The New Method of Measuring the Effects of Noise Reduction in Chaotic Data,

***Chaos Solitons and Fractals*, vol. 38, 2008, 1355-1368.**

The New Method of Measuring the Effects of Noise Reduction in Chaotic Data

WITOLD ORZESZKO

Nicolaus Copernicus University in Torun, Torun, Poland

E-mail: vit@uni.torun.pl

Abstract The presence of a noise, which is typical for real data, makes methods of chaotic signals analysis much more difficult to apply to. That is why algorithms of noise reduction in chaotic time series have been recently developed. A lot of existing algorithms require setting values of specified parameters and in consequence lead to many outputs. Thus one must additionally apply a supporting method which allows to indicate a "proper" output. In this paper such a new method is proposed and examined. As an example, the presented method is applied to support the Nearest Neighbours algorithm to reduce the noise in the time series from the Warsaw Stock Exchange. Next the cleaned data are investigated for the presence of chaos.

Keywords: Chaos, noisy chaos, noise reduction, Lyapunov exponent.

1. INTRODUCTION

Chaos theory has become a new approach to an analysis of economic processes. It deals with deterministic systems which due to complicated dynamics appear to be random. In consequence chaotic time series seem to be irregular but in fact they are generated by deterministic rules, which make them predictable in a short run. However, due to sensitive dependence on initial conditions long-run predictions of chaotic time series are strongly limited.

Distinguishing random dynamics from chaotic one is not an easy task and requires special methods. An empirical evidence of chaos in real data has been reported by various researchers. The most promising results have been obtained for financial time series, *e.g.* stock prices and indices, exchange rates or futures prices (see *e.g.*: [1], [4], [10], [20], [29], [26]).

The presence of a noise is typical for economic data. It makes chaos detection and forecasting much more difficult. That is why methods of chaotic time series analysis have been improved, to become more resistant to a noise. Simultaneously methods of noise reduction have been developed (*e.g.* [9], [12], [16], [19], [25]). Many of them lead to different outputs depending on considered values of specific parameters. Thus it is necessary to apply a supporting procedure verifying *a posteriori* the results of noise reduction. The task for such a procedure is to indicate which one from the obtained outputs is the best, *i.e.* the most similar to the original noise-free signal[1]. Since noise-free signals are usually unknown in practice, it must work without this information. Such a new method based on the new measure of a noise level is proposed in this paper. This technique is very easy to apply to and can be applied to noisy multivariate states or, by considering delay vectors, to noisy time series.

This paper is organized as follows. Section 1 contains the introduction. In section 2 noisy chaos is defined. The idea of noise reduction is described in section 3. In section 4 the new method of determining a level of a noise is introduced. The results of numerical experiments verifying this method are presented in section 5. Next the proposed technique is

[1] Some examples of such procedures are presented in *e.g.* [18].

applied to the time series from the Warsaw Stock Exchange. The results of this research and the largest Lyapunov exponent estimates are presented in section 6. The last section is a short summary.

2. NOISY CHAOS

In the modern econometric approach the idea of random forces in economic phenomena is fundamental and has made stochastic modeling dominant (see [2]). According to this approach, a stochastic term is regarded in chaos theory by introducing a class of models called “noisy chaos”.

The idea of noisy chaos is to distinguish two parts in a signal: the deterministic (chaotic) and the random ones. By the definition, a time series (x_t) has a smooth system explanation with a noisy observer and noisy law of motion, if there is a C^2 function $h: R^n \times R \to R$ and if there exists a C^2 function $f: R^n \times R \to R^n$ such that for all *t*:

$$s_t = f(s_{t-1}, \varepsilon_t), \tag{1}$$

$$x_t = h(s_t, \eta_t), \tag{2}$$

where $s_t \in R^n$ is a state of the *n*-dimensional dynamical system at the *t*-th moment, (η_t) and (ε_t) are independently and identically distributed stochastic processes (see [5]).

The typical sources of the observational noise (η_t) in economic time series are measurement errors, whereas the dynamical noise (ε_t) usually represents external influences perturbing systems. These factors are typical for economic phenomena, which makes introduction of noisy chaos to economics fully plausible.

The properties of noisy signals depend on a relative force of the random and deterministic part. Its measure is the signal-to-noise ratio (*SNR*) which is a ratio of the variance (or standard deviation) of the noise-free signal and that of the pure-noise signal. *SNR* greater than about 1000 can be regarded as a low noise and less than about 10 as a high noise (see [24]).

Due to sensitive dependence on initial conditions, even a very weak dynamical noise radically changes the evolution of chaotic systems. However, a weak noise may not alter qualitative properties of trajectories, since in noisy, chaotic systems fundamentality of chaos is remained (see [7], [22]), i.e.:

- erratic dynamics is caused by nonlinear interactions between endogenous forces (represented by deterministic part of a signal),
- systems are sensitive dependent on initial conditions, which implies that long-run predictions are strongly limited,
- systems are predictable in a short run,
- in dissipative systems an attracting set is remained (it looks like a "thicker" version of the noise-free attractor, *i.e*. for low resolution these two objects are indistinguishable).

On the other hand it should be marked that the presence of a noise may make methods of chaotic data analysis much more difficult to apply to or, sometimes, even useless (see [6], [11], [30]). Unfortunately, most of the existing methods are sensitive to the presence of a noise and that is why chaotic dynamics with a strong noise has been – so far – practically indistinguishable from truly random data (see [18], [28]).

3. NOISE REDUCTION

While analyzing real data a researcher must take into account the presence of a noise, which implies that investigated time series (x_t) can be decomposed as:

$$x_t = y_t + \varepsilon_t, \tag{3}$$

where (ε_t) is supposed to be a random noise with at least fast decaying autocorrelations and no correlation with the noise-free signal (y_t) (see [18]).

The aim of noise reduction is to recover the signal (y_t) based on the noisy data (x_t). The first step in this process is to establish a criterion for distinguishing (y_t) from (ε_t). The classical statistical tool providing this distinction is the spectral analysis, but this approach fails for chaotic data (see *e.g.* [4]). In chaos theory "randomness" is used to be associated with "high-dimensionality", "determinism" with "low-dimensionality" and the tool obtaining this distinction becomes correlation dimension (see [13]).

The issue of noise reduction is strictly associated with prediction, so methods of chaos forecasting are used to noise reduction. Very simple but simultaneously efficient prediction and noise reduction technique is the Nearest Neighbour method (*NN* hereafter) (see [18]). According to this method, to determine the value y_n one should form delay vectors $\hat{x}_t^m = (x_t, x_{t-1}, ..., x_{t-m+1)})$ and choose one of them with the coordinate x_n in the middle. For example, when *m* is even it can be the vector $\hat{x}_{n+m/2}^m$. Let $\hat{x}_{l_1+m/2}^m$, $\hat{x}_{l_2+m/2}^m$, ... , $\hat{x}_{l_k+m/2}^m$ denote in such a case *k* the nearest (in a sense of any *m*-dimensional metrics) neighbours of the vector

$\hat{x}^m_{n+m/2}$.[2] Then a value $\tilde{y}_n$ which denotes an estimate of y_n may be computed from the following formula:

$$\tilde{y}_n = \frac{1}{k}\sum_{i=1}^{k} x_{l_i} \ . \qquad (4)$$

4. MEASURING THE EFFECTS OF NOISE REDUCTION

A very important issue referring to noise reduction is to measure its effectiveness. Usually in noise reduction methods values of specified parameters must be set *a priori*, so each method may lead to many outputs. When the original noise-free signal (y_t) is known, to determine the best output $(\tilde{y}_t)$ one may calculate a distance between (y_t) and $(\tilde{y}_t)$, defined as:

$$e = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\tilde{y}_t - y_t)^2} \ , \qquad (5)$$

where N is the length of the time series.

