



Adaptation of Kohonen's Self-organizing Map to the Task of Constructing an Individual User Trajectory in an E-learning System

Alexey V. Bosov^(✉) 

Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences,
Vavilova 44/2, Moscow 119333, Russia

ABosov@frccsc.ru

Abstract. The problem of dynamic trainee classification for supporting the constructing process of an individual user trajectory in an e-learning system is considered. The education model is designed for a mixed form of activities with partial independent work, periodic control actions in distance tests, and full-time offline work with tests and offset tests. The purpose of the classification is to determine the trainee's knowledge category based on the current control event results. The semantics of the categories suggests the possibility of an individual task of different task complexity levels at the next step of learning. The direction of improving the existing classification methods is rejecting the gathering and use of statistics from previous (other) groups of students. The absence of samples of correct classification justified the use of self-organizing neural networks. For the solution, Kohonen's maps were used, the standard version adapted to the existing education model, and the task of considering the teacher's subjective evaluation policy. Three variants of the self-learning algorithm are described. The experimental results are presented and explained.

Keywords: E-learning system · Self-organizing neural network · Kohonen's map · Classification · Individual learning trajectory

1 Introduction

A typical tool of modern e-learning systems used both online and traditional forms of education is a tool for building individual trainee trajectories. As a rule, such a tool considers the intermediate learning outcomes and generates recommendations for setting up the system content, for example, to determine the control tasks' complexity level. The methods and algorithms implemented by these tools compose a significant part of the Item Response Theory (IRT). Probabilistic models are often used to describe the effectiveness of the learning process, which allows one to adapt the testing process to the level of knowledge of users [1–6]. The result of adaptation is an individual learning (testing) trajectory, which is a time-bound sequence of solutions to the problem of classifying trainees – assigning them to different categories of learning success. Based

on the classification results, different classes of users are offered different trajectories for further testing or training.

Such task of classification is typical. Therefore, the most common technics are used to solve it. Thus, the classical Bayesian classifier [7] has been successfully implemented in an e-learning system that uses the student's rating, which is calculated during the learning process as a convolution of the shown intermediate results [8]. The disadvantage of this option for categorizing students and other technics of constructing an individual trajectory based on statistics is that implementing the corresponding algorithms requires knowledge of some probabilistic characteristics. In [8], these are the conditional probabilities of maintaining/changing the rating after the next control event. It is impossible to know precisely these characteristics, so they are replaced by statistical estimates calculated from available or accumulated data. Accordingly, the disadvantages or difficulties of the Bayesian classifier usage when categorizing trainees include the following:

- frequency estimates of probabilities require a large amount of reliable data;
- any changes in the number, order, or nature of control measures lead to a distortion of the accumulated statistics, the impossibility of its use;
- it is assumed that the probabilistic characteristics coincide for different groups of trainees.

The article's purpose is to propose a trainees classification method based on using an e-learning tool without the above disadvantages. To do this, replacing the Bayesian classifier with a no less typical tool – a self-learning artificial neural network is suggested. Since the classification problem being solved is customary, the most common network variant was chosen – the Kohonen self-organization map [9]. The most well-known financial applications [10] of these maps have been replenished by now due to various applications. Without an aim to do a complete review, just a few examples, the most interesting in the context of the presented study, will be given. In general, apart from the field of financial analysis, practically any technical analysis might be named as a typical application for self-learning maps. It might be the analysis of production [11], trade [12], agriculture [13], geochemical research [14], monitoring of telecommunication system operation [15], generalization of psychological research [16], assessment of water resources [17], assessment of airport activities [18]. The versatility and expressiveness of self-organizing maps have given them applications in unexpected areas. For example, in [19], the Kohonen classifier is used to manage the information security of personal data; in [20] it is used to form a grid of nodes for numerical modeling. Also, there are medical applications [21, 22], seismology [23], fraud detection [24], and countering computer attacks [25], to name a few. This list can be continued, noting that applications to topics related to education are essential for the article. Categorizing enrolled students, trained employees, and active engineers [26–29] is a common practice.

