Robust Adaptive SOMs Challenges in a Varied Datasets Analytics

Alaa Ali Hameed[1(✉)], Naim Ajlouni[2], and Bekir Karlik[3]

[1] Department of Mathematics and Computer Science, Istanbul Aydin University, Istanbul, Turkey
aalihameed@aydin.edu.tr

[2] Department of Computer Engineering, Istanbul Aydin University, Istanbul, Turkey
naimajlouni@aydin.edu.tr

[3] Neurosurgical Simulation Research and Training Centre, McGill University, Montreal, QC, Canada
bkarlik@hotmail.com

Abstract. The advancement of available technology in use cause the production of huge amounts of data which need to be categorised within an acceptable time for end users and decision makers to be able to make use of the data contents. Present unsupervised algorithms are not capable to process huge amounts of generated data in a short time. This increases the challenges posed by storing, analyzing, recognizing patterns, reducing the dimensionality and processing Data. Self-Organizing Map (SOM) is a specialized clustering technique that has been used in a wide range of applications to solve different problems. Unfortunately, it suffers from slow convergence and high steady-state error. The work presented in this paper is based on the recently proposed modified SOM technique introducing a Robust Adaptive learning approach to the SOM (RA-SOM). RA-SOM helps to overcome many of the current drawbacks of the conventional SOM and is able to efficiently outperform the SOM in obtaining the winner neuron in a lower learning process time. To verify the improved performance of the RA-SOM, it was compared against the performance of other versions of the SOM algorithm, namely GF-SOM, PLSOM, and PLSOM2. The test results proved that the RA-SOM algorithm outperformed the conventional SOM and the other algorithms in terms of the convergence rate, Quantization Error (QE), Topology Error (TE) preserving map using datasets of different sizes. The results also showed that RA-SOM maintained an efficient performance on all the different types of datasets used, while the other algorithms a more inconsistent performance, which means that their performance could be data type-related.

Keywords: Robust · Adaptive-SOM · Clustering · Topology

A. Vellido et al. (Eds.): WSOM 2019, AISC 976, pp. 110–119, 2020.
https://doi.org/10.1007/978-3-030-19642-4_11

1 Introduction

The Self-Organizing Map (SOM) is an unsupervised learning algorithm introduced by Kohonen [1]. In the area of artificial neural networks, the SOM is an excellent data-exploring tool as well [2, 3]. It can project high-dimensional patterns onto a low-dimensional topology map. The SOM map consists of a one or two dimensional (2-D) grid of nodes. These nodes are also called neurons. Each neuron's weight vector has the same dimension as the input vector. The SOM obtains a statistical feature of the input data and is applied to a wide field of data classification [4–6]. SOM is based on competitive learning.

In SOM prior knowledge of the target output is not required for the recognition of process. This algorithm works by finding input features similarities within data objects to define their relation by calculating the distance between them [15]. The nodes output must map to the same weighed vector have been proposed by [7–9]. The winner output is defined as the node with the shortest distance between that node and the input vector. The weighted model continues to be updated to obtain the optimal cluster's topology [10]. The training time depends on dataset-size and the ability to find optimal weights within an acceptable time.

A number of modified SOM versions are developed and proposed for the improvement of vector quantization and the topology preservation performances [11–18]. Brugger et al., and Bogdan et al. proposed a method for detecting clusters by applying the different clustering algorithm to SOM [12, 19].

Berglund and Sitte [20, 21] proposed Parameter-Less SOM (PLSOM) and Parameter-Less SOM2 (PLSOM2) to overcome limitations with Kohonen SOM. PLSOM uses a Quadratic function for error fitting in place of the well-known neighbourhood size and learning parameters, this method suffers from initial weight distribution overreliance and oversensitivity to outliers. The PLSOM2 extended the work of PLSOM by updating the weights by scaling them according to input range observed instead of updating them based on the size of error relative to training maximum error.

In this paper, the performance of the RA-SOM algorithm, which employs a decreasing adaptive learning rate function, is to be tested using a number of different data types. The performance of the RA-SOM will then be compared against well-known algorithms which will be tested using the same datasets. It is expected that the RA-SOM will perform more efficiently than the other algorithms as it will require lower implementation run times to achieve the desired convergence, provide a lower Quantization Error QE, and maintain the topology of the clusters [22]. The test will be carried out on a number of datasets obtained from UCI and KEEL repository.