The issue is much more difficult when (y_t) is unknown. The new method proposed in this paper may be used in such a case. Its idea refers to an observable impact of an added noise on geometry of attractors[3]. It can be seen that the added noise makes attractors "thicker", *i.e.* an average distance between close points of the attracting sets is getting bigger. As an example, this effect is illustrated for the time series generated from the Henon system[4],

[2] The amount of the nearest neighbours – k is arbitrarily set. However one can reconstruct the algorithm by considering a radius of a neighborhood instead.
[3] Thus in principle the method is constructed to work with dissipative systems.
[4] The Henon system is defined in section 5.

with the observational noise of different amplitude.[5] To determine the noise level the ratio $SNR = \frac{\sigma_{Henon}}{\sigma_{noise}}$ was used[6]. The attractors reconstructed from each noisy time series by the technique of delay coordinates are presented in Figures 2a–5a.

The proposed method of measuring the effects of noise reduction consists in calculating the quantity *NRL* (*Noise Reduction Level*), which measures the level of a noise in data by regarding the property presented above.[7] In order to indicate the output which simultaneously is not distorted in comparison with the original noise-free signal, *NRL* consists of the two parts. The first one is the measure of mentioned "thickening", defined as $d_{\min} = \frac{1}{T}\sum_{i=1}^{T} d_i$, where *T* denotes the amount of approachable states (delay vectors) and d_i denotes a distance between *i*-th cleaned state (delay vector) and its closest neighbour. The biggest value of a relative decrease of $d_{\min}$ (in comparison to analyzing noisy data) indicates an output with the lowest level of a noise. However, considering only this factor is not sufficient, because such a procedure would always determine a time series with equal observations[8]. Thus the quantity *NRL* includes the second part – the value $\left|\frac{diam - diam^0}{diam^0}\right|$, where $diam^0$ and $diam$ denote maximal distances between states (delay vectors), respectively, before and after noise reduction. This part measures a relative change of a diameter of the attractor and may be interpreted as a level of a distortion, caused by noise reduction. Finally *NRL* is defined by the following formula:

[5] Pseudorandom numbers with uniform distribution were used in the research.

[6] Symbols σ_{Henon} and σ_{noise} denote standard deviations of, respectively, the noise-free data and the noise.

[7] As it was mentioned before the method may be used to multivariate states or, by considering delay vectors, to univariate time series.

[8] In the Nearest Neighbours method such an output is obtained for big values of the *k* parameter.

$$NRL = \frac{d_{\min} - d_{\min}^0}{d_{\min}^0} + \left| \frac{diam - diam^0}{diam^0} \right| \qquad (6)$$

According to the proposed method, a time series with the lowest level of the quantity *NRL* should be chosen from the obtained outputs of noise reduction.

As it is clearly seen, when delay vectors are being analyzed the quantity *NRL* depends on the considered value of the embedding dimension *m*. According to the Takens theorem an attractor may be properly reconstructed if $m > 2d$, where d denotes a dimension of the system. Thus the method is not expected to work properly for values of *m* which do not satisfy the Takens condition.

5. EXPERIMENTAL RESULTS

To examine the proposed method chaotic time series of length 2000 generated by the Henon map, logistic map and Lorenz system are used. These series are defined as follows:

a) the logistic map $f(x_t) \equiv x_{t+1} = 4x_t(1 - x_t)$ with the initial point $x_0 = 0{,}7$,

b) first coordinates of states from the Henon system generated by the equation $H(x_t, y_t) \equiv (x_{t+1}, y_{t+1}) = (1 + y_t - 1{,}4x_t^{\,2};\ 0{,}3x_t)$, with $(x_0, y_0) = (0{,}9, 0{,}9)$,

c) the Lorenz map, described by the set of differential equations:

$$\frac{dx}{dt} = 16(y - x)$$
$$\frac{dy}{dt} = -xz + 45{,}92x - y$$
$$\frac{dz}{dt} = xy - 4z$$

with $(x(0), y(0), z(0)) = (1,1,1)$. The analyzed time series was generated from the formula $x_t = x(t \cdot 0{,}01)$, for each $t \in N_{\geq 1}$.

The observational noise with $SNR = 50, 25, 10, 1$ was added to the each time series. Next the Nearest Neighbours method with $m = 5$ was used to reduce the noise from the obtained data.[9] The following values of a radius of neighbourhoods were considered: $r = 0.001$, 0.01, 0.1, 1 for the Henon and logistic map and $r = 0.02$, 0.2, 2, 20 – in case of Lorenz system[10].

At first, the time series generated by the Henon map was analyzed. In Figures 1–5 the attractors reconstructed from the data before and after noise reduction are presented[11].

[Figure 1]

[Figure 2a] [Figure 2b]

[Figure 3a] [Figure 3b]

[Figure 4a] [Figure 4b]

[Figure 5a] [Figure 5b]

In Tables 1–3 the computed values of the quantity $NRL(m')$ for $m' = 1, 2, 3, 5, 7, 10, 15$ are presented.[12] Since the method does not use information about the true noise-free signal (y_t), to verify its usefulness it is compared with values of the actual distances *e*. Each cell in the table contains the calculated value of *NRL* (at the top) and the relative change of the

[9] To apply the *NN* method software TISEAN (routine nrlazy.exe) created by H. Kantz and T. Schreiber was used. In this routine *m*=5 is a default value.

[10] The bigger values of *r* for the Lorenz system allow remaining the similar relation between *r* and the standard deviation of the investigated time series.

[11] Figures 2b, 3b, 4b and 5b present attractors reconstructed from the outputs for which the quantity *e* is the smallest (see Tables 1–3).

[12] The symbol *m*' denoting an embedding dimension of delay vectors considered in calculations of *NRL* is introduced to avoid confusion with the *m* parameter in the Nearest Neighbour method.

diameter – $\left|\frac{diam - diam^0}{diam^0}\right|$ (at the bottom, in brackets). For each *m*’ the smallest value of *NRL*, indicating “the best” output, is bolded. In these simulations only the *r* parameter was being changed, thus here the statements: “the best output” and “the best value of *r*” are equivalent.

[Table 1. Quantities *NRL*(*m*’) [%] and *e* for the Henon map]

[Table 2. Quantities *NRL*(*m*’) [%] and *e* for the logistic map]

[Table 3. Quantities *NRL*(*m*’) [%] and *e* for the Lorenz system]

As it can be seen from Tables 1–3, for most values $m' > 1$ the proposed method indicated the proper (in terms of the actual, “unknown” distance *e*) output. Its ineffectiveness for $m' = 1$ was expected, since this value does not satisfy the Takens criterion. Thus it can be summarized that the method led to wrong conclusions only for the Henon and Lorenz time series with *SNR*=10, and additionally can be misleading for big values of m'. Problems with a proper indication for *SNR*=10 may be avoided by the simultaneous inspection of the relative change of a diameter, since too big distortion in indicated data should be generally regarded as “suspicious” and can be justified only for noisy data with the strong random noise. However the presence of the strong noise may make true dynamics of noise-free signal uncovered at all and in consequence the applied procedure of noise reduction – useless. Thus choosing outputs with a high level of distortion seem to be doubtful.

6. APPLICATION TO REAL DATA

In this section the procedure of noise reduction is applied to daily log returns of the Warsaw Stock Exchange Index – WIG from 3.10.1994–18.05.2006 (2907 observations). WIG – the main index of the Polish stock market – reflects prices of all the quoted shares. Although the Warsaw Stock Exchange was developed in 1991, it was not until October 1994 that trading has taken place everyday. Therefore for data before that moment the basic condition of data homogeneity is not satisfied.

The Nearest Neighbours method with $m = 1, 2, 3, 5, 7, 10, 15$ was applied to reduce the noise in the investigated time series. For each *m* eight values of the *r* parameter were considered (see Table 4).

[Table 4. Values of the *r* parameter used in the *NN* method]

For each output (*i.e.* for every values of the *r* and *m* parameters) the quantity *NRL* with $m' = 1, 2, 3, 5, 7, 10, 15$ was calculated. The obtained results are presented in Table 8 in the appendix. For each *m*' the six smallest values of *NRL* are bolded.

