divergence or convergence of neighboring trajectories, i.e. the level of chaos in a dynamic system. The largest Lyapunov exponent allows to specify the extent of a change (an increase or a decrease) in the distance between the current state x_N of the system and its nearest neighbor x_i in the evolution of the system, and also estimate the distance between the vectors x_{N+1} and x_{i+1}. Based on this distance the value of the forecasts $\hat{x}_{N+1}$ is determined[8].

For real-time series, if you do not know a generator function *f*, the largest (maximal) Lyapunov exponent is estimated based on the relation[9]:

$$\Delta_n = \Delta_0 \cdot e^{n\lambda_{\max}} \tag{13}$$

as the direction component of the regression equation[10]:

$$\ln \Delta_n = \ln \Delta_0 + \lambda_{\max} n \tag{14}$$

where Δ_0 is the initial distance between two initially close (in the Euclidean distance sense) points of the reconstructed state space, Δ_n is the distance between these points after n iterations and $\lambda_{\max}$ is the largest Lyapunov exponent.

Consider a one-dimensional time series, composed of N observation $(s_1, s_2, ..., s_N)$. Of all the vectors s_t^d of reconstructed state space we choose the vector closest to the vector s_N^d (in terms of Euclidean distance) and it is denoted by $s_{\min}^d$. Let $\Delta_{\min}$ denote the distance between s_N^d and $s_{\min}^d$, and Δ_1 – the distance between s_{N+1}^d and $s_{\min+1}^d$. Assuming that $\Delta_1/\Delta_{\min}$ is a small change in the evolution of the system, the distance between vectors s_{N+1}^d and $s_{\min+1}^d$ is given by[11]:

$$\Delta_1 \approx \Delta_{\min} \cdot e^{\lambda_{\max}} \tag{15}$$

where $\lambda_{\max}$ is the largest Lyapunov exponent.

Because

$$s_{N+1}^d = \left(s_{N+1}, s_{N-\tau+1}, ..., s_{N-(d-1)\tau+1}\right) \tag{16}$$

the predicted value s_{N+1} can be determined from the relation (15) as the solution to the following equation:

$$\left(z - s_{i+1}\right)^2 + \left(s_N - s_i\right)^2 + ... + \left(s_{N-(d-1)\tau+1} - s_{i-(d-1)\tau+1}\right)^2 - \left(\Delta_{\min} e^{\lambda_{\max}}\right)^2 = 0 \tag{17}$$

Hence the forecast $\hat{s}_{N+1}$ can have two values[12]: $\hat{s}_{N+1}^+$ and $\hat{s}_{N+1}^-$, which are respectively the overestimated and the underestimated real value of s_{N+1}.
Futher forecast $\hat{s}_{N+T}$, $T = 2,3,....$ can be calculated directly from the formula:

$$\Delta_T \approx \Delta_{\min} \cdot e^{\lambda_{\max} \cdot T} \tag{18}$$

where Δ_T is the distance between vectors s_N^d and $s_{\min}^d$ after T iteration steps, i.e. between vectors s_{N+T}^d i $s_{\min+T}^d$, or by an iterative procedure described above for the vector s_{N+1}^d [13].

3. The empirical research

The author investigated the logarithms of daily returns on world's stock exchanges indices: NIKKEI225 – Tokyo Stock Exchange index (NKX), S & P 500 – New York Stock Exchange index (SPX) and WIG – Warsaw Stock Exchange index; currencies: the Swiss franc (CHF), the euro (EUR), the Japanese yen (JPY), and the US dollar (USD) against the Polish zloty; the prices of shares in the following companies: ING Bank Slaski (BSK), Mostostal Zabrze (MFA), Vistula (VST) and Wawel Castle (WWL); and prices of the following commodities: crude oil (SC), silver (XAG) and gold (XAU); using the following formula:

$$x_t = \ln s_t - \ln s_{t-1} \tag{19}$$

where: s_t – an observation of time series, traded in the period 3.01.2000 r. – 26.08.2013 r. Data come from the archive file soft the website stooq.com.