The remainder of the paper is organized as follows: Sect. 2 reviews the conventional SOM algorithm, Sect. 3 Reviews the RA-SOM algorithm, and Sect. 4 presents the simulation results and a performance comparison between the RA-SOM and other known algorithms including Kohonen SOM, PLSM, PLSOM2, and GF-SOM. The conclusions and future work are presented in Sect. 5.

2 SOM Algorithm

The SOM architecture is composed of input and output layers, connected by link-associated weights. The SOM map uses neuron connections topologies of the hexagonal and rectangular form [15, 16]. SOM output layers contain $n \times m$ neurons arranged as a two-dimensional grid. The original n-dimensional data are transferred to a two dimensional map in SOM. In this case the input vector $x_i = \{x_1, x_2, \ldots, x_n\}$, $i = 1, 2, \ldots, n$, where i is the number of input and n is the input units of the vector. Each i is associated to the map through a weight vector $w = \{w_{n1}, w_{n2}, \ldots, w_{nm}\}$.

SOM adapt a number of processes: First step, the $n \times m$ neuron weight vector is initialized randomly, the second step, an input vector x from the dataset is fed into the SOM network. Input vector x is fed to all neurons, at the same time. Third, the distance between the input and output neurons are calculated, then the closest neuron to the input identified (closest-distance) in this case using Euclidean Distance; this will be called the Best Matching Unit (BMU). The wining neuron is denoted by c.

$$c = arg\,\underset{i}{min}(\|w_i(t) - x(t)\|). \tag{1}$$

This process is iterated for entire input vectors in the dataset. In each iteration, the weight vector is updated by the winning neuron by:

$$w_i(t+1) = w_i(t) + \alpha(t).[x(t) - w_i(t)], \tag{2}$$

where $\alpha(t)$ is the learning rate. The GF-SOM algorithm utilizes a Gaussian-function which is given by:

$$w_i(t+1) = w_i(t) + h_{c,i}(t).[x(t) - w_i(t)], \tag{3}$$

where $h_{c,i}$ is the Gaussian neighborhood function given as

$$h_{c,i}(t) = \alpha(t).exp\left(-\frac{\|r_c - r_i\|}{2\sigma^2(t)}\right), \tag{4}$$

where $\|r_c - r_i\|$ is the Euclidean distance between the positions of the winning neuron c and the neuron i on the grid in each updated weight, and $\sigma(t)$ is the width of Gaussian. $\alpha(t) = \delta_\alpha.\alpha(t)$ and $\sigma(t) = \delta_\sigma.\sigma(t)$ are decreasing gradually during the learning process by constants factors δ_α and δ_σ, respectively.

3 RA-SOM Algorithm

The conventional Kohonen SOM algorithm uses a fixed learning α which is usually between 0–1. The choice of the learning rate affects the speed of the conversion and accuracy of the optimum model. It is known that the higher the learning rate, the faster the convergence. However, this will not guarantee the accuracy of the data topology (clustering), as data accuracy will require a lower leaning rate. Therefore, choosing a high learning rate will provide a high initial convergence, but once this is achieved the algorithm will be forced to diverge to a higher QE due to the inaccuracy in the data topology. On the other hand, choosing a small value for the learning rate will cause a slow divergence which will require many more iterations to achieve the required low QE. This cannot be acceptable in the case of big data.

For this reason, the RA-SOM introduces an adaptive learning rate $\alpha(t)$ [22, 23] which is of a decreasing form, it start by introducing a high learning rate $\alpha(t)$, which is decreased adaptively in subsequent iterations. The adaptively decreasing learning rate achieves high convergence during the first few iterations, this will be followed by a lower learning rate $\alpha(t)$, which will guarantee the continuous high convergence, and will help in defining the accuracy of the data clusters.

