FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 40, No 3 (2025), 563–596
https://doi.org/10.22190/FUMI240616041M
Original Scientific Paper

THE STUDY OF OBJECTS CLUSTERING ALGORITHMS BASED ON SELF-ORGANIZING KOHONEN CARDS USING METHODS OF EXTRACTING FACTORS *

Egor Markushin, Guzel Shkaberina, Natalya Rezova, Aleksey Popov and Lev Kazakovtsev

Reshetnev Siberian State University of Science and Technology
prosp. Krasnoyarskiy Rabochiy, 31, 660039, Krasnoyarsk, Russia

ORCID IDs:		
	Egor Markushin	N/A
	Guzel Shkaberina	https://orcid.org/0000-0001-8257-7329
	Natalya Rezova	https://orcid.org/0000-0002-1149-3299
	Aleksey Popov	N/A
	Lev Kazakovtsev	https://orcid.org/0000-0002-0667-4001

Abstract. We proposed algorithms for object clustering based on the self-organizing Kohonen maps using various methods of extracting factors (factor analysis: Principal Component Analysis, Maximum Likelihood Estimation, Principal Component Analysis based on Singular Value Decomposition) for preliminary reduction of the dimensionality of the initial data. The proposed algorithms reduce the error rate in automatic object grouping compared to conventional k-means models and self-organizing maps. We performed experiments with different distance measures (Mahalanobis, Euclidean, squared Euclidean, Manhattan), and various ways of neuron weight initialization (random, with a choice of weight coefficients from a dataset). The computational experiments showed that the used methods for extracting factors in the self-organizing maps algorithm improve the accuracy of clustering in most cases. Moreover, clustering accuracy decreases with increasing homogeneous batches in a mixed lot.

Keywords: self-organizing Maps, factor analysis, clustering algorithms.

1. Introduction

Self-Organizing Maps (SOM) [18] is a type of artificial neural network with unsu-

Received: June 16, 2024, revised: September 16, 2024, accepted: September 18, 2024
Communicated by Predrag Stanimirović
Corresponding Author: L. Kazakovtsev

E-mail addresses: mirrorsnight@gmail.com (E. Markushin), z_guzel@mail.ru (G. Shkaberina), natalyakl@yandex.ru (N. Rezova), ampopov1959@gmail.com (A. Popov), levk@bk.ru (L. Kazakovtsev)

2020 *Mathematics Subject Classification.* Primary 62P30; Secondary 62F10, 62M10

*This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. FEFE-2023-0004

pervised learning that performs the task of visualization and clustering. SOM performs vector quantization by dividing the continuous input space of n-dimensional data vectors into a reduced subset of prototype vectors organized into a regular grid (often two-dimensional). SOM provides several advantages over other clustering methods. First, it projects multidimensional data into an ordered 2D map, which makes it easier to detect hidden patterns in the input data [18]. Secondly, SOM preserves the topology of the data, as a result of which close elements in the input space are usually located in close neurons of the map [18]. Thirdly, SOM clustering has the advantage of grouping several prototypes into a cluster, which facilitates the description of complex cluster structures [22]. These functions can help identify hidden patterns in the data, detect similarities or dissimilarities between data groups, and visualize their natural structure, which cannot be observed using other clustering methods.

SOM has proved to be a very powerful and successful method of analyzing data from a wide variety of fields, such as ecology, engineering, biomedicine, etc. In [21], an analysis of the possibilities of clustering nodes of a wireless sensor network (BSS) using a self-organizing Kohonen map is presented. In the course of the study, a SOM model was synthesized with parameters adequate for the clustering of BSS nodes. As a result, it was revealed that SOM is effective for clustering BSS nodes, it informatively describes the distribution of BSS clusters in space.

In [30], an attempt was made to structure raw building materials by radioactivity using self-organizing Kohonen maps. As a result of training the network on data on the specific activity of raw materials, a cluster map with segmentation by the effective specific activity of natural radionuclides was obtained. Based on the results obtained, conclusions are drawn about the possibility and expediency of using the applied algorithm for the classification and analysis of data on the radioactivity of building materials.

SOM is widely used in the field of bioinformatics. For example, SOM was used in combination with the Markov model to characterize local phylogenetic relationships between aligned sequences [28], where it was found that a significant part of the genome ($\sim 3\%$) is associated with innate genomes. SOM was also used to analyze the correlation of genes with samples using gene expression data [9].

Other applications of SOM include the classification of medicinal/non-medicinal products [26] and the classification of bioanalysis of sediment toxicity [7].

SOM is also used for data analysis in sociology, culture and sports. In [23], using SOM as a method of machine learning without a teacher, based on survey data on individuals from 66 countries, individual cultural prototypes were identified around the world, and prototypes dominating in individual countries were also identified. Based on the data obtained using SOM, new measures have been developed to measure cultural heterogeneity within a country, cultural differences between countries, and cultural isolation. The results obtained in [23] have not only shown the usefulness of machine learning algorithms in inductive research of international business but also have managerial significance for international marketing and management. In the article [38], the authors proved the possibility of using Kohonen maps to simulate the training process of athletes.

In addition to these areas, SOM is used for data analysis, investment and financing decision-making. In conditions where decisions are made based on the analysis of stochastic, incomplete information, the use of methods of multidimensional statistical analysis and self-organizing Kohonen maps is not only justified but also necessary. In [4], the authors carried out neural network clustering of enterprises of the agricultural complex, based on the analysis of the groups obtained, conclusions were drawn about the level of their investment attractiveness of these enterprises.

In addition to the great possibilities, using SOM comes with a number of difficulties and limitations that should be taken into account. First, one of the most important factors is the choice of network architecture or, in other words, the number of neurons in its outer layer in which data elements will be projected and grouped. Depending on this decision, the resulting map and its interpretation will vary. There are several measures that can help determine the best clustering of SOM among different sizes and, therefore, the most appropriate number of neurons to use. Some of them are: the average quantization error [18], clustering confidence indices such as the Davies-Bouldin index (DBI) [8], the Dunn index [10] or cluster silhouette [27], as well as topographic error [17], topographic product [3], function Kaski-Lagusa [16] or topographic function [36]. Several models of dynamic self-organization have been proposed in the literature to overcome the static SOM architecture, such as "Growing Neural Gas" [11] (Growing Neural Gas, GNG), "growing Cell Structures" [12] (Growing Cell Structures, GCS) or "Growing hierarchical Bregman SOM" [20] (Growing Hierarchical Bregman SOM, GHBSOM). All these models are united by the fact that learning begins with a small number of units and new neurons are inserted in the learning process until the stopping criterion is met. Although the architecture of these methods is more flexible than that of SOM, it also assumes a larger number of parameters and a greater complexity of their configuration (i.e., the criterion for determining the finite number of neurons). In addition, most of them either do not provide a two-dimensional architecture of the output layer, or, if there is one, it is not obvious how to embed it into a two-dimensional plane. Therefore, they cannot be used to project multidimensional data onto two-dimensional maps.

The second issue to take into account is that SOM has a certain stochastic component. This is caused by the random initialization of prototypes representing neurons on the map. For this reason, it is highly recommended to run multiple SOMs of the same size and choose the best one by some measure or criterion.

Finally, the number of clusters in the SOM does not have to be chosen a priori, and the data elements are distributed over the neuron map. However, correctly identifying groups of neurons corresponding to natural clusters in the input data has always been a difficult task when using this method. Over the past few years, many SOM clustering algorithms based on distance matrices have been proposed. Using the distance between the SOM prototypes, the cluster centroids are identified. Thus, the neurons that will become part of each cluster are selected through an iterative process [35]. Some of these methods offer criteria for merging clusters, for example, using hierarchical clustering [37] or searching for depressions in the distance matrix using gradient analysis [5], although some are very sensitive to certain cluster shapes and some cr0itical SOM parameters, such as the number of the grid node or the topology of the grid. AutoSOME [24] is another distance-based method that applies a density equalization method to scale the SOM output lattice and uses a minimal spanning tree approach based on graph theory to identify data clusters and outliers. In [15], the author proposes a criterion of minimum variance instead of minimum distance. However, when combining two clusters, the algorithm needs to recalculate the new center, so it is adequate only when the clusters have a hyperspherical or hyperellipsoidal shape. Another alternative is the one proposed by Cabanes and Bennani in [6], where the data structure and segmentation are studied simultaneously using distance and density information.

Another strategy that has produced successful results is the clustering of prototypes. The idea is to be able to identify the true cluster of each prototype, making it easier to find and visualize complex data structures. In [34], the authors use the k-mean method and hierarchical clustering to accomplish this task and calculate the DBI to select the best clustering among several partitions with different numbers of clusters. Tasdemir and Mereni [31] follow a similar strategy. However, they propose a new index for selecting the best number of clusters - Conn_Index, which surpasses the results of other classical confidence indices.

In addition to the above remarks, the authors in [1] showed that the SOM learning algorithm is sensitive to the presence of noise and outliers. Due to the influence of outliers in the learning process, some neurons of the ordered map turn out to be far from most of the data, and therefore the network will not effectively represent the topological structure of the data under study. In [2], a Robust SOM (RSOM) learning algorithm was proposed, which is resistant to the presence of outliers in the data and is resistant to these deviations.

The structure of this paper is as follows: Section 2 presents an algorithm for automatic object grouping based on SOM using various distance measures, Section 3 describes the methods used to extract factors for preliminary reduction of the dimensionality of the initial data, and Section 4 presents the results of computational experiments. Section 5 presents the conclusions.

2. SOM Algorithm

The algorithm of functioning of self-learning Kohonen maps [18] (Self Organizing Maps, SOM) is one of the options for clustering multidimensional vectors. An important difference of the SOM algorithm is that in it all neurons (nodes, class centers) are ordered into some structure (usually a two-dimensional grid). At the same time, during training, not only is the winning neuron modified, but, to a lesser extent, its neighbors. Due to this, SOM can be considered one of the methods of projecting a multidimensional space into a space with a lower dimension. When using this algorithm, vectors that are similar in the original space turn out to be next to each other on the resulting map.

SOM implies the use of an ordered structure of neurons. One- and two-dimensional grids are usually used. In this case, each neuron is an n-dimensional vector, where n is determined by the dimension of the original space. The use of one- and two-dimensional grids is due to the fact that problems arise when displaying spatial structures of higher dimensions.

Neurons are usually located in nodes of a two-dimensional grid with rectangular or hexagonal cells. At the same time, as mentioned above, neurons also interact with each other. The magnitude of this interaction is determined by the distance between the neurons on the map. When implementing the SOM algorithm, the grid configuration (rectangular or hexagonal) is preset, as well as the number of neurons in the network.

First, SOM initializes the weights of each neuron. Then, going through the input data for each training example, the winning neuron is determined (the weight vector with the shortest distance (for example, Euclidean distance) from the training example). After the winning neuron is found, the weights of the neural network are adjusted. After training the SOM network, the trained weights are used to cluster new examples. The new example falls into the group of winning vectors.

`SOM algorithm`

`Step 0 Method of selection of informative features`

If flag=1 then Run The methods of extracting factors, else Step 1.

`Step 1 Initialization`

Step 1.1 Set the initial dimension of the neuron grid. $Width$ – width grid, $height$ – height grid, $M - width \cdot height$, total number of neurons.

Step 1.2 Choose the topology of neurons.

Step 1.3 Randomly set the weights $W = \{w_1, w_2, \dots, w_M\}$ each neuron.

Step 1.4 Set the learning rate α_0 and the radius of the neighborhood σ_0. Default $\alpha_0 = 0.7$,

$$\sigma_0 = \frac{\max(width, height)}{2}. \tag{2.1}$$

`Step 2 Learning algorithm`

Step 2.1 Set the number of training epochs, $t := 0$.

Step 2.2 Select a training sample $X_{train} = \{x_1, x_2, \dots, x_N\}$, where N is the amount of data.

