ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Methodology/Method*
Vol. 9, No. 4, December 2021, pp.1138-1145

Clusters partition algorithm for a self-organizing map for detecting resource-intensive database inquiries in a geo-ecological monitoring system

Tareq Nasser Mahdi[1], Jalal Qais Jameel[2], Konstantin A. Polshchykov[3], Sergej A. Lazarev[4], Ilya K. Polshchykov [n3], Vladimir Kiselev[n4].
[1]Mustansiriyah University, College of Pharmacy, Baghdad, Iraq, [2]Mustansiriyah University, College of Medicine, Baghdad, Iraq
[3],[4],[n3],[n4]Belgorod State University, Belgorod, 308015, Pobedy st. 85, Russia

ABSTRACT

The paper presents the research, aimed at improving the efficiency of automated software system for geo-ecological monitoring of agro-industrial sector resources. An algorithm of clusters partition in a self-organizing map was developed, in order to detect resource-intensive inquiries to databases of agricultural resources and objects. The algorithm is based on using fuzzy inference. The corresponding software for implementing the proposed algorithm was created. The carried-out experimental research has demonstrated that this algorithm allows considerably increasing the correctness of detecting resource-intensive inquiries to databases in comparison with other similar software applications. The algorithm, presented in this paper, can be recommended for practical application in an automated software system for geo-ecological monitoring of agricultural resources and objects.

Keywords: Geo-ecological monitoring, Agro-industrial sector, Resource-intensive inquiries, databases, Self-organizing map, Clusters partition.

corresponding author:

Tareq Nasser Mahdi
Mustansiriyah University, College of Pharmacy
Baghdad, Iraq
e-mail: tareq.nasser.m@gmail.com

1. Introduction

At present, agriculture is a production branch, which generates a large amount of information data from various sources, such as farming machinery, satellites and drones, weather stations, agrochemical soil monitoring results. The data, obtained from these sources, are used for forecasting and increasing crop yield, saving time and resources, monitoring soil fertility, taking decisions by agro-industrial sector management authorities. An important tool of optimizing various spheres of business activities, including agro-industrial sector, is geo-ecological monitoring [1-3]. The large amounts of transmitted, processed and analyzed information about the state of agricultural resources and objects are the main cause of implementing automated software systems for geo-ecological monitoring [4]. The databases of such systems are intensively accessed by a large number of users for performing required operations. Among such transactional accesses, resource-intensive inquiries, the processing of which by the server takes a prohibitive amount of time, processor, disc and memory resources, should be detected [5]. The search and conversion of resource-intensive inquiries is performed by data base administrators (dba).

For the search and identification of resource-intensive database inquiries various methods are used; the most efficient of them are based on neural networks, and among them, self-organizing map [6-16]. In the process of self-organizing map training, the weight values of its neurons are tuned in accordance with query parameters values, contained in the training set vectors. As a result, the clusters of neurons are formed, which have close weight values. To be able to use neuron clusters of a self-organizing map for classifying inquiries in terms of their resource-intensiveness, it is necessary to have data about these clusters' borders. The analysis has demonstrated that the known methods of clusters partition (k-means method, singling out clusters on the basis of unified distance matrix) has considerable limitations: the clusters should be of a certain shape; only the nearest

local minimum is searched; the result depends on which initial cluster center or initial coordinate was selected; small distances in an actual cluster can be perceived as distances between different clusters. Based on the foregoing, it can be affirmed that developing an algorithm for clusters partition in a self-organizing map for detecting resource-intensive database inquiries in a geo-ecological monitoring system is a relevant scientific and technological problem.

2. Main part

A considerable cause of drawbacks in cluster borders detection by the k-means method is its imperfection in cases of a large number of parameters in objects to be clustered [17]. In this regard, it is proposed to reduce the weight values set of each neuron in a self-organizing map to generalized quantities. The visual analysis of trained self-organizing maps gives indistinct, fuzzy boundaries of the obtained clusters. In this case it is very difficult to accurately determine a certain numeric values range, by which the neuron's weight vector uniquely determines its membership in this or that cluster. In this regard, in order to determine neuron clusters' borders the fuzzy inference method can be applied, which is successfully used for solving various scientific and applied problems [18-22].

