Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 18(8), 311–318
https://doi.org/10.12913/22998624/194428
ISSN 2299-8624, License CC-BY 4.0

Received: 2024.08.21
Accepted: 2024.10.09
Published: 2024.11.01

Statistical Long-Range Dependencies and Statistical Self-Similarity in Computer Systems Processing – The Case of Cache Bytes Counter

Bartosz Kowal[1*], Dominik Strzałka[1]

[1] Department of Complex Systems, Rzeszów University of Technology, Al. Powstańców Warszawy, 35-959 Rzeszów, Poland

* Corresponding author's e-mail: b.kowal@prz.edu.pl

ABSTRACT

The existence of long-range dependencies in many natural systems was a very important discovery that introduced many interesting challenges and explanations about the systems behaviour. In the case of man-made systems such dependencies can also be visible, and one example is computer systems. Because the studies focused on long-range statistical dependencies in computer systems, particularly in the context of system performance counters, are not very common in the literature, this paper undertook an investigation of statistical long-range dependencies present in cache memory data represented as time series. Based on the time series collected during computer system processing by internal system tools, it will be seen that in the case of cache memory modelling, statistical models with long-term dependencies should be used. The following paper sections show how to collect data, analyse, and build an appropriate model.

Keywords: Hurst exponent, long-range dependencies, systems performance, data analysis.

INTRODUCTION

The problem of system performance is the key issue starting from the beginning of information processing systems. Information processing systems are usually considered mass service systems; therefore, approaches based on MVA analysis can be taken [1], queueing systems [2], statistical performance analysis [3]. From a practical point of view, it is not only important to deal with their performance seen as an internal feature, but it is also expected to have appropriate statistical and reference models to see how they work in real conditions under different types of processed workload. A typical approach assumes that performance test can be done based on benchmarks giving the results represented by different point scores that can be further compared in order to have a feedback. However, it should be remembered that despite the benchmarks giving comparable results of performed tests, they also generate artificial workload that not always is seen in real computer systems that are used, for instance, by humans. In this paper, an approach was taken to collect long time series that represent the behaviour of Windows operating systems one counter during normal use of the computer system. The workload was generated by users who were asked to use different computer programmes similarly to their daily activities. Data were collected with a resolution of 1 second, and for analysed time series, the statistical analysis was performed. The main challenge was to determine the degree of statistical self-similarity. A motivation for this paper comes from the observation that the studies focusing on existence of statistical long-range dependencies (expressed by statistical self-similarity) in computer systems, particularly in the context of system performance counters, are not very common in the literature. There are some papers [4–6], where the authors explored the problem of long-range dependencies existence in specific areas of computer systems and computer networks. However, the most famous paper related to computer networks is a pioneer work of Paxson

[7]. But in the case of paper [7] there were no technical possibility to use built-in operating system solutions to collect big data sets. The family of Windows operating systems allow to collect different data sets that can be not only visualized but also analysed by different statistical methods. As a result a new existing phenomena can be discovered.

For example, paper [4] shows an analysis of memory errors per second and page faults per second in cache memory counters, indicating the existence of statistical self-similarity and long-range dependencies in these systems. However, this study was conducted on a single computer, which only allowed for the observation of cache memory behaviour in one hardware configuration.

On the other hand, paper [5] examined anomaly detection in IoT networks using the Hurst exponent and multifractal analysis. This research showed that multifractality and Hurst analysis can be effective in detecting anomalies in network traffic. Although this study focused on networks, the Hurst exponent method applied in this paper also confirms the usefulness of this tool in modelling long-range dependencies in computer systems.

Another application of long-range dependencies in computer networks is presented by the author [6], who describes a method for network traffic prediction based on wavelet transform and the fusion of multiple models. The Mallat algorithm decomposes the time series into approximate and detailed components, and the Hurst exponent analysis is used to classify these components.

COUNTERS IN WINDOWS OPERATING SYSTEMS

In the Windows operating system family (Windows OS) family, there is a special solution, called a 'performance counter' (or counter) that can be used to collect data on system performance [8]. This stands for special approach where internal management tools in operating system allowed to see, for example, % of CPU usage, the amount of used RAM, etc. [9]. The main use of these counters data is the detailed information about the system state. Depending on the version of Windows OS, there are several types of counters, and this determines how the information collected by the counter is calculated [10]. For example, in Table 1 there is one example of counter.

