ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse139922020.pdf
https://doi.org/10.30534/ijatcse/2020/139922020

Algorithm for Searching and Analyzing Abnormal Observations of Statistical Information Based on The Arnold – Kolmogorov – Hecht-Nielsen Theorem

[1,2]Leonid N. Yasnitsky
[1]Perm State University, Bukirev Street, 15, Perm, 614600, Russia
[2]National Research University Higher School of Economics, Studencheskaya Street, 38, Perm, 614070, Russia

ABSTRACT

An algorithm for detecting runouts in statistical information is proposed in this article. The idea of the algorithm is based on the property of some neural networks to demonstrate a large error in examples during training, which are runouts. For example, if a perceptron-type neural network has a relatively small number of hidden neurons, and if there are relatively few runouts in the training sample, then the neural network usually demonstrates a higher training error after the training procedure on the examples that are runouts than on nonrunout examples. However, two extreme cases are possible. On the one hand, if a neural network has too many degrees of freedom, it is usually well trained and demonstrates small values of the training error in all examples during training, including examples that are runouts. This is why a neural network with a large number of hidden neurons is not suitable for detecting runouts. On the other hand, if a neural network has too few degrees of freedom, it will demonstrate large values of the error both in runout examples and in examples that are not runouts after the training procedure. As such, it is also not suitable for detecting runouts. According to the proposed algorithm, a special neural network is designed using the formula obtained based on the relation derived from the Arnold – Kolmogorov – Hecht-Nielsen theorem. This special neural network is designed only for detecting and identifying outliers. Another neural network is being designed or other analysis methods are used for further data analysis. The proposed algorithm is intended for nonlinear subject areas described by small volumes of statistical samples that do not necessarily satisfy the normal distribution law. The application of the algorithm turned out to be efficient in solving a wide range of problems from various subject areas, such as medicine, economics, forensics, etc.

Key words: About four key words or phrases in alphabetical order, separated by commas.

1. INTRODUCTION

An abnormal observation or a "runout" is an observation that differs sharply from other sample members by its parameters. F.E. Grubbs [1] notes two types of runouts:

1. "Runout may be an extreme manifestation of the properties of the domain area under study. In this case, this observation should be saved and processed in the same way as all sample members." Let us call such observations runouts of type 1.2. "The runout may be a result of erroneous measurements or estimations, or errors in the recording of numerical values. It is advisable to conduct additional research to establish the cause of the anomalous value in such cases. If this is the reason, this observation can be removed from the sample." Let us call such observations runouts of type 2.

Many methods for controlling runouts are described in the modern literature, but there is no universal method suitable for all domain areas.

A diagnostic method is popular in linear regression analysis, according to which observations are selected that cause the greatest change in regression [2, 3], when this observation is excluded from the evaluation procedure.

Attempts to use neural networks are made in cases where there is no a priori information about the law of the probability distribution density of the process and its parameters (expectation, variance, correlation function), as well as when there are not enough data and the processes under study are described by a high degree of nonlinearity. Training methods and paradigms of neural networks insensitive to runouts are being developed [4-9]. Factographic search methods are being developed and applied [10]. Special neural network algorithms designed to detect runouts are offered. For example, neural networks with one hidden layer and sigmoid activation functions are used in [11] to detect runouts. The use of replicative neural networks for detecting runouts was reported in [12, 13]. A nonlinear runout detection procedure was presented in [14], based on the analysis of differences in the results obtained using the least square method and neural networks. Attention in the review writing [15] is drawn to the problem of determining the optimal number of neurons in neural networks of various types designed to detect runouts. However, specific recommendations for the optimal choice of the structure of neural networks are not provided in these writings, which makes it difficult to effectively use this method of detecting runouts. The fact is that neural networks with a large number of synaptic connections (neurons, degrees of freedom), as a rule, provide equally small learning errors both for examples that are runouts and for examples that are not runouts. In this regard, it is not possible to detect runout by analyzing neural network learning errors. Moreover, the use of neural networks with a small number of neurons also does not allow sufficient high-quality detection of runouts.

