Implementation of Selective Pressure Mechanism to Optimize Memory Consumption in the Synthesis of Neuromodels for Medical Diagnostics

Serhii Leoshchenko[1[0000-0001-5099-5518]], Andrii Oliinyk[2[0000-0002-6740-6078]], Sergey Subbotin[3[0000-0001-5814-8268]], Tetiana Zaiko[4[0000-0003-1800-8388]] and Nataliia Gorobii[5[0000-0003-2505-928X]]

[1,2,3,4,5] National University "Zaporizhzhia Polytechnic", Dept. of Software Tools, 69063 Zaporizhzhia, Ukraine
[1]sergleo.zntu@gmail.com [2]olejnikaa@gmail.com
[3]subbotin@zntu.edu.ua [4]nika270202@gmail.com
[5]gorobiy.natalya@gmail.com

Abstract. The introduction of artificial neural networks in the process of medical diagnosis meets a number of difficulties. The using of general methods for synthesis and training networks is difficult because of the complexity of the simulated system, that is the human. Neuroevolution approach for the synthesis of neural network models has proven itself well, but it is also not without difficulties. The paper proposes a new mechanism for modifying the genetic algorithm in the synthesis of neuromodels that can be used in medical diagnostics. Innovations allow to reduce time of synthesis, and also to solve a number of problems at a choice of the best individuals for formation of new population.

Keywords: medical diagnosis, prediction, neuromodels, synthesis, selective pressure.

1 Introduction

In the arsenal of modern medicine, there are many effective means of detecting a variety of diseases, but some of them are invasive, dangerous to the patient or difficult to implement and high-cost. Most of these techniques can afford only multi-commercial medical centers, and therefore inaccessible to the majority of population. Modern medicine, especially at the primary level, needs to be armed with inexpensive, safe for the patient, effective and reliable tools for the earliest possible detection of the most common forms of pathology. One of the ways to create such tools is the use of artificial neural network (ANN) technologies.

Neural networks are implemented according to the principles of construction and functioning of the human brain. From it that technologies inheritance the ability to learn and extract knowledge from statistical data, to generalize them in the form of rules and regularities, the property of intuition. Well-designed and properly trained

2019 IDDM Workshops.

ANN are able to build adequate mathematical models and use them to perform high-precision predictions (forecasts in many areas, including medicine.

ANNs are not programmed in the usual sense of the word, they are trained. The possibility of training is one of their main advantages over traditional algorithms. After training, ANNs become mathematical models of the subject areas under consideration. This means that virtual experiments can be performed on them, and ANNs will behave in exactly the same way as the subject area they are modeling.

The method of mathematical modeling in its classical sense has long been fruitfully used in many scientific fields. Today, no sufficiently complex technical object or process is created and launched without virtual computer experiments being performed on its mathematical model. Thanks to this, scientists and engineers know exactly how long the object they create will live, how it will behave in complex changing conditions, and what should be done to avoid trouble.

Experts note that the method of mathematical modeling for a long time was practically unavailable for use in the field of medical sciences due to the exceptional complexity of the object of modeling the human. But new ANN technologies allow to overcome this barrier and to construct mathematical models of patients and to carry out computer experiments on them: changing a way of life virtually, trying various courses of treatment, selecting medicines and observing on the computer screen to what it will lead.

Moreover, as the scientists note, cases have been repeatedly recorded when in the process of neural network modeling new, previously unknown knowledge and patterns were revealed and used. The results of neural network modeling-diagnoses and forecasts, eventually found confirmation, despite the apparent paradoxical nature of the detected patterns [1-4].

The facts discovered by the method of neural network mathematical modeling are not always consistent with the established practice of giving the same recommendations to all patients without exception: to follow a diet, abandon bad habits, limit the use of coffee and alcohol, lose weight, limit mental and physical activity, etc [5]. Virtual computer experiments have shown that these recommendations are really useful for most, but not for all patients. To identify atypical patients for whom these recommendations are not only useful, but also can cause harm, allows the intelligent system of diagnosis and prediction of diseases [6].

However, there are a number of difficulties in the union of ANN technology and health protection. Decision issued using ANN must to be accompany acceptable explanations and comments that ANN are not able to do because of their not verbality inherited from the prototype brain. Moreover, the theory of neural networks is still weak and only a very experienced mathematician can create a really adequate ANN model that provides high accuracy of diagnosis and forecasting.