The contribution of these studies and many others is not to develop the technology of self-organizing maps and clustering tools, but to expand the scope of known algorithms and to confirm their effectiveness in new applications. The article presents the results of a study of the exact nature. This study aims to apply the existing classification mechanism using Kohonen maps in the e-learning system CLASS.NET [30]. Several options for

adapting the description of the education process and the self-organization algorithm of the Kohonen map are proposed for this, and practical experiments are carried out.

2 Education Process Model and the Trainee Categorization Problem

The learning process model assumes blended distance learning and includes T offline control activities. A typical training scheme for the proposed model implies that the course available on the studied subject is divided into several parts. Each part ends with face-to-face activities (offline offset tests and full-time tests), and trainees perform several extramural tests between time intervals (online tests). After the next control gate (training step), a category is determined for each trainee, for example, “poor” (1), “low “ (2), “average “ (3), and “high level” (4). Within the framework of this work, these four categories of students will be used, which are denoted by the number $k \in K = \{1, 2, 3, 4\}$. This assumption does not impose any restrictions on generality, but fixing the number of categories and connecting them with explicit content is essential for the illustrative calculations below.

So, the classification task is solved at each training step: each trainee is assigned to one of the categories based on the demonstrated learning outcomes. The corresponding time points are denoted $t = 1, \dots, T$. A typical use of such categories in e-learning systems is determining the difficulty level of the next stage/step of training and/or control activity. This ensures the compliance of the control content with the status of the trainee’s preparation and the construction of an individual learning trajectory.

We will assume that for each t , the value $x_t \in \{\text{on, off, te}\}$ is given and determines the type of control measure: on – online test, off – offline offset test, te – offline full-time test. Each of S trainees at the moment t receives the next mark $e_t^s, s = 1, \dots, S$, which, depending on x_t , is characterized by its range of values. To be clear, we will assume that for $x_t = \text{on}$ the values $e_t^s \in \{0, 1, \dots, 100\}$, for $x_t = \text{off}$ the values $e_t^s \in \{0, 1\}$, for $x_t = \text{te}$ the values $e_t^s \in \{2, 3, 4, 5\}$. In other words, online testing is evaluated on a 100-point scale. In contrast, offset testing is estimated as traditional pass or fail, and full-time testing is considered on marks unsatisfactory, satisfactory, good, or excellent. Accordingly, the task of classifying trainees at time t is to determine the category $k_t^s \in K$ for each s -th trainee.

A solution to this problem proposed in [8] is to collapse the current vector of marks $(e_1^s, \dots, e_t^s)^T, x'$ is the transposed vector into some scalar-valued current rating, the assignment solution of a trainee with this rating to category k is provided by the Bayesian rule, which uses statistical characteristics (frequency estimates of posterior probabilities) calculated from the results demonstrated by previous groups of trainees. The principal proposal of this work is to use for classification only the results shown by a tiny group of trainees and classify them by comparing learning outcomes with each other. Formally, such a problem consists in determining the mapping

$$\left\{ \{e_\tau^s\}_{\tau=1}^t \right\}_{s=1}^S \rightarrow \{k_t^s\}_{s=1}^S, k_t^s \in K, t = 1, \dots, T. \quad (1)$$

If we do not take into account the scale and semantics of the input data set $\left\{ \left\{ e_{\tau}^s \right\}_{\tau=1}^t \right\}_{s=1}^S$, i.e., values $\{x_{\tau}\}_{\tau=1}^t$ of control types, then problem (1) is a typical setting for constructing a one-dimensional self-organizing Kohonen map [9]. Taking into account the order and types of control measures require a slight adaptation of the standard self-learning algorithm. In addition, it is also necessary to consider the dynamic nature of the learning model, i.e., that the classification is performed along with the next steps of learning and control. This feature also needs to be taken into account when building a map.