Step 2.3 Select a metric for calculating distances between neurons. Available metrics are Euclidean Distance (EuD), Euclidean Distance Square (SEuD), Manhattan Distance (ManD) and Mahalanobis Distance (MahD).

Step 2.4 Calculate by the selected metric $D(dN_M, dN_M)$ – the distances between neurons, where dN are the coordinates of the neurons.

Step 2.5 While t is less than epoch:

Step 2.5.1 Mix the training sample X_{train}.

Step 2.5.2 If there are more elements in the training sample X_{train}, take the input vector sequentially $x_i(t)$, otherwise, go to step 2.5.7.

Step 2.5.3 For the input vector $x_i(t)$ and the weights of the neuron $w_j(t)$ find the winning neuron $winner_c$. To do this, we calculate the distance from the vector to the neuron using formula (2.2):

$$winner_c = argmin(||x_i(t) - w_j(t)||^2). \tag{2.2}$$

Step 2.5.4 Let's determine the neighbors for the winning neuron $winner_c$ and the neighborhood function $h_i(t)$. The neighborhood function $h_i(t)$ is calculated using the formula (2.3):

$$h_i(t) = \alpha(t) \cdot e^{-(\frac{D(winner_c, dN_M)}{2\sigma(t)^2})}, \tag{2.3}$$

where $D(winner_c, dN_M)$ are the distance between the winning neuron and the neurons.

Defining neighbors: if $D(winner_c, dN_M)_i < \sigma(t)$, then the neuron is the neighbor of the winning neuron.

Step 2.5.5 Recalculate the weights for all neurons using formula (2.4):

$$w_j(t+1) = w_j(t) + \alpha(t) \cdot h_i(t) \cdot (x_i(t) - w_j(t)). \tag{2.4}$$

Step 2.5.6 Go to step 2.5.1.

Step 2.5.7 Calculate $\alpha(t) = \alpha_0 \cdot e^{-(\frac{t}{epoch})}$.

Step 2.5.8 Calculate $\sigma(t) = \sigma_0 \cdot e^{-(\frac{t}{epoch})}$.

Step 2.5.9 Shuffle the training sample X_{train}.

Step 2.5.10 $t := t + 1$.

`Step 3 The algorithm of SOM operation`

Step 3.1 While there are still items in the test sample $X_{test} = \{x_1, x_2, \ldots, x_N\}$, where N is the amount of data, select sequentially x_i, otherwise, terminate the algorithm.

Step 3.2 For the input vector x_i and the weights of the neuron w_j find the winning neuron $winner_c$. To do this, we calculate the distance from the vector to the neuron using formula (2.2).

Step 3.3 Refer x_i to the cluster corresponding to the neuron $winner_c$.

Step 3.4 Go to step 3.1.

3. Methods of Extracting Factors

When solving the tasks of the automatic grouping of objects, one has to deal with the data of a fairly large dimension-up to several thousand signs. Moreover, some

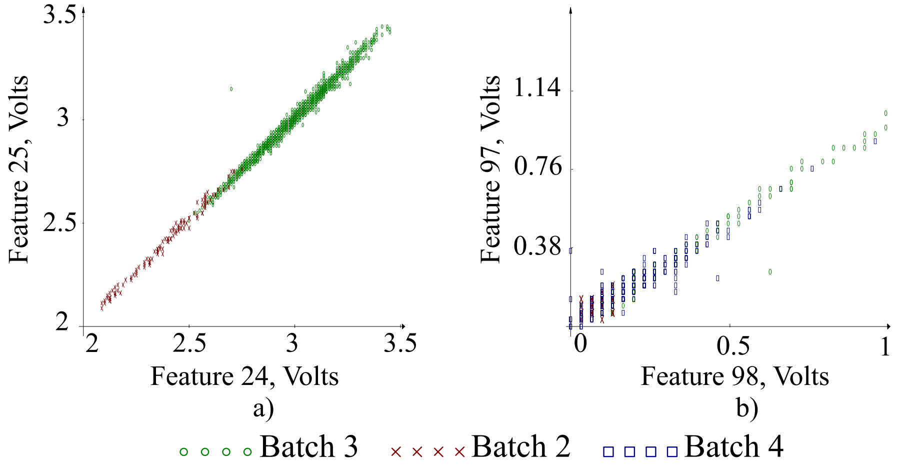


Fig. 3.1: An example of correlation between some characteristics of industrial products (Microchips 1526IE10_002) [29]

of these features are uninformative, and their inclusion in the automatic grouping model only worsens the accuracy of solving the problem. There are often obvious correlation dependencies between some of the characteristics (Figure 3.1).

The methods of extracting of factors can reduce the dimension of data by using these dependencies. Our approach based on SOM in combination with the selection of informative features will improve the accuracy of the solution for the problem of automatic object grouping. In our work to reduce the dimension of the data, we used Factor Analysis (Principal Component Analysis [25], Maximum Likelihood Estimation [33, 13]) and Principal Components Analysis based on Singular Value Decomposition [19].

Factor Analysis (Principal Component Analysis). The method of factor analysis, which is used to highlight the most important factors from a large number of variables. It is based on the search for linear combinations of variables that explain the largest share of data variability. These linear combinations are called the main components.

Factor Analysis (Maximum Likelihood Estimation). The method is aimed at evaluating the parameters of the statistical model based on the available data. It allows you to determine the value of the parameters that most likely corresponds to the observed data.

Principal Components Analysis based on Singular Value Decomposition. Singular value decomposition is a way of representing any matrix as the product of three other matrices: a left singular matrix U, a diagonal singular value matrix S, and a right singular matrix V, where the singular values are the square roots of the eigenvalues of the covariance matrix of the data (which is what this is for in this case pre-centering of the data is performed), the right singular matrix V will correspond to the eigenvectors of the covariance matrix of the data, and the left U will be the projection of the original data onto the principal components defined by the matrix V. Thus, the singular value decomposition also allows us to isolate the principal components, but without the need for calculation covariance matrix. In addition to being more efficient, this solution is considered more numerically stable because it does not require directly calculating the covariance matrix, which can be poorly conditioned in the case of strong feature correlation.

4. Computational Experiments

For the experiments, we considered the sample of industrial products (Microchips 1526IE10_002) [29] and Synthetic (artificial) datasets (with cluster labels) [14].

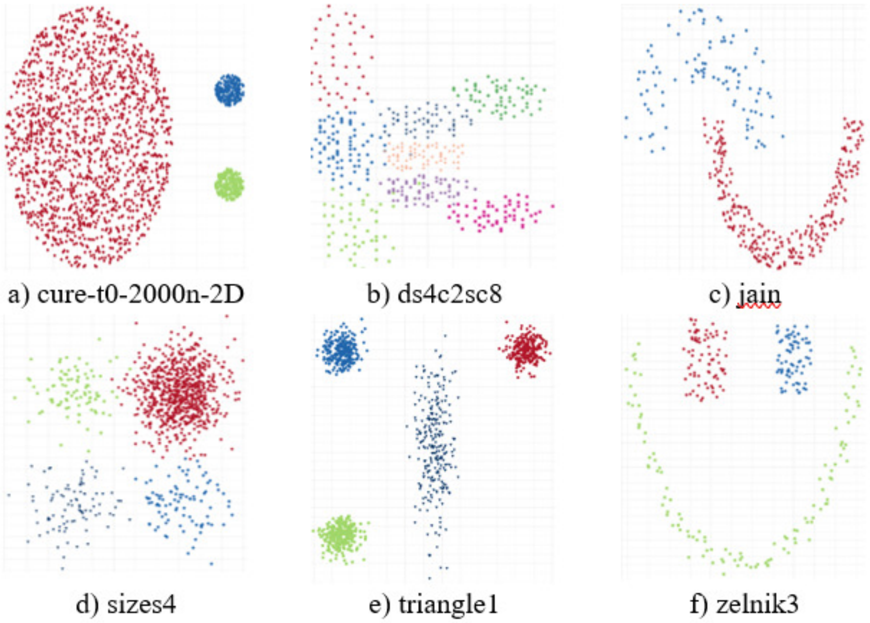
Microchips are set of results of test effects on electrical and radio products (ERI) for monitoring the current-voltage characteristics of input and output circuits of microcircuits (3987 data points, 67 dimensions, 7 clusters). Various combinations of batches were considered: full mixed lot (3987 data points, 67 dimensions, 7 clusters), four-batch mixed lot (446 data points, 62 dimensions, 4 clusters), three-batch mixed lot (300 data points, 41 dimensions, 3 clusters) and two-batch mixed lot (187 data points, 41 dimensions, 2 clusters). The complexity of the sample consists in the fact that the number of parameters in it is quite large in relation to the number of sample elements.

Various synthetic (artificial) datasets were considered: cure-t0-2000n-2D (2000 data points, 2 dimensions, 3 clusters) (Figure 4.1a), ds4c2sc8 (486 data points, 2 dimensions, 8 clusters) (Figure 4.1b), jain (373 data points, 2 dimensions, 2 clusters) (Figure 4.1c), sizes4 (1000 data points, 2 dimensions, 4 clusters) (Figure 4.1d), triangle1 (1000 data points, 2 dimensions, 4 clusters) (Figure 4.1e), zelnik3 (256 data points, 2 dimensions, 3 clusters) (Figure 4.1f).

We used various types of distance measures in the experiments: Mahalanobis distance (MahD), Euclidean distance (Eud), Square Euclidean distance (SEuD), and Manhattan distance (ManD). The choice of the method for initializing the weight coefficients was also different: random and with a choice of weight coefficients from the dataset (WCD). Also, various methods of reducing the dimension of data were applied to various combinations of batches. Each experiment was run 30 times.

The algorithm was implemented in Python. The following test system was used for computational experiments: AMD Radeon 9-7900X 12 C/24 T 4700MHz CPU, 32 GB RAM. Each experiment took an average of 10 minutes of machine time.

The initial initialization of the map was selected for each experiment: the grid configuration is hexagonal, the number of neurons in the grid for two batches is 81 neurons (9x9), for three batches – 100 neurons (10x10), for four batches – 121 neurons (11x11). The size of the map was determined by calculating the number of neurons based on the number of observations in the training data $5N^{1/2}$ [32].



a) cure-t0-2000n-2D

b) ds4c2sc8

c) jain

d) sizes4

e) triangle1

f) zelnik3

FIG. 4.1: Synthetic datasets

4.1. SOM and the choice of the method for initializing the weight coefficients

In this section, we compare the results of the experiment performed with k-means and SOM algorithm with the choice of the method for initializing the weight coefficients. Experiments with initial random initialization of weight coefficients is marked as SOM (random). Experiments with a choice of weight coefficients from the dataset (WCD) is marked as SOM (WCD).

Microchips 1526IE10_002

Computational experiments showed that the use of methods for initializing the weight coefficients in the SOM algorithm, in most cases, improves the accuracy of batch separation. Moreover, clustering accuracy decreases with an increasing number of homogeneous batches in a mixed lot (Table 4.1).

For the Mahalanobis distance, the best clustering accuracy was achieved with SOM (random) and SOM (WSD) algorithms for two batches in a mixed lot and a full mixed lot. For a three-batch mixed lot, the clustering accuracy shows the best result with SOM (random). For a four-batch mixed lot, the clustering accuracy shows the best result with SOM (WSD).