In the present work, an attempt is made to obtain a formula that allows determining the optimal number of neurons in an auxiliary neural network designed to detect runouts of statistical information. This formula is obtained using the corollary of the Arnold – Kolmogorov – Hecht – Nielsen theorem [16, 17]**.**

2. METHODS

The idea of the algorithm offered by the authors is based on the property of some neural networks to demonstrate a large error in examples during training, which are runouts. For example, if a perceptron-type neural network with sigmoid activation functions has a relatively small number of hidden neurons, and if there are relatively few runouts in the training sample, then the neural network usually demonstrates a higher training error after the training procedure on the examples that are runouts than on nonrunout examples.

Naturally, the following question arises: how many hidden neurons should a neural network have in order for its ability to detect runouts (i.e., to demonstrate the greatest training error in examples that are runouts) be demonstrated best? It is clear that the answer to this question depends on the individual characteristics of the sample, and therefore, it must be solved individually for each specific sample, which is difficult. The following technique is suggested to facilitate the solution of this issue.

A formula is known in the theory of neural networks [16, 17] that is a consequence of the Arnold – Kolmogorov – Hecht-Nielsen theorem:

$$\frac{N_y Q}{1+\log_2(Q)} \le N_w \le N_y\left(\frac{Q}{N_x}+1\right)\left(N_x+N_y+1\right)+N_y \tag{1}$$

Here N_x is the number of neurons of the input layer; N_y is the number of neurons of the output layer; Q is the number of elements of the training set; and N_w is the recommended number of synaptic connections of the neural network that ensure its optimal generalizing properties. If the neural network has one hidden layer, then the recommended number of hidden layer neurons can be found for it using the following formula:

$$N = \frac{N_w}{N_x + N_y} \tag{2}$$

Having expressed N_w from formula (2), substituting it into inequality (1) and dividing both parts of the inequality by $N_x + N_y$, a formula is obtained for determining the number of neurons in the hidden layer of a double layer neural network:

$$N_{\min} < N < N_{\max}\,, \tag{3}$$

where

$$N_{\min} = \frac{N_y Q}{\left(1+\log_2(Q)\right)\left(N_x + N_y\right)} \tag{4}$$

$$N_{\max} = \frac{N_y}{N_x + N_y}\left(\frac{Q}{N_x}+1\right)\left(N_x+N_y+1\right)+N_y \tag{5}$$

According to the relation derived from the Arnold – Kolmogorov – Hecht-Nielsen theorem, the optimal number of neurons in a neural network ensuring its best generalizing properties (minimum generalization error) lies in the interval between $N_{\min}$ and $N_{\max}$. The middle of this interval is often chosen in practice.

The observations indicated that a double layer perceptron was usually suitable for detecting runouts, if the number of its hidden neurons also lied within the interval determined by inequality (3) but closer to its lower boundary. In this regard, the authors have proposed to modify the Arnold – Kolmogorov – Hecht-Nielsen formula, replacing inequality (1) with the formula:

$$N = N_{\min} + \xi(N_{\max} - N_{\min}) \tag{6}$$

where ξ is an empirical coefficient, the optimal value of which is proposed to be selected for each specific sample by repeatedly executing the algorithm presented in Figure 1, sequentially setting the ξ value from 0 to 0.5 in increments of 0.1.