The new adaptive learning rate adapted in the RA-SOM is applied in this paper this will be of the form given by

$$w_i(t+1) = w_i(t) + \alpha(t).[x(t) - w_i(t)], \quad t = 0, 1, \ldots \tag{5}$$

where $w_i(t+1)$ is defined as the updating weights, and $\alpha(t)$ is a variable adaptive learning convergence rate is defined as

$$\alpha(t) = \frac{\lambda}{1 - \beta^t} \tag{6}$$

As a result, substitute (6) in (5) deriving a new format of RA-SOM as

$$w_i(t+1) = w_i(t) + \left(\frac{\lambda}{1 - \beta^t}\right).[x(t) - w_i(t)]. \tag{7}$$

In the RA-SOM, the weight vector w is randomly initialized as a grid of $n \times m$ neurons similar to the conventional SOM algorithm. Then, updating the weights is controlled adaptively through the proposed learning algorithm. The optimal weights are obtained in a shorter time compared to the conventional SOM, PLSOM and PLSOM2 algorithms. Moreover, the optimum weight vectors are also improved and provide lower quantization error. The logic behind the adaptive learning function (6) is quite simple: at the start of the function the value of β is large enough while the value of t is small, hence the term $(1 - \beta^t)$ will be relatively small, therefore $\alpha(t)$ will be relatively large, which will result in faster convergence of the updated weights in (7). As time t increases, the term $(1 - \beta^t)$ increases to a value close to unity, and hence $\alpha(t)$ will then be close to or equal to λ, which will result in low error performance in the updated weights of (7).

4 Simulation Results

The proposed algorithm has been tested in four different applications to assess its performance. In this paper, the methods were coded using MATLAB R2010b, and the tests were performed using a Core (TM) i7-3612QM CPU (2.10 GHz) PC equipped with 8,00 GB of RAM with Windows 7 Ultimate operating system. This involved a number of tests which were carried out on different datasets collected from UCI and KEEL repository. Data were divided into 70% training and 30% testing sets. Datasets used in this test were normalized using Min-Max normalization between 0 and 1. A comprehensive comparison between (QE), topology error (TE), and run time was carried out for all the test using all the different datasets. The algorithms tested during the test are conventional Kohonen SOM, GF-SOM, PLSOM, and PLSOM2. The results of these algorithms were then compared against the performance of the RA-SOM under the same tests conditions.

4.1 Balance Dataset

This test was carried out using the Balance dataset. The Balance dataset consists of 625 instances with 4 attributes and 3 classes. The dataset was collected from UCI and KEEL repository. This data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The Kohonen map, in this case, consists of 4 neurons for the input layer with a 2D grid of 4×3 neurons in the competitive layer. The experimental results are reported in Tables 1, 2, 3, 4 and 5. They show that the RA-SOM outperforms all rest of algorithms, with the PLSOM being the second best.

It must be noted that the conventional SOM had the worst performance in this test. The RA-SOM obtained the lowest QE using $\lambda = 0.5 \times 10^{-2}$ and $\beta = 0.992, 0.991$, however in the following parameters the QE increased, but still was much lower than the QE obtained by other algorithms as shown in Fig. 1. The test also shows that even through RA-SOM outperformed all other algorithms during all subsequent algorithm runs. From the result it can also be seen that the performance of the algorithms is parameter-dependent, this was very clear when considering the performance of SOM, GF-SOM and PLSOM2 algorithms, for example at parameter 1, PLSOM2 with parameters ($\beta = 1.3$, $QE = 0.222$) outperformed both SOM with parameters ($\delta_\alpha = 0.17$, and $QE = 0.24$) and GF-SOM with parameters ($\delta_\alpha = 1$, and $\delta_\sigma = 0.85 \times 10^{-2}, 0.87\times\ 10^{-2}$, and $QE = 0.242$), this later changed at parameter 3, PLSOM2 at ($\beta = 1.3$, $QE = 0.243$), SOM at parameters ($\delta_\alpha = 0.15$, and $QE = 0.24$), and GF-SOM at parameters ($\delta_\alpha = 0.8$, and $\delta_\sigma = 0.85 \times 10^{-2}, 0.87 \times 10^{-2}$, and $QE = 0.24$).

4.2 Dermatology Dataset

The test was carried out on the Dermatology dataset. The dataset was collected from UCI and KEEL repository. The differential diagnosis of erythemato-squamous diseases is a real problem in dermatology. The dataset consists of 366 instances, 33 attributes and 6 classes namely (psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis and pityriasis rubrapilaris). The structure of the Kohonen map used in this case consists of 33 neurons for the input layer and 2D grid size of 33×6 neurons in the competitive layer.