Additionally, to compare degrees of the outputs' distortion the relative changes of the diameter $\Delta diam \equiv \left| \frac{diam - diam^0}{diam^0} \right|$ are presented in Table 9.

An analysis of the obtained values of *NRL* almost univocally (for $m' > 1$) indicates six outputs. These are, in turn, the outputs corresponding to the following sets of the parameters $(m'; r)$: $(15; 0.05)$, $(10; 0.05)$, $(7; 0.05)$, $(5; 0.05)$, $(5; 0.025)$, $(7; 0.025)$. As it is seen from Table 9 the outputs from the first four places of this rank are quite heavily distorted, meanwhile the two latest ones are moderately distorted. However, to regard representative outputs in respect of the level of the distortion in further research, we decided to reject the outputs from the third and fourth positions (*i.e.* $(7;0.05)$ and $(5;0.05)$) and include one additional output with a small degree of the distortion (*i.e.* $\Delta diam \approx 0$) in considerations instead. According to the results from Tables 8 and 9 the output obtained for the parameters $m' = 7$, $r = 0.01$ has been chosen.

Finally the outputs summarized in Table 5 have been chosen to further research.

[Table 5. Outputs of the *NN* method chosen to further research]

Next, the largest Lyapunov exponent has been calculated to detect chaotic dynamics in the considered data.

Lyapunov exponents measure an average rate of converging or separating nearby points of the system. The presence of at least one positive exponent means a sensitive dependence on initial conditions and may be interpreted as a measure of chaos. Positivity of the Lyapunov exponent can be used as an operational definition of chaos (see *e.g.* [14], [20]).

In this paper the algorithm of determining the largest Lyapunov exponent introduced independently by Rosenstein et al. and Kantz has been applied. It proceeds as follows (see [18], [24]):

1. For each delay vector $\hat{x}_i^m = (x_i, x_{i-1}, ..., x_{i-m+1})$, $i = m, m+1, ..., N$ a set O_i which consists of k nearest neighbours $\hat{x}_{i_j}^m$ of the vector $\hat{x}_i^m$ is determined. To avoid the situation in which $\hat{x}_i^m$ and $\hat{x}_{i_j}^m$ are situated at the same trajectory, the condition $|i - i_j| > t^*$ for fixed t^* may be added[13].

2. For each $i = m, m+1, ..., N - n_{\max}$ and $n = 1, ..., n_{\max}$ one calculates the value:

[13] In this paper $t^*=10$, $k=1$, $n_{max}=5$ and the Euclidean distance were arbitrarily set.

$$d_n(i) = \frac{1}{k} \sum_{\hat{x}_{i_j}^m \in O_i} \left| x_{i+n} - x_{i_j+n} \right|, \tag{7}$$

where $n_{\max}$ is the fixed number of iterations while the divergence of states is analyzed.

3. The largest Lyapunov exponent is calculated using least-squares fit to the equation:

$$\ln(d_n) = \ln(d_0) + \lambda n, \tag{8}$$

where $d_n = \frac{1}{T}\sum_{i=1}^{T} d_n(i)$ denotes an average over the all delay vectors.

The algorithm allows to test for the presence of an exponential divergence between initially nearby trajectories. Only if for some range of n does the function $\ln(d_n)$ exhibit linear increase, its slope may be interpreted as an estimate of the largest Lyapunov exponent.

The results of the largest Lyapunov exponent estimates for $m = 1, 2, 3, 5, 7, 10, 15$ are summarized in Table 6.

[Table 6. Results of the largest Lyapunov exponent estimates]

Next the significance of the estimated exponents has been verified. To this end the blockwise bootstrap method has been applied (see [3]). In the test the hypotheses are formulated as:

$$H_0 : \lambda = 0 \quad vs. \quad H_1 : \lambda > 0, \tag{9}$$

what means that under the null hypothesis the investigated time series is not generated by chaotic system.

The blockwise bootstrap method consists of the following steps:

1. From *N*-point time series form delay vectors $\hat{x}_i^m$, $i = m, m+1, ..., N$ and estimate the largest Lyapunov exponent $\hat{\lambda}$,
2. Resample with replacement *k* delay vectors ξ_i, $i = 1, 2, ..., k$, where $k = N \bmod m$ and concatenate them into sequence $(\xi_1, ..., \xi_k)$, which constitutes the bootstrap sample,
3. Estimate the bootstrap value of the largest Lyapunov exponent $\tilde{\lambda}$ from the bootstrap sample and calculate $\tilde{\lambda} - \hat{\lambda}$,
4. Repeat steps 2)-3) a large number of times to construct an empirical distribution for $\tilde{\lambda} - \hat{\lambda}$,
5. Construct a one-sided confidence interval, by calculating the critical value as $\hat{\lambda} - q(\alpha)$ (*e.g.* for $\alpha = 90\%, 95\%, 99\%$), where $q(\alpha)$ is the quantile for the distribution in step 4), following from $\Pr\left\{\hat{\lambda} - \lambda \le q(\alpha)\right\} = \alpha$,
6. If $\hat{\lambda} - q(\alpha) > 0$ the null hypothesis is rejected.

As it can be seen the value of $\hat{\lambda}$ depends on the *m* parameter. Thus for each time series the maximum value of $\hat{\lambda}$ over all $m = 1, 2, 3, 5, 7, 10, 15$ was considered in the blockwise bootstrap method. The obtained results are presented in Table 7.

[Table 7. Results of the blockwise bootstrap method]

For the time series "WIG" (*i.e.* for the original time series before noise reduction) the null hypothesis was not rejected for any α. This is consistent with the hypothesis that investigated series is not chaotic. On the contrary, the positive Lyapunov exponents were detected in the all cleaned data. These results mean that the most likely reason, why the positive exponent was not found in "WIG" was the presence of a noise and that the investigated time series of daily log returns of the WIG index is generated by noisy chaos. However it should be marked that the estimated exponents of cleaned data, although being positive, are very small indeed, therefore this identified chaos is quite weak.

CONCLUDING REMARKS

In this paper the new method of determining a level of a noise in chaotic signals was proposed. It can be applied to indicate the best output of noise reduction algorithms, without using information about the true noise-free signal. The method is easy to apply to and leads to univocal results. The application to the chaotic time series with the added noise of a different level showed its good power. In most cases the indicated output was the proper one, *i.e*. its distance to the original noise-free data was the lowest.

The proposed method was applied to the time series of daily log returns of the Warsaw Stock Exchange Index. The Nearest Neighbours algorithm with different values of the parameters was used to reduce a noise. According to the proposed quantity *NRL* five indicated outputs were chosen to test for chaos. The largest Lyapunov estimates showed, that there is no evidence of chaos in the daily log returns of the Warsaw Stock Exchange Index. However, the applied noise reduction procedure led to time series with positive Lyapunov exponents. These results mean that investigated time series of daily log returns is a realization of noisy chaos.
Empirical evidence of chaos in economic data has been reported by some researchers, but in many cases this evidence seems weak. The presence of a noise may be responsible for such a situation. This inference seems to be confirmed by the results presented in this paper. Thus,

besides improving methods of chaotic time series to become more resistant to a noise, developing noise reduction techniques seem to be a promising area in chaos theory.

ACKNOWLEDGEMENTS

Financial support of Nicolaus Copernicus University in Toruń for the project UMK 403-E is gratefully acknowledged.

APPENDIX: RESULTS OF *NRL* CALCULATIONS

[Table 8. Values of *NRL* [%] calculated for the outputs from the *NN* method applied to the daily log returns of the Warsaw Stock Exchange Index]

[Table 9. Values of Δ*diam* [%] calculated for the outputs from the *NN* method to the daily log returns of the Warsaw Stock Exchange Index]

REFERENCES

[1] Barnett W. and Chen P. The Aggregation-theoretic monetary aggregates are chaotic and have strange attractors. in: W. Barnett, E. Berndt, H. White, eds., Dynamic Econometric Modeling, Cambridge: Cambridge University Press, 1986.

