Machine Learning Classification of Multifractional Brownian Motion Realizations

Lyudmyla Kirichenko[1[0000-0002-2780-7993]], Bulakh Vitalii[1[0000-0002-9177-8787]], Tamara Radivilova[1[0000-0001-5975-0269]]

Kharkiv National University of Radio Electronics, Kharkiv, 61166, Ukraine
lyudmyla.kirichenko@nure.ua
bulakhvitalii@gmail.com
tamara.radivilova@gmail.com

Abstract. A comparative analysis of machine learning classification of stochastic time series based on their multifractal properties is proposed. Multifractal time series were obtained by generating realizations of fractional Brownian motion in multifractal time. The features for classification were statistical, fractal and recurrent characteristics calculated for each time series. The various machine learning classifiers were chosen for classification: bagging with classification and regression decision trees, random forest with classification and regression decision trees, fully connected perceptron and recurrent neural network. Both cumulative time series of multifractal Brownian motion and time series increments were carried out. It was shown that in general, classification accuracy is higher when using series of increments. When classifying realizations of multifractional Brownian motion, bagging and recurrent neural network showed the best accuracy.

Keywords: multifractal, multifractional Brownian motion, classification of time series, features, random forest, bagging, recurrent neural network

1 Introduction

Over the past decades, it has become apparent that many complex objects and systems have fractal (self-similar) properties. This applies to time series that reflect the dynamics of complex nonlinear systems. Numerous studies show that changes in the structure or state of a system lead to changes in fractal properties of the corresponding time series. The results of time series fractal analysis are widely used in practice, in particular, for analysis of information systems with self-similar data flows to prevent system overload and to analyze and predict financial markets [1-4].

In these cases, models of fractal processes that are used for forecasting, modeling, etc. play an important role. One of the interesting and applied in practice models is the fractional Brownian motion in the multifractal time proposed by Mandelbrot [5,6]. Currently, multifractional Brownian motion is used to model various phenomena [7–

9], among which the prevailing place is occupied by financial series [10-14] and self-similar infocommunication traffic [15].

In recent years, to solve practical problems associated with the analysis and recognition of dynamic phenomena, time series classification by machine learning has been used [16–20]. Typically, time series are collected in classes based on whether they have a common attribute or property. Often a change in the state of a system entails a change in its fractal structure. For example, telecommunication traffic under DDoS attacks change their fractal properties [21-23]. Thus, the task of classifying time series based on their fractal properties is relevant. However, the classification of time series by machine learning methods is a fairly new area and most of the studies do not take into account their fractal properties.

The objective of the work is a comparative analysis of the fractal time series classification carried out by machine learning methods. Time series are realizations of the fractional Brownian motion in the multifractal time and time series of their increments, which are divided into classes according to their fractal properties. The ensembles of decision trees and neural networks are considered as classification methods.

2 Fractal Random Processes and Models

A random process $X(t)$ is self-similar if the process $a^{-H} X\left(at\right)$ has the same finite-dimensional distribution laws with $X(t)$. The parameter H, $0 < H < 1$, is called the Hurst exponent. It is the self-similarity degree and the measure of the long-term dependence of the process. The moments of the self-similar process satisfy the scaling relation $E\left[\left|X(t)\right|^{q}\right] \propto t^{qH}$.

Multifractal random processes are inhomogeneous fractal processes and have more flexible scaling relation: $E\left[\left|X(t)\right|^{q}\right] \propto t^{\tau(q)+1}$, where $\tau(q)$ is a nonlinear function of scaling exponent [24].

One of the most used characteristics of multifractal processes and time series is the generalized Hurst exponent $h(q)$, which is associated with the function $\tau(q)$ by the ratio [25]:

$$h(q) = \frac{\tau(q)+1}{q}.$$

The value $h(q)$ at $q = 2$ corresponds to the value of Hurst exponent H. The self-similar process are monofractal, their scaling exponent $\tau(q)$ is linear.

The popular models of the multifractal processes are the stochastic conservative binomial multiplicative cascades [24]. Such multifractal models are constructed using an iterative algorithm, where the values of the cascade realization are the values of some specially selected random variable. The conservatism of the cascade consists in the fact that for any number of iterations, the sum of the cascade values remains the same.

B. Mandelbrot proposed a multifractal model of financial time series which is based on fractional Brownian motion in multifractal time by operation of subordination [5,6]. The subordination is a random substitution of time and it can be represented in the form $Z(t) = Y(T(t))$, where $T(t)$ is a nonnegative nondecreasing random process called subordinator, $Y(t)$ is a random process, independent of $T(t)$.

