Detection of chaotic behavior in time series

Radim Pánis,[1,a] Martin Kološ[1,b] and Zdeněk Stuchlík[1,c]

[1]Research Centre for Theoretical Physics and Astrophysics, Institute of Physics,
 Silesian University in Opava, Bezručovo nám. 13, CZ-746 01 Opava, Czech Republic
[a]pan0010@slu.cz
[b]martin.kolos@physics.slu.cz
[c]zdenek.stuchlik@physics.slu.cz

ABSTRACT

Deterministic chaos is phenomenon from nonlinear dynamics and it belongs to greatest advances of twentieth-century science. Chaotic behavior appears apart of mathematical equations also in wide range in observable nature, so as in there originating time series. Chaos in time series resembles stochastic behavior, but apart of randomness it is totally deterministic and therefore chaotic data can provide us useful information. Therefore it is essential to have methods, which are able to detect chaos in time series, moreover to distinguish chaotic data from stochastic one. Here we present and discuss the performance of standard and machine learning methods for chaos detection and its implementation on two well known simple chaotic discrete dynamical systems - Logistic map and Tent map, which fit to the most of the definitions of chaos.

Keywords:
   chaos – fractal dimension – recurrence quantification analysis – machine learning – logistic map – time series – tent map

1   INTRODUCTION

As already mentioned chaotic behavior is natural phenomena and the first touch with chaos one assigns the study of the three-body problem by Henri Poincaré (1880). Mathematically, deterministic chaos is the phenomenon which can arise in dynamical systems. In discrete dynamical systems, chaos can appear even for one dimensional systems, while in continuous dynamical systems, deterministic chaos can only arise in three or more dimensions. Finite-dimensional linear systems cannot produce chaos, for a dynamical system to produce chaotic behavior, it must be either infinite-dimensional or nonlinear. The main properties of chaotic systems are e.g. the most known sensitivity to initial conditions or more "mainstream" term - butterfly effect, which means that, even the arbitrarily close initial points can evolve into significantly different trajectories. Other property one should mention are strange attractors. While most of the types of nice regular systems provide very simple attractors, such as points or circular curves called limiting cycles, chaotic motion leads to

something what is known as the strange attractor, which is attractor with magnificent details and great complexity.

With chaos it is not easy, so we have some dynamical systems, which are chaotic everywhere, but there are also cases, where chaotic behavior appears only in a subset of the phase space, then a large set of initial conditions leads to orbits that converge to this chaotic region. Chaotic behavior is for some systems as for example for logistic and tent map easy controllable and it depends on one parameter. Generally, bifurcation occurs when small smooth change of "chaotic parameter" (the bifurcation parameter) makes abrupt or topological change in behavior of the system. This dependency (or period-doubling transition from some N-point to an 2N-point attractor) on the chaotic parameter is nicely shown on the bifurcation diagram. For the purpose of presenting of the nonlinear methods of chaos detection we use superposition of graphics, where we display on the bifurcation diagram the estimates of chaotic behavior by later mentioned methods. We applied these methods on time series belonging to given "chaotic parameter" and we plot this estimation of chaoticity on the bifurcation diagram.

Strange attractors are special and they even have a fractal structure, and therefore one can calculate fractal dimension of them, which can provide useful information about the system even when the equations of the dynamical system are unknown and we can observe only one of its coordinates. For estimation of fractal dimension we present two methods, namely, Box-counting and Correlation dimension, classical approach of studying dynamical systems is the calculation of Lyapunov exponent, for this, one needs to know the rules of evolution for given dynamical system (equations), we however try to work only with one dimensional input, so we also use a numerical approach and compare both results. Another approach we use is recurrence quantification analysis (RQA), roughly speaking, it is numerical description of recurrence graph - graphical tool for investigating properties of dynamical systems. The last approach is lately again very popular Machine learning (ML), which has many advantages when dealing with nonlinear data and has proved its abilities in many useful applications. We use for our purpose all seven possible ML implementations build in Mathematica software, which is also used for all the calculations.

These methods are also used in (Pánis et al., 2019), where the information about the matter dynamics and electromagnetic field structure around compact object (black hole or neutron star) are provided, while the chaotic charged test particles dynamics around a Schwarzschild black hole immersed in an external uniform magnetic field is examined.