3 Implementation Options for Kohonen Maps

The *first* of the proposed *options* of the self-learning algorithm differs from the classical one in the determination of the distance between the input vectors and neurons of the map instead of the Euclidean norm $|x|^2 = xx'$, the normalization function $x_Q^2 = x'Qx$ is used. Here Q is the symmetric non-negative defined matrix, i.e., the identity matrix $Q = 1$ corresponds to the Euclidean norm $x_1^2 = |x|^2$. The Q matrix will be used for various purposes. In the first option, its task is to scale the marks. In all possibilities, this matrix will be diagonal; in the first one, there will be 100^{-2} on the diagonal, if the corresponding value is $x_t = \text{on}$, 1, if $x_t = \text{off}$, 5^{-2} , if $x_t = \text{te}$. Let us explain this with an example, using the following distribution of control measures types:

$$\begin{aligned} x_1, x_2, x_3, x_4 &= \text{on}, x_5 = \text{off}, x_6 = \text{te}, \\ x_7, x_8, x_9 &= \text{on}, x_{10} = \text{off}, x_{11} = \text{te}, \\ x_{12}, x_{13} &= \text{on}, x_{14} = \text{off}, x_{15} = \text{te}, x_{16}, x_{17} = \text{on}. \end{aligned} \quad (2)$$

That is, in the first part of the course, there are four online tests, offset test and full-time test; in the second part – three online tests, offset test and full-time test; and in the third part – two online tests, offset test, fulltime test, and two online tests. This order corresponds to the examples considered in [8]. Thereby, for 17 of these types of control activities, the matrix of normalization $Q = Q^{\frac{1}{2}} Q^{\frac{1}{2}}$:

$$Q^{\frac{1}{2}} = \text{diag} \left\{ \begin{array}{cccccccccccccccc} \frac{1}{100}, & \frac{1}{100}, & \frac{1}{100}, & \frac{1}{100}, & 1, & \frac{1}{5}, & & & & & & & & & & \\ & \frac{1}{100}, & \frac{1}{100}, & \frac{1}{100}, & & 1, & \frac{1}{5}, & & & & & & & & & \\ & & \frac{1}{100}, & \frac{1}{100}, & & & 1, & \frac{1}{5}, & & & & & & & & \\ & & & \frac{1}{100}, & & & & 1, & \frac{1}{5}, & & & & & & & \end{array} \right\}. \quad (3)$$

One chooses the moment t and the corresponding normalization matrix $Q_t = \text{diag}\{Q_{1,1}, \dots, Q_{t,t}\}$. Then we follow the standard algorithm. Denote by W the weight matrix of a map of size $4 \times t$, W_k is the k -th row of this matrix, i.e., vector of weights of the k -th neuron, $k = 1, 2, 3, 4$. Let us perform a sequence of iterations numbered $i = 1, \dots, I$. At the i -th iteration, successively, for each s -th trainee $s = 1, \dots, S$, the nearest neuron is fined as follows

$$k_{\min}^i(s) = \text{argmin}_{k=1,2,3,4} \|E_t^s - W_k(i)\|_{Q_t}^2,$$

where $E_t^s = (e_1^s, \dots, e_t^s)'$ is a vector composed of marks of the s -th trainee by the current moment t , $W_k(i)$ is the k -th row of the matrix W corresponding to the i -th iteration.

Next, the value of the neighborhood function $h(k, k_{min}^i(s))$ is calculated for each k -th neuron, and the weight vector is adjusted

$$\begin{aligned} h(k, k_{min}^i(s)) &= \exp\left\{-\frac{|k - k_{min}^i(s)|^2}{2\sigma_i^2}\right\}, \\ W_k(i) &= W_k(i) + \eta_i h(k, k_{min}^i(s)) (E_i^s - W_k(i)), \\ W_k(i+1) &= W_k(i), \text{ if } s = S, \text{ else } s = s + 1. \end{aligned} \quad (4)$$

Note that, unlike the usual algorithm, all input actions in (4) (marks of all trainees) are processed within one iteration and with the same learning parameters σ_i, η_i . The self-learning algorithm, as described, makes it possible to account for the marks of all trainees with the same weights, equalizing the contribution of all trainees to the classification. The step into the $i + 1$ -th iteration is performed when the weight matrix has been corrected by all available S marks.