In this case (Table 1) the method of retrieving data is immediate, i.e. when the command is invoked, the data are read and then processed by the mean() statistical function resulting as the average of the last two measurements over the period.

Thanks to internal operating system tools, for example perfuming [9], the data collected by the counters can be stored in files as time series and used for further data processing. In this paper, such an approach was taken, and the presented analysis is based on a set of 50 different time series from 50 different computers. These details are developed in Section 5.

ANALYSIS OF DEPENDENCIES IN TIME SERIES

From a practical point of view, there is a need to analyse time series not only by assessing typical statistics like minimum, maximum, or average values, but also by examining the long-term (long-range) dependencies. This approach allows for a deeper understanding and prediction of long-term system behaviour, which is extremely important in many scientific areas and technological applications [11, 12]. The second important aspect in the study of time series, enabling a deeper understanding of time series dynamics, is the study of their stationarity. Usually, in the simplest approach, the stationarity of a time series means that its statistical properties, such as mean and variance, do not change over time [13]. It is important to recognise whether a time series is stationary because most forecasting models and analyses used in statistics assume that the data are stationary. This refers especially to methods in which the statistical self-similarity of the data is considered [14].

Hurst exponent

The discovery and introduction to statistics of the Hurst exponent was a very important

Table 1. Type of counter: PERF_COUNTER_RAWCOUNT

Description	Shows the last observed counter value
Data read time	Instantaneous
Formula	Shows data in RAW form (as it was read)
Average	*SUM* (*N*)/*x* *N* is used to describe the raw counter data
Counter example	Memory\cache bytes

achievement in modelling science. It turned out that there exists an expansion of classical Brownian motion toward generalisation that includes the property of long-term memory [15]. Over the last 70 years, this interesting phenomenon was discovered in many systems [16] and in the case of computer systems the most important were discoveries from 1993 [17].

There are several methods that allow to calculate the Hurst exponent values. The most popular ones include: the R/S (Rescaled Range) method, periodogram analysis, and the detrended fluctuation method (DFA). Each of these methods has its own specific applications and may be preferred depending on the characteristics of the data and the objectives of the study. They also explain and interpret long-term (long range) dependencies (long memory) in time series. The Hurst exponent, denoted as H, helps to determine whether there are statistical patterns in a given time series that persist over long periods of time, beyond what would be expected from short-term correlations.

The Hurst exponent varies in the interval <0, 1> and is explained as:

- $H < 0.5$ indicates antipersistent behaviour (future values will probably tend to reverse trends),
- $H = 0.5$ suggests random walk-like or Brownian motion,
- $H > 0.5$ indicates persistent behaviour (future values are likely to follow the trend of past values).

Therefore, this analysis provides valuable information about the nature of phenomena, enabling them to better understand, forecast, and manage time-series data.

Hurst exponents are used in many fields, from finance [18] to hydrology [19], where they allow for the identification and forecasting of interesting events. In finance, for example, they can indicate potential market volatility [20, 21], whereas in hydrology, they help predict phenomena such as droughts [22, 23] or floods. Therefore, this analysis provides valuable tips that can be used to better understand the dynamics of the studied phenomena and to create more effective forecasting models [6].

There are many methods to estimate the Hurst exponent, e.g. the absolute value method, the aggregated variance method, Detrended Fluctuation Analysis [4] or Periodogram method [24]. From a historical point of view, the first and the oldest method of estimating the Hurst exponent was based on range and scale (R/S) analysis [15]. This method involves dividing the time series into segments, then calculating the quotient of the range of values and the standard deviation for each of them, and then determines the slope of the regression line for the logarithms of the range and scales [25].

Data stationarity

Different statistical tests and computing methods can be used for analysis of time-series stationarity. Among them, the following ones can be listed:

- a heuristic approach based on data plots with visual inspection about seasonality, trends and changing levels, the increasing variance, analysis of autocorrelation function [26],
- Augmented Dickey-Fuller (ADF) test [27],
- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [28],
- Phillips-Perron (PP) test [29].