For Euclidean distance (Eud), the best clustering accuracy was achieved for all algorithms for two batches in a mixed lot. The clustering accuracy has maximal values with the SOM (WCD) algorithm for three-batch mixed lot and four-batch

Table 4.1: Accuracy of ERI clustering for k-means, SOM(random) and SOM(WCD)

Algorithm	MahD	EuD	SEuD	ManD
Two-batch mixed lot				
k-means	0.677	**1.000**	**1.000**	**1.000**
SOM (random)	**1.000**	**1.000**	**1.000**	**1.000**
SOM (WCD)	**1.000**	**1.000**	**1.000**	**1.000**
Three-batch mixed lot				
k-means	0.416	0.982	0.982	0.985
SOM (random)	**0.989**	0.978	**0.989**	**0.989**
SOM (WCD)	0.985	**0.988**	0.985	**0.997**
Four-batch mixed lot				
k-means	0.461	0.748	0.748	0.741
SOM (random)	0.978	0.978	0.970	0.978
SOM (WCD)	**0.986**	**0.994**	**0.990**	**0.994**
Full mixed lot				
k-means	0.358	**0.418**	0.180	**0.527**
SOM (random)	**0.476**	0.369	**0.452**	0.514
SOM (WCD)	**0.476**	0.369	**0.452**	0.514

mixed lot. For a full mixed lot, the clustering accuracy has maximal values with the k-means algorithm.

For Square Euclidean distance (SEuD), the best clustering accuracy was achieved for all algorithms for two batches in a mixed lot. The clustering accuracy has maximal values with the SOM (random) algorithm for a three-batch mixed lot. For a four-batch mixed lot, the clustering accuracy has maximal values with the SOM (SWD) algorithm. For full mixed lot, the clustering accuracy has maximal values with SOM (random) and SOM (SWD) algorithms.

For Manhattan distance (ManD), the best clustering accuracy was achieved for all algorithms for two batches in a mixed lot. The clustering accuracy has maximal values with the SOM (WCD) algorithm for three-batch mixed lot and four-batch mixed lot. For a full mixed lot, the clustering accuracy has maximal values with the k-means algorithm.

Also, for various combinations of batches, the minimum (Min), maximum (Max), mean (Mean), standard deviation (σ), coefficient of variation (V), and span (R) of the objective function are calculated (Table 4.2, Table 4.3).

For all distances, the coefficient of variation and span factor have minimal values with SOM (random) algorithm (Figure 4.2 - Figure 4.4) for two, three and four batches in a mixed lot. In the case of the full mixed lot, the coefficient of variation and span factor have same values.

Synthetic datasets

A layer of neurons of SOM can be represented, in the form of a flexible grid, which is stretched over the space of the input vectors. Figure 4.5 show how neurons were distributed to the space of artificial data sets.

Computational experiment showed that the use methods for initializing the weight coefficients in SOM algorithm, in most cases, increases the accuracy in synthetic datasets (Table 4.4).

Table 4.2: The total value of the target function after 30 launch attempts for SOM(random) and SOM(WCD) algorithms. Microchips 1526IE10_002. MahD, EuD

Parameter	MahD random	MahD WCD	EuD random	EuD WCD
Two-batch mixed lot				
Min	47.06	140.17	38.70	44.94
Max	52.14	154.64	40.71	50.37
Mean	49.60	146.80	39.78	47.34
σ	1.31	3.96	0.47	1.31
V	2.65	2.70	1.19	2.76
R	5.08	14.47	2.01	5.43
Three-batch mixed lot				
Min	53.18	193.85	32.49	99.22
Max	66.70	245.39	34.37	105.47
Mean	58.85	212.90	33.51	102.31
σ	2.86	12.37	0.46	1.74
V	4.86	5.81	1.37	1.70
R	13.52	51.54	1.88	6.25
Four-batch mixed lot				
Min	177.26	669.56	123.81	402.22
Max	214.62	837.89	130.31	423.17
Mean	193.87	748.56	126.86	412.95
σ	9.50	45.11	1.41	4.49
V	4.90	6.03	1.11	1.09
R	37.36	168.33	6.50	20.94
Full mixed lot				
Min	4786.71	4786.71	1420.00	1420.00
Max	7676.67	7676.67	1525.33	1525.33
Mean	6217.95	6217.95	1457.12	1457.12
σ	771.09	771.09	23.37	23.37
V	12.40	12.40	1.60	1.60
R	2889.96	2889.96	105.33	105.3

For Mahalanobis distance (MahD), the best clustering accuracy was achieved with SOM (WSD) algorithm for the synthetic dataset “cure-t0-2000n-2D”. For synthetic datasets “ds4c2sc8”, “triangle1” and “sizes4”, the clustering accuracy shows the best result with k-means. For synthetic datasets “jain” and “zelnik3”, the clustering accuracy shows best result with SOM (random).

For Euclidean distance (Eud), the best clustering accuracy was achieved with SOM (WSD) algorithm for the synthetic dataset “cure-t0-2000n-2D”. For synthetic datasets “ds4c2sc8”, “sizes4” and “zelnik3”, the clustering accuracy shows the best result with SOM (random) and SOM (SWD) algorithms. For synthetic dataset “jain”, the clustering accuracy shows the best result with SOM (random). For the synthetic dataset “triangle1”, the clustering accuracy shows the best result with SOM (WCD).

For Square Euclidean distance (SEuD), the best clustering accuracy was achieved with the SOM (random) algorithm for the synthetic dataset “cure-t0-2000n-2D”. For synthetic datasets “ds4c2sc8”, “jain” and “zelnik3”, the clustering accuracy shows the best result with SOM (random) and SOM (WCD) algorithms. For the synthetic dataset “triangle1”, the clustering accuracy shows the best result with k-means and SOM (random) algorithms. For synthetic dataset “sizes4”, the clustering

Table 4.3: The total value of the target function after 30 launch attempts for SOM(random) and SOM(WCD) algorithms. Microchips 1526IE10_002. SEuD, ManD

Parameter	SEuD random	SEuD WCD	ManD random	ManD WCD
Two-batch mixed lot				
Min	38.70	40.39	39.43	42.41
Max	40.71	48.59	42.03	47.73
Mean	39.78	43.07	40.49	45.24
σ	0.47	1.73	0.60	1.30
V	1.19	4.01	1.47	2.87
R	2.01	8.20	2.60	5.32
Three-batch mixed lot				
Min	33.37	49.02	32.11	74.92
Max	35.35	54.73	33.77	78.72
Mean	34.23	52.18	33.04	76.55
σ	0.42	1.66	0.41	0.97
V	1.24	3.18	1.24	1.27
R	1.98	5.71	1.66	3.81
Four-batch mixed lot				
Min	124.43	194.41	122.05	307.15
Max	135.37	226.54	129.95	328.36
Mean	129.44	213.92	126.20	314.94
σ	2.22	6.78	1.79	4.63
V	1.71	3.17	1.42	1.47
R	10.94	32.13	7.90	21.21
Full mixed lot				
Min	1221.51	1221.51	1316.54	1316.54
Max	1286.37	1286.37	1389.70	1389.70
Mean	1253.82	1253.82	1345.18	1345.18
σ	13.76	13.76	15.97	15.97
V	1.10	1.10	1.19	1.19
R	64.86	64.86	73.16	73.16

accuracy shows the best result with the SOM (random) algorithm.

For Manhattan distance (ManD), the best clustering accuracy was achieved with k- the means algorithm for synthetic datasets “cure-t0-2000n-2D”, “ds4c2sc8”, “jain”. For the synthetic dataset “sizes4”, the clustering accuracy shows the best result with the SOM (WCD) algorithm. For the synthetic dataset “zelnik3”, the clustering accuracy shows the best result with SOM (random). For the synthetic dataset “triangle1”, the clustering accuracy shows best result with SOM (random) and SOM (WCD) algorithms.

Also, for various combinations of batches, the minimum (Min), maximum (Max), mean (Mean), standard deviation (σ), coefficient of variation (V) and span (R) of the objective function is calculated (Table 4.5, Figure 4.6 - Figure 4.7).

For Mahalanobis distance (MAhD), SOM (WCD) algorithm gives minimal values of the coefficient of variation and span factor for synthetic datasets “cure-t0-2000n-2D”, “sizes4”, “triangle1”. SOM (random) algorithm gives minimal values of the coefficient of variation and span factor for synthetic datasets “ds4c2sc8”, “jain”, “zelnik3”.

For Euclidean distance (Eud), SOM (WCD) algorithm gives minimal values of the coefficient of variation and span factor for synthetic dataset “cure-t0-2000n-2D”.

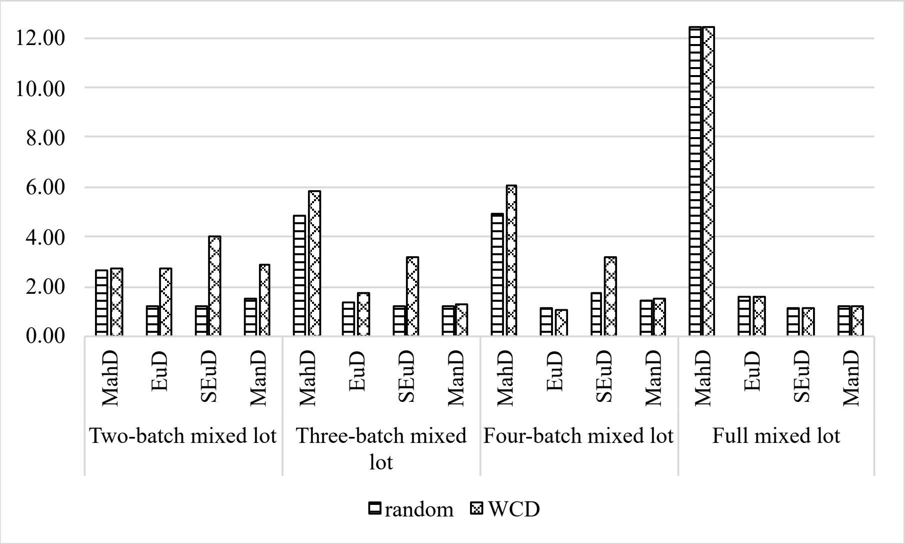


FIG. 4.2: Coefficient of variation (V) of the objective function value for two-batch mixed lot, three-batch mixed lot, four-batch mixed lot, full mixed lot

Table 4.4: Accuracy of Synthetic datasets clustering for k-means, SOM(random) and SOM(WCD)

Algorithm	MahD	EuD	SEuD	ManD
cure-t0-2000n-2D (2000 data point)				
k-means	0.489	0.839	0.839	**0.850**
SOM (random)	0.753	0.832	**0.848**	0.835
SOM (WCD)	**0.840**	**0.855**	0.829	0.836
ds4c2sc8 (486 data point)				
k-means	**0.945**	0.837	0.837	**0.952**
SOM (random)	0.741	**0.874**	**0.849**	0.849
SOM (WCD)	0.741	**0.874**	**0.849**	0.849
jain (373 data point)				
k-means	0.917	0.917	0.917	**0.912**
SOM (random)	**0.983**	**0.935**	**0.935**	0.903
SOM (WCD)	0.928	0.930	**0.935**	0.900
sizes4 (1000 data point)				
k-means	**0.986**	0.985	0.985	0.988
SOM (random)	0.903	**0.989**	**0.991**	0.985
SOM (WCD)	0.900	**0.989**	0.988	**0.986**
triangle1 (1000 data point)				
k-means	**0.990**	0.992	**0.992**	0.995
SOM (random)	0.964	0.998	**0.992**	**0.999**
SOM (WCD)	0.980	**1.000**	0.991	**0.999**
zelnik3 (256 data point)				
k-means	0.845	0.853	0.853	0.853
SOM (random)	**0.918**	**0.864**	**0.862**	**0.867**
SOM (WCD)	0.890	**0.864**	**0.862**	0.862

For synthetic dataset “ds4c2sc8”, SOM (random) and SOM (WCD) algorithms give same values of coefficient of variation. SOM (random) algorithm gives minimal val-