To be able to take into account all values of neurons' weights in the process of clusters partition, let us introduce the values of generalized neurons weights. To calculate value S_k – the generalized weight of neuron number k – the following fuzzy rules can be used:

$$If\ (w_{k1} = w_1^+)and(w_{k2} = w_2^+)and\ldots and(w_{ki} = w_i^+)and\ldots and(w_{kn} = w_n^+) \quad then(S_k = Y_1); \tag{1}$$

$$If\ (w_{k1} = w_1^+)and(w_{k2} = w_2^+)and\ldots and(w_{ki} = w_i^+)and\ldots and(w_{kn} = w_n^-) \quad then(S_k = Y_2); \tag{2}$$

…

$$If\ (w_{k1} = w_1^+)and(w_{k2} = w_2^+)and\ldots and(w_{ki} = w_i^-)and\ldots and(w_{kn} = w_n^-) \quad then(S_k = Y_y); \tag{3}$$

…

$$If\ (w_{k1} = w_1^-)and(w_{k2} = w_2^-)and\ldots and(w_{ki} = w_i^-)and\ldots and(w_{kn} = w_n^-) \quad then(S_k = Y_z), \tag{4}$$

where, w_{ki} – weight number i of neuron number k;

w_i^+ – fuzzy set «high value of the i-th weight of the neuron»;

w_i^- – fuzzy set «low value of the i-th weight of the neuron»;

$Y_1, Y_2, ..., Y_y, ..., Y_z$ – values of individual inferences of the corresponding fuzzy rules;

z – number of fuzzy rules, $z = n^2$.

the values of Y_y should be calculated by the formula:

$$Y_y = \frac{v_1 b_{1y} + v_2 b_{2y} + \ldots + v_i b_{iy} + \ldots + v_n b_{ny}}{v_1 + v_2 + \ldots + v_i + \ldots + v_n}, \tag{5}$$

where, v_i – the value of explained variance of the i-th parameter of query vectors.

Coefficients $b_{1y}, b_{2y}, \ldots, b_{iy}, \ldots, b_{ny}$ are calculated, taking into account to which fuzzy set (w_i^+ or w_i^-) the value w_{ki} in the fuzzy rule number y belongs:

$$b_{iy} = \begin{cases} 1, & w_{ki} = w_i^+; \\ 0, & w_{ki} = w_i^-. \end{cases} \quad (6)$$

The values $w_{k1}, w_{k2}, ..., w_{ki}, ..., w_{kn}$ can to a greater or a lesser extent correspond to fuzzy sets w_i^+ and w_i^-. To calculate the value of this correspondence let us use membership functions $U^+(w_{ki})$ and $U^-(w_{ki})$. They mean that value of function $U^+(w_{ki})$ indicates, what is the probability of value w_{ki} belonging to the fuzzy set «high value of the i-th weight of the neuron», and value of the function $U^-(w_{ki})$ indicates, what is the probability of value w_{ki} belonging to the fuzzy set «low value of the i-th weight of the neuron». The membership functions, presented in Figure. 1 and 2, have found wide application in research practice:

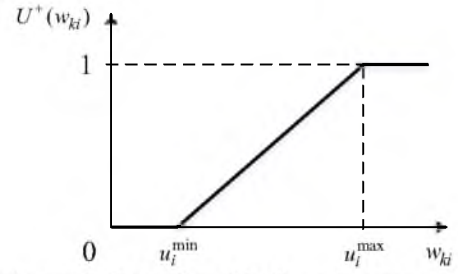

Figure 1. Membership function $U^+(w_{ki})$

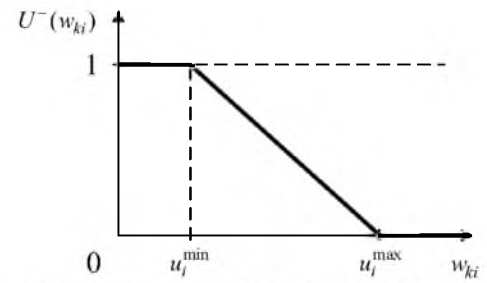

Figure 2. Membership function $U^-(w_{ki})$

In Figure. 1 and 2 the following values are represented:

w_i^{min} – minimum value of weight number i among all the neurons in a self-organizing map;

w_i^{max} – maximum value of weight number i among all the neurons in a self-organizing map.