The above-mentioned methods have their strengths and weaknesses that depend not only on methodology but also on processed input data sets. Moreover, sometimes it is not so obvious how to interpret obtained results despite the fact that usually they are expressed in terms of acceptance or rejection of a null hypothesis by the p-value. This is especially visible when used methods return divergent results that are not consistent.

Usually in the case of many examples of time series analysis, this step is omitted with a silent assumption that input data are stationary, but from methodological point of view it is mandatory for having final reliable results. In this particular paper, the authors used several methods to test data stationarity.

ANALYSED DATA

The study is focused on Cache Bytes counter collected data from 50 different PC computers with Windows 10 64-bit, of which 31 computers had DDR3 RAM and 19 had DDR4 RAM.

Data from the Bytes cache counter was collected at a sampling rate of 1 s. An important aspect of the conducted study was to take into account the current use of computers by normal users with moments of their activity and inactivity

including computers turning on and off. Computer users were asked to perform their typical activity based on the use of: web browsers, office software, multimedia, etc. There were no specific scenarios nor any assumed test in order to avoid any artificial patterns. Each final time series is a combination of data collected during different work sessions.

Each of tested computers was characterised by a unique configuration of components, as well as unknown software environments which introduced additional diversity to the collected data. An important methodological assumption was the rejection of benchmark tests as a research and experiment tool. This choice was caused by the desire to reflect real user behaviour, instead of generating data based on artificial, often extreme system load scenarios, which could provide an incorrect interpretation of the tested counter patterns. Another important aspect is the collection of the final time series, which contain at least 100,000 samples for each computer. This will enable detailed analysis of statistical properties expressed by the Hurst exponent, showing long-term dependencies in the data. Figure 1 shows basic statistical values, such as minimum, average, and median for the collected data.

Figs. 2 and 3 show the collected Cache Bytes counter values for PC1 and PC2. Each of the studied time series was unique, which makes impossible to find similarities between the series under study using simple methods. Nevertheless, despite these differences, it is likely that the systems may exhibit similar statistical properties, especially stationarity and long-range dependencies, which require more detailed analysis.

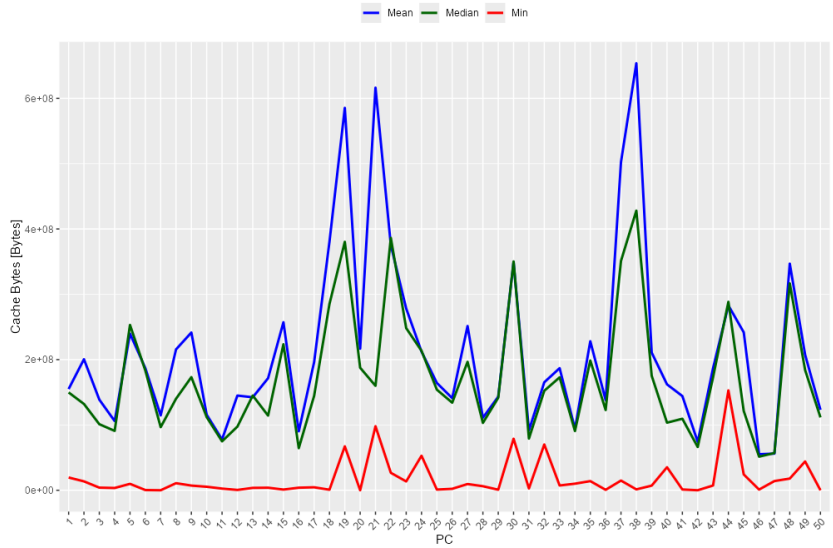

Figure 1. Basic statistical values such as minimum, average, median for the collected data sets