To solve this problem, often turned to strategies that allow not just to train, but to synthesize ANN in the same way as it happens in the real world these are neuroevolutionary methods. However, it should be emphasized that in this case, the developer instead of difficulties with the development and training of ANNs gets problems with the use of evolutionary algorithms. In this research, authors consider the moderniza-

tion of the previously proposed modified genetic algorithm [7], [8] using the selective pressure mechanism to optimize memory consumption.

2 The problems of synthesis of neuromodeles

Inasmuch the choice of ANN topology is, as a rule, a complex task solved by trial and error, the evolutionary search for a neural network structure is able to facilitate and to some extent automate the process of solving the problem of configuring and training ANNs. The simultaneous solution of two separate problems: setting the weights of connections and setting the structure of ANN allows to some extent compensate for the shortcomings inherent in each of them separately and combine their advantages. On the other hand, the payment for this is a huge search area, as well as the unification of a number of shortcomings caused by the use of the evolutionary approach. Summing up, let list the advantages and disadvantages [8-14].

Advantages [15], [16]:

- the ability to automatically search the topology of ANN and obtain a more accurate ANN model by considering the non-standard, irregular topologies;
- independence from the structure of ANN and characteristics of activation functions of neurons;
- the ability to automatically search the topology of the ANN and obtain a more accurate ANN model.

To simplify the task and improve the quality of the results, in the process of searching for the topology of ANN, it is possible to use additional regulatory restrictions that help to avoid excessive complication of the network, which is expressed in a rapid increase in the number of hidden neurons and connections between them [9].

Disadvantages [17], [18]:

- the complexity of estimating the structure of ANN without information on the values of the weights of the connections;
- the complexity of the search topology ANN;
- the complexity of fine-tuning the weights of connections in the later stages of evolutionary search;
- large requirements for the amount of RAM due to the use of populations of ANNs.

The first drawback is the main problem of evolutionary tuning of the ANN structure. It is mainly due to the sensitivity of training results to initial conditions and values of training algorithm parameters [19-22].

2.1 Using of selective and crossover operators during the synthesis of neuromodels

Selection of individuals consists in the selection (for the value of the fitness function calculated at the previous stage) of those individuals who will participate in the breeding of children for the next population, that is, for the next generation. This choice is

made according to the principle of natural selection, according to which the chromosomes with the highest values of fitness function have the greatest chances to participate in the creation of new individuals [23-26]. There are different methods of selection. The most popular is the so-called method of roulette wheel selection, which got its name by analogy with the famous gambling [23], [24]. Each chromosome can be mapped sector roulette wheel, the value of which is set proportional to the value of the fitness function of the chromosome. Therefore, the greater the value of the fitness function, the larger the sector on the roulette wheel. The entire roulette wheel corresponds to the sum of the fitness function of all individuals in the population in question. Each individual, Ind_i for $i = 1,2,...,n$ (where n is population size) corresponds to the wheel sector $v(Ind_i)$, expressed as a percentage, according to the formula

$$v(Ind_i) = p_s(Ind_i) \cdot 100\%,$$

where

$$p_s(Ind_i) = \frac{F(Ind_i)}{\sum_{i=1}^{n} F(Ind_i)}, \tag{1}$$

and $F(Ind_i)$ is a value of the fitness function of the individual Ind_i, and $p_s(Ind_i)$ is a probability of selection of individuals Ind_i. Selection of an individual can be represented as the result of turning the roulette wheel, since the selected individual (that is, the winner) refers to the sector of the wheel that fell out. Obviously, the larger the sector, the greater the likelihood of selecting the appropriate individual. Therefore, the probability of choosing this chromosome is proportional to the value of its fitness function.

The roulette wheel selection method is considered by genetic algorithms to be the main method of selecting individuals for the parent population with a view to their subsequent transformation by genetic operators, such as crossing and mutation [24], [25]. Despite the random nature of the selection procedure, parent individuals are selected in proportion to the values of their fitness function, that is, according to the probability of selection, determined by the formula . Each individual gets in the parent pool is the number of copies, which is set by the expression $p_s(Ind_i) = \frac{F(Ind_i)}{\sum_{i=1}^{n} F(Ind_i)}$.