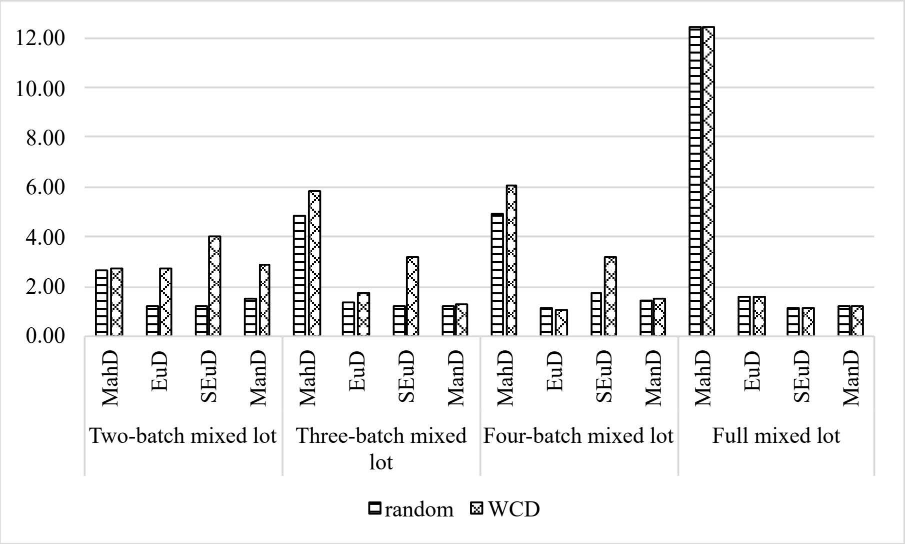


Fig. 4.3: The span coefficient (R) of the objective function value for two-batch mixed lot, three-batch mixed lot, four-batch mixed lot

ues of the coefficient of variation and span factor for synthetic datasets “ds4c2sc8”, “jain”, “size”, “triangle1”. For synthetic dataset “zelnik3”, SOM (random) algorithm gives minimal values of the coefficient of variation, while SOM (WCD) algorithm gives minimal values of the span factor.

For Square Euclidean distance (SEuD), SOM (WCD) algorithm gives minimal values of the coefficient of variation for synthetic datasets “cure-t0-2000n-2D”, “ds4c2sc8” “sizes4”, while SOM (random) algorithm gives minimal values of the span factor. For synthetic dataset “jain”, SOM (random) algorithm gives minimal values of the coefficient of variation, but SOM (random) and SOM (WCD) algorithms give the same values of span factor. For synthetic datasets “triangle1”, “zelnik”, SOM (random) algorithm gives minimal values of the coefficient of variation and the SOM (WCD) algorithm gives minimal values of the span factor.

For Manhattan distance (ManD), SOM (random) algorithm gives minimal values of the coefficient of variation and span factor for synthetic datasets “cure-t0-2000n-2D”, “jain”, “sizes4”, “triangle1”. Also, SOM (random) and SOM (WCD) algorithms give the same values of span factor for the synthetic dataset “jain”. For synthetic dataset “ds4c2sc8”, SOM (WCD) algorithm gives minimal values of the coefficient of variation, while SOM (random) and SOM (WCD) algorithms give the same values of span factor. For synthetic dataset “zelnik3”, SOM (random) algorithm gives minimal values of the coefficient of variation and SOM (WCD) algorithm gives minimal values of the span factor.

Computational experiments showed that the coefficient of variation for any type of mixed lot composition was higher (worse) with SOM (WCD) initialization. For

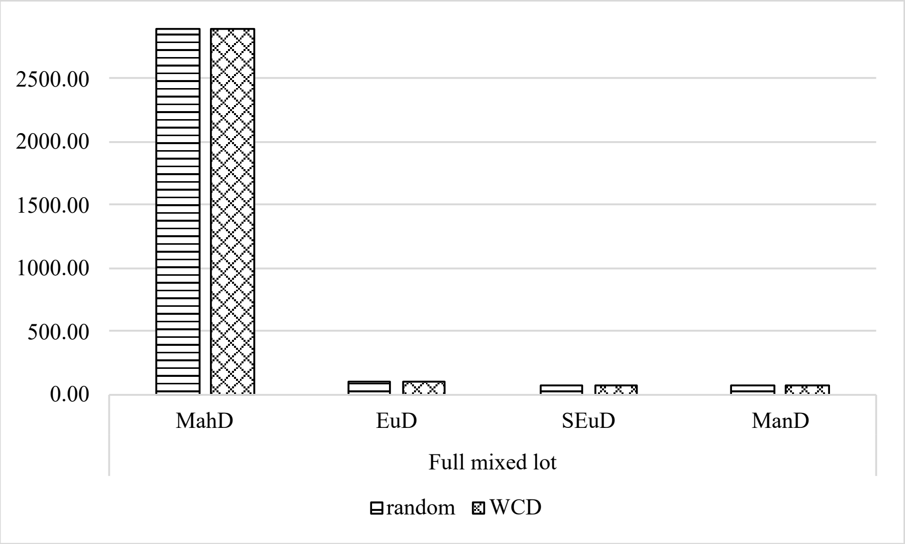


Fig. 4.4: The span coefficient (R) of the objective function value for full mixed lot

the next computational experiments, we used SOM with the initial random initialization of weight coefficients.

4.2. SOM and methods of extracting factors

In this section, we compare the results of the experiment performed with k-means and SOM algorithm with various methods of extracting factors. Various combinations of batches were subjected to factor analysis (Principal Component Analysis, Maximum Likelihood Estimation) and Principal components analysis based on Singular Value Decomposition. Experiments with the factor analysis (Principal component analysis) is marked as FA_PA+SOM. Experiments with the factor analysis (Maximum Likelihood Estimation) is marked as FA_ML+SOM. Experiments with the principal components analysis based on Singular Value Decomposition is marked as PCA+SOM.

In this work, the number of factors was determined by the Kaiser criterion, and the total proportion of variance reproduced by these factors should be at least 70% (Figure 4.8).

Computational experiment showed that the use methods of extracting factors in SOM algorithm, in most cases, improves the accuracy of batch separation. Moreover, clustering accuracy decreases with increasing number of homogeneous batches in a mixed lot (Table 4.6).

For two-batch mixed lot (Figure 4.9), the best clustering accuracy was achieved for all algorithms, except k-means with Mahalanobis distance (MahD). k-means algorithm showed worse result for Mahalanobis distance (MahD).

Table 4.5: The total value of the target function after 30 launch attempts for SOM(random) and SOM(WCD) algorithms. Synthetic datasets

Parameter	MahD	MahD	EuD	EuD	SEuD	SEuD	ManD	ManD
	random	WCD	random	WCD	random	WCD	random	WCD
cure-t0-2000n-2D								
Min	35.37	29.78	1.23	1.35	0.62	0.63	1.23	1.24
Max	92.46	81.49	1.60	1.61	0.68	0.69	1.40	1.52
Mean	57.25	56.30	1.32	1.43	0.65	0.65	1.31	1.41
σ	13.41	12.72	0.07	0.06	0.02	0.02	0.04	0.07
V	23.42	**22.60**	5.04	**4.06**	2.59	**2.39**	**3.34**	4.71
R	57.08	**51.71**	0.37	**0.27**	**0.06**	0.07	**0.17**	0.28
ds4c2sc8								
Min	6.14	6.19	6.14	6.19	6.14	6.19	0.54	0.60
Max	15.42	16.50	15.42	16.50	15.42	16.50	0.64	0.70
Mean	10.15	10.57	10.15	10.57	10.15	10.57	0.59	0.64
σ	2.09	2.46	0.03	0.03	0.01	0.01	0.03	0.02
V	**20.56**	23.24	**0.26**	**0.26**	0.10	**0.08**	4.73	**3.73**
R	**9.28**	10.31	**9.28**	10.31	**9.28**	10.31	**0.10**	**0.10**
jain								
Min	3.18	3.11	0.35	0.37	0.13	0.13	0.23	0.24
Max	6.44	7.62	0.45	0.51	0.16	0.17	0.29	0.31
Mean	4.82	5.23	0.40	0.44	0.15	0.14	0.27	0.27
σ	0.91	1.21	0.02	0.03	0.01	0.01	0.01	0.01
V	**18.84**	23.21	**5.64**	7.81	**6.70**	6.79	**4.82**	5.36
R	**3.25**	4.50	**0.09**	0.14	**0.04**	**0.04**	**0.06**	**0.06**
sizes4								
Min	11.18	13.03	0.92	0.98	0.34	0.32	0.89	0.93
Max	38.16	33.18	1.12	1.25	0.42	0.42	1.11	1.23
Mean	22.87	22.91	1.00	1.09	0.37	0.35	0.98	1.05
σ	6.85	5.29	0.04	0.06	0.02	0.02	0.04	0.06
V	29.93	**23.08**	**4.18**	5.35	6.15	**5.13**	**4.54**	5.92
R	26.98	**20.15**	**0.19**	0.26	**0.08**	0.10	**0.22**	0.30
triangle1								
Min	4.23	4.04	0.38	0.41	0.14	0.14	0.41	0.42
Max	44.54	28.18	0.50	0.61	0.17	0.16	0.49	0.58
Mean	13.32	11.89	0.45	0.47	0.15	0.15	0.45	0.47
σ	7.84	6.78	0.03	0.04	0.01	0.01	0.02	0.03
V	58.88	**57.05**	**6.29**	8.42	**3.44**	3.48	**3.83**	7.41
R	40.31	**24.15**	**0.12**	0.20	0.03	**0.02**	**0.08**	0.16
zelnik3								
Min	53.18	4.04	32.49	0.41	33.37	0.14	32.11	0.42
Max	66.70	28.18	34.37	0.61	35.35	0.16	33.77	0.58
Mean	58.85	11.89	33.51	0.47	34.23	0.15	33.04	0.47
σ	2.86	6.78	0.46	0.04	0.42	0.01	0.41	0.03
V	**4.86**	57.05	**1.37**	8.42	**1.24**	3.48	**1.24**	7.41
R	**13.52**	24.15	1.88	**0.20**	1.98	**0.02**	1.66	**0.16**

For three-batch mixed lot (Figure 4.10), the best clustering accuracy was achieved for SOM algorithm for Mahalanobis distance (MahD), Square Euclidean distance (SEuD), Manhattan distance (ManD) and k-means algorithm for Euclidean distance (Eud).

For four-batch mixed lot (Figure 4.11), the best clustering accuracy was achieved for FA_ML+SOM algorithm for all distances.

For full mixed lot (Figure 4.12), the best clustering accuracy was achieved for SOM algorithm for Mahalanobis distance (MahD) and Square Euclidean distance (SEuD) and FA_ML+SOM algorithm for Euclidean distance (Eud) and Manhattan distance (ManD).

Table 4.6: Accuracy of ERI clustering in homogeneous batches

Algorithm	MahD	EuD	SEuD	ManD
Two-batch mixed lot				
k-means	0.677	**1**	**1**	**1**
SOM	**1**	**1**	**1**	**1**
PCA+SOM	**1**	**1**	**1**	**1**
FA_PA+SOM	**1**	**1**	**1**	**1**
FA_ML+SOM	**1**	**1**	**1**	**1**
Three-batch mixed lot				
k-means	0.416	**0.982**	0.982	0.985
SOM	**0.989**	0.978	**0.989**	**0.989**
PCA+SOM	0.985	0.979	0.973	0.981
FA_PA+SOM	0.959	0.971	0.968	0.968
FA_ML+SOM	0.948	0.905	0.908	0.902
Four-batch mixed lot				
k-means	0.461	0.748	0.748	0.741
SOM	0.978	0.978	0.97	0.978
PCA+SOM	0.976	0.979	0.977	0.979
FA_PA+SOM	**0.985**	0.983	0.985	0.99
FA_ML+SOM	**0.985**	**0.998**	**0.998**	**1**
Full mixed lot				
k-means	0.358	0.418	0.18	0.527
SOM	**0.476**	0.369	**0.452**	0.514
PCA+SOM	0.34900000	0.35100000	0.41300000	0.36700000
FA_PA+SOM	0.351	0.421	0.422	0.424
FA_ML+SOM	0.42	**0.516**	0.423	**0.657**

Also, for various combinations of batches, the minimum (Min), maximum (Max), mean (Mean), standard deviation (σ), coefficient of variation (V) and span (R) of the objective function are calculated (Table 4.7 - Table 4.10).