In [6] it is proved that if $X(t)$ is the process of subordination

$$X(t) = B_H(\theta(t)) \tag{1}$$

where $B_H(t)$ is fractional Brownian motion with Hurst exponent H and $\theta(t)$ is conservative binomial multiplicative cascades, then $X(t)$ is the multifractal process. The scaling function $X(t)$ is defined by

$$\tau_X(q) = \tau_\theta(Hq), \tag{2}$$

where $\tau_\theta(Hq)$ is the scaling function of the multiplicative cascade $\theta(t)$.

Quite often, for practical purposes, it is not interested in the time series itself, but in its increments. The series of increments $Xdif$ for time series $X(t)$ is determined by the formula

$$Xdif(t) = X(t) - X(t-1). \tag{3}$$

3 Features for classification by machine learning

One of the most important issues of the time series classification task is the selection of features by which the partition into classes is carried out. Changes the time series fractal properties entails the changes the statistical and correlation properties. Therefore, statistical, fractal and recurrent characteristics calculated from the time series were chosen as features.

The studies presented in [26, 27] showed that the statistical characteristics that reflect the change in the fractal properties of the time series are variance, coefficient of variation, median, asymmetry coefficient, etc. As features representing fractal properties, it is convenient to use the values of the Hurst exponent H and the generalized Hurst exponent $h(q)$ such as the mean and standard deviation of the generalized Hurst exponent, the specific values $h(q)$ and the range Δh.

A fairly new approach to the selection and use of time series features in machine learning is the calculation of recurrence characteristics. The recurrence plot of a time series $X(t)$ is a matrix, where an element with coordinates (i, j) characterizes the proximity of points $X(t_i)$ and $X(t_j)$ in phase space [28]. A numerical analysis of recurrence plots allows calculating the quantitative characteristics of recurrence such as a measure of recurrence, a measure of determinism, a measure of entropy, etc. These characteristics are advisable to use as features in machine learning classification [23,29,30].

4 Classification Methods

The ensemble methods of decision trees bagging and random forest, as well as neural networks, were chosen as classifiers.

The ensemble of models is a complex model consisting of separate basic models. Component models can be of the same type, or different. One of the first and most famous ensembles is bagging, based on the statistical bootstrap aggregating: multiple sample generation based on a single sample [31]. In this classification method, all elementary classifiers are trained and work independently of each other. Several samples of the same size are extracted from a single training sample, each of which is used to train one of the ensemble models. The decision is made either by voting: the class that was chosen by a simple majority of models is selected by averaging, which is defined as the average of all outputs.

The decision tree method is one of the simple and effective solutions to classification tasks in many different areas. Decision trees change quite a lot with a small change in the data sample, however, when several trees are combined into an ensemble, the spread in the values of the target variable becomes much smaller. In this paper, as one of the methods for classifying time series, was used the bagging with classification and regression decision trees. When using regression trees, the result of the classification model is the probability of matching the time series with the class.

Random Forest is a bagging method with regression or classification decision trees. However, unlike its main version, it has several features, in particular, in addition to randomly selecting learning objects, features are also randomly selected [32]. In this work, for comparison, along with the bagging, a random forest was also used with classification and regression decision trees.

Neural networks are widely used as classifiers. There are many neural network architectures designed to classify various objects. During the experiment, different neural networks were investigated and two neural network architectures were selected.

The first neural network was a fully connected perceptron of seven large layers with activation function of the ReLU type. Using the ReLU activation function is less expensive and significantly accelerates the convergence of gradient descent. After each full connected layer, the level of regularization was included in the network. As a regularization method, batch normalization was chosen [33], which is used to stabilize the neural network and prevent the effect of overtraining. The second network contained recurrent levels to take into account the relationship between elements. For both neural networks, the Adam stochastic optimization method (Adaptive Moment Estimate) [34] was used.

5 Experiment Description

To simulate multifractional Brownian motion $X(t)$ realizations the generation of multifractal cascades $\theta(t)$ with $Beta(\alpha, \alpha)$-distribution [27] were used. They were subordinates for the fractional Brownian motion $B_H(t)$ by (1). In the case when the cascade weight coefficients are the values of $Beta(\alpha, \alpha)$-distribution, the scaling exponent $\tau(q)$ is uniquely determined by the value of the Hurst exponent H, $0.5 < H < 1$ [35].