Each individual gets in the parent pool is the number of copies, which is set by the expression

$$c(Ind_i) = p_s(Ind_i) \cdot n, \tag{2}$$

where n are the number of the individuals Ind_i for $i = 1,2,...,n$ in the population, and $p_s(Ind_i)$ is a probability of selection of an individual Ind_i, what is calculated by

$p_s(Ind_i) = \frac{F(Ind_i)}{\sum_{i=}^{n} F(Ind_i)}$. Strictly speaking, the number of copies of a given individual in the parent pool is equal to an integer part of $c(Ind_i)$. When using formulas (1) and (2) it is necessary to pay attention to the fact that $c(Ind_i) = \frac{F(Ind_i)}{\overline{F}}$, where $\overline{F}$ is the average value of the fitness function in the population. Obviously, the roulette method can be used when the value of the fitness function is positive. This method can only be used in function maximization problems (not minimization).

At tournament selection all individuals of population are divided into subgroups with the further choice in each of them of an individual with the best fitness [24], [26]. There are two ways to make this choice: deterministic tournament selection and stochastic tournament selection. Deterministic choices have a probability of 1, and random choices have a probability of: <1. Subgroups can be of arbitrary size, but most often the population is divided into subgroups of 2-3 individuals each.

The tournament method is suitable for solving problems of both maximization and minimization of the function. In addition, it can be easily extended to problems related to multi-criteria optimization, that is, to the case of simultaneous optimization of several functions. In the tournament method, it can be changed the size of the subgroups into which the population is divided (tournament size). Studies confirm that the tournament method is more effective than the roulette method.

During ranking selection individuals of the population are ranked according to the values of their fitness function. This can be thought of as a sorted list of individuals, ordered in the direction from the most adapted to the least adapted (or vice versa), in which each individual is assigned a number, which determines its place in the list and is called a rank. The number of copies of each individual introduced into the parent population is calculated by a priori given function depending on the rank of the individual.

The advantage of the rank method is that it can be used both to maximize and minimize the function. It also does not require scaling due to the problem of premature convergence relevant to the roulette method [23-25].

The application of genetic operators to individuals selected by selection leads to the formation of a new population of children from the parent population created at the previous stage.

The crossover operation consists in the exchange of fragments of chains between two parent individuals [27-32]. A pair of parents for mating are selected from parents pool at random so that the probability of selecting a particular individual for breeding equal to the probability p_c. For example, if two individuals from the parent population are randomly selected as parents n, то $p_c = 2/n$.

Two-point crossover, as its name implies, differs from point crossing in that descendants inherit fragments of parent individuals determined by two randomly selected crossing points [28-30]. For a pair of individuals crossing at points 4 and 6 is shown

in Fig. 1. Note that such crossing does not lead to the destruction of the scheme, which is the parent individual 2.

$$\begin{matrix}\{parent\,1:[0011001110\;10]\\ parent\,2:[1010110110\;11]\}\end{matrix}\xrightarrow{\text{crossover}}\begin{matrix}[0011111110\;10]:child\,1\\ [1010000110\;11]:child\,2\end{matrix}$$
$$l_k:\ 4\ \ 6$$

Fig. 1. An example of two-point crossover

Multiple-point crossover is a generalization of previous operations and is characterized by a correspondingly large number of crossing points [30], [31]. For example, for three crossing points equal to 4, 6, and 9, and the same number of parents as in Fig. 1, the crossing results are shown in Fig. 2.

$$\begin{matrix}\{parent\,1:[0011001110\;10]\\ parent\,2:[1010110110\;11]\}\end{matrix}\xrightarrow{\text{crossover}}\begin{matrix}[001111111011]:child\,1\\ [101000011010]:child\,2\end{matrix}$$
$$l_k:\ 4\ \ 6\ \ 9$$

Fig. 2. An example of multiple-point crossover

Uniform crossover is performed according to a randomly selected standard that specifies which genes should be inherited from the first father (other genes are taken from the second parent) [28-31]. That is the general rule of uniform crossing can be represented as follows:

$$\begin{aligned}&Crossover\,(Ind_1, Ind_2, DataofCros) = Ind_3\\ &g_{Ind_3} = \{g_1 = Rand\left(g_{Ind_1}, g_{Ind_2}\right),\\ &g_2 = Rand\left(g_{Ind_1}, g_{Ind_2}\right),\ldots,\\ &g_i = Rand\left(g_{Ind_1}, g_{Ind_2}\right)\}\end{aligned}\quad . \tag{3}$$