2   METHODS OF DETECTION OF CHAOTIC BEHAVIOR IN TIME SERIES

2.1   Box-counting method

Box-counting (D_0) or box dimension is one of the most widely used estimations of fractal dimension. The calculation and empirical estimation of this method is quiet simple compared to another ones. We present the general idea behind the algorithm, for more detailed description one can look at (Schroeder, 1991). For a set S in a Euclidean space $\mathbb{R}^n$ we

define Box-counting as

$$D_0 = \lim_{\epsilon \to 0} \frac{\ln N(\epsilon)}{\ln \frac{1}{\epsilon}}, \tag{1}$$

where $N(\epsilon)$ is the number of boxes of side length ϵ required to cover the set. Dimension of S is estimated by seeing how the logarithmic rate of $N(\epsilon)$ increase as $\epsilon \to 0$, or in words as we make the grid finer.

2.2 Correlation dimension

Very popular tool for detecting chaos in experimental data is calculation of the Correlation dimension (D_2). The general idea behind computing correlation dimension is to find out for some small ϵ the number of points $C(\epsilon)$ (correlation sum), which Euclidean distance is smaller than ϵ.

$$D_2 = \lim_{\epsilon \to 0} \frac{\ln C(\epsilon)}{\ln \epsilon}, \tag{2}$$

one computes this for various number of ϵ and D_2 can be then approximated again by fitting of the logarithmic values.

It is worth mentioning that D_2 and D_0 is part of D_q family of fractal dimensions (Schroeder, 1991) defined as

$$D_q = \lim_{\epsilon \to 0} \frac{1}{q - 1} \frac{\ln \sum_k p_k^q}{\ln \epsilon} \quad -\infty \leq q \leq \infty, \tag{3}$$

where, p_k denotes relative frequency with which fractal's points are falling inside the k-th cell.

For $q = 0$ we obtain already mentioned Box-counting dimension, for $q \to 1$ we obtain information dimension, which numerator is denoted as the Shannon's entropy and for $q = 2$ we obtain correlation dimension.

There is several algorithm approaches of calculation correlation dimension, lets just mention the approximation of $C(\epsilon)$ published in (Grassberger and Procaccia, 1983):

$$\hat{C}(\epsilon) = \lim_{N \to \infty} \frac{2}{N(N - 1)} \sum_{i < j} H(\epsilon - |x_i - x_j|), \tag{4}$$

where H is Heaviside step function.

When using the nonlinear methods one should not omit importance of embedding dimension and then also closely connection to Takens's theorem about reconstruction of state space from sequence of observations (Takens, 1981). Embedding dimension creates from series of length $N + m - 1$ for some given m series of N vectors, where i-th component looks like:

$$x_i = (x_{i-m+1}, x_{i-m+2}, ..., x_i) \in \mathbb{R}^m. \tag{5}$$

One of the purposes of embedding dimension, in context of Correlation dimension is to distinguish between chaotic and random time series. Where by chaotic series for increasing $m \in \mathbb{N}$ fractal dimension estimation stabilizes at some value $D < m$, while for random series, dimension goes along with m to infinity.

2.3   Lyapunov exponent

Lyapunov exponents apart of previous methods are originally used for investigation of dynamical systems, or rather said not any fractals. In short, it is a number that describe the amount of separation of trajectories which are infinitesimally close. Near trajectories in chaotic systems diverge exponentially, what leads to positive Lyapunov exponents. The amount of separation can differ for various directions of initial separation vector. Because of this fact, there is a spectrum of Lyapunov exponents, which corresponds to the phase space dimension.