When specific $Beta(\alpha, \alpha)$ - distribution for the multifractal cascades was set and the specific Hurst exponent of fractional Brownian motion was selected, subordinated processes $X(t)$ with needed Hurst exponent H which is determined by (2) was obtained.

In this paper, each class was a set of model time series of multifractional Brownian motion $X(t)$ with the Hurst exponent H belonging to the same range of values. For each time series, the Hurst exponent was chosen randomly within the appropriate range. The values of the Hurst exponent are changing in the range from 0.5 to 1 with a step 0.05. The minimum and maximum values of the Hurst exponent were selected 0.51 and 0.99, respectively. Thus, the training of models was carried out in 10 classes, where $H \in \{[0.51, 0.55), [0.55, 0.6), [0.6, 0.65), \dots, [0.9\ 0.95), [0.95, 0.99]\}$.

Figure 1 shows the realizations of cascade processes $\theta(t)$ (top) and corresponding realizations of multifractional Brownian motion $X(t)$ of different classes (bottom).

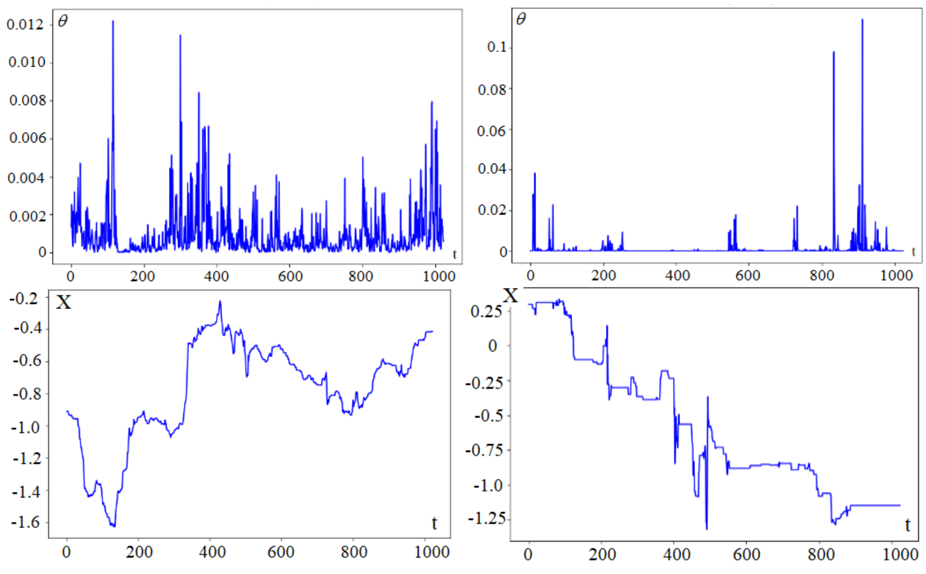

Fig. 1. Realizations of cascade processes (top) and corresponding realizations of multifractional Brownian motion (bottom)

For each multifractional Brownian motion realization, the realization of its increments was obtained by (3) and classification of increment time series was also carried out by a separate experiment. Figure 2 shows corresponding realizations of increments of multifractional Brownian motion, which shown in Fig. 1.

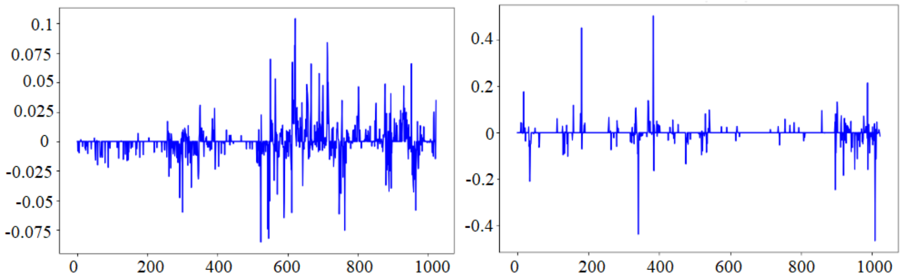

Fig. 2. Realizations of increments of multifractional Brownian motion

Thus, each class of time series is a set of realizations of multifractional Brownian motion or their increments with the same multifractal properties. To classify the statistical, fractal and recurrent characteristics of time series were used as features. The obtained features were the inputs of each of 6 classifiers: bagging with classification and regression decision trees, random forest with classification and regression decision trees, fully connected perceptron and recurrent neural network.