To calculate value S_k on the basis of fuzzy rules (1) – (4), first it is necessary to perform fuzzification, i.e. To calculate the membership functions of values $w_{k1}, w_{k2}, ..., w_{ki}, ..., w_{kn}$ to fuzzy sets w_i^+ and w_i^- :

$$U^+(w_{ki}) = \begin{cases} 0, & w_{ki} < u_i^{min}; \\ \dfrac{w_{ki} - u_i^{min}}{u_i^{max} - u_i^{min}}, & u_i^{min} \le w_{ki} < u_i^{max}; \\ 1, & w_{ki} \ge u_i^{max}; \end{cases} \quad (7)$$

$$U^-(w_{ki}) = \begin{cases} 1, & w_{ki} < u_i^{min}; \\ \dfrac{u_i^{max} - w_{ki}}{u_i^{max} - u_i^{min}}, & u_i^{min} \le w_{ki} < u_i^{max}; \\ 0, & w_{ki} \ge u_i^{max}. \end{cases} \quad (8)$$

The next stage is aggregation:

$$G_{k1} = U^+(w_{k1}) \wedge U^+(w_{k2}) \wedge \ldots \wedge U^+(w_{ki}) \wedge \ldots \wedge U^+(w_{kn}), \quad (9)$$

$$G_{k2} = U^+(w_{k1}) \wedge U^+(w_{k2}) \wedge \ldots \wedge U^+(w_{ki}) \wedge \ldots \wedge U^-(w_{kn}), \quad (10)$$

$$\ldots$$

$$G_{ky} = U^+(w_{k1}) \wedge U^+(w_{k2}) \wedge \ldots \wedge U^-(w_{ki}) \wedge \ldots \wedge U^-(w_{kn}), \quad (11)$$

$$\ldots$$

$$G_{kz} = U^-(w_{k1}) \wedge U^-(w_{k2}) \wedge \ldots \wedge U^-(w_{ki}) \wedge \ldots \wedge U^-(w_{kn}). \quad (12)$$

The final stage of calculating S_k value is defuzzification:

$$S_k = \frac{G_{k1}Y_1 + G_{k2}Y_2 + \ldots + G_{ky}Y_y + \ldots + G_{kz}Y_z}{G_{k1} + G_{k2} + \ldots + G_{ky} + \ldots + G_{kz}}. \quad (13)$$

As a result of performing the fuzzy inference, each neuron in a self-organizing map would have only one weight S_k. In this case, in order to divide the neurons into C clusters and to determine clusters borders, the k-means algorithm for one-dimensional space can be used. In Figure. 3 and 4 the fragments of block diagram of an algorithm for clustering neurons in a self-organizing map for database inquiries classification are presented.