[2] Barnett W.A., Medio A and Serletis A. Nonlinear and Complex Dynamics in Economics. working paper, Washington University, 1997.

[3] Bask M. A positive Lyapunov exponent in Swedish exchange rates? Chaos, Solitons and Fractals, 2002, 14:1295-1304.

[4] Blank S.C. Chaos in Futures Markets? A Nonlinear Dynamical Analysis. The Journal of Futures Markets, 1991, 11:711–728.

[5] Brock W.A. Distinguishing Random and Deterministic Systems: Abridged Version. Journal of Economic Theory, 1986, 40:168–195.

[6] Brock W.A. and Sayers C. Is the Business Cycle Characterized by Deterministic Chaos? Journal of Monetary Economics, 1988, 22:71–90.

[7] Bullard J.B. and Butler A. Nonlinearity and Chaos in Economic Models: Implications for Policy Decisions. working paper 1991–002B, The Federal Reserve Bank of St. Louis, 1991.

[8] Castagli M., Eubank S., Farmer J.D. and Gibson J. State Space Reconstruction in the Presence of Noise. Physica D, 1991, 51:52–98.

[9] Cawley R. and Hsu G.H. Local-geometric-projection Method for Noise Reduction in Chaotic Maps and Flows. Physical Review A, 1992, 46:3057–3082.

[10] Chavas J.-P. and Holt M.T. On Nonlinear Dynamics: The Case of the Pork Cycle. American Agricultural Economics Association, 1991, 73:819–828.

[11] Dechert W. and Gencay R. Lyapunov Exponents as a Nonparametric Diagnostic for Stability Analysis. Journal of Applied Econometrics, 1992, 7:S41–S60.

[12] Farmer J.D. and Sidorowich J.J. Exploiting Chaos to Predict the Future and Reduce Noise. in: Evolution, Learning and Cognition. ed. Y.C. Lee, World Scientific, Singapore, 1988.

[13] Farmer J.D. and Sidorowich J.J. Optimal Shadowing and Noise Reduction. Physica D, 1991, 47:373–392.

[14] Frank M. and Stengos T. Chaotic dynamics in economic time series. Journal of Economic Surveys, 1988, 2:103-133.

[15] Grassberger P. and Procaccia I. Measuring the Strangeness of Strange Attractors. Physica D, 1983, 9:189–208.

[16] Hammel S.M. Noise Reduction for Chaotic Systems. Naval Surface Warfare Center, Silver Spring, 1989.

[17] Hsieh D.A. Chaos and Nonlinear Dynamics: Application to Financial Markets. Journal of Finance, 1991, 5:1839–1877.

[18] Kantz H. and Schreiber T. Nonlinear Time Series Analysis. Cambridge University Press, Cambridge, 1997.

[19] Kostelich E.J. and Yorke J.A. Noise Reduction in Dynamical Systems. Physical Review A, 1988, 38:1649–1652.

[20] Mayfield E.S. and Mizrach B. On Determining the Dimension of Real-Time Stock-Price Data. Journal of Business & Economic Statistics, 1992, 10:367-374.

[21] Nychka D., Ellner S., Gallant R. and McCaffrey D. Finding Chaos in Noisy Systems. Journal of the Royal Statistical Society B, 1992, 54:399–426.

[22] Ott E., Yorke E.D. and Yorke J.A. A Scalling Law: How an Attractor's Volume Depends on Noise Level. Physica D, 1985, 16:62–78.

[23] Ramsey J.B., Sayers C.L. and Rothman P. The Statistical Properties of Dimension Calculations Using Small Data Sets: Some Economic Applications. International Economic Review, 1990, 31:991–1020.

[24] Rosenstein M.T., Collins J.J. and De Luca C.J. A Practical Method for Calculating Largest Lyapunov Exponents From Small Data Sets. Physica D, 1993, 65:117–134.

[25] Sauer T. A Noise Reduction Method for Signals from Nonlinear Systems. Physica D, 1992, 58:193–201.

[26] Scheinkman J. and LeBaron B. Nonlinear dynamics and stock returns, Journal of Business, 1989, 62:311-337.

[27] Schreiber T. Determination of the noise level of chaotic time series. Physical Review E, 1993, 48:R13–R16.

[28] Shintani M. and Linton O. Is There Chaos in the World Economy? A Nonparametric Test Using Consistent Standard Errors. International Economic Review, 2003, 44:331-357.

[29] Wei A. and Leuthold R.M. Long Agricultural Futures Prices: ARCH, Long Memory, or Chaos Processes? working paper, University of Illinois, 1998.

[30] Whang Y.-J. and Linton O. The Asymptotic Distribution of Nonparametric Estimates of the Lyapunov Exponent for Stochastic Time Series. Journal of Econometrics, 1999, 91:1–42.

FIGURES

Fig. 1. Noise-free Henon attractor

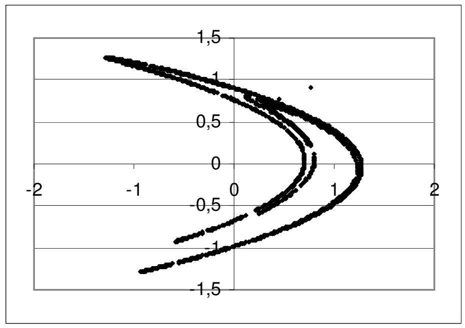


Fig. 2a. Henon attractor with noise (*SNR*=50)

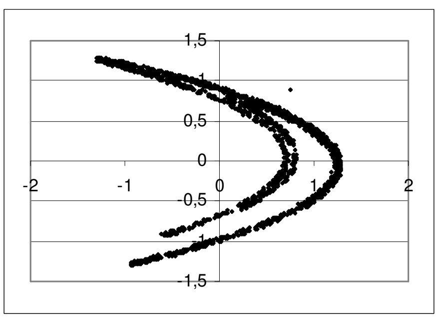


Fig. 2b. Henon attractor after noise reduction (*r*=0.1)

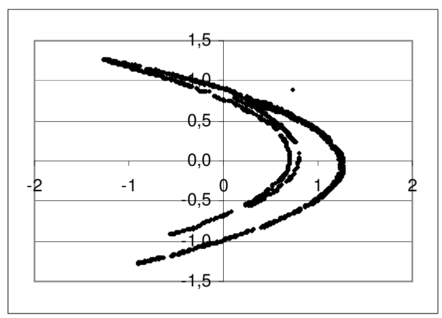


Fig. 3a. Henon attractor with noise (*SNR*=25)

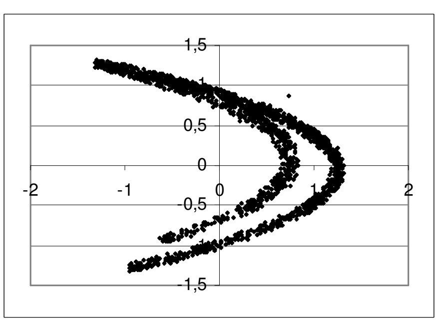


Fig. 3b. Henon attractor after noise reduction (*r*=0.1)

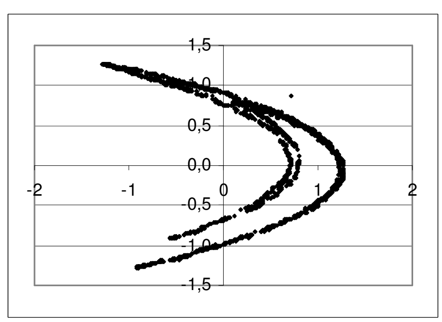


Fig. 4a. Henon attractor with noise (SNR=10)