The research was conducted for a time series of different lengths: 512, 1024, 2048 and 4096 values. Such length is associated with the method of generating realizations of the binomial stochastic cascade. Model training for each class was carried out on 300 examples of training time series and was tested on 150 test ones.

6 Results and discussion

To implement bagging and random forest methods and neural networks, Python with libraries of machine learning methods was used [36]. The results of the classification of multifractional Brownian motion realizations indicate different classification accuracy for different classifiers. The best practice was the bagging with regression trees and a recurrent neural network. The worst results were shown by the Random forest with classification trees and a fully connected perceptron. Fig. 3 shows the histograms of the probability distribution of matching to class number for each value of the Hurst exponent for classifying time series with a length of 1024 values by the recurrent neural network. Such distributions are typical for all variants of classification.

Table 1 presents the average probabilities of class determining depending on the length of time series and the method of classification. The dependence of the classification accuracy on the length of the time series is obvious since the longer the series, the more accurately its fractal and recurrent characteristics are considered. Starting from the length of 2048 values, the probability of a correct class definition for bagging, random forest with regression trees and the recurrent neural network becomes greater than 0.9.

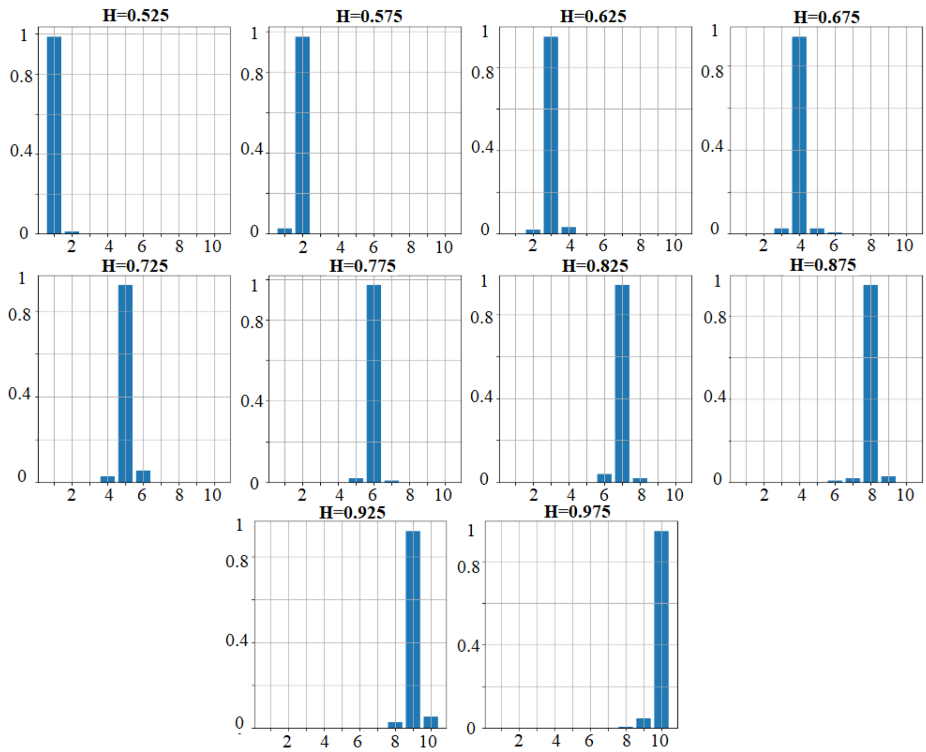

Fig. 3. Distribution of probabilities of the class number determining depending on the value of the Hurst exponent

The classification performed on the increments realizations of the multifractional Brownian motion showed better results. In this case, it was sufficient to use only the statistical and fractal characteristics of time series without building recurrence plots and calculating recurrence characteristics. This significantly reduces the training time and the structure of classifiers.

Table 1. The average probability of class determination for the cumulative realizations

Length of time series	Bagging		Random forest		Neural network	
	classification trees	regression trees	classification trees	regression trees	perceptron	recurrent
512	0.547	0.564	0.575	0.592	0.587	0.594
1024	0.781	0.796	0.631	0.685	0.627	0.785
2048	0.918	0.936	0.769	0.933	0.698	0.940
4096	0.956	0.961	0.957	0.962	0.764	0.969

Table 2 presents the average probabilities of class determination depending on the time series length and the method of classification for the increments realizations. The high probability of the correct class determination quickly is achieved even for relatively small lengths of time series.