In the first stage of the study, we estimated the parameters of the state space reconstruction for the selected time series using the delays method: the time delay τ was estimated by means of the autocorrelation function *ACF* and the embedding dimension d – with the false nearest neighbor method *FNN* (Table 1). Then, the analyzed time series underwent the process of random noise reduction by the nearest neighbors method for the estimated embedding dimension d and the delay time $\tau = 1$. The filtered time series were designated with the symbol NameofTimeSeries_red and for those time series we also carried out the reconstruction of state space. Table 1 gives the parameters of the reconstruction d and τ for the analyzed time series before and after filtration.

Table 1. The parameters of state space reconstruction for the analyzed time series

Time series	τ	d	Time series	τ	d	Time series	τ	d
CHF	10	7	WIG	16	7	NKX_red	3	6
EUR	21	8	WWL	5	7	SC_red	21	6
ING	6	8	XAG	4	8	SPX_red	9	6
JPY	2	6	XAU	22	7	USD_red	25	7
MSZ	7	8	ZWC	17	10	VST_red	18	7
NKX	6	6	CHF_red	22	6	WIG_red	8	7
SC	2	6	EUR_red	5	8	WWL_red	3	7
SPX	15	7	ING_red	4	8	XAG_red	7	8
USD	23	6	JPY_red	23	8	XAU_red	16	7
VST	19	9	MSZ_red	2	7	ZWC_red	3	7

Source: own work.

In the next stage of research we estimated forecasts for the selected time series using the nearest neighbor method – NNM – and the method based on the value of the largest Lyapunov exponent – LEM. In order to determine the forecast using the NNM method, we took into account the $k = 2(d+1)$ nearest neighbors of the point x_n^d, assuming the weight of the i-th neighbor was an arithmetic average of the first coordinates of the nearest neighbors (NNM_A), the weight given by the formula (9) – NNM_B, equation (10) – NNM_C and formula (11) – NNM_D. The assessment of designated forecasts was made with the following metrics: d – the average forecast error ME, q – the average absolute forecast error MAE, σ-root mean square error RMSE, and I – Theil coefficient. Tables 2 and 3 show the prediction errors over the entire verification range for the forecast horizon equal to 10, obtained by the NNM method in four above cases estimating the weights of the nearest neighbors.

Table 4 shows the prediction errors over the entire verification range for the forecast horizon equal to 10, obtained by the LEM method for revalued (LEM+) and undervalued (LEM–) forecasts.

Analyzing the results obtained using methods NNM_A, NNMB, NNM_C and NNM_D (Tables 2 and 3), it can be seen that for the majority of analyzed time series the errors (the mean absolute forecast error q and the root mean squared error σ) obtained for the time series in which the reduction of noise was used are lower than for the forecasts obtained for the original time series. The only exception is a series of ING for all versions of NNM method. The values of Theil coefficient after applying noise reduction using the nearest neighbors method are also lower for most of the analyzed time series. The exceptions are the time series JPY, NKX, SC, USD, XAG for methods MMN_A, MMN_B and MMN_D and time series JPY, NKX, VST for MMN_C methods.

Based on the data in Table 4 it can be concluded (as in the case of forecasts obtained by the nearest neighbors method), that the reduction of random noise allowed to improve the accuracy of the obtained predictions. The ex-post errors of the time series which were filtered by the nearest neighbor method are lower than the forecasts obtained for the unfiltered series. The exceptions are the time series JPY, SC, SPX and XAG for the method LEM+ and JPY, NKX and SPX for the method LEM–, for which the reductions increased the mean absolute forecast error q and the root mean square error σ. After applying random noise reduction for more than 50% of the time series, the values of Theil coefficient estimates obtained by methods LEM+ and LEM– are also lower. The exceptions are CHF, JPY, MFA, SPX, WWL, XAG for the method LEM+, and JPY, SPX, USD, WWL for the method LEM–.

Comparing the results obtained using the largest Lyapunov exponent method (Table 4) it can be seen that in the entire verification range the overestimated forecasts proved to be more accurate for most of the analyzed series. However, the analysis of forecast errors received by NNM methods indicates that forecasts using the weighted average of the first coordinates of the nearest neighbors (MMN_B, MMN_C and MMN_D) proved to be more accurate than forecasts based on the arithmetic mean of the first coordinates (method MMN_A) for most of the analyzed time series.