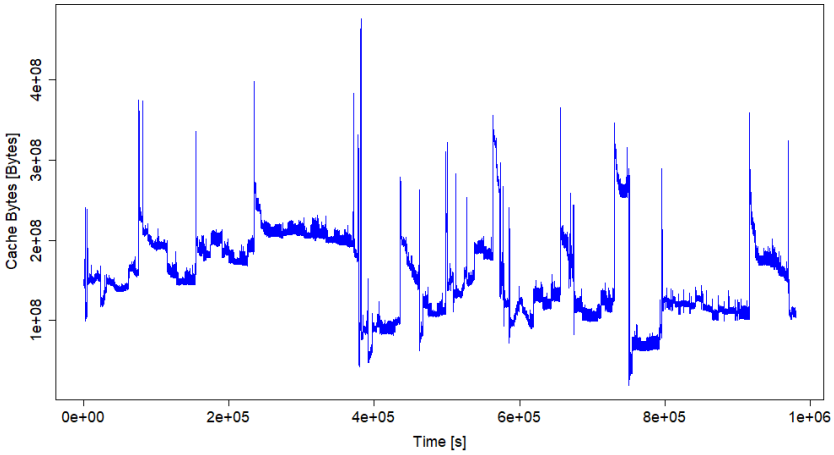

Figure 2. Collected data for Cache Bytes – computer PC1

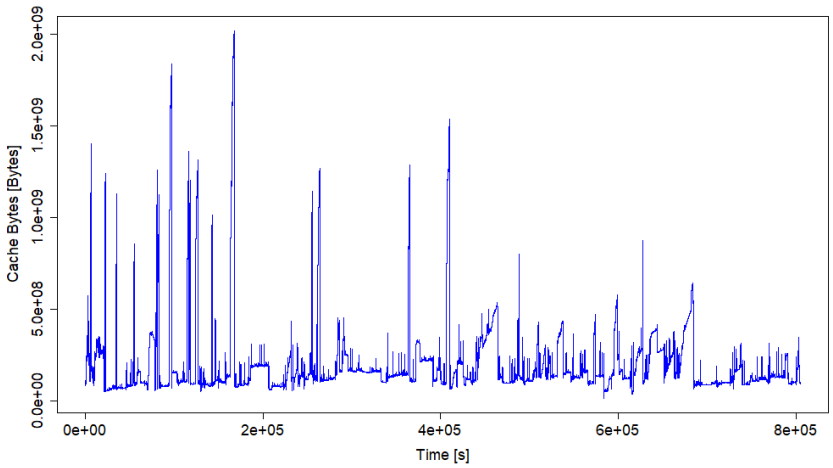

Figure 3. Collected data for Cache Bytes – computer PC2

RESULTS OF DATA ANALYSIS

The aim of the chapter is to analyse time series in terms of time series stationarity verification. The following subsections present the analysis of time series by ADF, KPSS, and PP methods to eliminate the problem of estimating stationarity by only one or two methods (the problem of incompatibility of the results of ADF and KPSS methods). The analysis was carried out for real data in the time series and for their increments.

The ADF and KPSS methods can return one of four combinations:

- ADP and KPSS indicate stationarity. Stationarity can be assumed;
- ADP indicates nonstationarity, and KPSS indicates stationarity. Therefore, the series is stationary in trend,
- ADP indicates stationarity, and KPSS indicates nonstationarity. Therefore, the series is differentially stationary.
- ADP and KPSS indicate nonstationary. Nonstationary can be assumed.

Analysis of stationarity in time series

In calculations regarding the stationarity of time series, the t-series library in the R statistical environment was used [30]. The results of the stationarity analysis are presented in Figure 4.

Stationarity analysis using the ADF, KPSS and PP tests for the time series showed large differences between the results of the ADF and KPPS tests for real data. Out of 50 time series, 44 were found to be Difference-Stationary (results of ADF and PP test indicated stationarity, whereas for KPSS tests some time series were non-stationary), while 6 series were non-stationary.

On the other hand, stationarity analysis for time series increments, except for one series, confirmed their stationarity, suggesting that the original time series probably have nonstationary parts, such as trends or seasonality, which were removed by applying increments. The conclusions drawn from this analysis suggest the necessity to consider the possible presence of time series nonstationarity for analyzed data. Its presence can