The choice of training parameters is made according to the most common recommendations [31–33]: $\sigma_i = \sigma_0 \exp\{-i/\delta\}$, $\delta = 1000/\ln(\sigma_0)$, $\eta_i = \eta_0 \exp\{-i/\Delta\}$. Moreover, the value δ was not even used in the calculations and was supposed to $\sigma_i = \text{const} = 1$. The remaining values $\eta_0 = 0.1$ and $\Delta = 1000$ are usual recommendations for $I = 1000$ iterations.

As a part of the same algorithm option, the ability to prescribe significance weights for different types of control activities can be added. For example, considering a full-time nature, a mark for the offset test with double weight and a mark for the full-time test with triple weight could be taken. This does not seem very important.

The next *second option* of the algorithm or, more precisely, the classification model eliminates two possible drawbacks of the first variant. Firstly, it is necessary to ensure a correct categorization of trainees in a group from the teacher's point of view, in which all four categories of trainees are not clearly represented. For example, in a group with no underperformers, category $k = 1$ should be left empty. At the same time, applying algorithm (4) may still give four categories well, distributing trainees without considering the evaluative context nested in the categories. In this regard, one can see the second drawback as follows: the context "poor," "low," "average," and "high" put into the categories by the teacher may not be guaranteed by the algorithm (4). Another statement of the same problem consists in the fact that the algorithm (4) does not consider the expert's (teacher's) idea of the content of the concepts "poor," "low level," "average," and "high".

Modifying (4) to eliminate the indicated drawbacks, we introduce "expert" trajectories $(IE_T^s)^k, k \in K$, the meaning of which is to represent an "ideal" instance of the k -th category:

$$\begin{aligned} (IE_{17}^s)^1 &= (30, 30, 30, 30, 0, 2, 30, 30, 30, 0, 2, 30, 30, 0, 2, 30, 30)', \\ (IE_{17}^s)^2 &= (50, 50, 50, 50, 0, 3, 50, 50, 50, 0, 3, 50, 50, 0, 3, 50, 50)', \\ (IE_{17}^s)^3 &= (70, 70, 70, 70, 1, 4, 70, 70, 70, 1, 4, 70, 70, 1, 4, 70, 70)', \\ (IE_{17}^s)^4 &= (90, 90, 90, 90, 1, 5, 90, 90, 90, 1, 5, 90, 90, 1, 5, 90, 90)'. \end{aligned} \quad (5)$$

Note, these "ideal" trajectories solve the issue with initial conditions because they can be used in (4) to start iterations, i.e., to define $W_k(0)$ values. In the second option,

trajectories (5) are used as input vectors:

$$\begin{aligned} W_k(i) &= W_k(i) + \eta_i h(k, k_{\min}^i(s)) (E_t^s - W_k(i)), s = 1, \dots, S, \\ W_k(i) &= W_k(i) + \eta_i h(k, k_{\min}^i(s)) \left((IE_t^s)^k - W_k(i) \right), \\ k &= 1, \dots, K, s = 1, \dots, [S/k], \\ W_k(i+1) &= W_k(i), \text{ if } s = S \text{ and } k = K. \end{aligned} \quad (6)$$

Thus, for the considered case $k = 1, 2, 3, 4$, each of the four “ideal” trajectories $(IE_{17}^s)^k$ participates in the i -th iteration of the algorithm, repeating as an input in the amount of 25%, $[S/k] \cdot 100\%$, of the number of available trajectories, i.e., number of trainees.