An example of uniform crossover is shown in Fig. 3.

$$\begin{matrix}\{parent\,1:[0011001110\;10]\\ parent\,2:[1010110110\;11]\}\end{matrix}\xrightarrow{\text{crossover}}\begin{matrix}[1011011110\;10]:child\,1\\ [0010100110\;11]:child\,2\end{matrix}$$

locus :	1	2	3	4	5	6	7	8	9	10	11	12
etalon :	0	1	0	1	1	0	1	1	1	0	1	1

Fig. 3. An example of uniform crossover

3 The idea of implementing and using selective pressure

The main factor of evolution is natural selection, which leads to the fact that among genetically different individuals of the same population survive and leave offspring only the most adapted to the environment. In genetic algorithms is also highlighted stage of selection, which from the current population are selected and included in the

parent population of individuals who have the greatest value of a fitness function. The next step, sometimes called evolution, involves genetic crossover and mutation operators that recombine genes on chromosomes.

However, it should be recognized that the classical genetic algorithm emulates natural evolution is not fully, so it may exist wish to look at the mechanism of selective pressure, which will more effectively perform crossover.

We introduce selective pressure at the crossover stage by extending the selection operation. It is established the relationship between the probability of gene transmission to a descendant and the knowledge of the suitability of parents. To do this, we will expand the rank selection by introducing additional criteria for evaluating individuals.

The first criterion will be used to assess memory redundancy. As mentioned above, neural networks have memory that implemens as the weights connections. The less memory ANN has, the fewer images it can remember. However, in a situation where two ANNs with different memory provide the necessary accuracy of recognition (evaluation), the network with less memory, of course, has the best generalizing properties [33]. The network memory redundancy will be characterized by the redundancy factor for the training sample storage:

$$crit_m = \frac{W_{FFc} + W_{FBc}}{samp_{Inst} \cdot samp_{Feat}}, \quad (4)$$

where W_{FFc} is the number of direct ANN connections ($W_{FFc} = \{w_1, w_2, ..., w_i\}$); W_{FBc} is the number of feedback ANN connections ($W_{FBc} = \{w_1, w_2, ..., w_j\}$); $samp_{Inst}$ is the number of instances at training set; $samp_{Feat}$ is the number of features at training set.

If $crit_m > 1$, then the ANN memory is redundant (the ANN memory dimension is greater than the sample size). If $crit_m = 1$, then the ANN can remember the entire training sample (the memory dimension of the ANN is equal to the size of the training sample). If $crit_m < 1$, then the ANN will not be able to remember exactly the entire training sample (the memory dimension of the ANN is less than the dimension of the training sample), but the ANN will show generalizing and approximating abilities.

The use of the second criterion is related to the approximation properties of the ANN. One of the most important characteristics of ANN models is the quality of approximation. In the case where the error level of the models is one-to-one, the approximation quality is higher in the model that uses fewer links [33]. The quality coefficient of approximation of the neural network model is defined as the average share of error attributable to the non-zero weights of the network:

$$crit_a = \frac{Error}{W_{FFc} + W_{FBc} - W_{w=0}}, \quad (5)$$

where $Error$ is the aggregate error allowed by the network (for example, root mean square error) is such that $Error < \xi$, where ξ is the maximum allowable error (learn-

ing objective). As an *Error* it can be used a sample error ($E_{нав.}$) or a test sample error as an error ($E_{тест.}$); W_{FFc} is the number of direct connections of the ANN ($W_{FFc} = \{w_1, w_2, ..., w_i\}$); W_{FBc} is the number of feedbacks for recurrent ANN ($W_{FBc} = \{w_1, w_2, ..., w_j\}$); $W_{w=0}$ are zero weights (ANN connections whose weight is 0).

Thus, we consider the modification of rank selection using criteria for evaluating ANN moles.