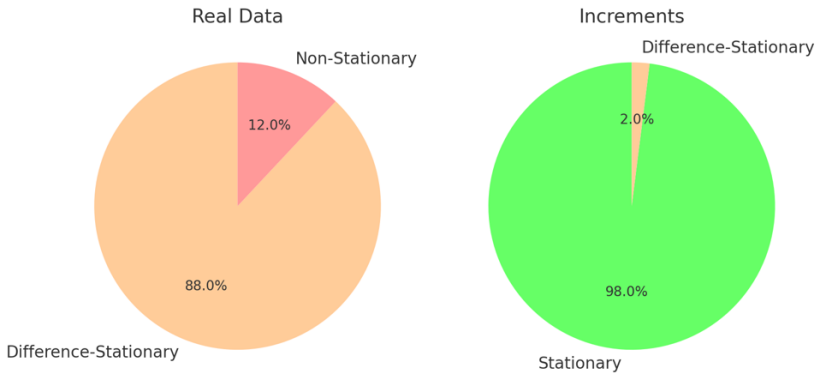

Figure 4. Results of the stationarity analysis of real data and their increments

lead to problems with results interpretation, for example when predictive models will be created.

Results of data analysis

As a part of the next study step, each time series with cache byte counter data from personal computers was subjected to detailed analysis using several Hurst exponent estimation methods.

To examine statistical long-range dependencies, i.e. the Hurst exponent, the Pracma [31] and fractal [32] packages from the CRAN repository of the R language and an original script for periodogram analysis in the R environment were used. From these two libraries, the following Hurst exponent estimation methods were selected.

- Absolute Values of Aggregated Series (aggabs() method from the Fractal package) [30],
- Aggregated Variance Method (aggVar() method from the Fractal package) [30],
- *Hs* (R/S), i.e. simplified rescaled range (R/S) approach (hurstexp() method from the Pracma package) [31],
- Periodogram method [33].

The second stage of the study was the use of the external bucket random permutation method [34], which allowed a random change in the order of individual samples in the time series. The purpose of this method was to check whether the analysed datasets contain long-range dependencies, and this is not the effect of random aggregation. This method allows distinguishing long-term dependencies in a time series from those that appear as a result of a random distribution of data. The authors of the work [2] showed that the

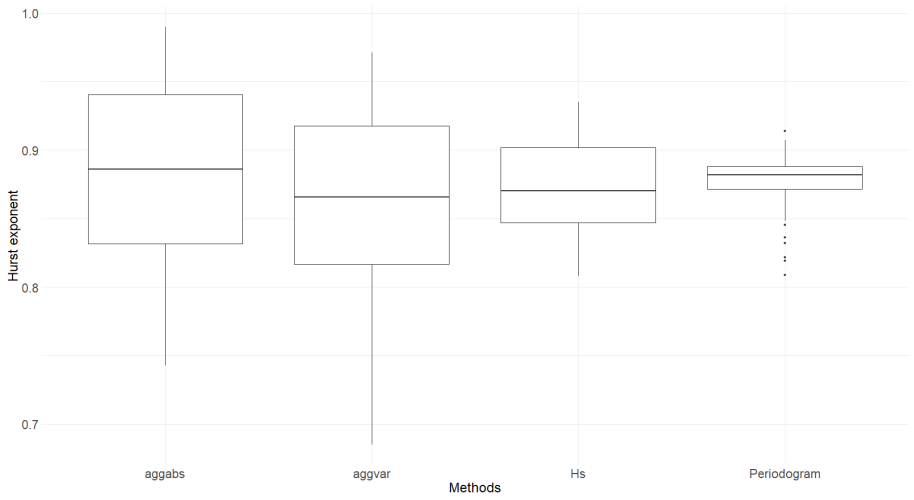

Figure 5. Hurst exponent for real data, calculations were based on four different methods: aggabs(), aggvar(), hurstexp() – Hs, periodogram