Table 2. The forecast errors received by the NNM_A method and the NNM_B method or analyzed time series

NNM_A	*d*	*q*	*σ*	*I*	NNM_B	*d*	*q*	*σ*	*I*
CHF	0.00046	0.00377	0.00451	1.28887	CHF	0.00070	0.00338	0.00399	1.01102
CHF_red	–0.00421	0.00239	0.00280	0.81978	CHF_red	–0.00104	0.00262	0.00305	0.97768
EUR	0.00164	0.00363	0.00415	1.46383	EUR	0.00102	0.00306	0.00347	1.02000
EUR_red	–0.00407	0.00174	0.00244	1.07792	EUR_red	–0.00104	0.00175	0.00234	0.99435
ING	0.00050	0.01162	0.01376	0.98714	ING	–0.00056	0.01084	0.01381	0.99478
ING_red	0.04213	0.01132	0.01585	0.87084	ING_red	0.00944	0.01216	0.01692	0.99218
JPY	–0.00045	0.00462	0.00549	0.74514	JPY	–0.00162	0.00515	0.00628	0.97331
JPY_red	0.00738	0.00391	0.00477	0.86469	JPY_red	0.00091	0.00404	0.00510	0.98846
MSZ	–0.00265	0.02966	0.03451	1.63511	MSZ	–0.00385	0.02294	0.02731	1.02420
MSZ_red	–0.01651	0.01426	0.01585	1.20701	MSZ_red	–0.00271	0.01290	0.01449	1.00767
NKX	0.00067	0.01416	0.01783	1.19776	NKX	0.00036	0.01376	0.01636	1.00850
NKX_red	–0.01362	0.01526	0.01796	1.28950	NKX_red	–0.00200	0.01332	0.01594	1.01544
SC	0.00249	0.00695	0.00834	1.62015	SC	0.00284	0.00550	0.00662	1.02092
SC_red	–0.02343	0.00784	0.00895	1.86489	SC_red	–0.00287	0.00561	0.00669	1.04234
SPX	–0.00073	0.00614	0.00719	1.21326	SPX	–0.00161	0.00553	0.00655	1.00820
SPX_red	–0.00258	0.00244	0.00309	0.81436	SPX_red	0.00046	0.00256	0.00337	0.97249
USD	–0.00018	0.00477	0.00546	0.99715	USD	0.00047	0.00482	0.00546	0.99578
USD_red	0.00481	0.00315	0.00387	1.28364	USD_red	–0.00031	0.00300	0.00341	0.99712
VST	0.01359	0.02212	0.02977	1.30056	VST	0.01257	0.01853	0.02626	1.01211
VST_red	0.08552	0.01835	0.02129	1.14886	VST_red	0.01442	0.01642	0.01992	1.00602
WIG	0.00362	0.00838	0.00933	1.40097	WIG	0.00219	0.00661	0.00794	1.01468
WIG_red	0.01741	0.00525	0.00630	0.85292	WIG_red	0.00421	0.00567	0.00677	0.98727
WWL	0.00925	0.02201	0.02652	1.33293	WWL	0.00797	0.02000	0.02315	1.01551
WWL_red	0.00011	0.00651	0.00848	1.23757	WWL_red	0.00213	0.00526	0.00764	1.00543
XAG	0.01074	0.01440	0.02006	0.87473	XAG	0.01268	0.01530	0.02136	0.99165
XAG_red	0.02538	0.00827	0.01042	1.01219	XAG_red	0.00565	0.00809	0.01033	0.99432
NS A	0.00357	0.00765	0.00942	0.78543	XAU	0.00478	0.00885	0.01053	0.98086
XAU_red	0.00547	0.00654	0.00767	0.69056	XAU_red	0.00236	0.00754	0.00912	0.97717
ZWC	–0.00080	0.00709	0.00923	1.39440	ZWC	0.00003	0.00549	0.00787	1.01365
ZWC_red	0.01773	0.00615	0.00716	1.00762	ZWC_red	0.00408	0.00624	0.00713	0.99834

Source: own work.