Selection begins by sorting (ranking) individuals based on their availability so that $F(Ind_i) \geq F(Ind_j)$ for $i > j$. Each individual is then assigned a probability of being selected p_s, taken from a given restricted division $\sum_i p_s = 1$. The probability of selection is calculated by the form:

$$p_{s_i} = \frac{1}{n}\left(a - (a-b)\frac{rank + crit_m + crit_a - 1}{n-1}\right), \quad (6)$$

where $a \in [1;2]$, $b = 2 - a$, *rank* is a rank of individuals in the sorted list of individuals.

The use of criteria in determining the probability of selection solves several problems, namely:

- advance convergence of the method;
- reducing the variety;
- selection of the best individuals, at the same rank (at the same value of the fitness function).

It has long been known that setting the probability of transmission of the parent gene to the offspring in uniform crossing can significantly increase its efficiency, and also allows to emulate other crossover operators (single-point, two-point). It is also known that the use of a uniform crossing operator allows to apply the so-called multiparent recombination, when to cross one child more than two parents. Uniform crossing gives greater flexibility when combining strings, which is an important advantage when working with genetic algorithms.

Therefore, a uniform crossing with a specified parent pool size will be used as the crossing operator. The pool will be filled with individuals selected using modified rank selection. This approach adds flexibility to the method and allows to hope for a change in the behavior of the method.

4 Experiments

Data for testing were taken from the open repository – UC Irvine Machine Learning Repository. Data sample was used: Breast Cancer Coimbra Data Set [34]. Clinical features were observed or measured for 64 patients with breast cancer and 52 healthy controls. There are 10 predictors, all quantitative, and a binary dependent variable,

indicating the presence or absence of breast cancer. The predictors are anthropometric data and parameters which can be gathered in routine blood analysis. Prediction models based on these predictors, if accurate, can potentially be used as a biomarker of breast cancer. Table 1 shows the main characteristics of the data sample. 75% of the sample was used for training, 25% of the sample was used for testing.

Table 1. Main characteristics of the Breast Cancer Coimbra Data Set

Criterion	Characteristic	Criterion	Characteristic
Data Set Characteristics	Multivariate	Number of Instances	116
Attribute Characteristics	Integer	Number of Attributes	10

During the evaluation of the test results we will pay attention to the following criteria:

– the spent time, s;
– average error of final network (E);
– the size of parent pool.

The relative error value in this case will be calculated as the ratio of the classification error to the total sample size (number of instances).

$$E = \frac{error_{class}}{Number_{sampl}} \cdot 100\%, \tag{7}$$

where E is relative error; $error_{class}$ is classification error; $Number_{sampl}$ the number of instances in the sample.

The following hardware and software have been used for experimental verification of the proposed method for ANN synthesis: the computing system of the Department of software tools of Zaporizhzhia Polytechnic National University (ZPNU), Zaporizhzhya: Xeon processor E5-2660 v4 (14 cores), RAM 4x16 GB DDR4, the programming model of Java threads.

5 The results analysis

Table 2 shows the results of testing the modified genetic algorithm (MGA) in comparison with the modified genetic algorithm with using selective pressure (MGA with SP).

Table 2. Results of testing

	Time, s	E	Size of parent pool
Modified GA	631.373	2.96%	–
MGA with SP	627.879	2.04%	2
MGA with SP	649.216	2.63%	3

MGA with SP	657.395	2.74%	4
MGA with SP	686.775	2.76%	5
MGA with SP	696.98	2.81%	6
MGA with SP	660.592	2.14%	7
MGA with SP	712.053	2.85%	8
MGA with SP	735.768	2.92%	9
MGA with SP	787.813	2.97%	10

From the results of the experiment it can be seen that the most acceptable performance of the method is observed at the size of the parent pool 2 and 7. In other cases of parents, the ratio of resources used and time spent in the exact initial neuromodels is not satisfactory. It can also be noted that the number of parent individuals for crossing >10 does not make sense, because with large values of the execution time and the overhead of sending data, the accuracy deteriorates significantly.

Moreover, it can be concluded that the use of selective pressure and uniform crossing reduce the size of the population, without taking into account and without considering those individuals of the population that are characterized by a small value of the fitness function. Also, selective pressure allows to take into account additional quality indicators of neural network models [35]. This avoids the problem of identical ranks for models with the same fitness function score.