The results of the test are provided in Tables 6, 7, 8, 9 and 10 for the conventional SOM, GF-SOM, PLSOM, PLSOM2, and RA-SOM, respectively. Figure 2 shows the relevant QE against the run time. From this result, it can be concluded that the RA-SOM outperform all algorithms by obtaining lowest initial QE = 0.174 with the following parameters ($\lambda = 0.7 \times 10^{-2}$, and $\beta = 0.992$), this was further improved to obtain an optimal QE = 0.173 at ($\lambda = 0.5 \times 10^{-2}$, and $\beta = 0.99$); these values remained consistent in all subsequent runs. The result shows that the RA-SOM defined a dataset cluster topology in early run times and managed to maintain this topology throughout.

Table 1. QE results of the conventional SOM algorithm for balance dataset

δ_α	0.17	0.16	0.15
QE	0.24	0.25	0.24

Table 2. QE results of the GF-SOM algorithm for balance dataset

δ_σ	δ_α		
	1	0.9	0.8
0.0087	0.242	0.239	0.24
0.0086	0.242	0.24	0.24
0.0085	0.242	0.238	0.24

Table 3. QE results of the PLSOM algorithm for balance dataset

B	4	3	2
QE	0.22	0.22	0.23

Table 4. QE results of the PLSOM2 algorithm for balance dataset

B	1.5	1.4	1.3
QE	0.222	0.224	0.243

Table 5. QE results of the RA-SOM algorithm for balance dataset

β	λ		
	0.005	0.004	0.003
0.992	0.2	0.21	0.2
0.991	0.2	0.21	0.216
0.99	0.22	0.22	0.2

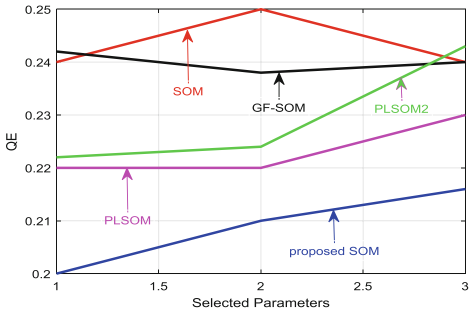

Fig. 1. Comparison of QE measures with various test parameters for balance dataset.

However, the rest of the algorithms did not manage to maintain the topology, as shown in Fig. 2. The PLSOM started at a high QE = 0.188 in the first run at $\beta = 13.7$; this was reduced to QE = 0.177 at the second run at $\beta = 13.6$, and slightly improved by the third run to QE = 0.176 at $\beta = 13.5$. No more improvement was obtained for any further runs and variations of β. The PLSOM2 also started with a QE = 0.182 at $\beta = 7.5$; this was further reduced to QE = 0.175 at $\beta = 8$; however, QE was increased at the third run to QE = 0.179 at $\beta = 8.5$, which indicates that the algorithm had difficulty to maintain the dataset cluster topology. The conventional SOM in this test provided a lower initial QE = 0.181 at $\delta_\alpha = 0.6$, which is better than the performance of both PLSOM and PLSOM2. However, this was not maintained as both of the algorithms performed much better at subsequent runs. The optimal QE for the conventional SOM is QE = 0.18 at $\delta_\alpha = 0.4$, which was the worse between all algorithms. The GF-SOM started at QE = 0.179 at $\delta_\alpha = 0.6$ and $\delta_\sigma = 0.4$. This improved to an optimum of QE = 0.177 at $\delta_\alpha = 0.4$ and $\delta_\sigma = 0.02$. The algorithm best QE was much

Table 6. QE results of the conventional SOM algorithm for dermatology dataset

δ_α	0.6	0.5	0.4
QE	0.181	0.181	0.18

Table 7. QE results of the GF-SOM algorithm for dermatology dataset

δ_σ	δ_α		
	0.6	0.5	0.4
0.04	0.179	0.177	0.177
0.03	0.181	0.181	0.182
0.02	0. 18	0.18	0.177

Table 8. QE results of the PLSOM algorithm for dermatology dataset

B	13.7	13.6	13.5
QE	0.188	0.177	0.176

Table 9. QE results of the PLSOM2 algorithm for dermatology dataset

B	8.5	8	7.5
QE	0.179	0.175	0.182

Table 10. QE results of the RA-SOM algorithm for dermatology dataset

β	λ		
	0.007	0.006	0.005
0.992	0.174	0.179	0.184
0.991	0.176	0.176	0.179
0.99	0.18	0.1732	0.173