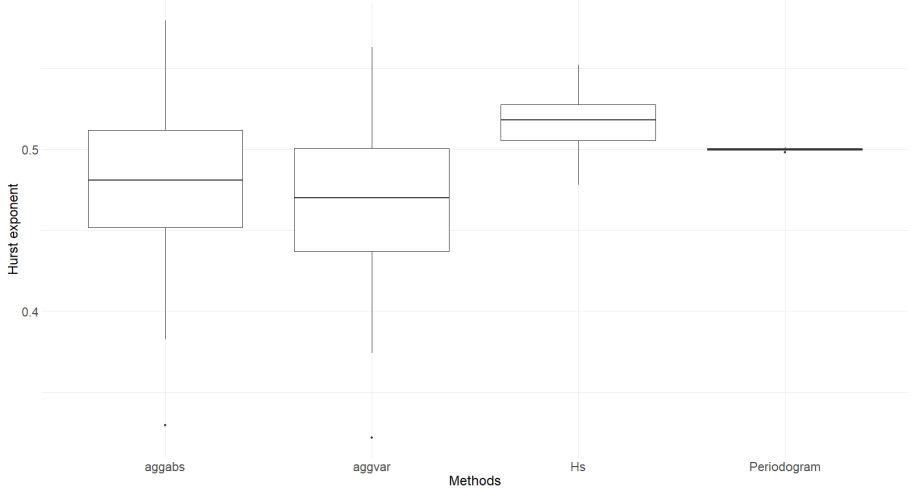

Figure 6. Hurst exponent for shuffled data

arrangement of data has an impact on the properties of a time series, such as the Hurst exponent.

Analysis of the time series showed that the Hurst exponent values for all data are above the 0.5 H-index value, suggesting strong trends and long-term dependencies in the data (Fig. 5). Depending on the used methods, the estimated values of the Hurst exponents are significantly greater than the 0.5 threshold confirming that in all investigated time series there are seep statistical long-term dependencies. This observation can also be confirmed when for the original time series a bucket shuffling will be done.

The External Bucket Random Permutation method was used to test whether the order of values in the time series influences the estimation of the Hurst exponent. In this case, the results of the Hurst exponent analysis indicate a significant reduction in Hurst values compared to the original data (Fig. 5 versus Fig. 6). For all-time series, the H values vary around the value of 0.5, suggesting the lack of long-term memory in the rearranged time series. Such series can be compared to processes similar to the white noise.

In summary, the external bucket random permutation method confirms that the high values of the Hurst exponent in the original time series are due to their specific inner structure and the order of the data, rather than a random distribution of values.

CONCLUSIONS

As was shown in the paper, it is possible to collect long time series that represent the behaviour of computer systems. In this particular case, the cache memory counter was recorded for 50 different personal computers. Having this data, selected statistical tests were done and they confirmed that long-range dependencies ($H > 0.5$) are present in the case of all analysed computers. In all cases, very high values of Hurst exponent were shown and this is another example of a system where such dependencies exist. Usually, they were observed in many natural systems but this time there are strong evidences that they exist inside computer systems.

Although the analysis of these long-term dependencies provides new insights into the behaviour of computer systems, direct optimization of cache memory management and resource allocation is not possible due to the closed architecture of MS Windows. Nevertheless, understanding these dependencies can be useful in designing performance monitoring methods and in predicting potential system loads, which could be also applicable in research on other operating systems. The whole discovery needs further studies that will be the subject of next research.

REFERENCES

1. Reiser M., Lavenberg S.S., Mean-value analysis of closed multichain queuing networks. Journal of the Association for Computing Machinery, 1980, 27(2), 313-322.
2. Ross S., Introduction to Probability Models (Eleventh Edition). Academic Press, 2014.
3. Cox D.R., A use of complex probabilities in the theory of stochastic processes. Proc. Cambridge Phil. Sac. 1955, 51, 313-319.
4. Strzalka D., Long-range dependencies and statistical self-similarity in computer memory system. Journal of Circuits, Systems and Computers, 2015, 24(3), 1550031.
5. Dymora P., Mazurek M., Anomaly Detection in IoT Communication Network Based on Spectral Analysis and Hurst Exponent. Appl. Sci. 2019, 9, 5319. https://doi.org/10.3390/app9245319
6. Tian Z., Network traffic prediction method based on wavelet transform and multiple models fusion. International Journal of Communication Systems, 2020, e4415. https://doi.org/10.1002/dac.4415
7. Paxson V., Floyd S., Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking, 1995, 3(3), 226-244, https://doi.org/10.1109/90.392383
8. Microsoft. Performance Counters. https://learn.microsoft.com/en-us/windows/win32/perfctrs/performance-counters-portal (Accessed: 26.03.2024).
9. Microsoft. Windows Performance Monitor. https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/cc749249(v=ws.11) (Accessed: 26.03.2024).
10. Microsoft. Counter Types. https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785636(v=ws.10) (Accessed: 26.03.2024).
11. Zhi-Jie Zhou, Chang-Hua Hu, Dong-Ling Xu, Jian-Bo Yang, Dong-Hua Zhou, New model for system behavior prediction based on belief rule based systems. Information Sciences, 2010, 180(24), 4834-4864.
12. Shou Z., Wang Z., Han K., Liu Y., Tiwari P. and

Di X., Long-Term Prediction of Lane Change Maneuver Through a Multilayer Perceptron. IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020.