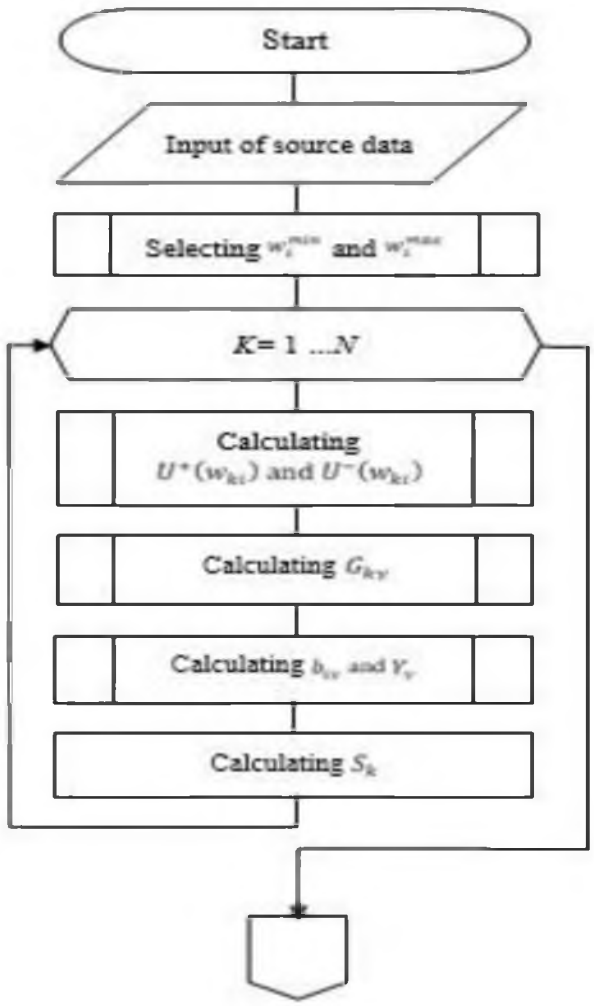

Figure 3. Block diagram of an algorithm for clustering neurons of a self-organizing map

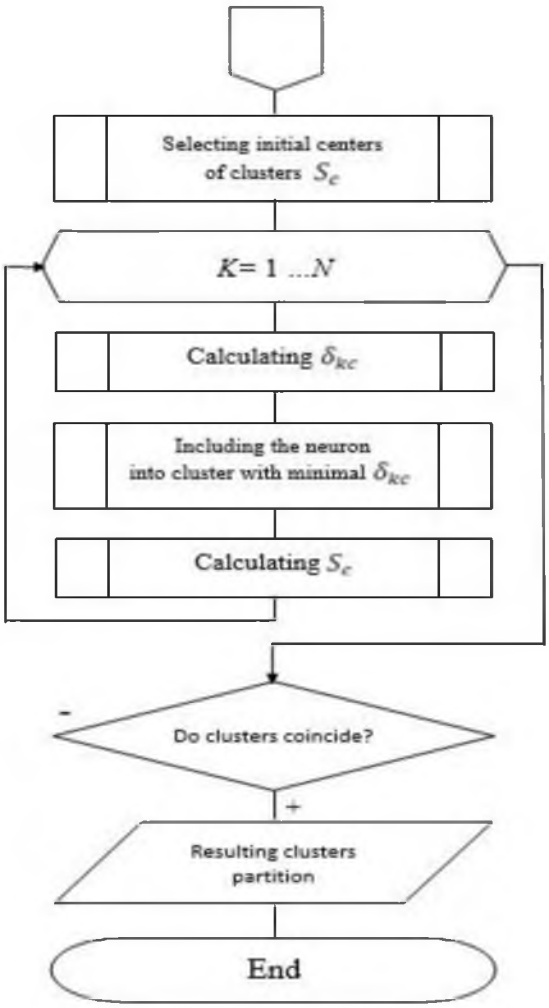

Figure 4. Block diagram of an algorithm for clustering neurons of a self-organizing map (continued)

The algorithm consists of the following steps:
Step 1. Input of source data: the values of neurons weights and the values of explained variance of query vectors parameters.

Step 2. In the values set of a self-organizing map's neurons weights the values $w_1^{\min}, w_2^{\min}, \ldots, w_n^{\min}$ and $w_1^{\max}, w_2^{\max}, \ldots, w_n^{\max}$ are selected.

Step 3. Neuron number $k = 1$ is selected.

Step 4. According to formulas (7) and (8) fuzzification results are calculated –membership functions values of weights of neuron number k to fuzzy sets w_i^+ and w_i^-.

Step 5. According to formulas (9) – (12) aggregation results are calculated G_{ky}.

Step 6. According to formulas (5) and (6) values Y_y and b_{iy} are calculated.