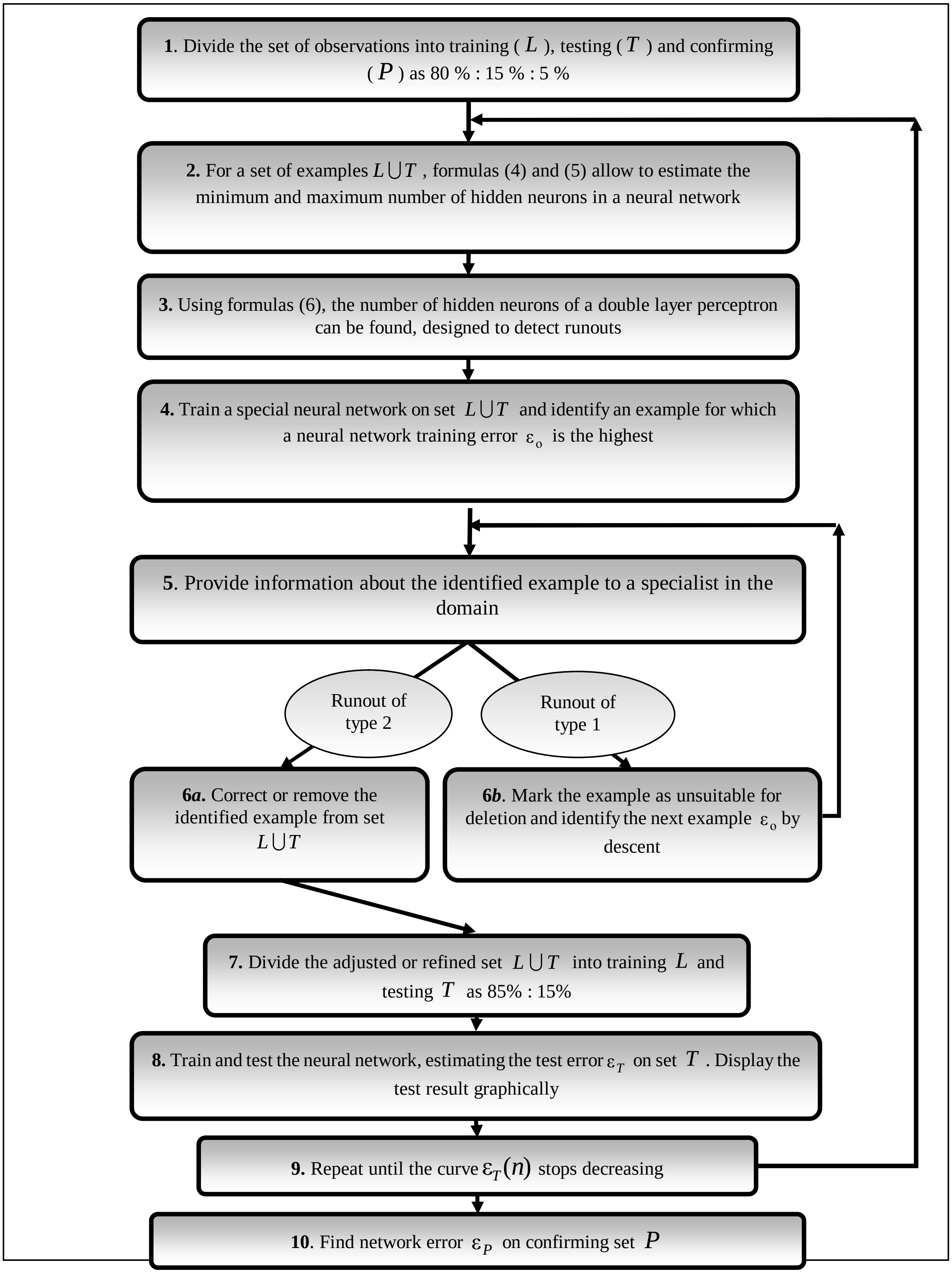
1. Divide the set of observations into training (L), testing (T) and confirming (P) as 80 % : 15 % : 5 %
2. For a set of examples $L \cup T$, formulas (4) and (5) allow to estimate the minimum and maximum number of hidden neurons in a neural network
3. Using formulas (6), the number of hidden neurons of a double layer perceptron can be found, designed to detect runouts
4. Train a special neural network on set $L \cup T$ and identify an example for which a neural network training error ε_o is the highest
5. Provide information about the identified example to a specialist in the domain
Runout of type 2
Runout of type 1
6a. Correct or remove the identified example from set $L \cup T$
6b. Mark the example as unsuitable for deletion and identify the next example ε_o by descent
7. Divide the adjusted or refined set $L \cup T$ into training L and testing T as 85% : 15%
8. Train and test the neural network, estimating the test error ε_T on set T . Display the test result graphically
9. Repeat until the curve $\varepsilon_T(n)$ stops decreasing
10. Find network error ε_P on confirming set P

Figure 1: Flowchart for sequential runout detection and analysis

According to this flowchart, the algorithm includes the following items:

1. Divide the set of observations into training (L), testing (T) and confirming (P) as 80 % : 15 % : 5 %.

2. For a set of examples $L \cup T$, formulas of the relation derived from the Arnold – Kolmogorov – Hecht-Nielsen theorem (4), (5) allow to estimate the minimum and

maximum number of hidden neurons in a neural network.

3. Using formulas (6), the number of hidden neurons of a double layer perceptron can be found, designed to detect runouts.

4. Train a special neural network on set $L \cup T$ and identify an example for which a neural network training error ε_o is the highest.

5. Provide information about the identified example to a specialist in the domain to solve the question of whether the detected abnormal observation is a runout of type 1 or 2.