13. Vaidyanathan P.P., Low-Noise and Low-Sensitivity Digital Filters. In: Douglas F. Elliott (Ed.), Handbook of Digital Signal Processing, Academic Press, 1987, 359-479.
14. Bal A., Ganguly D., Chatterjee K., Stationarity and self-similarity determination of time series data using hurst exponent and R/S ration analysis. In: Hassanien A.E., Bhattacharyya S., Chakrabati S., Bhattacharya A., Dutta,S. (Eds) Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, Vol. 1300, 2021.
15. Hurst H.E., Long-term storage capacity of reservoirs. Trans. Amer. Soc. Civil Eng, 1951, 116, 770-808.
16. Beran J. Statistics for Long-Memory Processes. New York: Chapman & Hall, 1994.
17. Leland W.E., Willinger W., Wilson D.V. and Taqqu M.S., On the self-similar nature of Ethernet traffic. ACM/SIGCOMM'93. Computer Communication Review, 1993, 23, 183-193.
18. Cont R., Long range dependence in financial markets. In: Lévy-Véhel J., Lutton E. (Eds) Fractals in Engineering. Springer, London 2005.
19. Taqqu S.M., Note on evaluation of R/S for fractional noises and geophysical records. Water Resources Research, 1970, 6, 349-350.
20. Zournatzidou G., Floros C., Hurst exponent analysis: Evidence from volatility indices and the volatility of volatility indices. J. Risk Financial Manag, 2023, 16, 272. https://doi.org/10.3390/jrfm16050272
21. Cont R., Das P., Rough volatility: Fact or artefact? Sankhya B, 2024, 86, 191-223. https://doi.org/10.1007/s13571-024-00322-2
22. Raczynski K., Dyer J., Variability of annual and monthly streamflow droughts over the Southeastern United States. Water, 2022, 14, 3848. https://doi.org/10.3390/w14233848
23. Tatli H., Detecting persistence of meteorological drought via the Hurst exponent. Meteorological Applications, 2015, 22(4), 763-769. https://doi.org/10.1002/met.1519
24. Tovkach S., Self-similarity of operating modes of aviation engine with the use of wireless data transmission. Advances in Science and Technology Research Journal, 2019, 13(2), 176-185.
25. Chronopoulou A., Viens F.G., Hurst index estimation for self-similar processes with long-memory. Recent Development in Stochastic Dynamics and Stochastic Analysis, 2010, 91-117.
26. Hyndman R., Athanasopoulos G., Forecasting: principles and practice. 2nd edition. OTexts, Australia, 2018.
27. Fuller W., Introduction to Statistical Time Series. New York: John Wiley and Sons, 1976.
28. Kwiatkowski D., PC Phillips, P. Schmidt, Y. Shin, Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? Journal of Econometrics, 1992, 54(1), 159-178.
29. Perron P., Ng S., Useful modifications to some unit root tests with dependent errors and their local asymptotic properties. The Review of Economic Studies, 1996, 63(3), 435-463.
30. Tseries, CRAN repository, https://cran.r-project.org/web/packages/tseries/index.html (Accessed 12.03.2024)
31. Pracma, CRAN repository, https://cran.r-project.org/web/packages/pracma/index.html (Accessed 15.03.2024)
32. Fractal, CRAN repository, https://cran.r-project.org/web/packages/fractal/index.html (Accessed 14.03.2024)
33. Yingjun L., Yong L., Kun W., Tianzi J., Lihua Y., Modified periodogram method for estimating the Hurst exponent of fractional Gaussian noise. Phys Rev E, 2009, 80, 066207.
34. Zhou Y., Taqqu M., Applying bucket random permutations to stationary sequences with long-range dependence. Fractals, 2007, 15(2), 105-126.