Table 4.7: The total value of the target function after 30 launch attempts. Two-batch mixed lot

Parameter	MahD	EuD	SEuD	ManD
k-means				
Min	1176.07	186.08	198.28	864.67
Max	1177.60	186.08	198.28	864.67
Mean	1176.8	2 186.08	198.28	864.67
σ	0.39	0.00	0.00	0.00
V	0.03	0.00	0.00	0.00
R	1.52	0.00	0.00	0.00
SOM, PCA+SOM, FA_PA+SOM, FA_ML+SOM				
Min	47.06	38.70	39.43	39.06
Max	52.14	40.71	42.03	41.74
Mean	49.60	39.78	40.49	40.25
σ	1.31	0.47	0.60	0.73
V	2.65	1.19	1.47	1.81
R	5.08	2.01	2.60	2.68

For two-batch mixed lot, the coefficient of variation and span factor have minimal values with k-means algorithm (Figure 4.13 - Figure 4.14) for all distances.

For three-batch mixed lot, the coefficient of variation has minimal values with k-means algorithm (Figure 4.15) for Mahalanobis distance (MahD) and Manhattan distance (ManD). In this case, the span factor has minimal values with FA_ML+SOM

Table 4.8: The total value of the target function after 30 launch attempts. Three-batch mixed lot

Parameter	MahD	EuD	SEuD	ManD
k-means				
Min	1872.21	241.88	227.47	1165.06
Max	1876.46	309.03	327.71	1165.06
Mean	1874.90	244.12	230.81	1165.06
σ	0.84	12.05	17.99	0.00
V	0.04	4.94	7.80	0.00
R	4.25	67.16	100.25	0.00
SOM				
Min	53.18	32.49	33.37	32.11
Max	66.70	34.37	35.35	33.77
Mean	58.85	33.51	34.23	33.04
σ	2.86	0.46	0.42	0.41
V	4.86	1.37	1.24	1.24
R	13.52	1.88	1.98	1.66
PCA+SOM				
Min	4.03	0.68	0.20	0.39
Max	9.05	0.93	0.37	0.53
Mean	6.68	0.78	0.24	0.45
σ	1.34	0.05	0.03	0.03
V	20.07	6.53	11.89	6.78
R	5.02	0.24	0.18	0.14
FA_PA+SOM				
Min	4.92	0.80	0.22	0.46
Max	10.40	1.37	0.35	0.56
Mean	7.10	0.92	0.27	0.50
σ	1.17	0.14	0.05	0.03
V	16.43	14.80	18.18	6.54
R	5.48	0.57	0.13	0.10
FA_ML+SOM				
Min	5.86	1.01	0.30	0.60
Max	8.78	1.33	0.45	0.80
Mean	7.47	1.15	0.37	0.68
σ	0.85	0.06	0.05	0.05
V	11.37	5.36	14.68	6.97
R	2.92	0.32	0.15	0.20

algorithm for Mahalanobis distance (MahD), PCA+SOM algorithm for Euclidean distance (Eud), FA_PA+SOM algorithm for Square Euclidean distance (SEuD) and Manhattan distance (ManD) (Figure 4.16).

For four-batch mixed lot, the coefficient of variation has minimal values with k-means algorithm (Figure 4.17) for Mahalanobis distance (MahD) and SOM algorithm for Euclidean distance (Eud), Square Euclidean distance (SEuD) and Manhattan distance (ManD). In this case, the span factor has minimal values with PCA+SOM algorithm for Mahalanobis distance (MahD), PCA+SOM, FA_PA+SOM algorithms for Euclidean distance (Eud), PCA+SOM, FA_PA+SOM, FA_ML+SOM algorithms for Square Euclidean distance (SEuD) and PCA+SOM, FA_ML+SOM algorithms for Manhattan distance (ManD) (Figure 4.18).

For full mixed lot, the coefficient of variation has minimal values with k-means algorithm (Figure 4.19) for Mahalanobis distance (MahD) and SOM algorithm for Euclidean distance (Eud), Square Euclidean distance (SEuD) and Manhattan distance (ManD). In this case, the span factor has minimal values with FA_PA+SOM algo-

Table 4.9: The total value of the target function after 30 launch attempts. Four-batch mixed lot

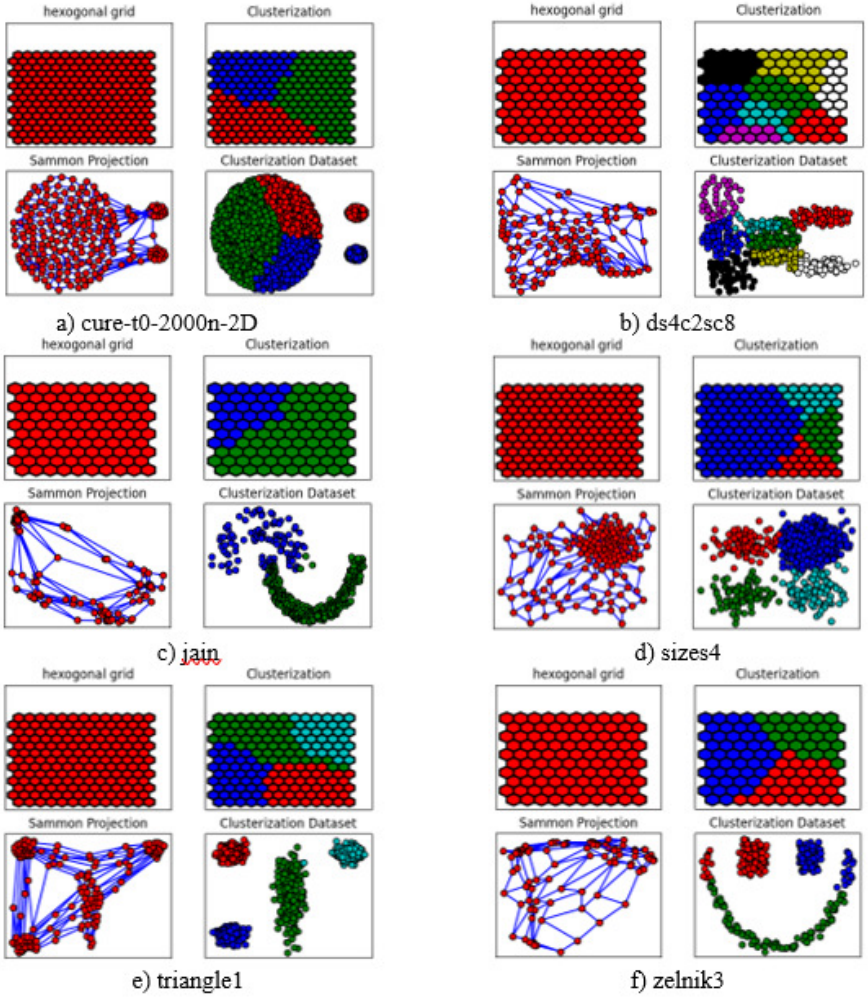
Parameter	MahD	EuD	SEuD	ManD
k-means				
Min	3355.59	554.55	791.65	2865.33
Max	3361.44	673.27	1170.43	3653.22
Mean	3359.36	569.59	841.71	2938.36
σ	1.27	35.30	112.97	219.55
V	0.04	6.20	13.42	7.47
R	5.85	118.72	378.79	787.90
SOM				
Min	177.26	123.81	124.43	122.05
Max	214.62	130.31	135.37	129.95
Mean	193.87	126.86	129.44	126.20
σ	9.50	1.41	2.22	1.79
V	4.90	1.11	1.71	1.42
R	37.36	6.50	10.94	7.90
PCA+SOM				
Min	7.07	1.07	0.26	0.53
Max	12.55	1.35	0.37	0.75
Mean	8.76	1.19	0.30	0.65
σ	1.35	0.06	0.04	0.06
V	15.47	5.35	12.37	8.85
R	5.49	0.28	0.11	0.22
FA_PA+SOM				
Min	6.37	1.09	0.27	0.55
Max	12.65	1.36	0.38	0.85
Mean	9.18	1.21	0.34	0.67
σ	1.63	0.07	0.04	0.06
V	17.74	5.81	10.97	8.34
R	6.28	0.27	0.11	0.31
FA_ML+SOM				
Min	8.45	1.30	0.38	0.76
Max	22.59	1.80	0.49	1.00
Mean	11.53	1.50	0.42	0.86
σ	3.13	0.12	0.03	0.06
V	27.18	7.74	7.59	6.60
R	14.14	0.50	0.11	0.23

rithm for Mahalanobis distance (MahD) and Euclidean distance (Eud), FA_ML+SOM algorithm for Square Euclidean distance (SEuD) and for Manhattan distance (ManD) (Figure 4.20).

Figures 4.21 - 4.23 show how neurons were distributed to the space of ERI and a visual representation of the algorithms for three-batch mixed lot, four-batch mixed lot and full mixed lot.

Table 4.10: The total value of the target function after 30 launch attempts. Full mixed lot

Parameter	MahD	EuD	SEuD	ManD
k-means				
Min	28605.72	2786.10	2714.11	14669.07
Max	28851.14	3146.37	3275.19	21356.05
Mean	28710.57	2851.43	2873.83	16243.14
σ	63.83	90.53	168.74	1316.66
V	0.22	3.17	5.87	8.11
R	245.42	360.27	561.08	6686.98
SOM				
Min	4786.7	1 1420.00	1221.51	1316.54
Max	7676.67	1525.33	1286.37	1389.70
Mean	6217.95	1457.12	1253.82	1345.18
σ	771.09	23.37	13.76	15.97
V	12.40	1.60	1.10	1.19
R	2889.96	105.33	64.86	73.16
PCA+SOM				
Min	332.41	47.95	37.22	41.12
Max	485.19	56.82	42.25	46.89
Mean	376.89	52.13	39.03	44.08
σ	29.39	1.85	1.07	1.38
V	7.80	3.55	2.74	3.12
R	152.78	8.87	5.03	5.77
FA_PA+SOM				
Min	336.03	45.11	34.75	38.91
Max	406.30	52.84	39.04	45.15
Mean	372.28	48.78	36.29	40.90
σ	19.26	2.28	1.04	1.32
V	5.17	4.67	2.86	3.23
R	70.27	7.72	4.29	6.24
FA_ML+SOM				
Min	392.72	53.64	42.39	46.82
Max	548.58	63.11	46.18	52.51
Mean	483.81	58.41	44.27	50.16
σ	35.30	1.99	0.86	1.37
V	7.30	3.41	1.94	2.73
R	155.86	9.48	3.79	5.68



a) cure-t0-2000n-2D

b) ds4c2sc8

c) jain

d) sizes4

e) triangle1

f) zelnik3

Fig. 4.5: Distribution of neurons over the space of artificial data sets

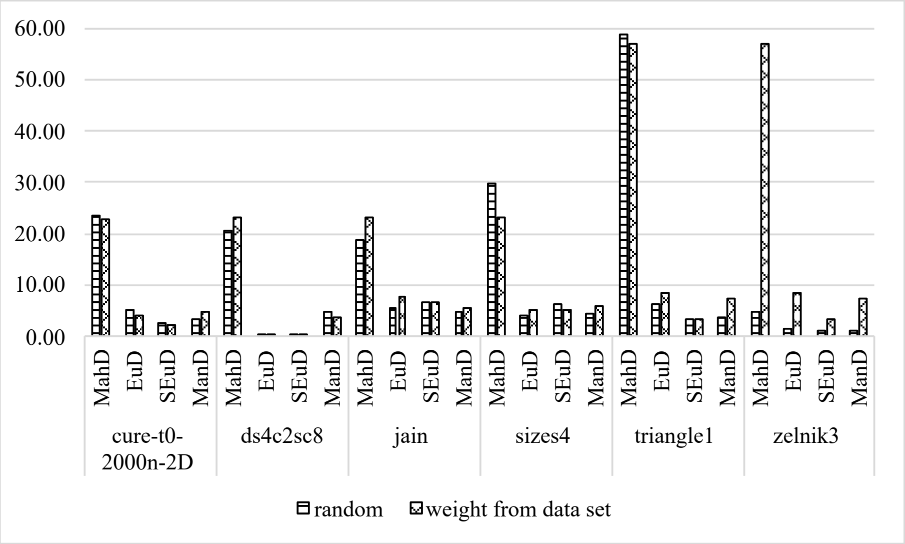


Fig. 4.6: Coefficient of variation (V) of the objective function value for Synthetic datasets