For example if we consider logistic map $f(x) = rx(1-x), r \in [0, 4]$ as an typical example of simple chaotic system, which is more precisely one-dimensional nonlinear difference equation. The Lyapunov exponent can be calculated directly from the expression of the f function from the formula

$$\lambda(r) = \lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln \left[\left| f'\left(f^n(x_0)\right) \right| \right]. \tag{6}$$

However we try to work only with time series inputs which is not allowing us to use such a formula. We assumed in future to apply our methods on observational data from the telescope. This approach leads us to use method, which is determining Lyapunov exponents from time series. The maximal Lyapunov exponent characterizes the spectra and therefore denotes amount of predictability for some dynamical system. It can be calculated without knowledge of a model which produces the time series. We use the method based on the statistical properties of the divergence of neighboring trajectories approach introduced by Kantz, (Kantz, 1994). Our algorithm applied to logistic map is very similar to formula 6 which could be found for example also in (Enns, 2001).

2.4   Recurrence quantification analysis

The recurrence quantification analysis is quiet widely used tool for investigating the state space trajectories. Simply said it determines the number and duration of recurrences of a dynamical system. RQA is developed since 1992 (Zbilut and Webber, 1992; Marwan, 2008), where the novel approach based on averaging along with the way of setting the correct input parameters, which provide more accurate RQA measures is presented in (Bhatta et al., 2020). Recurrence plot provides a graphical tool for observing periodicity of phase space trajectories and was introduced in (Eckmann et al., 1987). This observing is possible through visualization of a symmetrical square matrix, in which the elements correspond to times at which a state of a dynamical system recurs.

One can define RP which measures recurrences of a trajectory $x_i \in R^d$ in phase space

$$R_{i,j} = H\left(\epsilon - \|x_i - x_j\|\right) \quad i, j = 1, ..., N, \tag{7}$$

where N is the number of measured points x_i, ϵ is a threshold distance and $\| \cdot \|$ is a norm. From this equation we obtain the already mentioned symmetrical square matrix of zeroes and ones. When we will represent this two repeating elements with different colors in a plot we obtain the discussed RP. Threshold value parameter determinate density of RP plot.

RQA tools with well established short forms, which we use for investigating of chaotic trajectories are:

(1)   RR - The recurrence rate is simplest tool, which measures density of recurrence points in the recurrence plot, or in another words, it counts the number of ones in RP and divides them by number of all elements in the matrix. RR reflects the chance that some state of the system will recur

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j}. \tag{8}$$

(2)   DET - Determinism is rate of recurrence points which build diagonal lines. DET determines how predictable the system is

$$DET = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{i,j=1}^{N} R_{i,j}}, \tag{9}$$

where $P(l)$ denotes the frequency distribution of lengths l of the diagonal lines.

(3)   LL - Is average diagonal line length, which is in relation with the time of predictability of the system. It reflects the average time for which any two parts of trajectory are close, this time can be denoted as mean prediction time

$$LL = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=l_{min}}^{N} P(l)}. \tag{10}$$

(4)   ENTR - Entropy or the Shannon entropy of the probability distribution of the diagonal line lengths p(l), which are reflecting complexity of the system's deterministic structure

$$ENTR = - \sum_{l=l_{min}}^{N} p(l) \ln p(l), \tag{11}$$

where $p(l)$ is probability that a diagonal line is exactly of the length l can be estimated from the frequency distribution $P(l)$ with $p(l) = \frac{P(l)}{\sum_{l=l_{min}}^{N} P(l)}$.

2.5   Machine learning

Machine learning is very powerful tool, which founds application in many fields, lets just mention language translating algorithms, computer vision, beating best Go player in the world (Silver et al., 2017b), or chess programs of different architecture (Silver et al., 2017a) in an incredible fashion. In this work only basic principles of machine learning are presented. Roughly speaking machine learning is field of computer science, strongly connected to another fields as optimization, statistics, linear algebra, etc.

Its beginning goes to 1950's and as many inventions in computer science or better said in science in general. Machine learning was not invented by single person, let's only mention

Table 1. Comparison of time in seconds required for different nonlinear methods applied on time series generated by logistic map. For parameter r varying from 2 to 4 with the step of 0.01 leads to 200 time series, with initial value $x_0 = 0.1$ we did set up the iterations to length 100, 1000 and 10000 and compared the time needed for the calculation for given methods.