Finally, the last *third option* of the classification model takes into account the factor of obsolescence of the marks received by the trainee or the greater relevance (priority) of the latest, older outcomes. This factor can be considered as simple as possible by limiting the number of assessments, setting $Q_t = \text{diag}\{0, \dots, 0, Q_{t, t-\varepsilon+1}, \dots, Q_{t, t}\}$, i.e., taking into account only ε of the most recent marks. A more flexible way is to use the aging parameter α , $0 < \alpha \leq 1$, and the normalization matrix

$$Q_t^{\frac{1}{2}} = \text{diag}\left\{\alpha^{t-1} Q_{1,1}^{\frac{1}{2}}, \dots, \alpha^2 Q_{t-2,t-2}^{\frac{1}{2}}, \alpha^1 Q_{t-1,t-1}^{\frac{1}{2}}, Q_{t,t}^{\frac{1}{2}}\right\}. \quad (7)$$

Such of normalization means a dynamic change in the input data: the multiplication of marks up to the current control actions t by the value α .

4 Practical Study Results

To illustrate the results of the proposed classifiers, we used a slightly adapted learning model implemented in the e-learning system CLASS.NET [30] and the same data set as in [8]. In the calculations, marks were used according to the control scheme (2) for $S = 36$ students. For a graphical illustration of the calculation results, the trajectories of three students were chosen. Firstly, selected trajectories are the most dynamic for the graphs to reflect better the results of applying the algorithms. Secondly, they are typical for students of different levels of knowledge. The actual data for these trajectories are given in Table 1.

Table 1. Examples of learning outcome trajectories.

s	e_1^s	e_2^s	e_3^s	e_4^s	e_5^s	e_6^s	e_7^s	e_8^s	e_9^s	e_{10}^s	e_{11}^s	e_{12}^s	e_{13}^s	e_{14}^s	e_{15}^s	e_{16}^s	e_{17}^s
1	58	73	54	49	1	3	56	58	69	0	2	45	67	1	3	51	72
2	54	69	55	65	1	4	71	69	69	1	4	67	55	1	4	66	61
3	69	67	74	68	1	4	99	91	92	1	5	87	93	1	5	94	87

The marks from the Table 1, normalized by the Q matrix from (3), are included in the graphs of all the calculations presented below.

The entire set of marks in the trajectories selected for the table was subjectively assessed as typical for students with a “low level” ($s = 1$), an “average level” ($s = 2$) and a “high level” ($s = 3$). At the same time, all three trajectories illustrate explicit transitions between classes $k = 2, 3, 4$, so they should help demonstrate the difference between the proposed classifiers. In the graphs below, the trajectories of marks and the corresponding trajectories of categories calculated by one of the proposed classification options are shown as lines of the same type. Namely, the trajectories for the user of the e-learning system with the number $s = 1$ are shown as a simple black line, the trajectories for the user with the number $s = 2$ are shown as a broken black line, the trajectories for the user with the number $s = 3$ are shown as a gray line. Moreover, on the same graph can be seen both the normalized marks of students (these trajectories are located in the lower part of the graphs, since their absolute values are in the interval $(0, 1)$), and the calculated categories (these trajectories are located in the upper part of the graphs, their absolute values are integers $k \in \{2, 3, 4\}$). Note that in the illustrative materials, there is no example with a clearly expressed trajectory for the category “poor”, $k = 1$. It can be said that such a learning outcome is not typical, although, for the entire set of students, there were rare decisions to assign the current category $k = 1$ to the student, which is in the illustration below.

Figure 1 shows the classification result of the first option of the Kohonen classifier, i.e., algorithm (4) with the normalization matrix (3). The result is relatively consistent with the subjective assessment of the selected student’s knowledge level. Particularly noteworthy is the trajectory $s = 1$, which shows how the classifier changes the category in the case when the correspondence between the set of marks and the knowledge level is not quite obvious.