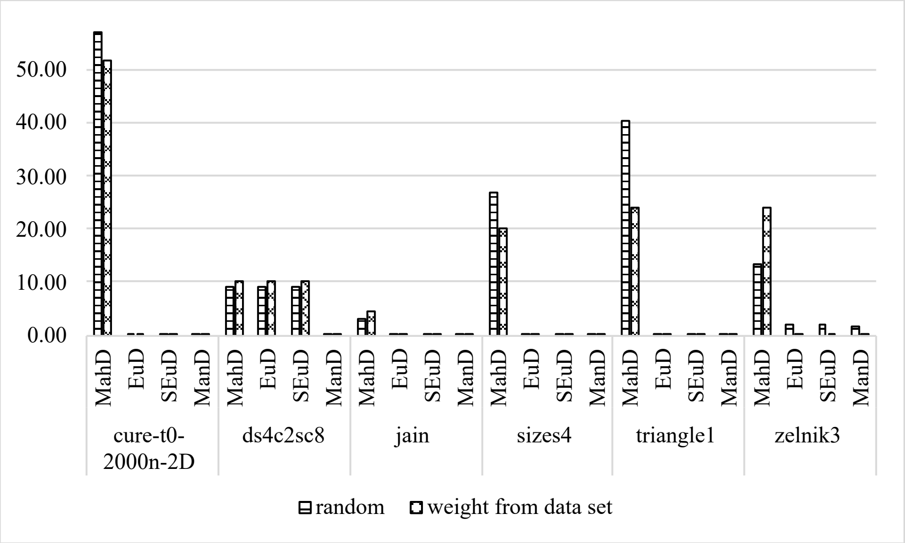
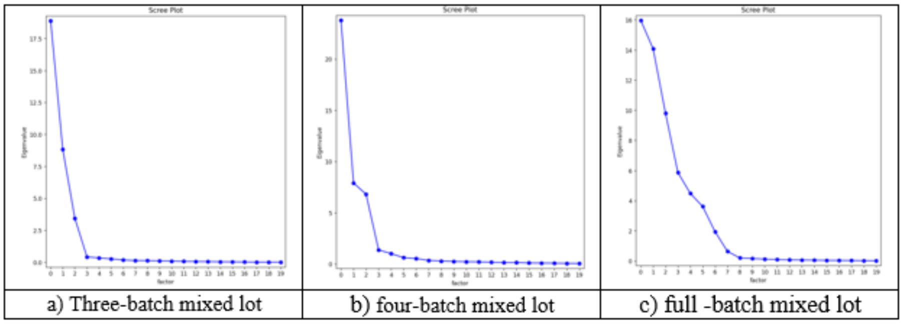


Fig. 4.7: The span coefficient (R) of the objective function value for Synthetic datasets



a) Three-batch mixed lot

b) four-batch mixed lot

c) full -batch mixed lot

Fig. 4.8: Scree plots

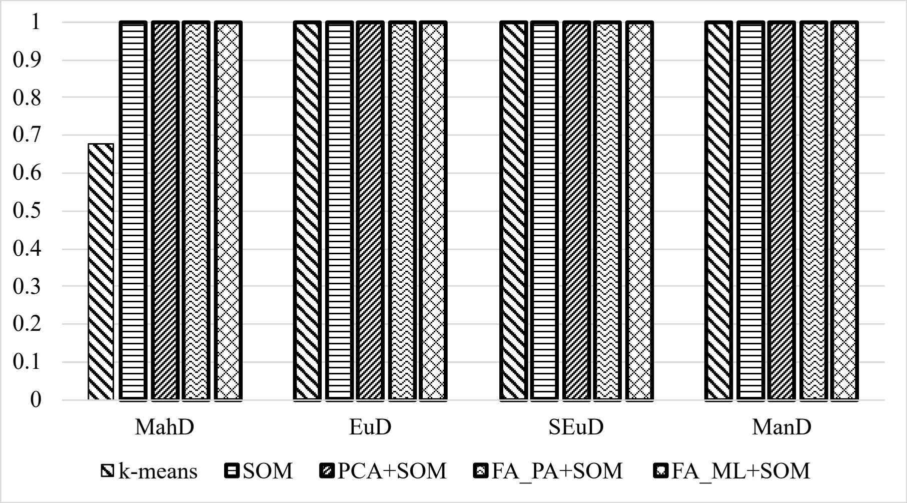


Fig. 4.9: Accuracy of ERI clustering. A mixed sample consisting of two batches

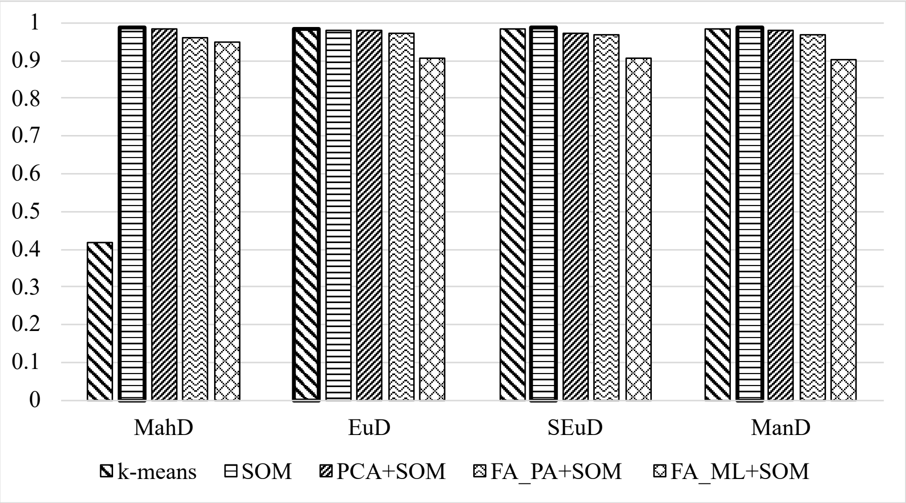


Fig. 4.10: Accuracy of ERI clustering. A mixed sample consisting of three batches

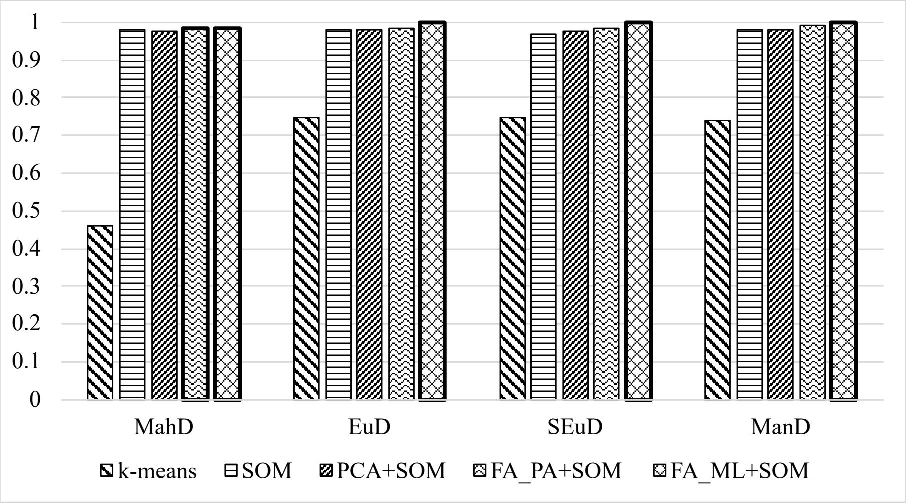


Fig. 4.11: Accuracy of ERI clustering. A mixed sample consisting of four batches

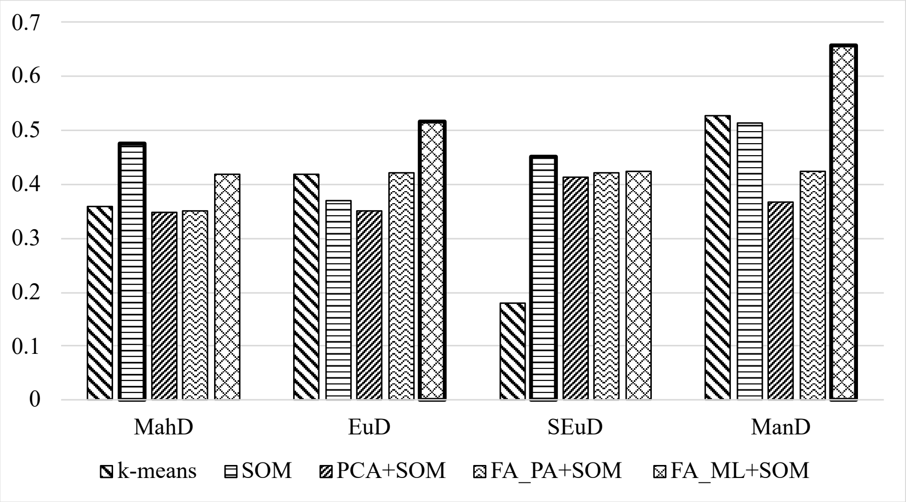


Fig. 4.12: Accuracy of ERI clustering. Total sample

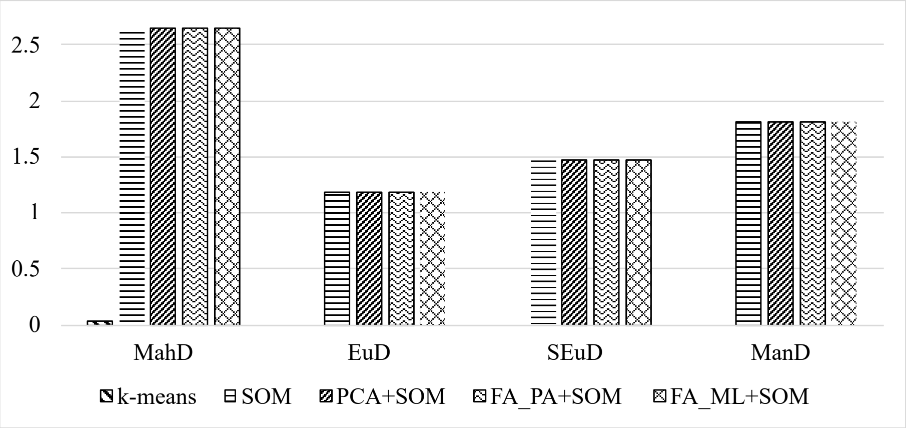


Fig. 4.13: Coefficient of variation (V) of the value of the objective function for two-batch mixed lot