Table 2. The average probability of class determination for the increments realizations

Length of time series	Bagging		Random forest		Neural network	
	classification trees	regression trees	classification trees	regression trees	perceptron	recurrent
512	0.899	0.900	0.903	0.899	0.791	0.744
1024	0.985	0.986	0.982	0.986	0.949	0.957
2048	0.995	0.995	0.996	0.995	0.970	0.973
4096	0.998	0.998	0.998	0.998	0.979	0.981

Conclusion

The multifractal time series were classified by machine learning methods. Time series were obtained by generating realizations of fractional Brownian motion in multifractal time. Both cumulative time series of multifractal Brownian motion and series of increments were carried out.

Time series were divided into classes according to their multifractal properties. The classification was carried out on the basis of quantitative features calculated for each time series. The features for classification were statistical, fractal and recurrent characteristics. Such classifiers were chosen for classification: bagging with classification and regression decision trees, random forest with classification and regression decision trees, fully connected perceptron and recurrent neural network.

The results of the research have shown that the accuracy of the classification is higher when the increments time series were classified. When classifying cumulative realizations of multifractional Brownian motion, method of bagging with regression trees and recurrent neural network showed the best accuracy.

In future research it worth to concentrate on the classification of real multifractal time series using different classification algorithms.

References

1. Brambila, F.: Fractal Analysis – Applications in Physics, Engineering and Technology. https://www.intechopen.com/books/fractal-analysis-applications-in-physics-engineering-and-technology

2. Shelukhin, O.I.; Smolskiy, S.M.; Osin, A.V.: Self-Similar Processes in Telecommunications. John Wiley & Sons, New York, USA, 320 p. (2007).
3. Peters, E. E.: Fractal Market Analysis: applying chaos theory to investment and economics. Wiley (2003).
4. Daradkeh, Y. I., Kirichenko, L., Radivilova, T.: Development of QoS methods in the information networks with fractal traffic. International Journal of Electronics and Telecommunications **64(1)**, 227-232 (2018). doi: 10.24425/118142
5. Mandelbrot, B. B., Fisher, A., Calvet, L.: A multifractal model of asset returns Cowles Foundation Discussion. Sauder School of Business Working Paper, Paper **1164**, 1–33 (1997).
6. Calvet, L. E., Fisher, A. J., Mandelbrot, B. B.: Large deviations and the distribution of price changes. Cowles Foundation Discussion Paper, Yale University **1165**, 1–28 (1997).
7. Ayache, A., Cohen, S., Véhel, J. L.: The covariance structure of multifractional Brownian motion, with application to long range dependence. In: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (Cat. No. 00CH37100), vol. 6. IEEE (2000). doi: 10.1109/ICASSP.2000.860233
8. Pierre, R. B., Dury, M. E., Haouas, N.: Selection of sparse multifractional model. https://hal.archives-ouvertes.fr/hal-01194347.
9. Ahn, K. I., Lee, K.: Identification of nonstandard multifractional brownian motions under white noise by multiscale local variations of its sample paths. Mathematical Problems in Engineering **2013** (2013). doi: https://doi.org/10.1155/2013/794130
10. Corlay, S., Lebovits, J., Véhel, J. L.: Multifractional stochastic volatility models. Mathematical Finance **24(2)**, 364-402 (2014). doi: https://doi.org/10.1111/mafi.12024
11. Bianchi, S.: Pathwise identification of the memory function of multifractional Brownian motion with application to finance. International Journal of Theoretical and Applied Finance **8(02)**, 255-281 (2005). doi: https://doi.org/10.1142/S0219024905002937
12. Muniandy, S. V., Lim, S. C.: Modeling of locally self-similar processes using multifractional Brownian motion of Riemann-Liouville type. Physical Review E **63(4)**, 046104 (2001). doi: 10.1103/PhysRevE.63.046104
13. Fauth, A., Tudor, C. A.: Multifractal random walks with fractional Brownian motion via Malliavin calculus. IEEE Transactions on Information Theory **60(3)**, 1963-1975 (2014).
14. Günay, S.: Performance of the Multifractal Model of Asset Returns (MMAR): evidence from emerging stock markets. International Journal of Financial Studies **4(2)**, 11 (2016). doi:10.3390/ijfs4020011
15. Li, M., Zhao, W., Chen, S.: MBm-based scalings of traffic propagated in internet. Mathematical Problems in Engineering **2011** (2011). doi:10.1155/2011/389803
16. Esling, P.; Agon, C.: Time series data mining. ACM Computing Surveys **45**, 12:1–12:34 (2012).
17. Kirichenko, L., Radivilova, T., Zinkevich, I.: Comparative Analysis of Conversion Series Forecasting in E-commerce Tasks. In: Shakhovska N., Stepashko V. (eds) Advances in Intelligent Systems and Computing II. CSIT 2017. Advances in Intelligent Systems and Computing, vol 689. Springer, Cham (2018). doi: https://doi.org/10.1007/978-3-319-70581-1_16
18. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., Muller, P. A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33(4), 917-963 (2019).
19. Kirichenko, L., Radivilova, T., Tkachenko, A.: Comparative Analysis of Noisy Time Series Clustering. In: 3rd International Conference on Computational Linguistics and Intelli-