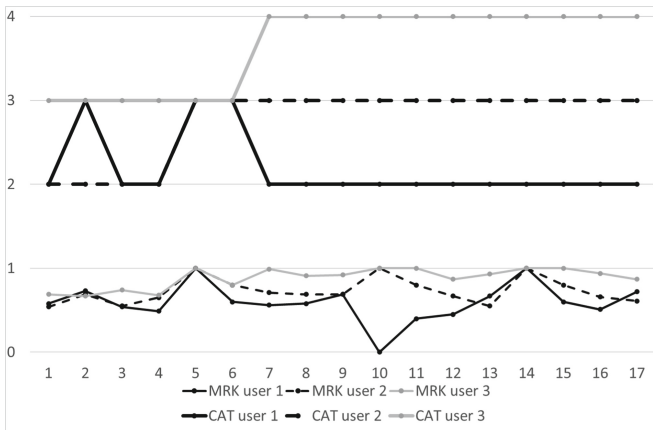


Fig. 1. Option one of the classifier – algorithm (3), (4)

Let’s evaluate not only these shown in Fig. 1 trajectories but also the entire set of classification results according to the first option of the algorithm. It must be stated that, in general, this classifier somewhat overestimates the results. This is, of course, a subjective assessment. But indeed, according to the data used for the calculation,

the classifier cannot estimate the upper limit of the marks. The second option of the algorithm should “help” him in this. The same calculations performed by the algorithm (6) with “ideal” trajectories (5) are shown in Fig. 2. This algorithm turned out to be more conservative in respect of following the categories given by the “ideal” trajectories. This is seen from the category trajectory for $s = 3$.

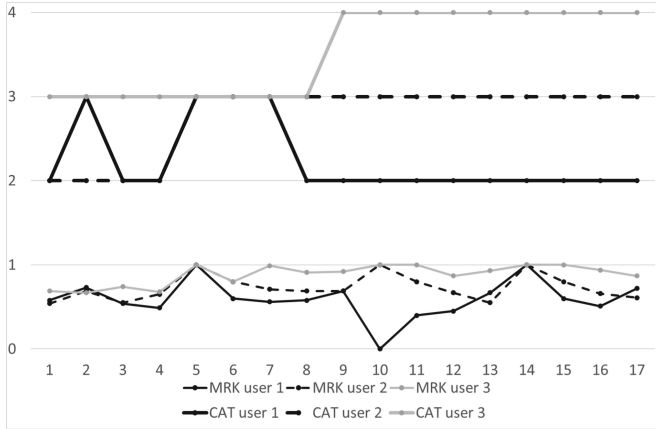


Fig. 2. Option two of the classifier – algorithm (5), (6)

If the conservatism of the second algorithm option is not needed, but on the contrary, a more dynamic solution is required that responds faster to current results, then the third algorithm option works. The same calculations performed with the normalization Q matrix from (7) for $\alpha = 0.75$ are shown in Fig. 3. The classification results have become very dynamic as expected. At the same time, the overall final knowledge level assessment remained consistent with the subjective expert view, at least on the considered trajectories.

5 Final Remarks

All three proposed options of the classification algorithm show consistent results. The choice between one or another option depends to a greater extent on the priorities of a particular teacher. The main target for further research should be the teacher's presentation and visualization form of the classification results, especially for the practical implementation of the described methodology. A suitable view to represent the categories of trainees in the e-learning system will undoubtedly be among the many computer graphics tools.

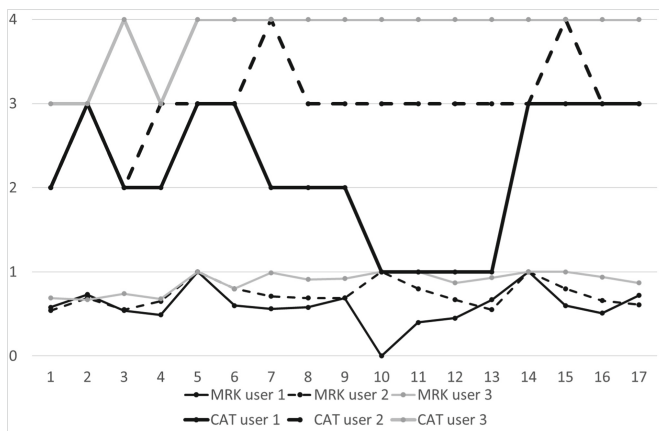


Fig. 3. Option three of the classifier – algorithm (4), (7), $\alpha = 0.75$

Acknowledgments. The study was supported by the Russian Science Foundation Grant No. 22–28–00588, <https://rscf.ru/project/22-28-00588/>. The research was carried out using the infrastructure of the Shared Research Facilities «High Performance Computing and Big Data» (CKP «Informatics») of FRC CSC RAS (Moscow).