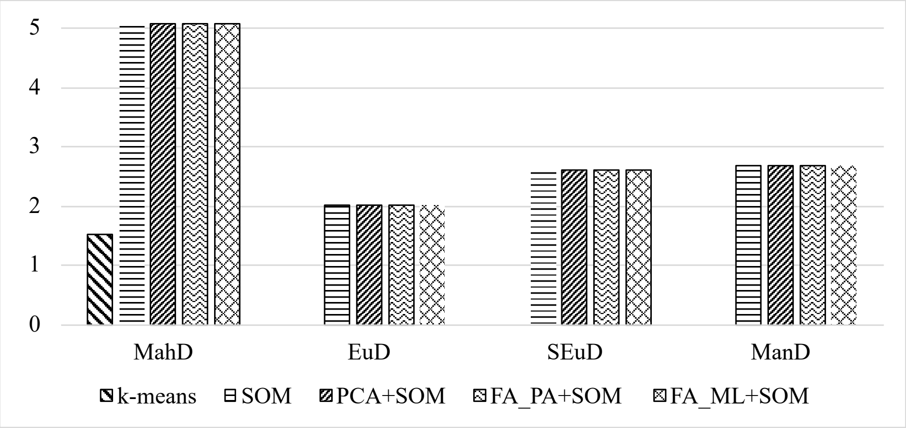


Fig. 4.14: The span coefficient (R) of the value of the objective function for two-batch mixed lot

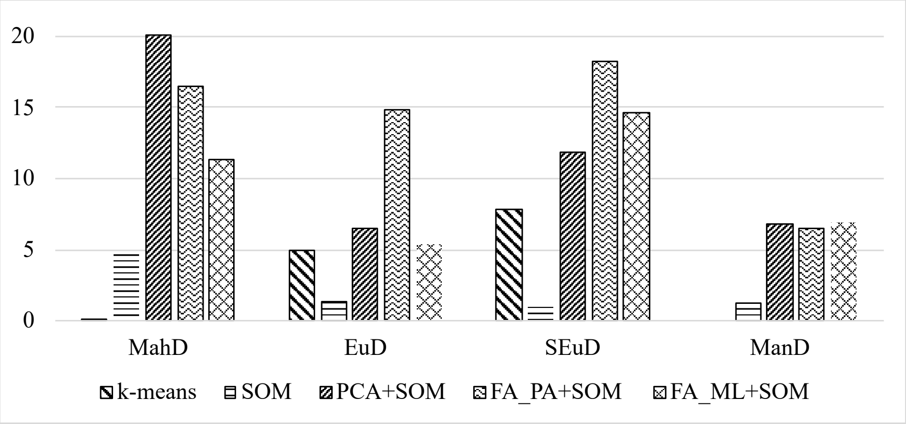


Fig. 4.15: Coefficient of variation (V) of the value of the objective function for three-batch mixed lot

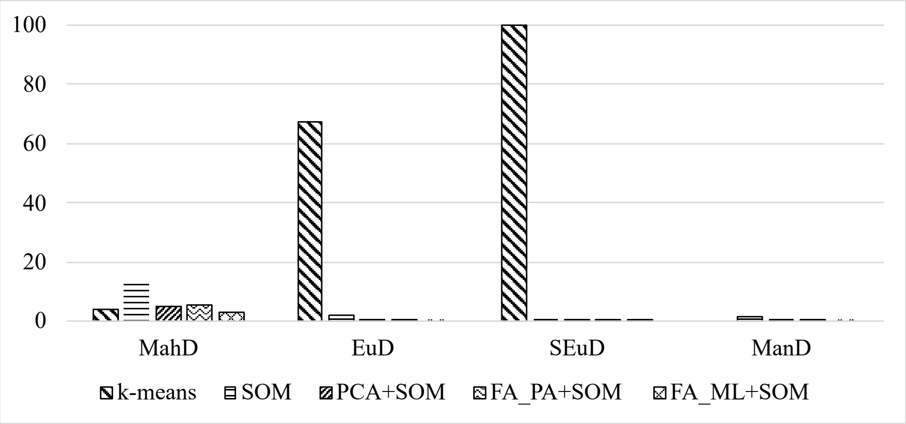


Fig. 4.16: The span coefficient (R) of the value of the objective function for three-batch mixed lot

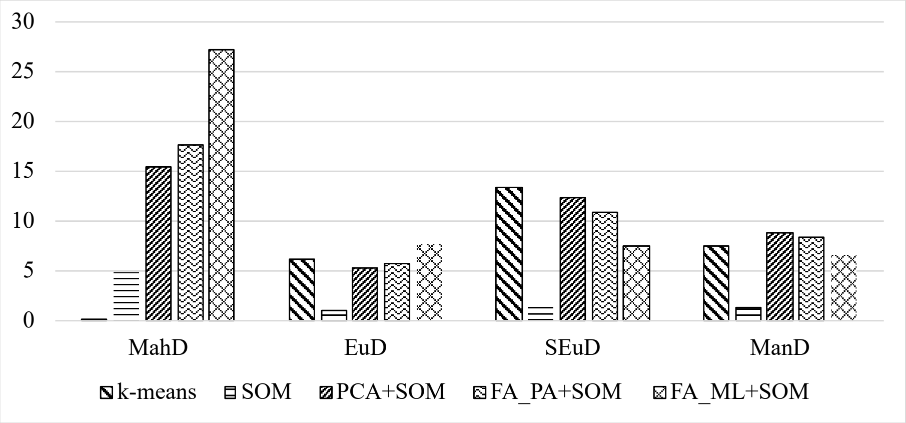


Fig. 4.17: Coefficient of variation (V) of the value of the objective function for four-batch mixed lot

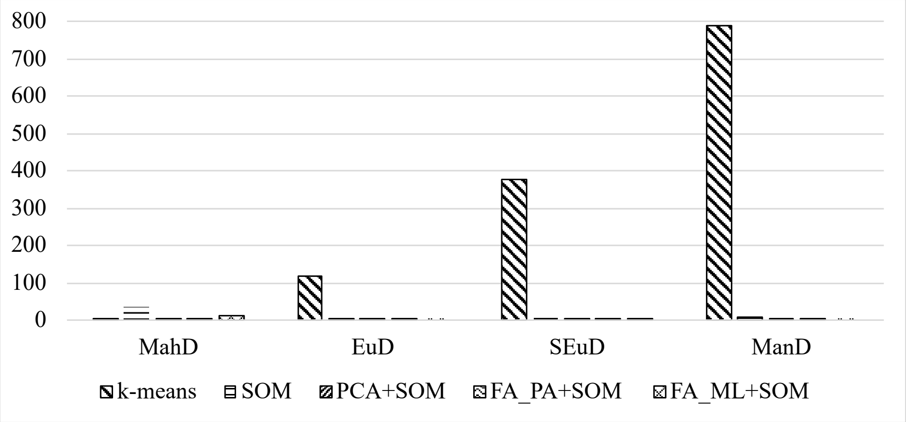


Fig. 4.18: The span coefficient (R) of the value of the objective function for four-batch mixed lot

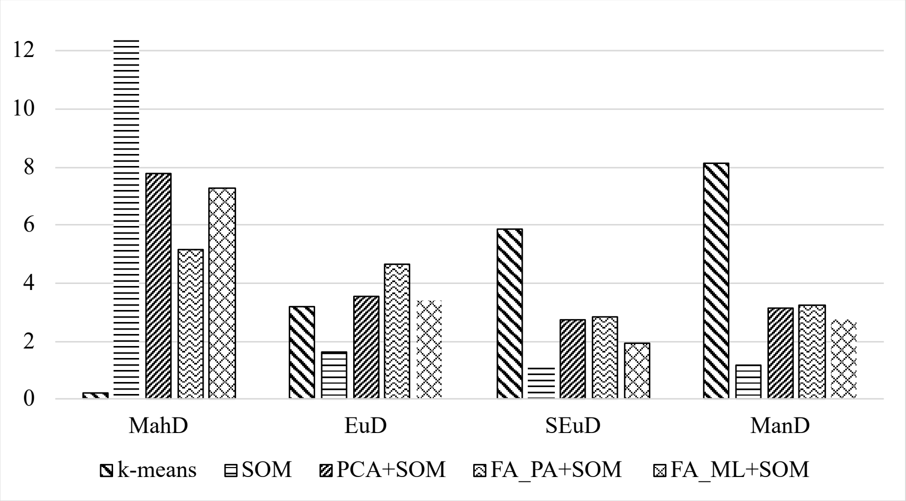


Fig. 4.19: Coefficient of variation (V) of the value of the objective function for full mixed lot

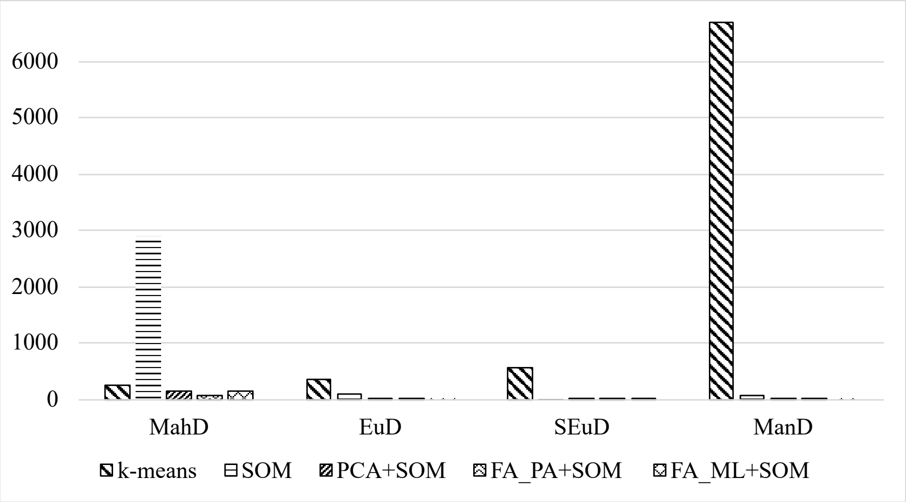
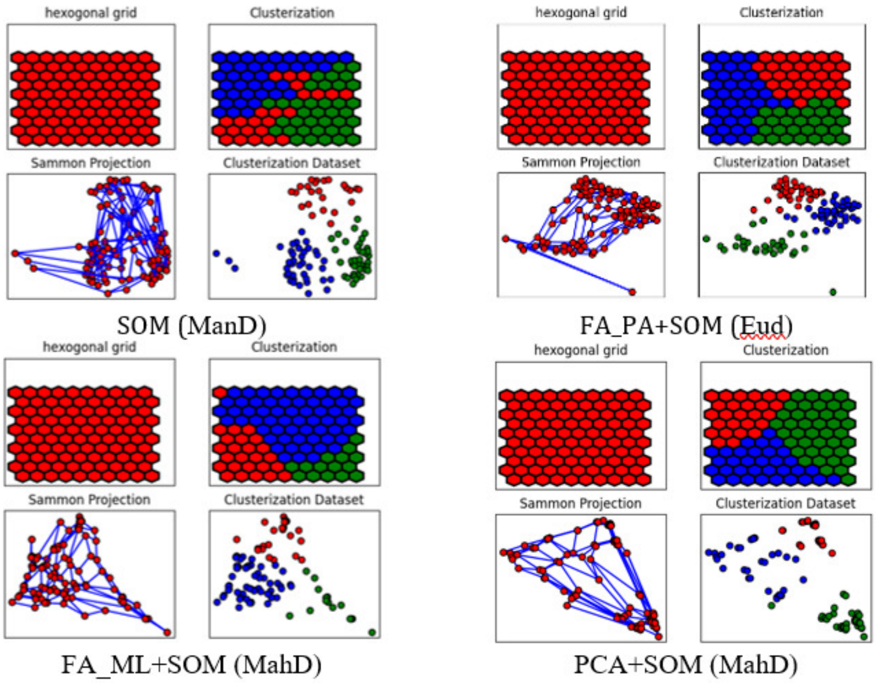


Fig. 4.20: The span coefficient (R) of the value of the objective function for full mixed lot