Table 3. The forecast errors received by the NNM_C method and the NNM_D method for analyzed time series

NNM_C	*d*	*q*	*σ*	*I*	NNM_D	*d*	*q*	*σ*	*I*
CHF	0.000706	0.003387	0.003993	1.011116	CHF	0.00070	0.00338	0.00399	1.01099
CHF_red	–0.005225	0.002612	0.003056	0.979940	CHF_red	–0.00104	0.00262	0.00305	0.97764
EUR	0.001014	0.003058	0.003463	1.018249	EUR	0.00102	0.00306	0.00347	1.02007
EUR_red	–0.005196	0.001762	0.002350	1.000334	EUR_red	–0.00104	0.00175	0.00234	0.99432
ING	–0.000595	0.010851	0.013828	0.996686	ING	–0.00056	0.01084	0.01381	0.99476
ING_red	0.047111	0.012150	0.016902	0.990035	ING_red	0.00944	0.01216	0.01692	0.99219
JPY	–0.001602	0.005132	0.006251	0.964391	JPY	–0.00162	0.00515	0.00628	0.97342
JPY_red	0.004532	0.004039	0.005114	0.992766	JPY_red	0.00091	0.00404	0.00510	0.98845
MSZ	–0.003852	0.022896	0.027231	1.018090	MSZ	–0.00385	0.02294	0.02731	1.02420
MSZ_red	–0.013354	0.012857	0.014388	0.994018	MSZ_red	–0.00271	0.01290	0.01449	1.00781
NKX	0.000307	0.013744	0.016351	1.007477	NKX	0.00036	0.01376	0.01636	1.00851
NKX_red	–0.009873	0.013292	0.015896	1.010276	NKX_red	–0.00200	0.01332	0.01594	1.01544
SC	0.002869	0.005551	0.006671	1.036627	SC	0.00283	0.00550	0.00662	1.02066
SC_red	–0.014235	0.005545	0.006593	1.010972	SC_red	–0.00287	0.00561	0.00670	1.04285
SPX	–0.001621	0.005514	0.006560	1.009904	SPX	–0.00161	0.00553	0.00655	1.00822
SPX_red	0.002331	0.002566	0.003375	0.972676	SPX_red	0.00046	0.00256	0.00337	0.97242
USD	0.000434	0.004823	0.005470	1.000541	USD	0.00047	0.00482	0.00546	0.99573
USD_red	–0.001414	0.002989	0.003407	0.994127	USD_red	–0.00031	0.00300	0.00341	0.99716
VST	0.012458	0.018354	0.026084	0.998683	VST	0.01257	0.01853	0.02626	1.01211
VST_red	0.071990	0.016383	0.019902	1.003802	VST_red	0.01442	0.01642	0.01992	1.00603
WIG	0.002155	0.006593	0.007912	1.007598	WIG	0.00219	0.00661	0.00794	1.01481
WIG_red	0.021128	0.005652	0.006764	0.984419	WIG_red	0.00421	0.00567	0.00677	0.98732
WWL	0.007942	0.019974	0.023105	1.011965	WWL	0.00797	0.02000	0.02315	1.01555
WWL_red	0.010697	0.005312	0.007650	1.008235	WWL_red	0.00213	0.00526	0.00764	1.00541
XAG	0.012709	0.015252	0.021341	0.990060	XAG	0.01268	0.01530	0.02136	0.99166
XAG_red	0.028341	0.008065	0.010285	0.985645	XAG_red	0.00565	0.00809	0.01033	0.99438
XAU	0.004838	0.008844	0.010554	0.985202	XAU	0.00478	0.00885	0.01053	0.98079
XAU_red	0.011830	0.007539	0.009111	0.974574	XAU_red	0.00236	0.00754	0.00912	0.97720
ZWC	0.000033	0.005491	0.007884	1.018457	ZWC	0.00003	0.00549	0.00787	1.01362
ZWC_red	0.020454	0.006240	0.007134	0.998884	ZWC_red	0.00408	0.00624	0.00713	0.99831

Source: own work.