References

1. Van der Linden, W.J., Scrams, D.J., Schnipke, D.L., et al.: Using response-time constraints to control for differential speededness in computerized adaptive testing. *Appl. Psych. Meas.* **23**(3), 195–210 (1999)
2. Rasch, G.: *Probabilistic Models for Some Intelligence and Attainment Tests*. The University of Chicago Press, Chicago (1980)
3. Kibzun, A.I., Inozemtsev, A.O.: Using the maximum likelihood method to estimate test complexity levels. *Autom. Remote Control* **75**(4), 607–621 (2014)
4. Naumov, A.V., Mkhitarian, G.A.: On the problem of probabilistic optimization of time-limited testing. *Autom. Remote Control* **77**(9), 1612–1621 (2016)
5. Kuravsky, L.S., Marmalyuk, P.A., Yuryev, G.A., Dumin, P.N., Panfilova, A.S.: Probabilistic modeling of the process of carrying out test tasks on the basis of the modified function Rush. *Issues of psychology* **4**, 109–118 (2015)
6. Kuravsky, L.S., Margolis, A.A., Marmalyuk, P.A., Panfilova, A.S., Yuryev, G.A., Dumin, P.N.: A probabilistic model of adaptive training. *Appl. Math. Sci.* **10**(48), 2369–2380 (2016)
7. Callan, R.: *The Essence of Neural Networks*. Prentice Hall Europe (1999)
8. Bosov, A.V., Martyushova, Ya.G., Naumov, A.V., Sapunova, A.P.: Bayesian approach to the construction of an individual user trajectory in the system of distance learning. *Inform. Appl.* **14**(3), 89–96 (2020)
9. Kohonen, T.: *Self-Organizing Maps*. Springer, Third Extended Edition), Berlin, Heidelberg (2001)
10. Deboeck, G., Kohonen, T.: *Visual Explorations in Finance*. Springer, London (1998)
11. Kulikova, O.M., Kalugin, V.E., Pil'nik, N.B., Gushhina, A.A.: Study of innovative development of production in the regions of the Russian Federation using Kohonen maps. *Fundam. Res.* **25**(2), 5639–5643 (2015)