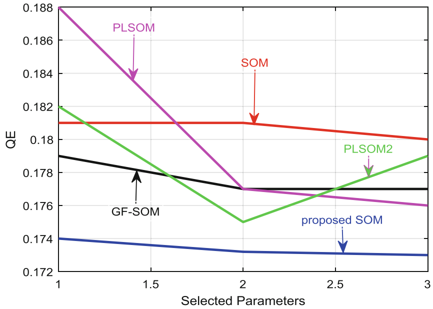

Fig. 2. Comparison of QE measures with various test parameters for dermatology dataset

higher than the global best which was obtained by the RA-SOM which is QE = 0.173. However, from Fig. 2, it can be seen that the GF-SOM did not undergo too many topology changes compared to the PLSOM and PLSOM2 algorithms.

4.3 Arcene Dataset

The dataset was collected from UCI and KEEL repository. ARCENE was obtained by merging three mass-spectrometry datasets to obtain enough training and test data for a benchmark. Another dataset we used to examine the efficiency and investigate the performance of the RA-SOM algorithm against other algorithms was the Arcene dataset. It consists of 100 instances, 10000 attributes and 2 classes. The structure of the used Kohonen maps consists of 100 neurons for the input layer with a 2D grid of 100×2 neurons in the competitive layer.

The basic parameters used in this experiment are: for the conventional SOM algorithm: $\delta_\alpha = 0.5 \times 10^{-5}$, for the PLSOM2 algorithm, $\beta = 0.5 \times 10^{-5}$ and for the proposed RA-SOM: $\lambda = 0.5 \times 10^{-10}$, and $\beta = 0.8$. The test results are provided in Table 11. The number of iterations used in this test the same for all three algorithms (SOM, PLSOM2, and RA-SOM). The test results show that RA-SOM outperformed the other two algorithms by obtaining the lowest QE = 0.067, with an accuracy of 66.67%. The CPU time shows that the PLSOM2 was the worst, which is expected as the algorithms require many more iterations to complete a cycle compared to both Conventional SOM and RA-SOM, RA-SOM needed extra CPU time as more iterations are needed to calculate the Adaptive learning rate compared to conventional SOM.

Table 11. Performance comparison of the conventional SOM, GF-SOM, PLSOM, PLSOM2 and RA-SOM for Arcene dataset

Appendicitis dataset	Accuracy (%)	# iteration	QE	CPU time
Conventional SOM	60.00	100	0.0645	2.88
PLSOM2	60.00	100	0.0634	3.21
RA-SOM	66.67	100	0.0607	2.90

4.4 Gisette Dataset

The dataset was collected from UCI and KEEL repository. The digits have been size-normalized and centered in a fixed-size image of dimension 28×28. The original data were modified for the purpose of the feature selection challenge. The final dataset used to examine the efficiency and investigate the performance of the proposed SOM algorithm against other algorithms is the Gisette dataset. The Gisette dataset consists of 6000 instances, 5000 attributes and 2 classes. The structure of the used Kohonen maps consists of 100 neurons for the input layer with a 2D grid of 100×2 neurons in the competitive layer.

The basic parameters used in this experiment are: for the conventional SOM algorithm: $\delta_\alpha = 0.5 \times 10^{-7}$, for the PLSOM2 algorithm, $\beta = 0.5 \times 10^{-5}$ and for the proposed SOM: $\lambda = 0.5 \times 10^{-8}$, and $\beta = 0.9$. The test results are provided in Table 12 below. The number of iterations used in this test the same for all three algorithms (SOM, PLSOM2, and RA-SOM). The test results show that RA-SOM has outperformed the other two algorithms by obtaining the lowest QE = 0.0419, with an accuracy of 60.44%. The CPU time shows that the PLSOM2 was again the worst. RA-SOM needed extra CPU time as more iterations are needed to calculate the Adaptive learning rate compared to conventional SOM.