Step 7. For neuron number k according to formula (13) the generalized weight S_k is calculated.

Step 8. The number of the selected neuron increments by 1.
Step 9. Fulfillment of the following condition is checked:

$$k \le N.$$

If this condition is fulfilled, jump to step 4, if not – jump to step 10.
Step 10. From the set of neurons, C neurons are selected, the weights of which would be initial values of cluster centers. At the first stage each cluster would contain only one neuron, the weight of which is selected as the center of this cluster.

Step 11. Neuron number $k = 1$ is selected.
Step 12. For the selected neuron number k proximity measures of its weight S_k to each cluster's center values are calculated by the formula:

$$\delta_{kc} = |S_c - S_k|, \tag{14}$$

where, S_c – center value of cluster number c; c can take a value from 1 to C.

Step 13. The minimum value of δ_{kc} is determined. The selected neuron is included into the cluster, for which the δ_{kc} value is minimal.

Step 14. The center value of the cluster, into which the new neuron was included, is recalculated:

$$S_c = \frac{1}{L_c} \sum_{l=1}^{L_c} S_l, \tag{15}$$

where, S_l – weight value of the l-th neuron of cluster number c;
L_c – number of neurons, contained in cluster number c.
Step 15. The number of the selected neuron increments by 1.
Step 16. Fulfillment of the following condition is checked:

$$k \le N. \tag{16}$$

If this condition is fulfilled, jump to step 12, if not – jump to step 17.

Step 17. The contents of clusters, obtained in the current iteration, are saved. The clusters contents, obtained in the current and in the previous iterations, are compared. If the contents don't coincide, the jump to step 11 is

performed, and the next iteration of neurons clustering is carried out. If the contents coincide, a decision, concerning the current neurons clustering, is taken. End of the algorithm.

The above-mentioned algorithm is realized in the form of software for clustering sql-inquiries with the use of self-organizing map (som-clustering). The software applications were developed in python language on the basis of tensor calculations by means of tensorflow library.

The carried-out analysis has demonstrated that the inquiries, incoming to databases, can be grouped into four clusters:

1st cluster – «heavy» resource-intensive inquiries, which are characterized with the intensive usage of system memory, processors, storage space and output channels in the process of their execution;

2nd cluster – «slow» resource-intensive inquiries, characterized with frequent execution and recompilation procedures;

3rd cluster – spontaneous problematic inquiries, occurring due to occasional temporary resource shortages, caused by peak loads in the network, servers, processors, or resource shortages, occurring in the process of executing other inquiries;

4th cluster – other, not resource-intensive (not problematic), inquiries.

To classify various types of inquiries, over 500 experiments were carried out, in the course of which the developed method of detecting resource-intensive inquiries was used. The obtained findings were used for calculating the correctness indices of detecting resource-intensive inquiries.

Correctness of detecting inquiries to databases was evaluated by two indicators:

1) detection probability of resource-intensive inquiries to databases;

2) probability of error detection of resource-intensive inquiries to databases.

Detection probability value of resource-intensive inquiries to databases was calculated by the formula:

$$P_{\text{det}} = \frac{Q_{\text{det}}}{Q_{\Sigma}}, \tag{17}$$

where, Q_{det} – the number of detected resource-intensive inquiries, incoming to databases;

Q_{Σ} – the total number of resource-intensive inquiries, incoming to databases.

To calculate the probability of error detection of resource-intensive database inquiries the following expression was used:

$$P_{\text{err}} = \frac{Q_{\text{err}}}{Q_{\text{det}}}, \tag{18}$$

where, Q_{err} – the number of incoming inquiries to databases, which were erroneously classified as resource-intensive.

3. Results and discussion

Besides, a number of experiments in detecting resource-intensive inquiries by using up-to-date sql query tuning and monitoring applications tuning advisor [23, 24] and oracle cost-based optimizer [25, 26] were carried out. The findings of the research results are presented in Table 1.