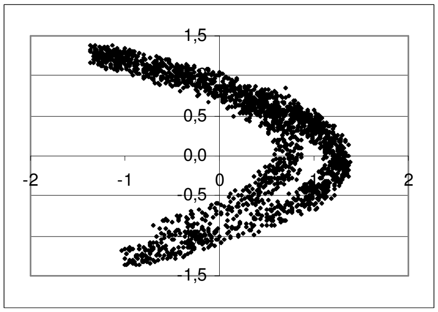


Fig. 4b. Henon attractor after noise reduction (r=0.1)

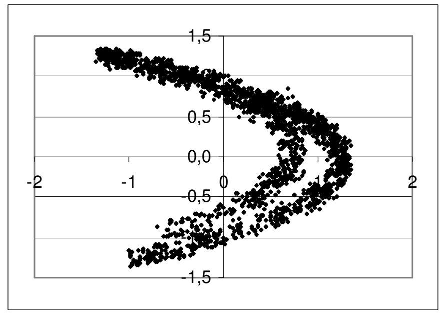


Fig. 5a. Henon attractor with noise (SNR=1)

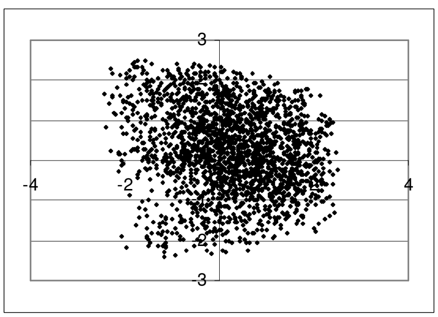
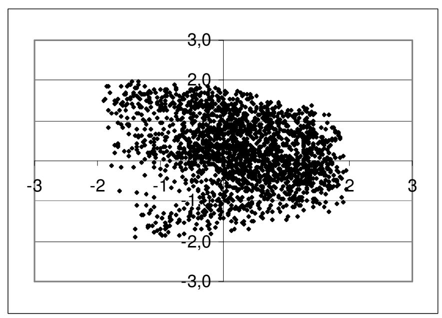


Fig. 5b. Henon attractor after noise reduction (r=1)



TABLES

Table 1. Quantities *NRL*(*m*') [%] and *e* for the Henon map

Radius *r*	***e***	***NRL*(1)**	***NRL*(2)**	***NRL*(3)**	***NRL*(5)**	***NRL*(7)**	***NRL*(10)**	***NRL*(15)**
				***SNR*=50**				
0.001	0.01468	-0.0007 (5.9E-06)	-4.7E-05 (1.8E-05)	-7.5E-06 (1.3E-05)	1.5E-05 (5.2E-06)	2.7E-05 (6.0E-06)	1.2E-05 (3.1E-06)	6.9E-06 (5.7E-06)
0.01	0.01467	-1.2183 (5.9E-06)	-0.7571 (0.0279)	-0.7515 (1.3E-05)	-0.3991 (5.2E-06)	-0.1506 (6.0E-06)	-0.0654 (0.0019)	-0.0026 (0.0089)
0.1	**0.01029**	-0.4840 (1.8407)	**-44.3977** (1.1578)	**-52.4965** (1.2760)	**-51.4974** (1.5243)	**-43.6779** (0.9922)	**-23.8308** (1.1207)	-5.5998 (0.3580)
1	0.14354	**-3.3979** (12.0588)	-31.7682 (10.9801)	-36.3733 (10.2278)	-28.9374 (15.3200)	-21.4811 (14.9719)	-13.4605 (16.7386)	**-17.6959** (17.5354)
before reduction	0.01468	0	0	0	0	0	0	0
				***SNR*=25**				
0.001	0.02937	-0.0006 (1.9E-05)	2.2E-05 (1.8E-05)	3E-05 (5.9E-06)	2.5E-05 (1.0E-05)	7.9E-06 (3.7E-06)	1.2E-05 (2.0E-06)	5.2E-06 (7.7E-07)
0.01	0.02937	-0.0006 (1.9E-05)	2.2E-05 (1.8E-05)	3E-05 (5.9E-06)	2.5E-05 (1.0E-05)	7.9E-06 (3.7E-06)	1.2E-05 (2.0E-06)	5.2E-06 (7.7E-07)
0.1	**0.01479**	-3.3090 (3.2037)	**-42.9889** (2.6869)	**-54.1038** (2.5693)	**-56.8414** (2.4585)	**-53.0518** (2.0608)	**-37.4352** (1.6085)	-12.4759 (0.4583)
1	0.14418	**-4.3662** (13.2392)	-33.5175 (12.3726)	-42.0554 (11.4756)	-38.5766 (13.6458)	-33.9058 (15.8930)	-24.8093 (15.9481)	**-21.9537** (17.6750)
before reduction	0.02937	0	0	0	0	0	0	0
				***SNR*=10**				
0.001	0.07341	0.0005 (4.7E-06)	-8.2E-06 (8.4E-07)	1.8E-05 (9.5E-06)	6.0E-06 (6.4E-06)	-2.5E-06 (5.3E-07)	2.7E-06 (1.7E-06)	2.0E-06 (2.5E-06)
0.01	0.07341	0.0005 (4.7E-06)	-8.2E-06 (8.4E-07)	1.8E-05 (9.5E-06)	6.0E-06 (6.4E-06)	-2.5E-06 (5.3E-07)	2.7E-06 (1.7E-06)	2.0E-06 (2.5E-06)
0.1	**0.06520**	0.3903 (1.8416)	-7.0338 (2.1068)	-15.0061 (1.5912)	-21.1362 (1.4716)	-16.0755 (1.4489)	-11.2505 (0.9368)	-6.4818 (0.7386)
1	0.14706	**-1.7854** (16.8279)	**-34.6290** (16.7948)	**-45.5679** (15.3430)	**-46.1866** (18.3951)	**-42.7728** (19.9092)	**-37.3932** (20.1559)	**-32.0302** (18.4711)
before reduction	0.07341	0	0	0	0	0	0	0

SNR=1								
0.001	0.73413	**-1.1E-05** (4.8E-06)	1.3E-05 (3.4E-06)	-1.2E-06 (2.3E-06)	2.4E-07 (4.1E-06)	1.8E-06 (4.0E-06)	2.3E-06 (3.2E-06)	1.6E-06 (1.9E-06)
0.01	0.73413	**-1.1E-05** (4.8E-06)	1.3E-05 (3.4E-06)	-1.2E-06 (2.3E-06)	2.4E-07 (4.1E-06)	1.8E-06 (4.0E-06)	2.3E-06 (3.2E-06)	1.6E-06 (1.9E-06)
0.1	0.73413	0.3691 (4.8E-06)	-0.0504 (3.4E-06)	-0.0141 (2.3E-06)	-0.0065 (4.1E-06)	-0.0037 (4.0E-06)	-0.0008 (3.2E-06)	-0.0015 (1.9E-06)
1	**0.57762**	0.2583 (21.6299)	**-4.4669** (20.4535)	**-3.5693** (22.3104)	**-7.2804** (19.4009)	**-6.6697** (19.9088)	**-5.4544** (19.8175)	**-5.2039** (19.1304)
before reduction	0.73413	0	0	0	0	0	0	0