6 Conclusion

The increasing of accuracy and reduction of memory and computing power costs for storing and crossing the total population volume confirm the high efficiency of the proposed modification. However, the growth of input parameters should be noted. Therefore, the next step may be to automate the selection of input parameters, depending on the problem and its boundaries.

References

1. Einarson, T.R., Acs, A., Ludwig, C., Panton, U.H.: Prevalence of cardiovascular disease in type 2 diabetes: a systematic literature review of scientific evidence from across the world in 2007–2017. Cardiovasc Diabetol 17 (1), 1-19 (2018). DOI: 10.1186/s12933-018-0728-6
2. Petrie, J.R., Guzik, T.J., Touyz, R.M.: Diabetes, Hypertension, and Cardiovascular Disease: Clinical Insights and Vascular Mechanisms. The Canadian journal of cardiology 34(5), 575-584 (2017). DOI: 10.1016/j.cjca.2017.12.005
3. Thanassoulis, G., Vasan, R.S.: Genetic Cardiovascular Risk Prediction – Will We Get There? Circulation 122(22), 2323-2334 (2010). DOI: 10.1161/CIRCULATIONAHA.109.909309
4. Jungen, C., Scherschel, K., Eickholt, C., Kuklik, P., Klatt, N. Disruption of cardiac cholinergic neurons enhances susceptibility to ventricular arrhythmias. Nature Communications 8 (2017). DOI: 10.1038/ncomms14155

5. Leoshchenko, S., Oliinyk, A., Subbotin, S., Zaiko, T.: Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems. In: Proceedings of the 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, pp. 411-416 (2018). DOI: 10.1109/INFOCOMMST.2018.8632015
6. Volchek, Y.A., Shyshko, V.M., Spiridonova, O.S., Mokhort, T.V.: Position of the model of the artificial neural network in medical expert systems. Juvenis scientia (9), 4–9. Scientia, Saint Petersburg (2017).
7. Leoshchenko, S., Oliinyk, A., Subbotin, S., Gorobii, N., Zaiko, T.: Synthesis of artificial neural networks using a modified genetic algorithm. Proceedings of the 1st International Workshop on Informatics & Data-Driven Medicine (IDDM 2018), 1-13 (2018). dblp key: conf/iddm/PerovaBSKR18
8. Leoshchenko, S., Oliinyk, A., Subbotin, S., Gorobii, N., Shkarupylo, V.: Modification of the Genetic Method for Neuroevolution Synthesis of Neural Network Models for Medical Diagnosis. In: Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019), pp. 143-158 (2019). dblp key: conf/cmis/LeoshchenkoOSGS19
9. Cao, W., Wang, X., Ming, Z., Gao, J.: A review on neural networks with random weights. Neurocomputing, 275, 278-287 (2018)
10. Kieffer, B., Babaie, M., Kalra, S., Tizhoosh, H.R.: Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 1-6 (2017)
11. Oliinyk, A., Zaiko, T., Subbotin, S.: Training sample reduction based on association rules for neuro-fuzzy networks synthesis. Optical Memory and Neural Networks (Information Optics) 23(2), 89-95 (2014). DOI: 10.3103/S1060992X14020039
12. Shkarupylo, V., Skrupsky, S., Oliinyk, A., Kolpakova T.: Development of stratified approach to software defined networks simulation. EasternEuropean Journal of Enterprise Technologies 89(5/9) 67-73 (2017). DOI: 10.15587/1729-4061.2017.110142
13. Kolpakova, T., Oliinyk, A., Lovkin, V.: Improved method of group decision making in expert systems based on competitive agents selection. In: Proceedings of the IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pp. 939-943 (2017). DOI: 10.1109/UKRCON.2017.8100388
14. Stepanenko, O., Oliinyk, A., Deineha, L., Zaiko, T.: Development of the method for decomposition of superpositions of unknown pulsed signals using the second-order adaptive spectral analysis. EasternEuropean Journal of Enterprise Technologies 2(9), 48-54 (2018). DOI: 10.15587/1729-4061.2018.126578
15. Yadav, J., Rani, A., Singh, V., Murari, B.: Prospects and limitations of non-invasive blood glucose monitoring using near-infrared spectroscopy. Biomedical Signal Processing and Control, Vol. 18 (2015). doi: 10.1016/j.bspc.2015.01.005
16. Jiang, J., Trundle, P. Ren, J.: Medical image analysis with artificial neural networks. Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society 34. 617-631 (2010). DOI: 10.1016/j.compmedimag.2010.07.003.
17. Sun, Q., Zhang, M., Mujumdar, A.S.: Recent developments of artificial intelligence in drying of fresh food: A review. Critical Reviews in Food Science and Nutrition 59(14), 2258-2275 (2019). DOI: 10.1080/10408398.2018.1446900
18. Neuroevolution — evolving Artificial Neural Networks topology from the scratch, https://becominghuman.ai/neuroevolution-evolving-artificial-neural-networks-topology-from-the-scratch-d1ebc5540d84 – last accessed 2019/09/04