6*a*. If the identified example is a runout of type 2, then correct it (if it is possible to correct the error) or remove the identified example from set $L \cup T$ and proceed to the next step 7.

6*b*. If the identified example is a runout of type 1, then mark it as unsuitable for deletion and identify the next example $L \cup T$ by descent using a special perceptron and proceed to step 5.

7. Divide the adjusted or refined set $L \cup T$ into training L and testing L as 85 % : 15 %.

8. Train and test the neural network, estimating the test error ε_T on set T. Display the test result graphically as on Figure 2.

9. Repeat steps 2 – 9 until the curve in Figure 2 stops decreasing with each new iteration *n*.

10. Find network error ε_P on confirming set P.

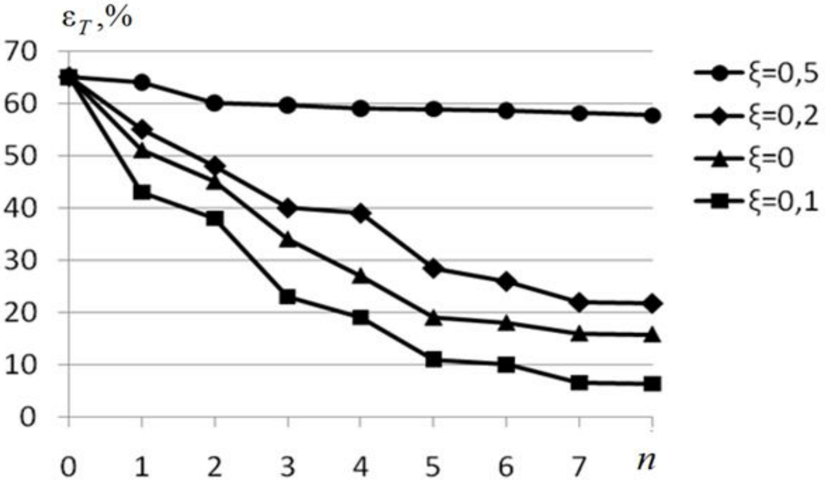

Figure 2: Approximate dependences of the test error on the empirical coefficient and on the number of iterations in steps 2 – 8 of the proposed algorithm

As can be seen from Figure 2, the quality of the neural network obtained this way depends on the value of the coefficient ξ. In the example shown in the figure, the best coefficient ξ value turned out to be 0.1. This means that any deviation from this optimal coefficient ξ value upward or downward results in an increase in network test error ε_T (as well as ε_P). This optimal coefficient ξ value was obtained in the writings on the creation of a neural network lie detector [18], where the application of the proposed algorithm in the analysis of polygraph surveys allowed to reduce the error of neural networks by 20 % to 80 %, depending on the statistical sample used. The same optimal coefficient ξ value was recorded when creating a neural network system for diagnosing and predicting the course of diseases of the cardiovascular and gastroenterological systems [19-23].

For other domain areas explored in [24-29], the optimal value of the coefficient ξ differs from 0.1, but usually does not go beyond the interval [0; 0.2]. In any case, it can be refined by building curves similar to the curves of Figure 2.

3. RESULTS

It must be noted that the experience of implementing projects by the Perm branch of the Scientific Council of the Russian Academy of Sciences on the method of artificial intelligence (www.PermAi.ru) [28] indicated that the attempts to build neural network models without detecting, correcting or eliminating runouts of type 2 in some cases had failed to give positive results, i.e., the errors of neural networks could not be reduced to any values acceptable for practical application. As such, the authors conclude that the algorithm for detecting, correcting, and eliminating runouts proposed in this article is useful not only as a tool to increase the accuracy of neural network models, but also as a way to expand the application capabilities of neural network technologies. Some well-established computer programs were created based on this algorithm and partially made available on the website www.PermAi.ru. Their creation secured a scientific priority in the application of neural network technologies in industry, economics, medicine, psychology, sociology, forensics, sports, etc. [28].