Method / Length	100	1000	10000
Box-count	0.354	0.416	3.519
Correlation dim.	4.155	26.54	2353
Lyapunov exp.	3.294	119.5	12574
RQA- RR	0.164	6.332	507.8
RQA-DET	0.209	6.069	916.9
RQA-ENTR	0.197	7.247	1128
RQA-LAM	0.197	7.248	1128
ML-Rand. For.	0.677	1.212	7.479

A. Samuel, who used first the term "Machine learning". Machine learning is using algorithms on data samples to discover known or unknown patterns in data, this dividing of patterns leads to basic divisions of machine learning and namely supervised, semi-supervised and unsupervised learning. The wide range of applications announces the good ability of handling nonlinear data. Our intention of using machine learning is to decide whether a trajectory of a particle is chaotic or not. For this purpose we use supervised machine learning, where we train various ML algorithms with samples calculated by classical methods already described, namely, (Box-counting, Correlation dimension, Lyapunov exponent, RQA - RR, DET, LL, ENTR) with effort to use all the different properties of them and the training set consists overall of 100 examples.

3   CONCLUSION

Deterministic chaos is hardly predictable and apparently random behavior which can appear in dynamical systems. Classical example of such nonlinear dynamical system is logistic map, denoted by quadratic recurrence equation

$$x_{n+1} = rx_n (1 - x_n). \tag{12}$$

Given the initial value $x_0 \in (0, 1)$, the logistic map (12) will generate sequence of real numbers $x_n \in (0, 1)$. The behavior of such sequence x_n strongly depends on logistic map parameter $r \in [2, 4]$. Roughly speaking is the behavior on the interval $r \in (2, r_0)$ regular (rather predictable) and on the interval $r \in [r_0, 4]$ is chaotic (hardly predictable or rather unpredictable) with some occasional "islands of regularity". The transition between regular and chaotic behaviors happens for parameter $r = r_0 \approx 3.56995$. Bifurcation diagram for logistic map, with asymptotically approached values of the sequence, is shown in Fig. 1 depicted by gray background points.