Table 4. The forecast errors received by the LEM+ method and the LEM– method for analyzed time series

LEM+	*d*	*q*	*σ*	*I*	LEM–	*d*	*q*	*σ*	*I*
CHF	–0.02084	0.02084	0.02266	32.56648	CHF	0.02277	0.02338	0.02839	51.11110
CHF_red	–0.08416	0.01683	0.01897	37.75099	CHF_red	0.01452	0.01452	0.01531	24.59909
EUR	–0.01767	0.01767	0.02004	34.10042	EUR	0.02004	0.02004	0.02138	38.82504
EUR_red	–0.03694	0.00865	0.00960	16.70128	EUR_red	0.00735	0.00735	0.00830	12.49471
ING	–0.06339	0.06540	0.07618	30.25237	ING	0.06341	0.06406	0.07216	27.14216
ING_red	–0.13304	0.03186	0.03877	5.20789	ING_red	0.04657	0.04657	0.05442	10.26355
JPY	–0.01309	0.01309	0.01684	7.00140	JPY	0.01291	0.01385	0.01798	7.98132
JPY_red	–0.11626	0.02387	0.02709	27.86056	JPY_red	0.02887	0.02887	0.03128	37.15900
MSZ	–0.09000	0.09607	0.11544	18.29636	MSZ	0.09411	0.10494	0.11259	17.40529
MSZ_red	–0.35436	0.07087	0.08955	38.50319	MSZ_red	0.02512	0.03758	0.05510	14.57891
NKX	–0.04457	0.04589	0.05244	10.36302	NKX	0.03644	0.03691	0.04366	7.18380
NKX_red	0.19625	0.04057	0.04967	9.86250	NKX_red	0.03536	0.04920	0.05735	13.14906
SC	–0.03068	0.04352	0.05350	66.67117	SC	0.05229	0.05229	0.06372	94.58058
SC_red	0.14463	0.04177	0.05321	65.84891	SC_red	0.02347	0.04705	0.05570	72.16218
SPX	–0.01790	0.01790	0.02018	9.55222	SPX	0.01318	0.01462	0.01760	7.26509
SPX_red	–0.09880	0.01976	0.02235	42.65151	SPX_red	0.01895	0.01895	0.02149	39.45796
USD	–0.02975	0.02975	0.03384	38.29524	USD	0.02494	0.02494	0.02617	22.89498
USD_red	–0.07094	0.01514	0.01715	25.18411	USD_red	0.02331	0.02331	0.02446	51.25740
VST	–0.08730	0.08730	0.09813	14.13341	VST	0.10372	0.10372	0.11381	19.01144
VST_red	–0.20142	0.04028	0.04556	5.26004	VST_red	0.07433	0.07433	0.07628	14.74526
WIG	–0.02804	0.02833	0.03138	15.84918	WIG	0.02560	0.02560	0.03135	15.81507
WIG_red	–0.09255	0.01883	0.01990	8.52458	WIG_red	0.02136	0.02136	0.02220	10.60354
WWL	–0.06349	0.07867	0.08377	13.30231	WWL	0.08527	0.08813	0.09729	17.94371
WWL_red	–0.18440	0.04005	0.04815	39.93762	WWL_red	0.04110	0.04110	0.04572	36.01424
XAG	–0.04202	0.04224	0.04916	5.25412	XAG	0.06711	0.06994	0.07912	13.60913
XAG_red	–0.16008	0.04783	0.05011	23.39302	XAG_red	0.04754	0.04754	0.05247	25.64752
XAU	–0.03529	0.03529	0.04089	14.78573	XAU	0.05239	0.05239	0.06104	32.95746
XAU_red	–0.14021	0.02804	0.03213	12.12172	XAU_red	0.03181	0.03181	0.03353	13.19534
ZWC	–0.04220	0.04220	0.04477	32.84296	ZWC	0.03145	0.04138	0.04343	30.90992
ZWC_red	–0.03598	0.01033	0.01256	3.09854	ZWC_red	0.01498	0.01525	0.01894	7.03986

Source: own work.