For example, the representatives of the above scientific school of artificial intelligence achieved the following:

- they first created a neural network lie detector and proved the efficiency of its use [18];
- they first created intelligent medical systems capable of not only diagnosing diseases, but also predicting their appearance and development over time, as well as selecting the best courses of treatment and prevention of diseases [19-23];
- they first created a neural network system for assessing the value of urban apartments that was self-adaptive to space and time, i.e., suitable for use in various regions of Russia and capable of adapting to volatile economic environment in the region, country, and the world [25, 27];
- they first demonstrated the possibility of using neural networks in investigative practice to detect serial killer maniacs [24];
- they were among the first to use neural networks for predicting and optimizing box office for movies [26]; and
- they were among the first to use neural networks to predict the results of voting and develop recommendations for improving the ranking of political figures; to diagnose aircraft engines; to identify an individual's abilities in business, scientific and managerial activities, predispositions for drug addiction and alcoholism, etc. [28, 29].

4. CONCLUSION

An algorithm for detecting runouts of statistical information has been proposed, which is characterized by the use of an auxiliary neural network specially developed for this purpose and designed using the mathematical formula proposed by the authors on the basis of the corollary of the Arnold-Kolmogorov-Hecht-Nielsen theorem.

The proposed algorithm is intended for detecting runouts of statistical information in domain areas described by small volumes of statistical samples not necessarily satisfying the law of normal distribution and not necessarily obeying linear laws.

As indicated in the article, the experience of the authors in the implementation of neural network projects has revealed that the proposed algorithm was useful not only as a tool to increase the accuracy of neural network models, but also as a way to expand the application capabilities of neural network technologies.

ACKNOWLEDGEMENT

The material was funded from the RFBR grant 19-010-00307 "Development of an intelligent self-adaptive system for mass valuation and scenario forecasting of the market value of residential housing in the regions of the Russian Federation".

REFERENCES

1. F.E. Grubbs, “Procedures for Detecting Outlying Observations in Samples”, Technometrics, vol. 11, no. 1, 1969, pp. 1-21. https://doi.org/10.1080/00401706.1969.10490657
2. A. Atkinson, Plots, Transformations and Regression, Oxford University Press, Oxford, 1985.
3. R.D. Cook and S. Weisberg, Residuals and Influence in Regression, Chapman & Hall, London, 1982.
4. C.H. Aladag, E. Egrioglu, and U. Yolcu, “Robust multilayer neural network based on median neuron model”, Neural Computing and Applications, vol. 24, 3-4, 2014, pp. 945-956. https://doi.org/10.1007/s00521-012-1315-5
5. Z. Wang and B.S. Peterson, “Constrained least absolute deviation neural networks”, IEEE Transactions on Neural Networks and Learning Systems, vol. 19, no. 2, 2008, pp. 273-283. https://doi.org/10.1109/TNN.2007.905840
6. C.C. Chuang and J.T. Jeng, “CPBUM neural networks for modeling with outliers and noise”, Applied Soft Computing Journal, vol. 7, no. 3, 2007, pp. 957-967.
7. H. Ferdowsi, S. Jagannathan, and M. Zawodniok, “An online outlier identification and removal scheme for improving fault detection performance”, IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 5, 2014, pp. 908-919.
8. D.S. Chen and R.C. Jain, “A robust back-propagation learning algorithm for function Approximation”, IEEE Trans. Neural Networks, vol. 5, 1994, pp. 467-479. https://doi.org/10.1109/72.286917
9. G. Beliakov, A. Kelarev, and J. Yearwood, “Derivative-free optimization and neural networks for robust regression”, Optimization: A Journal of Mathematical Programming and Operations Research, vol. 61, no. 12, 2012, pp. 1467-1490. https://doi.org/10.1080/02331934.2012.674946
10. S. Kulik, “Model for evaluating the effectiveness of search operations”, Journal of ICT Research and Applications (ITB Journal of Information and Communication Technology), vol. 9, no. 2, 2015, pp. 177-196.
11. P. Sykacek, “Equivalent error bars for neural network classifiers trained by Bayesian inference”, in Proceedings of the European Symposium on Artificial Neural Networks, Bruges, 1997, pp. 121-126.
12. S. Hawkins, H. He, G. Williams, and R. Baxter, “Outlier detection using neural networks”, in Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK02), 2002, pp. 170-180. https://doi.org/10.1007/3-540-46145-0_17
13. G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, “A comparative study of RNN for outlier detection in

data mining", in Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM02), 2002, pp. 709-712.