For chaos detection and chaotic behavior description in sequences of numbers following methods have been tested: Box-counting method (section 2.1), Correlation dimension (2.2),
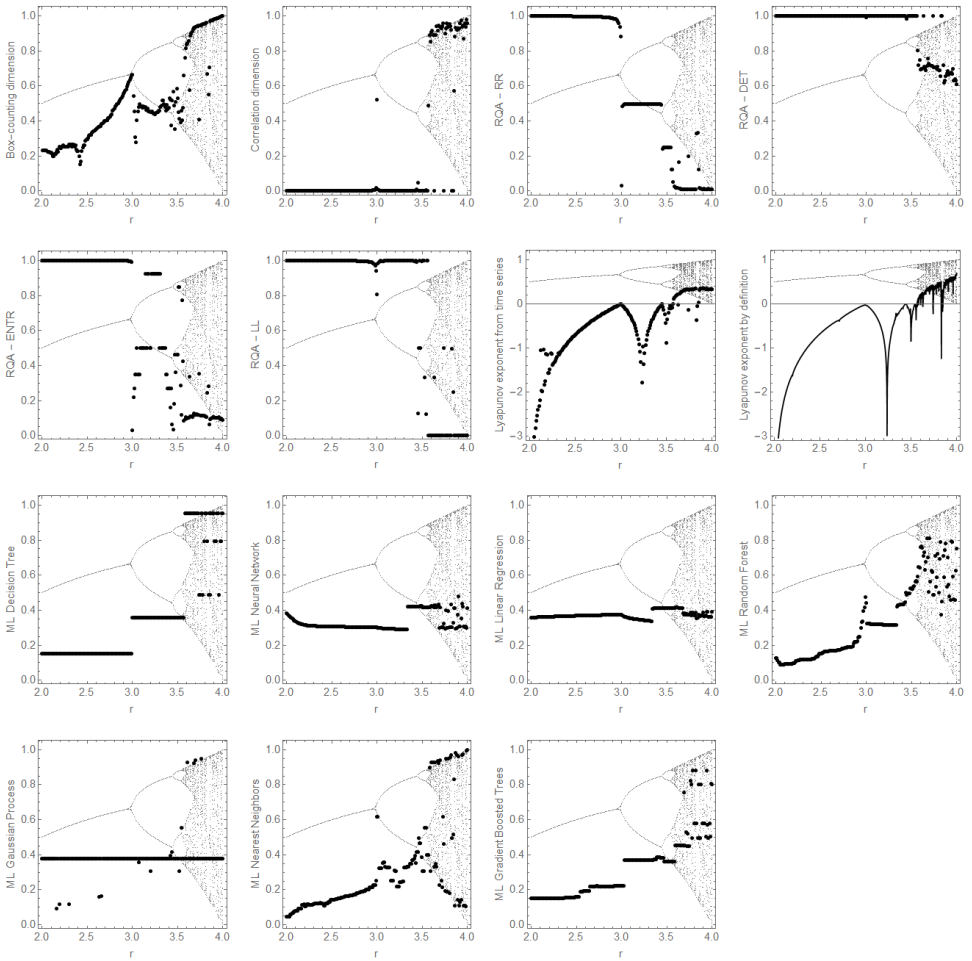
Figure 1. Comparison of different nonlinear methods for time series generated by logistic map $x_{n+1} = rx_n(1 - x_n)$. For different parameter $r \in [2, 4]$ we generate 10000 points series with initial value $x_0 = 0.1$. All the methods detect more chaoticity when we enter the chaotic region for value $r > r_0 \approx 3.56995$. RQA tools however, are working in reverse fashion as one can see, they denote more chaotic regions with lower numbers and vice versa, the data (chaoticity estimations) for RQA-LL and RQA-ENTR have been transformed into interval $[0, 1]$.
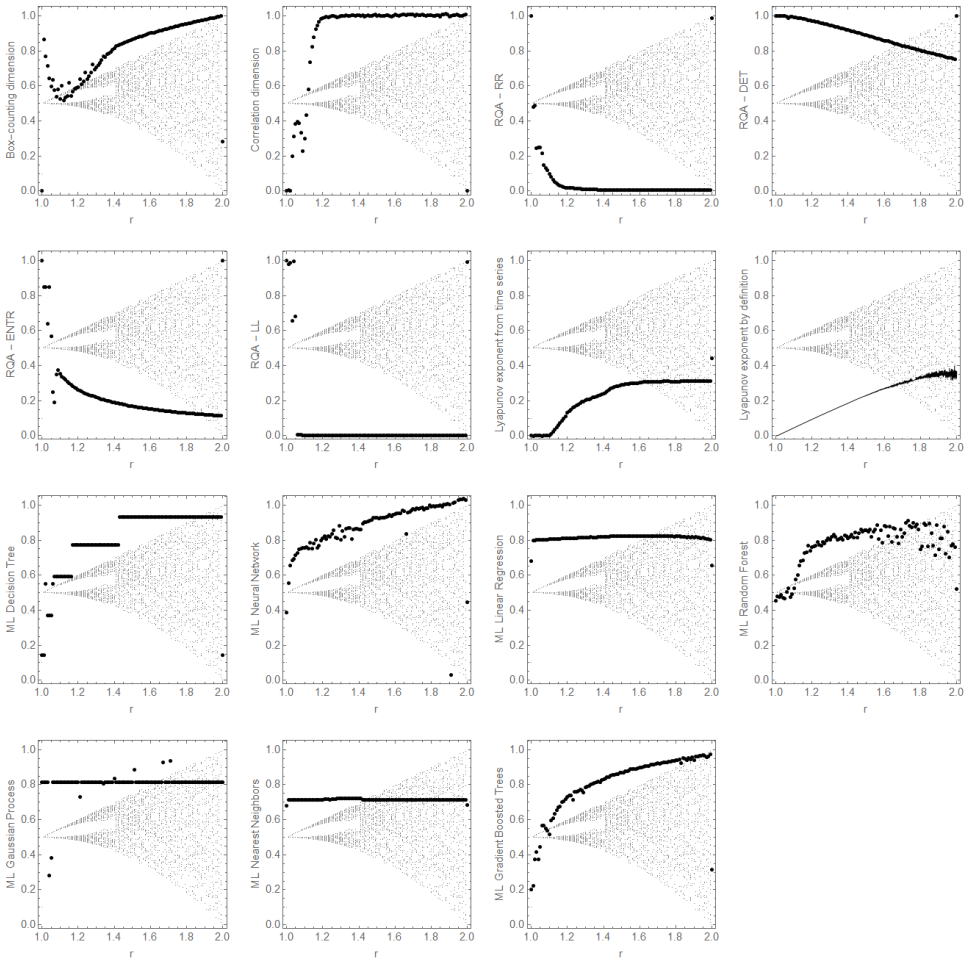
Figure 2. Comparison of different nonlinear methods for time series generated by tent map defined as $x_{n+1} = f_r(x_n) = \begin{cases} rx_n & \text{for } x_n < \frac{1}{2} \\ r(1-x_n) & \text{for } x_n \geq \frac{1}{2} \end{cases}$. For different parameter $r \in [1,2]$ we generate 10000 points series with initial value $x_0 = 0.1$.