Table 12. Performance comparison of the conventional SOM, GF-SOM, PLSOM, PLSOM2 and RA-SOM for Gisette dataset

Appendicitis dataset	Accuracy (%)	# iteration	QE	CPU time
Conventional SOM	50.83	100	0.0443	5.56
PLSOM2	55.89	100	0.0431	9.29
RA-SOM	60.44	100	0.0419	5.69

5 Conclusion and Future Work

In this work, several alternative algorithms we tested to the proposed RA-SOM under the same conditions. Results showed that the RA-SOM performed more efficiently than the other algorithms in all the datasets tested. It was noticed that the RA-SOM not just outperformed the other algorithms, but it also maintained the dataset variations. The increase or reduction of the number of classes, instances and attributes had no effects on the abilities of the RA-SOM to efficiently converge the QE end the algorithms ability to maintain the dataset topology. It is well known that selecting suitable learning parameters is key to obtain an optimum model with lower clustering topology error. This is one of the main drawbacks in model estimation and bound to be even a bigger issue in big data contexts, as selecting the optimum parameters one needs to run the program many times and each run may be extremely time-consuming. RA-SOM offers more flexibility to obtain the different selection of parameters and thus obtain relevant optimum model quickly and more efficiently.

References

1. Kohonen T (1990) The self-organizing maps. Proc IEEE 78(9):1464–1480
2. Kohonen T, Oja E, Simula O, Visa A, Kangas J (1996) Engineering applications of the self-organizing map. Proc IEEE 84(9):1358–1384
3. Shieh SL, Liao IE (2012) A new approach for data clustering and visualization using self-organizing maps. Expert Syst Appl 39:11924–11933
4. Kohonen T (2001) Self-organizing maps. Springer, Berlin
5. Vesanto J, Alhoniemi E (2002) Clustering of the self-organizing map. IEEE Trans Neural Netw 11(3):586–600
6. Lapidot I, Guterman H, Cohen A (2002) Unsupervised speaker recognition based on competition between self-organizing maps. IEEE Trans Neural Netw 13(4):877–887
7. Kohonen T (1989) Self-organization and associative memory process. Springer, Berlin

8. Hoffmann M (2005) Numerical control of Kohonen neural network for scattered data approximation. Numer Algorithms 39(1):175–186
9. Marini F, Zupan J, Magri AL (2005) Class-modeling using Kohonen artificial neural networks. Anal Chim Acta 544(1):306–314
10. Astel A, Tsakovski S, Barbieri P, Simeonov V (2007) Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets. Water Res 41(19):4566–4578
11. Appiah K, Hunter A, Dickinson P, Meng H (2012) Implementation and applications of tri-state self-organizing maps on FPGA. IEEE Trans Circuits Syst Video Technol 22(8):1150–1160
12. Brugger D, Bogdan M, Rosenstiel W (2008) Automatic cluster detection in Kohonen's SOM. IEEE Trans Neural Netw 19(3):442–459
13. Chi SC, Yang CC (2008) A two-stage clustering method combining ant colony SOM and k-means. J Inf Sci Eng 24(5):1445–1460
14. Cottrell M, Gaubert P, Eloy C, Francois D, Hallaux G, Lacaille J, Verleysen M (2009) Fault prediction in aircraft engines using self-organizing maps. In: International workshop on self-organizing maps, vol 5629. Springer, Heidelberg, pp 37–44
15. Tasdemir K, Milenov P, Tapsall B (2011) Topology-based hierarchical clustering of self-organizing maps. IEEE Trans Neural Netw 22(3):474–485
16. Wong MLD, Jack LB, Nandi AK (2006) Modified self-organising map for automated novelty detection applied to vibration signal monitoring. Mech Syst Signal Process 20 (3):593–610
17. Yang L, Ouyang Z, Shi Y (2012) A modified clustering method based on self-organizing maps and its applications. Procedia Comput Sci 9:1371–1379
18. Yen GG, Wu Z (2008) Ranked centroid projection: a data visualization approach with self-organizing maps. IEEE Trans Neural Netw 19(2):245–259
19. Bogdan M, Rosenstiel W (2001) Detection of cluster in self-organizing maps for controlling a prostheses using nerve signals. In: Proceedings of 9th European symposium on artificial neural networks. ESANN, pp 131–136
20. Berglund E, Sitte J (2006) The parameterless self-organizing map algorithm. IEEE Trans Neural Netw 17(2):305–316
21. Berglund E (2010) Improved PLSOM algorithm. Appl Intell 32(1):122–130
22. Hameed AA, Karlik B, Salman MS, Eleyan G (2019) Robust adaptive learning approach to self-organizing maps. Knowl-Based Syst 171:25–36
23. Hameed AA, Karlik B, Salman MS (2016) Back-propagation algorithm with variable adaptive momentum. Knowl Based Syst 114:79–87