14. H. Garces and D. Sbarbaro, "Outliers detection in environmental monitoring databases", Engineering Applications of Artificial Intelligence, vol. 24, no. 2, 2011, pp. 341-349. https://doi.org/10.1016/j.engappai.2010.10.018
15. C. Beguin, R. Chambers, and B. Hulliger, "Evaluation of edit and imputation using robust methods", in Methods and Experimental Results from the Euredit Project, chapter 2, 2002.
16. Hecht-Nielson, R.: Kolmogorov's maping neural network existence theorem. In: Proceedings of the First IEEE International Conference on Neural Networks (San Diego, CA. 1987), vol. 3, pp. 11-14.
17. S. Haykin, Neural networks: A comprehensive foundation (2nd ed.), Prentice Hall International, Inc., New Jersey, 1999.
18. L.N. Yasnitsky, Z.I. Sichinava, and F.M. Cherepanov, Neyrosetevoy detektor lzhi: printsipy postroyeniya i opyt razrabotki [Neural network lie detector: principles and development experience], LAP LAMBERT Academic Publishing GmbH & Co. KG., Saarbrucken (Germany), 2012.
19. L.N. Yasnitsky, A.A. Dumler, K.V. Bogdanov, A.N. Poleschuk, F.M. Cherepanov, T.V. Makurina, and S.V. Chugaynov, "Diagnosis and Prognosis of Cardiovascular Diseases on the Basis of Neural Networks", Biomedical Engineering, vol. 47, no. 3, 2013, pp. 160-163. https://doi.org/10.1007/s10527-013-9359-0
20. L.N. Yasnitsky, A.A. Dumler, A.N., Poleshchuk, C.V. Bogdanov, and F.M. Cherepanov, "Artificial Neural Networks for Obtaining New Medical Knowledge: Diagnostics and Prediction of Cardiovascular Disease Progression", Biology and Medicine, vol. 7, no. 2, 2015, BM-095-15.
21. L.N. Yasnitsky, A.A. Dumler, and F.M. Cherepanov, "The Capabilities of Artificial Intelligence to Simulate the Emergence and Development of Diseases, Optimize Prevention and Treatment Thereof, and Identify New Medical Knowledge", Journal of Pharmaceutical Science and Research, vol. 10, no. 9, 2018, pp. 2192-2200.
22. L.N. Yasnitsky, A.A. Dumler, and F.M. Cherepanov, "Dynamic Artificial Neural Networks as Basis for Medicine Revolution", Advances in Intelligent Systems and Computing, vol. 850, 2019, pp. 351-358.
23. O.V. Khlinova, L.N. Yasnitsky, and I.V. Skachkova, "Neural Network System for Medical Diagnostic of Gastrointestinal Diseases", Advances in Intelligent Systems and Computing, vol. 850, 2019, pp. 359-365.
24. L.N. Yasnitsky, S.V. Vauleva, D.N. Safonova, and F.M. Cherepanov, "The use of artificial intelligence methods in the analysis of serial killers' personal characteristics", Criminology Journal of Baikal National University of Economics and Law, vol. 9, no. 3, 2015, pp. 423-430.
25. L.N. Yasnitsky and V.L. Yasnitsky, "Technique of design for integrated economic and mathematical model for mass appraisal of real estate property. Study case of Yekaterinburg housing market", Journal of Applied Economic Sciences, vol. 11, no. 8, 2016, pp. 1519-1530.
26. L.N. Yasnitsky, I.A. Mitrofanov, and M.V. Immis, "Intelligent System for Prediction Box Office of the Film", Lecture Notes in Networks and Systems, vol. 78, 2020, pp. 18-25. https://doi.org/10.1007/978-3-030-22493-6_3
27. A.O. Alexeev, I.E. Alexeeva, L.N. Yasnitsky, and V.L. Yasnitsky, "Self-adaptive Intelligent System for Mass Evaluation of Real Estate Market in Cities", Advances in Intelligent Systems and Computing, vol. 850, 2019, pp. 81-87. https://doi.org/10.1007/978-3-030-02351-5_11
28. L.N. Yasnitsky, "O nauchnom prioritete permskikh uchenykh v oblasti iskusstvennogo intellekta [On the scientific priority of Perm scientists in artificial intelligence]", Artificial intelligence in solving urgent social and economic problems of the 21st century: collection of articles based on the proceedings of the Fourth All-Russian Research-to-Practice Conference. (Perm, May 21 – 24, 2019) Part I, Perm State National Research University, Perm, 2019, pp. 7-25.
29. L.N. Yasnitsky, Intellektualnyye sistemy [Intelligent systems], Laboratory of knowledge, Moscow, 2016, p. 221.