Lyapunov exponent (2.3), RQA (2.4) and diferent machine learning algorithms (2.5). All of the algorithms we use in this article are capable to use one dimensional sequence of real numbers (time series) as input. From the point of observation is important to distinguish between chaotic sequences ruled by some (unknown) laws and random sequences obeying for example some stochastic distribution, which can contribute to the noise part in the detected signal. The theoretical boundary of distinguishing between chaos and random sequences is sequence length (Ott, 1993). If the nonlinear dynamical system has many degrees of freedom, then for short sequences is impossible to tell.

To clear up how the nonlinear methods works is shown on four representative sequences $x_n^{(1)}, x_n^{(2)}, x_n^{(3)}, x_n^{(4)}$ of length 10^4, where $x_n^{(1)}$ is regular sequence, $x_n^{(2)}$ is weakly chaotic sequence generated by logistic map with $r = 3.6$, $x_n^{(3)}$ is strongly chaotic sequence generated by logistic map with $r = 4$, and $x_n^{(4)}$ is sequence of pseudo-random numbers. The results are presented in Tab. 2.

In detail we have tested the nonlinear methods on sequences of numbers generated by logistic and tent map for various values of parameter r, see Fig. 1 and 2, the application/testing of the methods on a specific task can be found in (Pánis et al., 2019). The important common sign of all these methods is that, they are able to detect more chaoticity when divergence of trajectories in bifurcation diagram occurs, in other words, when there is more than one fixed point. This fact we can observe for example when $r = 3$ for logistic map in Fig. 1 and for r little above value 1 for tent map in Fig. 2. However, this is not undeniable true for all the cases, what leads to different estimations of chaoticity for different methods. Box-counting and Lyapunov exponent method show approximately linear behavior when moving to point, where the divergence of the trajectories begins, while when moving to such a point chaoticity is increasing and decreasing by leaving it. Correlation dimension shows little bit different behavior, when nonchaotic region is denoted by values close to zero and the level of chaos starts rapidly grow only from small distance from the region of divergence of trajectories in bifurcation diagram, this behavior is also observable by RQA tools, however in reversed fashion. Satisfying is the detection of chaoticity for logistic map for the $r \geq 3.56995$, when all these methods are showing the highest values of chaoticity, in this region are also "islands of regularity" though, what means not all the estimations of chaos in this region should be high. By tent map is the behavior of the methods also satisfying, we should not omit the fact the divergence of trajectories starts from very small values above $r = 1$ and grows with r moving to value $r = 2$, where from our definition of tent map and initial condition value $x_0 = 0.1$ the behavior gets suddenly quiet regular again.

As expected different machine learning algorithms produce different results. Random Forest algorithm seems to produce quiet reliable result - our estimation of chaoticity is getting higher most of the time as the r parameter grows, the exceptions could be explained by the fact there are also "islands of regularity" in the chaotic region.

The time spent for calculation of the individual graphics in Fig. 1 is presented in Tab. 1, where we in addition present also the times consumed in the cases for shorter time series in order to catch the nonlinear growth of time consumption with larger input. From here we can see that the Box-counting method is the fastest, but when looking on the results the other methods seem to be more precise. The time spent also strongly depends on the sequence length - such dependence is nonlinear in the case of almost all the methods. For

Table 2. Comparison of values of different nonlinear methods applied on regular, chaotic and pseudo-random generated series of the length 10 000.