Table 2. Quantities *NRL*(*m*') [%] and *e* for the logistic map

Radius *r*	***e***	***NRL*(1)**	***NRL*(2)**	***NRL*(3)**	***NRL*(5)**	***NRL*(7)**	***NRL*(10)**	***NRL*(15)**
***SNR*=50**								
0.001	0.00703	**0.0016** (3.0E-05)	0.0001 (1.3E-05)	7.4E-05 (1.2E-05)	2.7E-05 (4.6E-06)	1.5E-05 (3.3E-07)	5.7E-06 (2.5E-06)	3.0E-06 (2.4E-06)
0.01	0.00672	2.0744 (0.1633)	-4.0256 (0.2803)	-10.3170 (0.2300)	-9.7381 (0.0972)	-4.0643 (0.0342)	-0.5611 (0.0370)	-0.0005 (0.0374)
0.1	**0.00592**	2.0064 (3.0840)	**-44.7431** (2.9513)	**-52.8140** (2.4919)	**-50.0907** (2.1513)	**-33.8730** (1.6497)	**-8.8553** (1.0258)	-0.4265 (0.7200)
1	0.35224	0.3891 (94.5038)	2.6862 (94.6522)	2.6313 (94.9453)	2.2099 (95.5106)	1.6892 (96.0580)	-0.9103 (96.4631)	**-2.3007** (96.6425)
before reduction	0.00703	0	0	0	0	0	0	0
***SNR*=25**								
0.001	0.01405	0.0011 (4.4E-05)	-2.7E-06 (1.5E-05)	-2E-05 (5.1E-06)	-8.8E-06 (5.7E-06)	-1.1E-05 (7.9E-07)	-5.6E-06 (1.0E-06)	-8.2E-08 (9.7E-07)
0.01	0.01403	**-0.1498** (0.0556)	-0.9017 (1.5E-05)	-1.7604 (5.1E-06)	-0.6588 (5.7E-06)	-0.2160 (0.0117)	-0.0507 (0.0016)	0.0213 (0.0273)
0.1	**0.00710**	1.9456 (4.7304)	**-48.8873** (4.8348)	**-58.7867** (3.9522)	**-60.3410** (3.7847)	**-49.9355** (2.4174)	**-17.1284** (1.6307)	-1.1552 (0.8177)
1	0.35235	0.2343 (92.2182)	1.5085 (92.4053)	0.9652 (92.8347)	0.7166 (93.7220)	1.0035 (94.2610)	-0.9388 (94.6404)	**-2.9226** (95.0957)
before reduction	0.01405	0	0	0	0	0	0	0
***SNR*=10**								
0.001	0.03513	-0.0011 (2.0E-05)	-6.6E-06 (5.0E-06)	2.7E-06 (5.3E-06)	6.9E-06 (6.7E-06)	8.2E-06 (6.3E-06)	7.4E-06 (6.5E-06)	5.7E-06 (6.2E-06)
0.01	0.03513	-0.0315 (2.0E-05)	-0.0148 (5.0E-06)	-0.0193 (5.3E-06)	-0.0031 (6.7E-06)	0.0012 (6.3E-06)	-0.0011 (6.5E-06)	-0.0006 (6.2E-06)
0.1	**0.01514**	**-6.4746** (7.8158)	**-38.0378** (8.6298)	**-47.7701** (7.3435)	**-53.8826** (7.1690)	**-53.7599** (4.3353)	**-32.8362** (3.1025)	**-4.7891** (1.2972)
1	0.35025	1.2471 (84.2245)	-2.4985 (84.8853)	-2.7062 (86.2465)	-2.6009 (87.8562)	-1.7341 (88.9405)	-1.7650 (89.6840)	-4.5276 (90.3072)
before reduction	0.03513	0	0	0	0	0	0	0
***SNR*=1**								
0.001	0.35130	**0.0005** (1.9E-05)	-1.2E-05 (9.0E-06)	1.4E-05 (1.2E-05)	1.1E-05 (1.2E-05)	1.5E-05 (1.4E-05)	5.4E-06 (3.5E-06)	1.2E-05 (1.2E-05)

0.01	0.35130	**0.0005** (1.9E-05)	-1.2E-05 (9.0E-06)	1.4E-05 (1.2E-05)	1.1E-05 (1.2E-05)	1.5E-05 (1.4E-05)	5.4E-06 (3.5E-06)	1.2E-05 (1.2E-05)
0.1	0.35129	2.4990 (1.9E-05)	-0.1141 (9.0E-06)	-0.6419 (1.2E-05)	-0.3871 (1.2E-05)	-0.0532 (1.4E-05)	-0.0001 (3.5E-06)	-0.0067 (1.2E-05)
1	**0.25267**	0.6972 (53.5282)	**-3.3927** (56.9274)	**-3.9366** (57.8384)	**-3.7838** (58.9435)	**-4.0285** (59.5164)	**-3.8178** (60.1248)	**-3.6061** (60.4158)
before reduction	0.35130	0	0	0	0	0	0	0

Table 3. Quantities *NRL*(*m'*) [%] and *e* for the Lorenz system

Radius *r*	***e***	***NRL*(1)**	***NRL*(2)**	***NRL*(3)**	***NRL*(5)**	***NRL*(7)**	***NRL*(10)**	***NRL*(15)**
				***SNR*=50**				
0.02	0.25706	0.0002 (8.6E-06)	-3.6E-05 (6.5E-06)	-2.3E-06 (4.8E-06)	7.2E-06 (4.2E-06)	-3.9E-06 (1.6E-06)	-1.7E-05 (1.0E-06)	-1.1E-05 (1.3E-06)
0.2	0.25681	-1.2059 (8.6E-06)	-0.1491 (6.5E-06)	-0.5439 (4.8E-06)	-0.3071 (4.2E-06)	-0.1304 (1.6E-06)	-0.0941 (1.0E-06)	-0.0861 (1.3E-06)
2	**0.22635**	2.5773 (1.0671)	**-8.4983** (1.1788)	**-30.3092** (1.1135)	**-37.8543** (0.8351)	**-37.3158** (0.7591)	**-35.2303** (0.7671)	**-31.7086** (0.6755)
20	4.66168	**-1.8027** (46.6171)	-0.9174 (46.3099)	-13.9762 (46.0364)	-18.9447 (45.2045)	-19.5569 (44.0110)	-19.7320 (42.0856)	-20.0896 (39.2118)
before reduction	0.25706	0	0	0	0	0	0	0
				***SNR*=25**				
0.02	0.51413	-0.0008 (3.3E-06)	-7.8E-05 (8.3E-08)	-1.8E-06 (1.1E-06)	7.6E-07 (1.6E-06)	-5.1E-06 (4.1E-07)	6.2E-06 (5.5E-08)	2.7E-06 (1.4E-07)
0.2	0.51406	0.1405 (3.3E-06)	-0.1154 (8.3E-08)	-0.1382 (1.1E-06)	-0.0677 (1.6E-06)	-0.0233 (4.1E-07)	-0.0102 (5.5E-08)	-0.0143 (1.4E-07)
2	**0.29672**	0.3916 (0.6252)	**-15.2572** (0.8171)	**-37.3999** (0.9941)	**-46.5354** (0.6066)	**-47.7607** (0.5440)	**-47.0001** (0.6787)	**-45.5992** (0.5101)
20	4.59838	**-1.7748** (46.5307)	-5.6262 (46.0921)	-22.4071 (45.8256)	-28.6054 (45.0101)	-30.4874 (43.7353)	-31.4565 (41.9276)	-32.3224 (39.0802)
before reduction	0.51413	0	0	0	0	0	0	0
				***SNR*=10**				
0.02	1.28531	0.0003 (8.3E-06)	7.4E-06 (2.1E-06)	7.9E-07 (5.2E-06)	1.1E-06 (1.1E-06)	-2.5E-06 (1.4E-06)	2.1E-06 (1.3E-06)	5.8E-06 (1.3E-06)
0.2	1.28531	0.0003 (8.3E-06)	7.4E-06 (2.1E-06)	7.9E-07 (5.2E-06)	1.1E-06 (1.1E-06)	-2.5E-06 (1.4E-06)	2.1E-06 (1.3E-06)	5.8E-06 (1.3E-06)
2	**1.02212**	-0.6801 (0.6565)	-12.4599 (0.5894)	-19.5989 (0.1762)	-25.0840 (0.1719)	-25.2618 (0.1092)	-23.5408 (0.1010)	-21.8643 (0.1960)
20	4.32607	**-0.7606** (46.2775)	**-16.8876** (45.3678)	**-31.2357** (44.6167)	**-37.6060** (44.1126)	**-40.3037** (42.4684)	**-41.9160** (40.8214)	**-43.7330** (38.0821)
before reduction	1.28531	0	0	0	0	0	0	0
				***SNR*=1**				
0.02	12.8531	-0.0003 (8.2E-06)	-1.5E-05 (5.1E-06)	9.2E-06 (3.0E-06)	2.6E-06 (2.0E-06)	3.8E-06 (2.3E-06)	1.2E-06 (2.2E-07)	4.9E-06 (4.2E-06)