gent Systems (COLINS-2019) Proceedings 2362, 2019 April 18-19, pp.184-196. Kharkiv, Ukraine (2019)

20. Buza, K.: Time series classification and its applications. In: 8th International Conference on Web Intelligence, Mining and Semantics Proceedings, pp.1-4 (2018). doi: https://doi.org/10.1145/3227609.3227690
21. Kaur, G., Saxena, V., Gupta, J.: Detection of TCP targeted high bandwidth attacks using self-similarity. Journal King Saud University Computer and Information Sciences (2017).
22. Deka, R., Bhattacharyya, D.: Self-similarity based DDoS attack detection using Hurst parameter. Security Communication Networks **9**, 4468–4481 (2016).
23. Radivilova, T., Kirichenko, L., Ageiev, D., Bulakh, V.: Classification Methods of Machine Learning to Detect DDoS Attacks. In: 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) Proceeding Vol.1, pp. 207-210. IEEE, Metz (2019). doi: 10.1109/IDAACS.2019.8924406
24. Riedi, R. H.: Multifractal Processes. https://www.researchgate.net/publication/2839202_Multifractal_Processes
25. Kantelhardt, J.W.: Fractal and Multifractal Time Series. http://arxiv.org/abs/0804.0747
26. Bulakh, V., Kirichenko, L., Radivilova, T.: Time Series Classification Based on Fractal Properties. In: 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) Proceedings, 21–25 August 2018; pp. 198–20. IEEE, Lviv (2018). doi: 10.1109/DSMP.2018.8478532
27. Kirichenko, L., Radivilova, T., Bulakh, V.: Machine Learning in Classification Time Series with Fractal Properties. Data **4(1) 5**, 1-13 (2019). doi:10.3390/data4010005
28. Marwan, N., Romano, M., Thiel, M., Kurths, J.: Recurrence plots for the analysis of complex system. Physics Reports **438(5–6)**, 237-329 (2007). doi: https://doi.org/10.1016/j.physrep.2006.11.001
29. Kirichenko, L., Radivilova, T., Bulakh, V.: Binary Classification of Fractal Time Series by Machine Learning Methods. In: Lytvynenko V., Babichev S., Wójcik W., Vynokurova O., Vyshemyrskaya S., Radetskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. Advances in Intelligent Systems and Computing 1020, 701-711 (2020). doi: https://doi.org/10.1007/978-3-030-26474-1_49
30. Kirichenko, L., Radivilova, T., Bulakh, V.: Classification of Fractal Time Series Using Recurrence Plots. In: 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), pp. 719-724. IEEE, Kharkiv (2018). doi: 10.1109/INFOCOMMST.2018.8632010
31. Breiman, L.: Bagging predictors. Machine Learning **24 (2)**, 123–140 (1996).
32. Breiman, L.: Random Forests. Machine Learning **45 (1)**, 5–32 (2001).
33. Ioffe, S. Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: 32nd International Conference on Machine Learning Proceeding, PMLR 37, pp.448-456. Lille, France (2015). https://arxiv.org/abs/1502.03167
34. Kingma, D. P., Ba, J.: Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations (ICLR) Proceeding, San Diego, USA (2015). https://arxiv.org/abs/1412.6980
35. Bulakh, V., Kirichenko, L., Radivilova, T.: Classification of Multifractal Time Series by Decision Tree Methods. In: 14th International Conference ICTERI 2018 ICT in Education, Research, and Industrial Applications Proceeding, 14–17 May 2018, pp. 1–4. Kyiv, Ukraine (2018).
36. Cielen, D., Meysman, A., Ali, M.: Introducing data science: big data, machine learning, and more, using Python tools. Manning Publications Co Manning Publications (2016).