Met./ Type	Regular	Weak ch.	Strong ch.	P.-rand.
Box-count	0.0969	0.8343	1	1
Corr. dim.	$5 * 10^{-15}$	0.8943	0.8735	0.9978
Lyap. exp.	$-7 * 10^{-16}$	0.1889	0.3289	0.0842
RQA-RR	0.4999	0.0183	0.0116	0.0057
RQA-DET	1	0.6959	0.6114	0.0113
RQA-ENTR	8.5152	0.8951	0.8202	0.0296
RQA-LAM	5001	5.6299	2.8507	2.0006
ML-Rand. For.	0.5915	0.6917	0.7869	0.3788

most of the methods we have programmed and tested several variants for each of them according to algorithm architecture and settings of adjustable parameters. The goal was to obtain good and also computing time acceptable results, the code in Mathematica is available on the GitHub repositary at (Pánis, 2019).

In previous texts and presented Figures 1, 2, and tables 2 and 1 we provide brief overview about the common and novel methods used in nonlinear time series analysis, which can provide useful hints for potential users in the sense of finding suitable method when studying the nonlinear phenomena. Proclaim any here discussed method as the best would be misleading, and this ambiguity expresses aptly the saying, that every complicated question has a simple answer which is wrong. When choosing a method for particular purpose, one should have a good idea about the data structure as well as have in mind the definitions of the methods when discussing the results in the context of the underlying physics. Considering the main properties of data, when choosing a method, one should definitely have in mind the available hardware capabilities along with the dimensions of the data.

For very large data sets definitely the Box-counting and ML methods should be suitable, however when considering ML, the available data for training set constriction have to be mentioned. Lyapunov exponents are popular in physics from the point of interpretation and connection to the direct physical properties of the systems. RQA in the trade-off with longer computational time needed provide more numerical descriptions of the data, the connection to the physical properties in terms of Blazars can be found in (Bhatta et al., 2020) and the enhancing of the computational time needed could be the point of future research in a context of GPU usage.

ACKNOWLEDGMENTS

REFERENCES

Bhatta, G., Pánis, R. and Stuchlík, Z. (2020), Deterministic Aspect of the γ-ray Variability in Blazars, *arXiv e-prints*, arXiv:2010.15085, arXiv: 2010.15085.

Eckmann, J.-P., Oliffson Kamphorst, S. and Ruelle, D. (1987), Recurrence plots of dynamical systems, *EPL (Europhysics Letters)*, **4**, p. 973.

Enns, R. (2001), *Nonlinear physics with Mathematica for scientists and engineers*, Birkhauser, Boston, ISBN 978-0-8176-4223-5.

Grassberger, P. and Procaccia, I. (1983), Characterization of strange attractors, *Physical Review Letters*, **50**, pp. 346–349.

Kantz, H. (1994), A robust method to estimate the maximal Lyapunov exponent of a time series, *Physics Letters A*, **185**, pp. 77–87.

Marwan, N. (2008), A historical review of recurrence plots, *European Physical Journal Special Topics*, **164**, pp. 3–12, arXiv: 1709.09971.

Ott, E. (1993), *Chaos in dynamical systems*, Cambridge University Press.

Pánis, R., Kološ, M. and Stuchlík, Z. (2019), Determination of chaotic behaviour in time series generated by charged particle motion around magnetized Schwarzschild black holes, *European Physical Journal C*, **79**(6), 479, arXiv: 1905.01186.

Pánis, R. (2019), Chaos detection, https://github.com/radim525/Chaos-detection.

Schroeder, M. (1991), *Fractals, chaos, power laws. Minutes from an infinte paradise*.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D. (2017a), Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, *ArXiv e-prints*, arXiv: 1712.01815.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. and Hassabis, D. (2017b), Mastering the game of Go without human knowledge, *Nature*, **550**, pp. 354–359.

Takens, F. (1981), *Detecting strange attractors in turbulence*, volume 898, p. 366.

Zbilut, J. P. and Webber, C. L. (1992), Embeddings and delays as derived from quantification of recurrence plots, *Physics Letters A*, **171**, pp. 199–203.