12. Vrbka, J., Nica, E., Podhorska, I.: The application of Kohonen networks for identification of leaders in the trade sector in Czechia. *Equilibrium. Q. J. Econ. Econ. Policy* **14**(4), 739–761 (2019)
13. Boniecki, P., Nowakowski, K., Tomczak, R., Kujawa, S., Piekarska-Boniecka, H.: The application of the Kohonen neural network in the nonparametric-quality-based classification of tomatoes. In: *Proceedings of SPIE 8334. Fourth International Conference on Digital Image Processing (ICDIP 2012)*, p. 833427 (2012)
14. Yu, X., Xiao, F., Zhou, Y., Wang, Y., Wang, K.: Application of hierarchical clustering, singularity mapping, and Kohonen neural network to identify Ag-Au-Pb-Zn polymetallic mineralization associated geochemical anomaly in Pangxidong district. *J. Geochem. Explor.* **203**, 87–95 (2019)
15. Babenko, G.V.: Kohonen's self-organizing maps as the instrument of detection the deviations in TCP/IP traffic flow. *IT Security* **18**(4), 60–66 (2011)
16. Dylykova, A.Zh.: Application of the Kohonen network for the classification of personality according to psychological characteristics. *Mod. Trends Dev. Sci. Prod.* 291–294 (2016)
17. Aguilera, P.A., Garrido Frenich, A., Torres, J.A., Castro, H., Martinez Vidal, J.L., Canton, M.: Application of the Kohonen neural network in coastal water management: methodological development for the assessment and prediction of water quality. *Water Res.* **35**(17), 4053–4062 (2001)
18. Rahmalia, D., Herlambang, T.: Application Kohonen network and fuzzy C means for clustering airports based on frequency of flight. *Kinetik: Game Technol. Inf. Syst. Comput. Netw. Comput. Electron. Control* **3**(3), 229–236 (2018)
19. Kornev, P.A., Malysh, V.N.: Application of the Kohonen neural network for information security control of personal data. *Control Syst. Inf. Technol.* **46**(4), 53–58 (2011)
20. Voytishek, A.V., Khmel, D.S.: Analytical description for application of 1D Kohonen scheme for constructing adaptive meshes. *Siberian J. Num. Math. Sib. Branch Russ. Acad. Sci. Novosibirsk*, **14**(2), 131–140 (2011)
21. Mazing, M.S., Zaitceva, A.Y., Davydov, R.V.: Application of the Kohonen neural network for monitoring tissue oxygen supply under hypoxic conditions. *J. Phys. Conf. Ser.* **2086**(012116) (2021)
22. Ben Nouma, B., Mitiche, A., Mezghani, N.: A sample-encoding generalization of the Kohonen associative memory and application to Knee Kinematic data representation and pathology classification. *Appl. Sci.* **9**(9), 1741 (2019)
23. Volkova, A.: Application of self-organizing Kohonen maps for problems of seismic facies analysis. In: *ProGREss'21 of Conference Proceedings*, vol. 2021, pp. 1–5. European Association of Geoscientists & Engineers (2021)
24. Marković, V.S., Marjanović Jakovljević, M., Njeguš, A.: Anomalies detection in the application logs using Kohonen SOM machine learning algorithm. In: *Sinteza 2020. International Scientific Conference on Information Technology and Data Related Research*. Belgrade. Singidunum University. Serbia, pp. 275–282 (2020)
25. Barletta, V.S., Caivano, D., Nannavecchia, A., Scalera, M.: A Kohonen SOM architecture for intrusion detection on in-vehicle communication networks. *Appl. Sci.* **10**(15), 5062 (2020)
26. Zarubina, N.K., Ovchinkin, O.V., Pykhtin, A.I.: Exploratory analysis of the results of university admissions using the Kohonen neural network for planning the contingent of students. *Inf.-Measuring Control Syst.* **14**(6), 65–69 (2016)
27. Syabrenko, A.P., et al.: Analysis of the results of certification of employees of enterprises using Kohonen maps. *Sci. Tech. Bull. Volga Region* **6**, 159–162 (2018)
28. Akhterov, A., Lezina, O., Shastina, A.: Diagnostics of development of organizational and managerial competencies of the engineers using the self-organizing Kohonen maps. *Autom. Control Techn. Syst.* **42**, 35–45 (2013)

29. Tynchenko, V.S., Tynchenko, V.V., Bukhtoyarov, V.V., Kukartsev, V.V., Kukartsev, V.A., Eremeev, D.V.: Application of Kohonen self-organizing maps to the analysis of enterprises' employees certification results. *IOP Conf. Ser. Materials Sci. Eng.* **537**(4), 4042010 (2019)
30. Naumov, A.V., Dzhumurat, A.S., Inozemtsev, A.O.: The distance learning system for mathematical disciplines CLASS.NET. *J. Comp. Inf. Tech.* **10**, 36–40 (2014)
31. Ritter, H., Martinetz, T., Schulten, K.: *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wiley, Reading, MA (1992)
32. Kohonen, T.: Exploration of very large database by self-organizing maps. In: *International Conference on Neural Networks*. Houston, vol. 1, pp. PL1–PL6 (1997)
33. Kohonen, T.: Self-Organized formation of topologically correct feature maps. *Biol. Cybern.* **43**, 59–69 (1982)