19. Oliinyk, A.O., Oliinyk, O.O., Subbotin, S.A.: Software-hardware systems: Agent technologies for feature selection. Cybernetics and Systems Analysis 48(2), 257-267 (2012). DOI: 10.1007/s10559-012-9405-z
20. Oliinyk, A.A., Subbotin, S.A.: The decision tree construction based on a stochastic search for the neuro-fuzzy network synthesis. Optical Memory and Neural Networks (Information Optics) 24(1), 18-27 (2015). DOI: 10.1007/s10559-012-9405-z
21. Alsayaydeh, J.A.J., Shkarupylo, V., Bin Hamid, M.S., Skrupsky, S., Oliinyk, A.: Stratified model of the Internet of Things infrastructure. Journal of Engineering and Applied Sciences 13(20), 8634-8638 (2018). DOI: 10.3923/jeasci.2018.8634.8638
22. Oliinyk, A., Subbotin, S.: A stochastic approach for association rule extraction. Pattern Recognition and Image Analysis 26(2), 419-426, 2016. DOI: 10.1134/S1054661816020139.
23. Exploring Weight Agnostic Neural Networks, https://ai.googleblog.com/2019/08/exploring-weight-agnostic-neural.html – last accessed 2019/09/01
24. Saini, N. Review of Selection Methods in Genetic Algorithms. International Journal of Engineering and Computer Science 6(12), 22261-22263 (2017).
25. Jebari, K., Madiafi, M.: Selection Methods for Genetic Algorithms. International Journal of Emergency Services 3(4), 333-344 (2013).
26. Ma, Y., Zhang, C.: Quick convergence of genetic algorithm for QoS-driven web service selection. Computer Networks 52(5), 1093-1104 (2008).
27. Gao W.: An improved fast-convergent genetic algorithm. IEEE International Conference on Robotics, Intelligent Systems and Signal Processing 2, 1197-1202 (2003). DOI: 10.1109/RISSP.2003.1285761
28. Genetic Algorithms – Crossover, https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_crossover.htm – last accessed 2019/09/01
29. Mendes, J.: A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. WSEAS Transactions on Computers 12, 164-173 (2013).
30. Varun Kumar, S.G., Panneerselvam, Dr.R.: A Study of Crossover Operators for Genetic Algorithms to Solve VRP and its Variants and New Sinusoidal Motion Crossover Operator. International Journal of Computational Intelligence Research 13(7), 1717-1733 (2017).
31. Hassanat, A., Alkafaween, E. On Enhancing Genetic Algorithms Using New Crossovers, https://arxiv.org/ftp/arxiv/papers/1801/1801.02335.pdf – last accessed 2019/09/02
32. Umbarkar, Dr.A., Sheth, P.: Crossover operators in genetic algorithms: a review. ICTACT Journal on Soft Computing 6(1), 1083-1092 (2015). DOI: 6. 10.21917/ijsc.2015.0150.
33. Subbotin S.A.: Criteria for Comparison of Recognition Models Based on Neural Networks and Analysis of Their Mutual Relations. Artificial Intelligence 1, 142-152 (2014).
34. Breast Cancer Coimbra Data Set, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra – last accessed 2019/09/01
35. Leoshchenko, S., Oliinyk, A., Skrupsky, S., Subbotin, S., Lytvyn V.: Parallel Genetic Method for the Synthesis of Recurrent Neural Networks for Using in Medicine. In: Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019), pp. 1-17 (2019). dblp key: conf/cmis/LeoshchenkoOSSL19