0.2	12.8531	-0.0003 (8.2E-06)	-1.5E-05 (5.1E-06)	9.2E-06 (3.0E-06)	2.6E-06 (2.0E-06)	3.8E-06 (2.3E-06)	1.2E-06 (2.2E-07)	4.9E-06 (4.2E-06)
2	12.8532	-0.4620 (8.2E-06)	0.0171 (5.1E-06)	-0.0627 (3.0E-06)	-0.0322 (2.0E-06)	-0.0044 (2.3E-06)	-0.0025 (2.2E-07)	-0.0008 (4.2E-06)
20	**8.34946**	**-6.5525** (25.6417)	**-5.2089** (25.2863)	**-9.8332** (23.2964)	**-14.2985** (22.3302)	**-17.0138** (19.9970)	**-17.2966** (19.5290)	**-17.6282** (18.8746)
before reduction	12.8531	0	0	0	0	0	0	0

Table 4. Values of the *r* parameter used in the *NN* method

***m*=1**	***m*=2**	***m*=3**	***m*=5**	***m*=7**	***m*=10**	***m*=15**
0.0005	0.00075	0.001	0.0025	0.005	0.005	0.005
0.00075	0.001	0.0025	0.005	0.0075	0.0075	0.0075
0.001	0.0025	0.005	0.0075	0.01	0.01	0.01
0.0025	0.005	0.0075	0.01	0.025	0.025	0.025
0.005	0.0075	0.01	0.025	0.05	0.05	0.05
0.0075	0.01	0.025	0.05	0.075	0.075	0.075
0.01	0.025	0.05	0.075	0.1	0.1	0.1
0.025	0.05	0.075	0.1	0.25	0.25	0.25

Table 5. Outputs of the *NN* method chosen to further research

Name of series (parameters)	***NRL*(1) (Δ*diam*)**	***NRL*(2) (Δ*diam*)**	***NRL*(3) (Δ*diam*)**	***NRL*(5) (Δ*diam*)**	***NRL*(7) (Δ*diam*)**	***NRL*(10) (Δ*diam*)**	***NRL*(15) (Δ*diam*)**
WIG1 (15; 0.05)	3.729 (7.033)	-54.501 (9.976)	-61.233 (15.075)	-72.02 (11.244)	-73.208 (11.947)	-73.560 (12.696)	-69.557 (17.024)
WIG2 (10; 0.05)	15.018 (10.091)	-53.511 (14.140)	-59.616 (19.630)	-67.639 (18.030)	-67.947 (19.342)	-67.800 (20.250)	-62.475 (25.574)
WIG3 (5; 0.025)	-3.195 (1.336)	-33.452 (1.042)	-42.351 (0.431)	-49.399 (1.569)	-51.625 (1.998)	-51.91 (2.717)	-47.184 (7.354)
WIG4 (7; 0.025)	-1.870 (0.000)	-30.778 (0.412)	-39.520 (0.303)	-47.983 (0.538)	-51.141 (0.795)	-52.288 (0.990)	-48.440 (5.108)
WIG5 (7; 0.01)	-1.011 (0.000)	-1.700 (0.000)	-4.880 (0.054)	-7.754 (0.000)	-9.576 (0.000)	-8.972 (0.000)	-8.483 (0.000)

Table 6. Results of the largest Lyapunov exponent estimates

series \ *m*	***m*=1**	***m*=2**	***m*=3**	***m*=5**	***m*=7**	***m*=10**	***m*=15**
WIG	-0.0117	-0.0053	-0.0005	0.0039	0.0053	0.0036	0.0022
WIG1	0.0332	0.0418	0.0472	0.0540	0.0213	0.0372	0.0384
WIG2	0.0261	0.0352	0.0386	0.0331	0.0186	0.0320	0.0332
WIG3	-0.0011	0.0154	0.0209	0.0249	0.0182	0.0166	0.0257
WIG4	0.0107	0.0184	0.0200	0.0331	0.0255	0.0140	0.0217
WIG5	0.0097	0.0050	0.0128	0.0216	0.0368	0.0124	0.0171

Table 7. Results of the blockwise bootstrap method

	WIG	**WIG1**	**WIG2**	**WIG3**	**WIG4**	**WIG5**
$\hat{\lambda}$	0.0053	0.0540	0.0386	0.0257	0.0331	0.0368
$\hat{\lambda}-q(90\%)$	-0.010	0.044	0.040	0.012	0.026	0.065
$\hat{\lambda}-q(95\%)$	-0.012	0.041	0.037	0.009	0.022	0.057
$\hat{\lambda}-q(99\%)$	-0.019	0.033	0.030	0.003	0.017	0.050

Table 8. Values of *NRL* [%] calculated for the outputs from the *NN* method applied to the daily log returns of the Warsaw Stock Exchange Index

Series (parameters *m*/*r*)		***NRL*(1)**	***NRL*(2)**	***NRL*(3)**	***NRL*(5)**	***NRL*(7)**	***NRL*(10)**	***NRL*(15)**
m'=1	*r*=0.0005	-13.541	-0.903	-0.063	0.003	-0.005	-0.004	0.009
	r=0.00075	-17.060	-1.408	-0.195	-0.008	-0.035	-0.034	-0.042
	r=0.001	**-19.092**	-1.748	-0.355	-0.099	-0.131	-0.130	-0.124
	r=0.0025	**-21.234**	-0.643	-1.172	-0.383	-0.407	-0.500	-0.430
	r=0.005	**-19.969**	-3.423	-2.019	-2.668	-2.999	-3.070	-2.807
	r=0.0075	**-18.632**	-6.313	-4.893	-5.954	-6.258	-6.417	-6.078
	r=0.01	**-19.153**	-8.128	-8.584	-10.469	-11.015	-11.126	-10.348
	r=0.025	-13.011	-25.458	-29.163	-33.764	-35.056	-34.945	-31.719
m'=2	*r*=0.00075	0.155	-7.518	-0.912	-0.101	-0.046	-0.023	-0.029
	r=0.001	-1.125	-7.998	-1.362	-0.168	-0.128	-0.059	-0.080
	r=0.0025	-0.034	-7.071	-3.181	-0.949	-0.628	-0.562	-0.590
	r=0.005	2.501	-5.943	-4.273	-3.625	-3.723	-3.739	-3.248
	r=0.0075	7.343	-8.157	-8.107	-7.596	-7.910	-7.961	-7.470
	r=0.01	-1.115	-11.919	-11.859	-12.729	-13.528	-13.541	-12.655
	r=0.025	**-20.984**	-31.553	-37.837	-40.482	-42.013	-41.846	-38.180
	r=0.05	13.249	**-35.367**	-41.751	-36.593	-37.187	-36.716	-33.267
m'=3	*r*=0.001	-0.560	-1.362	-1.683	-0.060	-0.020	-0.003	-0.012
	r=0.0025	-1.186	-1.733	-6.635	-1.229	-0.492	-0.432	-0.471
	r=0.005	-1.261	-3.043	-6.555	-4.132	-3.311	-3.327	-2.944
	r=0.0075	2.991	-5.566	-9.639	-8.070	-7.559	-7.724	-7.708
	r=0.01	1.716	-8.286	-12.257	-12.853	-13.402	-13.561	-12.732
	r=0.025	-12.592	-32.745	-41.998	-45.989	-47.844	-47.664	-43.316
	r=0.05	-7.700	**-38.803**	**-45.976**	-39.464	-40.126	-39.675	-35.929
	r=0.075	3.843	-22.243	-23.299	-20.609	-20.377	-20.051	-18.382
m'=5	*r*=0.0025	-0.446	-0.239	-0.360	-0.455	-0.042	0.010	-0.009
	r=0.005	-1.271	0.058	-1.480	-4.506	-2.329	-1.472	-1.298
	r=0.0075	0.695	-2.622	-4.196	-8.417	-6.654	-5.875	-5.513
	r=0.01	0.203	-3.669	-7.988	-12.311	-12.147	-11.977	-11.728
	r=0.025	-3.195	-33.452	**-42.351**	**-49.399**	**-51.625**	**-51.91**	-47.184
	r=0.05	10.920	**-43.497**	**-50.740**	**-52.826**	**-52.739**	**-53.865**	**-49.334**
	r=0.075	6.964	-21.389	-22.851	-21.992	-21.793	-21.458	-19.685
	r=0.1	1.643	-7.602	-8.042	-6.774	-6.600	-6.473	-5.956
m'=7	*r*=0.005	0.037	0.003	0.047	-1.119	-1.156	-0.401	-0.245
	r=0.0075	-1.009	-0.043	-1.473	-3.694	-4.691	-3.437	-2.913
	r=0.01	-1.011	-1.700	-4.880	-7.754	-9.576	-8.972	-8.483
	r=0.025	-1.870	-30.778	-39.520	**-47.983**	**-51.141**	**-52.288**	**-48.440**
	r=0.05	11.554	**-49.500**	**-54.787**	**-60.398**	**-60.261**	**-59.731**	**-54.595**