Table 1. Evaluation results of detection correctness of resource-intensive database inquiries

Studied software	Values of P_{det}	Values of P_{err}
sql tuning advisor	0.825	0.094
Oracle cost-based optimizer	0.844	0.102
Som-clustering	0.951	0.082

Analysis of the data, presented in tab. 1, demonstrates that the proposed algorithm allows increasing the probability of detecting resource-intensive database inquiries by 12.68% – 15.27% and reducing the probability of erroneous detecting of resource-intensive database inquiries by 12.20% – 24.39% in comparison with the currently used sql-inquiries tuning and optimization applications.

4. Conclusions

The presented research is aimed at solving the problem of clustering of a self-organizing map for detecting resource-intensive inquiries to databases of an automated software system for geo-ecological monitoring of agro-industrial resources.

An algorithm for clusters partition in a self-organizing map, designed for database inquiries classification, was developed. The novelty of the algorithm consists in using fuzzy inference for calculating generalized weights of neurons. The usage of this algorithm allows determining clusters' borders for the subsequent detecting of resource-intensive database inquiries by means of neural network. The corresponding software was developed for implementing this algorithm.

The experimental research has demonstrated that using the proposed algorithm allows considerably increasing the probability of detecting resource-intensive inquiries to databases and reducing the probability of erroneous detecting of resource-intensive inquiries to databases in comparison with using other similar applications. Based on the foregoing, the algorithm, presented in this paper, can be recommended for using in practice for improving the efficiency of agro-industrial sector's automated geo-ecological monitoring software systems functioning.

Acknowledgements

the theory was prepared within the framework of the state task of the russian federation fzwg-2020-0029 "development of theoretical foundations for building information and analytical support for telecommunications systems for geoecological monitoring of natural resources in agriculture".

References

[1] O. Kuzichkin, R. Romanov, N. Dorofeev, and A. Grecheneva, "Organization and application of information and analytical support for geological monitoring of water use," *IIOAB Journal,* vol. 11, pp. 20-26, 2020.

[2] R. Romanov, O. Kuzichkin, G. Vasiliev, and E. Mikhaleva, "The use of the geoelectric method of the express control during the geoecological monitoring of the decentralized water supply," in *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2019, pp. 1-5.

[3] G. S. Vasilyev, O. R. Kuzichkin, A. V. Grecheneva, N. V. Dorofeev, and D. S. Surzhik, "Analysis of the combined transfer functions for geotechnical control," *International Multidisciplinary Scientific GeoConference: SGEM,* vol. 18, pp. 43-50, 2018.

[4] K. Polshchykov, A. Shabeeb, S. Lazarev, and V. Kiselev, "Justification for the decision on loading channels of the network of geoecological monitoring of resources of the agroindustrial complex," *Periodicals of Engineering and Natural Sciences,* vol. 9, pp. 781-787, 2021.

[5] S. M. Alghazali, K. Polshchykov, A. M. Hailan, and L. Svoykina, "Development of Intelligent Tools for Detecting Resource-intensive Database Queries," *Development,* vol. 12, 2021.

[6] I.-T. Chen, L.-C. Chang, and F.-J. Chang, "Exploring the spatio-temporal interrelation between groundwater and surface water by using the self-organizing maps," *Journal of Hydrology,* vol. 556, pp. 131-142, 2018.

[7] L.-C. Chang, W.-H. Wang, and F.-J. Chang, "Explore training self-organizing map methods for clustering high-dimensional flood inundation maps," *Journal of Hydrology,* vol. 595, p. 125655, 2021.

[8] G. Chamundeswari, G. Varma, and C. Satyanarayana, "Contact Distribution Function based Clustering Technique with Self-Organizing Maps," *International Journal of Image, Graphics and Signal Processing,* vol. 12, p. 9, 2018.

[9] D. Chushig-Muzo, C. Soguero-Ruiz, A. P. Engelbrecht, P. D. M. Bohoyo, and I. Mora-Jiménez, "Data-driven visual characterization of patient health-status using electronic health records and self-organizing maps," *IEEE Access,* vol. 8, pp. 137019-137031, 2020.