Conclusions

In the paper we studied the effect of random noise reduction by the nearest neighbors method on the accuracy of the forecasts of selected financial time series. The research results show that for the majority of analyzed time series the ex-post forecast errors obtained for the series that used noise reduction are much lower than the forecasts obtained for the unfiltered series.

In addition, basing on the selected financial time series we compared two methods of forecasting: the nearest neighbor method (four versions) and the largest Lyapunov exponent method (two versions). The research showed that three versions (B, C, D) of the nearest neighbor method using a weighted average to estimate forecasts were the most effective. These forecasts were characterized with the smallest values of forecast errors for the majority of the analyzed financial time series.

It should be noted that the values of forecasts determined by these methods to a large extend depend on the adopted metric, the weights of the nearest neighbors, the values of parameters of the reconstructed state space and the number of nearest neighbors. Thus, it seems that in order to improve the quality of the forecasts, additional studies should be performed with changed parameters.

Notes

1 Nowiński (2007), p. 24.
2 Stawicki (1993).
3 Takens (1981).
4 Abarbanel et al. (1992)
5 Ibidem, s. 51; Nowiński (2007), pp. 248–249.
6 Orzeszko (2005).
7 Zawadzki (1996).
8 Guégan, Leroux (2009), p. 2401; Zhang et al. (2004), p. 3.
9 Zawadzki (1996).
10 Kantz, Schreiber (2004), p. 192.
11 Guégan, Leroux (2009), p. 2402.
12 Miśkiewicz-Nawrocka (2012).
13 Zhang (2004), p. 3; Guégan, Leroux (2009), p. 2402.

References

Abarbanel H.D., Brown, R. & Kennel, M.B. (1992). Determining Embedding Dimension for Phase Space Reconstruction Using a Geometrical Construction. *Physical Review A*, 45 (6), 3404–3411, DOI: 10.1103/PhysRevA.45.3403.

Cao, L. (2001). *Method of false nearest neighbors*. Soofi A.S., Cao L. (Eds.), *Modeling and Forecasting Financial Data*. Boston: Kluwer.

Guégan, D. & Leroux, J. (2009). Forecasting chaotic systems: The role of local Lyapunov exponents. *Chaos, Solitons & Fractals*, 41, 2401–2404. DOI: 10.1016/j.chaos.2008.09.01.

Kantz, H. & Schreiber, T.(2004). *Nonlinear time series analysis* (second edition). Cambridge: Cambridge University Press.

Miśkiewicz-Nawrocka, M. (2012). *Zastosowanie wykładników Lapunowa do analizy ekonomicznych szeregów czasowych*. Katowice: Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach.

Nowiński, M. (2007). *Nieliniowa dynamika szeregów czasowych*. Wrocław: Wydawnictwo Akademii Ekonomicznej we Wrocławiu.

Orzeszko, W. (2005). *Identyfikacja i prognozowanie chaosu deterministycznego w ekonomicznych szeregach czasowych*. Warszawa: Polskie Towarzystwo Ekonomiczne.

Ramsey, J.B., Sayers, C.L.& Rothman, P. (1990). The Statistical Properties of Dimension Calculations Using Small Data Sets: Some Economic Applications, *International Economic Review*, 31 (4), pp. 991–1020, DOI: 10.2307/2527026.

Stawicki, J. (1993). *Metody filtracji w modelowaniu procesów ekonomicznych*. Toruń: Wydawnictwo Uniwersytetu Mikołaja Kopernika.

Takens, F. (1981). *Detecting strange attractors in turbulence*. D.A. Rand, L.S. Young (Eds.), *Lecture Notes in Mathematics* (pp. 366–381), Berlin: Springer.

Zawadzki, H. (1996). *Chaotyczne systemy dynamiczne*. Katowice: Wydawnictwo Akademii Ekonomicznej w Katowicach.

Zhang, J., Lam, K.C., Yan, W.J., Gao, H. & Li, Y (2004). Time series prediction using Lyapunov exponents in embedding phase space. *Computers and Electrical Engineering*, 30, 1–15, DOI: 10.1109/ICOSP.1998.770189.