	r=0.075	5.864	-21.815	-23.228	-22.744	-22.625	-22.297	-20.469
	r=0.1	1.551	-7.468	-7.995	-6.741	-6.578	-6.455	-5.937
	r=0.25	0.003	-0.011	-0.013	-0.011	-0.010	-0.008	-0.007
m'=10	r=0.005	0.023	-0.022	-0.025	-0.034	-0.057	-0.039	-0.011
	r=0.0075	-0.259	-0.251	-0.068	-0.475	-1.107	-1.343	-0.777
	r=0.01	-1.244	-1.512	-1.829	-2.976	-4.033	-4.837	-4.335
	r=0.025	2.972	-27.037	-36.079	-44.474	-48.114	-49.852	**-48.923**
	r=0.05	15.018	**-53.511**	**-59.616**	**-67.639**	**-67.947**	**-67.800**	**-62.475**
	r=0.075	8.161	-22.607	-24.183	-23.369	-23.318	-23.051	-21.227
	r=0.1	1.71	-7.383	-7.964	-6.699	-6.544	-6.433	-5.918
	r=0.25	-0.002	-0.005	-0.005	-0.004	-0.005	-0.004	-0.004
m'=15	r=0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.0075	-0.134	0.117	0.003	-0.003	-0.013	-0.050	-0.032
	r=0.01	-0.198	-0.357	-0.078	-0.293	-0.615	-0.947	-1.097
	r=0.025	-4.329	-21.641	-30.714	-39.53	-43.205	-45.551	-46.785
	r=0.05	3.729	**-54.501**	**-61.233**	**-72.02**	**-73.208**	**-73.560**	**-69.557**
	r=0.075	9.504	-23.277	-24.795	-23.67	-23.709	-23.447	-21.625
	r=0.1	1.554	-7.291	-7.903	-6.622	-6.471	-6.395	-5.889
	r=0.25	0.001	-0.012	-0.014	-0.014	-0.015	-0.015	-0.013

Table 9. Values of Δ*diam* [%] calculated for the outputs from the *NN* method to the daily log returns of the Warsaw Stock Exchange Index

Series (parameters *m*/*r*)		**Δ*diam* *m*=1**	**Δ*diam* *m*=2**	**Δ*diam* *m*=3**	**Δ*diam* *m*=5**	**Δ*diam* *m*=7**	**Δ*diam* *m*=10**	**Δ*diam* *m*=15**
m'=1	*r*=0.0005	0.000	0.015	0.015	0.026	0.025	0.026	0.035
	r=0.00075	0.000	0.085	0.010	0.054	0.04	0.038	0.031
	r=0.001	0.000	0.085	0.003	0.051	0.003	0.003	0.011
	r=0.0025	0.000	0.702	0.141	0.464	0.444	0.423	0.498
	r=0.005	2.048	0.097	1.18	0.808	0.681	0.697	0.94
	r=0.0075	2.808	0.339	2.012	1.546	1.596	1.713	2.065
	r=0.01	3.479	1.404	2.812	2.332	2.439	2.604	3.415
	r=0.025	11.935	12.869	17.150	17.222	17.523	18.085	21.271
m'=2	*r*=0.00075	0.000	0.000	0.000	0.000	0.002	0.002	-0.029
	r=0.001	0.000	0.000	0.046	0.000	0.008	0.007	-0.080
	r=0.0025	0.000	0.000	0.055	0.107	0.151	0.204	-0.590
	r=0.005	0.000	0.21	0.11	0.107	0.012	0.079	-3.248
	r=0.0075	0.000	0.73	0.071	0.06	0.063	0.253	-7.470
	r=0.01	0.000	0.277	0.215	0.128	0.104	0.424	-12.655
	r=0.025	5.189	7.306	7.781	10.95	11.461	12.257	-38.180
	r=0.05	35.982	40.727	41.886	50.560	50.878	51.667	-33.267
m'=3	*r*=0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.005
	r=0.0025	0.000	0.000	0.070	0.000	0.000	0.000	0.005
	r=0.005	0.000	0.000	0.005	0.000	0.079	0.038	0.226
	r=0.0075	0.000	0.000	0.167	0.000	0.050	0.07	0.093
	r=0.01	0.000	0.059	0.176	0.038	0.028	0.245	1.017
	r=0.025	2.653	3.407	3.038	5.410	6.019	6.916	11.042
	r=0.05	34.268	37.334	37.68	47.892	48.181	48.973	52.589
	r=0.075	70.023	70.922	72.615	76.342	76.819	77.239	78.894
m'=5	*r*=0.0025	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.0075	0.000	0.000	0.000	0.000	0.000	0.013	0.104
	r=0.01	0.000	0.000	0.023	0.000	0.027	0.008	0.006
	r=0.025	1.336	1.042	0.431	1.569	1.998	2.717	7.354
	r=0.05	20.438	29.459	32.048	34.338	35.569	34.863	39.266
	r=0.075	67.415	71.157	72.811	74.915	75.398	75.842	77.602
	r=0.1	90.674	90.980	91.116	92.598	92.806	92.937	93.439
m'=7	*r*=0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.0075	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.01	0.000	0.000	0.054	0.000	0.000	0.000	0.000
	r=0.025	0.000	0.412	0.303	0.538	0.795	0.990	5.108
	r=0.05	14.113	21.282	26.737	26.344	27.826	28.899	33.937

	r=0.075	66.023	70.43	72.333	74.136	74.573	75.019	76.836
	r=0.1	90.709	91.074	91.138	92.612	92.818	92.948	93.455
	r=0.25	99.989	99.987	99.987	99.989	99.990	99.992	99.993
m'=10	*r*=0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.0075	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.01	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.025	0.000	0.000	0.264	0.193	0.182	0.359	1.973
	r=0.05	10.091	14.140	19.630	18.030	19.342	20.250	25.574
	r=0.075	64.824	69.324	71.247	73.474	73.872	74.274	76.096
	r=0.1	90.746	91.153	91.161	92.631	92.834	92.960	93.468
	r=0.25	99.995	99.994	99.994	99.995	99.995	99.996	99.996
m'=15	*r*=0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.0075	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.01	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	r=0.025	0.000	0.000	0.182	0.000	0.000	0.011	0.202
	r=0.05	7.033	9.976	15.075	11.244	11.947	12.696	17.024
	r=0.075	64.051	68.563	70.512	73.103	73.442	73.855	75.701
	r=0.1	90.809	91.229	91.199	92.676	92.877	92.976	93.482
	r=0.25	99.988	99.986	99.985	99.985	99.985	99.985	99.987