[10] F. J. Abarca-Alvarez, M. L. Navarro-Ligero, L. M. Valenzuela-Montes, and F. S. Campos-Sánchez, "European strategies for adaptation to climate change with the Mayors adapt initiative by self-organizing maps," *Applied Sciences,* vol. 9, p. 3859, 2019.

[11] B. Girau and C. Torres-Huitzil, "Fault tolerance of self-organizing maps," *Neural Computing and Applications,* pp. 1-17, 2018.

[12] D. Guamán, S. Delgado, and J. Pérez, "Classifying Model-View-Controller Software Applications Using Self-Organizing Maps," *IEEE Access,* vol. 9, pp. 45201-45229, 2021.

[13] A. Alqudah, H. R. Al-Zoubi, M. A. Al-Khassaweneh, and M. Al-Qodah, "Highly Accurate Recognition of Handwritten Arabic Decimal Numbers Based on a Self-Organizing Maps Approach," *INTELLIGENT AUTOMATION AND SOFT COMPUTING,* vol. 24, pp. 493-505, 2018.

[14] T. H. Nguyen, "Metagenome-based disease classification with deep learning and visualizations based on self-organizing maps," in *International Conference on Future Data and Security Engineering*, 2019, pp. 307-319.

[15] H. Pen, Q. Wang, and Z. Wang, "Boundary precedence image inpainting method based on Self-organizing Maps," *Knowledge-Based Systems,* vol. 216, p. 106722, 2021.

[16] X. Qu, L. Yang, K. Guo, L. Ma, M. Sun, M. Ke, *et al.*, "A survey on the development of self-organizing maps for unsupervised intrusion detection," *Mobile networks and applications,* pp. 1-22, 2019.

[17] E. F. Galutira, A. C. Fajardo, and R. P. Medina, "A novel Kohonen self-organizing maps using exponential decay average rate of change for color clustering," in *Intelligent and Interactive Computing*, ed: Springer, 2019, pp. 23-33.

[18] K. Polshchykov, S. Lazarev, O. Polshchykova, and E. Igityan, "The algorithm for decision-making supporting on the selection of processing means for big arrays of natural language data," *Lobachevskii Journal of Mathematics,* vol. 40, pp. 1831-1836, 2019.

[19] K. Polshchykov, S. Lazarev, I. Konstantinov, O. Polshchykova, L. Svoikina, E. Igityan, *et al.*, "Assessing the Efficiency of Robot Communication," *Russian Engineering Research,* vol. 40, pp. 936-938, 2020.

[20] I. Konstantinov, K. Polshchykov, S. Lazarev, and O. Polshchykova, "Model of neuro-fuzzy prediction of confirmation timeout in a mobile ad hoc network," 2017.

[21] K. Polshchykov, S. Lazarev, and A. Zdorovtsov, "Neuro-fuzzy control of data sending in a mobile ad hoc network," *Journal of Fundamental and Applied Sciences,* vol. 9, pp. 1494-1501, 2017.

[22] O. Ivaschuk, K. Polschykov, S. Lazarev, O. Ivaschuk, and V. Fedorov, "Integral estimate of terrestrial compartment condition in management of Biotechnosphere of Rural and Urban Areas," *International Journal of Pharmacy and Technology,* vol. 8, pp. 27032-27038, 2016.

[23] R. Cornejo, "Performance Tuning Basics," in *Dynamic Oracle Performance Analytics*, ed: Springer, 2018, pp. 3-16.

[24] D. Kuhn, S. R. Alapati, and B. Padfield, "SQL Tuning Advisor," in *Expert Indexing in Oracle Database 11g*, ed: Springer, 2012, pp. 205-231.

[25] H. Rahman, "Oracle Database Design and Development in Ador Composite Ltd," 2020.

[26] M. Sharma, G. Singh, and R. Singh, "A review of different cost-based distributed query optimizers," *Progress in Artificial Intelligence,* vol. 8, pp. 45-62, 2019.