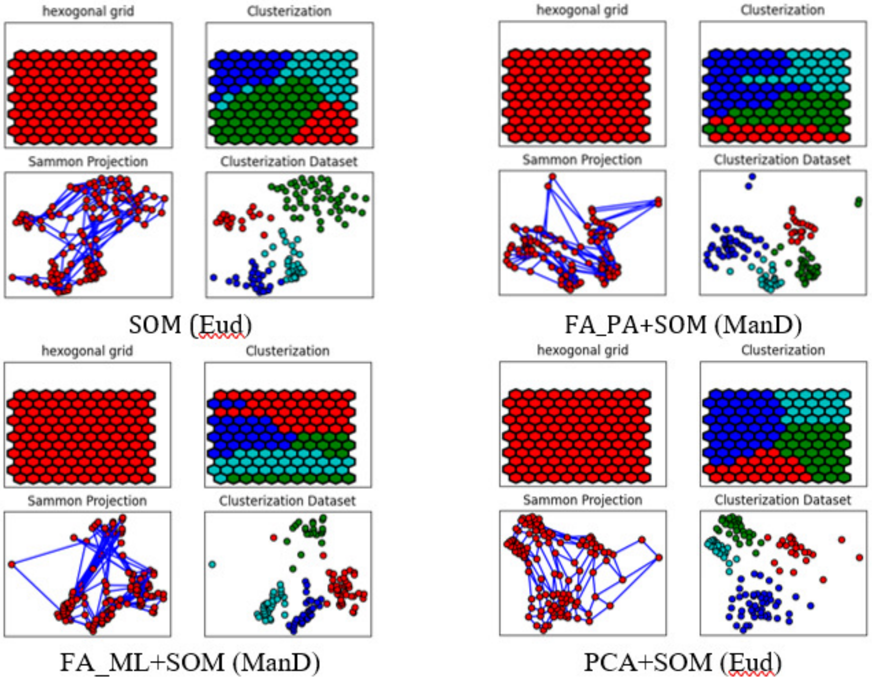
SOM (ManD)

FA_PA+SOM (Eud)

FA_ML+SOM (MahD)

PCA+SOM (MahD)

Fig. 4.21: Graph of the placement of scales. Three-batch mixed lot



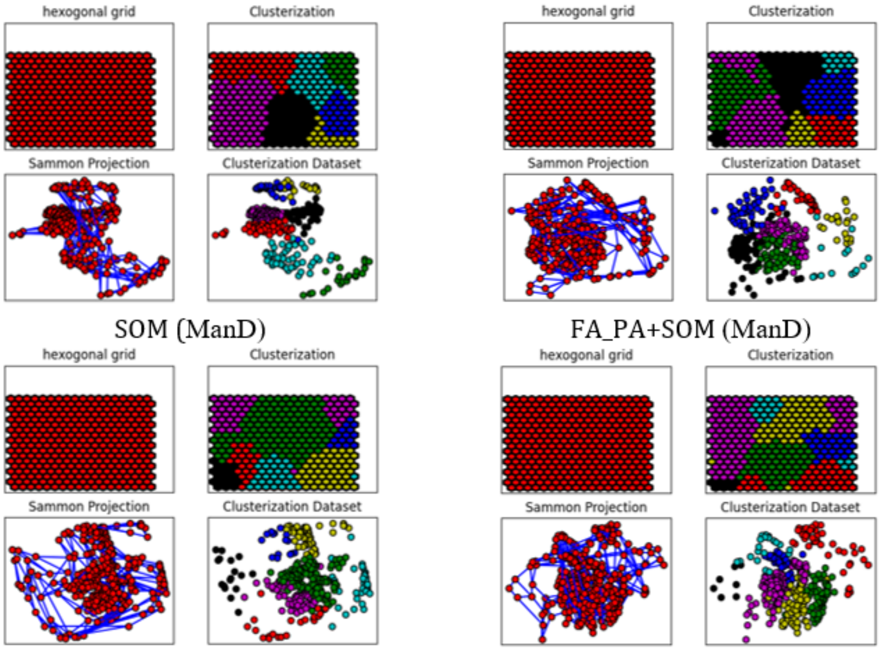
SOM (Eud)

FA_PA+SOM (ManD)

FA_ML+SOM (ManD)

PCA+SOM (Eud)

Fig. 4.22: Graph of the placement of scales. Four-batch mixed lot



SOM (ManD)

FA_PA+SOM (ManD)

FA_ML+SOM (ManD)

PCA+SOM (SEud)

FIG. 4.23: Graph of the placement of scales. Full mixed lot

5. Conclusions

In our work, we proposed algorithms for clustering of industrial products based on self-organizing Kohonen maps using various methods of extracting factors (factor analysis: Principal Component Analysis, Maximum Likelihood Estimation, Principal Component Analysis based on Singular Value Decomposition). We performed experiments with various distance measures (Mahalanobis, Euclidean, squared Euclidean, Manhattan), and different ways of neuron weight initialization (random, with a choice of weight coefficients from the dataset).

Our studies have shown that the used distance measure, in most cases, does not significantly affect the clustering accuracy.

The way neuron weights initialization plays a role in the stability of the objective function: the coefficient of variation for any type of mixed lot composition was higher (worse) with SOM (WCD) initialization.

The computational experiments showed that the use of methods of extracting factors in the SOM algorithm improves the accuracy of batch separation in most cases. Moreover, clustering accuracy decreases with an increasing number of homogeneous batches in a mixed lot.

REFERENCES

1. H. Allende, C. Moraga and R. Salas: *Robust estimator for the learning process in neural networks applied in time series.* ICANN, LNCS. **2415** (2002), 1080–1086.
2. H. Allende, S. Moreno, C. Rogel, and R. Salas: *Robust Self-organizing Maps.* In: Progress in Pattern Recognition, Image Analysis and Applications, CIARP 2004, Lecture Notes in Computer Science (A. Sanfeliu, J.F. Martinez Trinidad, J.A. Carrasco Ochoa, eds.), Springer, Berlin, Heidelberg, 2004, 3287.
3. H. Bauer, K. Pawelzik and G. Theo: *A Topographic Product for the Optimization of Self-Organizing Feature Maps.* Neural Information Processing Systems. **4** (1991), 1140–1147.
4. M. N. Belousova and V. A. Belousov: *Cluster analysis of the investment attractiveness of enterprises.* Financial analytics: problems and solutions. **10**(2) (2017), 181–191.
5. D. Brugger, M. Bogdan and W. Rosenstiel: *Automatic cluster detection in Kohonen's SOM.* IEEE Trans Neural Netw. **92** (1981), 161–172.
6. G. Cabanes and Y. Bennani: *Unsupervised Topographic Learning for Spatiotemporal Data Mining.* Advances in Artificial Intelligence. **832542** (2010), 1–12.
7. E. Coz, B. Artíñano, A. L. Robinson, G. S. Casuccio, T. L. Lersch and S. N. Pandis: *Individual Particle Morphology and Acidity.* Aerosol Science and Technology. **42**(3) (2008), 224–232.
8. D. L. Davies and D. W. Bouldin: *A Cluster Separation Measure.* IEEE Transactions on Pattern Analysis and Machine Intelligence. **1**(2) (1979), 224–227.
9. S. C. Dinger, M. A. Van Wyk and S. Carmona: *Clustering gene expression data using a diffraction-inspired framework.* BioMed Eng OnLine. **11** (2012), 85.
10. J. C. Dunn: *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.* Journal of Cybernetics. **3** (1973), 32–57.

11. B. Fritzke: *A Growing Neural Gas Network Learns Topologies.* Neural Information Processing Systems. **7** (1994), 625–632.

12. B. Fritzke: *Growing cell structures—a self-organizing network for unsupervised and supervised learning.* Neural networks. **7**(9) (1994), 1441–1460.

13. H. Harman: *Modern factor analysis.* The university of Chicago press, Chicago, 1967.

14. https://github.com/milaan9/Clustering-Datasets (last access: 14.06.2024).

15. S. Kaski, T. Honkela, K. Lagus and T. Kohonen: *WEBSOM – Self-organizing maps of document collections.* Neurocomputing. **21**(1-3) (1998), 101–117.

16. S. Kaski and K. Lagus: *Comparing self-organizing maps.* In: Partificial Neural Networks — ICANN 1996. Lecture Notes in Computer Science (C. von der Malsburg, W. von Seelen, J.C. Vorbrüggen, B. Sendhoff, eds.), Springer, Berlin, Heidelberg, 1996, 1112.

17. K. Kiviluoto: *Topology preservation in self-organizing maps.* In: Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, 1996, 294–299.

18. T. Kohonen: *Self-Organizing Maps.* Springer-Verlag Berlin Heidelberg, Berlin, 2001.

19. D. Lawley and A. Maxwell: *Factor analysis as a statistical method.* Butterworths, London, 1963.

20. E. López-Rubio, EJ. Palomo and E. Domínguez: *Bregman divergences for growing hierarchical self-organizing networks.* Int J Neural Sys. **24**(4) (2014), 1450016.

21. S. S. Makhrov: *Neural network clustering of wireless sensor network nodes.* T-Comm. **6** (2014).

22. E. Merenyi, B. Csatho and K. Tasdemir: *Knowledge discovery in urban environments from fused multi-dimensional imagery.* Urban Remote Sensing Joint Event, Paris, France, 2007, 1–13.

23. W. Messner: *Advancing our understanding of cultural heterogeneity with unsupervised machine learning.* Journal of International Management. **28**(2) (2022), 100885.

24. A. M. Newman and J. B. Cooper: *AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number.* BMC Bioinformatics. **11** (2010), 117.

25. W. Pearson: *LIII. On lines and planes of closest fit to systems of points in space.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. **2**(11) (1901), 559—572.

26. A. C. Pehlivanli, O. K. Ersoy and T. Ibrikci: *Drug/nondrug classification with consensual Self-Organising Map and Self-Organising Global Ranking algorithms.* International Journal of Computational Biology and Drug Design (IJCBDD). **1** (2008), 4.

27. P. J. Rousseeuw: *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis.* BMC Bioinformatics. **20** (1987), 53–65.

28. P. J. Saheb-Al-Zamani at ai.: *Limited regeneration in long acellular nerve allografts is associated with increased Schwann cell senescence.* Experimental Neurology. **247** (2013), 165–177.

29. G. Shkaberina, N. Rezova, E. Tovbis and L. Kazakovtsev: *Visual Assessment of Cluster Tendency with Variations of Distance Measures.* Algorithms. **16** (2023), 5.

30. P. E. Sokolov: *The use of Kohonen maps to assess the radioactivity of building materials*. Bulletin of Science and Education of the North-West of Russia. **1** (2017).

31. K. Tasdemir and E. Merenyi: *A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures*. IEEE Trans Syst Man Cybern B Cybern. **41**(4) (2011), 1039–53.

32. J. Tian, M. N. Azarian and M. Pecht: *Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm*. Proceedings of the European Conference of the PHM Society. **2**(1) (2014).

33. K. Uberla: *Factorenanalyse*. Springer-Verlag, Berlin, 1977.

34. J. Vesanto and E. Alhoniemi: *Clustering of the self-organizing map*. IEEE Transactions on Neural Networks. **11**(3) (2023), 586–600.

35. J. Vesanto and M. Sulkava: *Distance Matrix Based Clustering of the Self-Organizing Map*. In: Artificial Neural Networks — ICANN 2002, Lecture Notes in Computer Science (J.R. Dorronsoro, eds.), Springer, Berlin, Heidelberg, 2022, 2415.

36. T. Villmann, R. Der, M. Herrmann and T. M. Martinetz: *Topology preservation in self-organizing feature maps: exact definition and measurement*. IEEE Trans Neural Netw. **8**(2) (2023), 256–266.

37. S. Wu and T. W. S. Chow: *Clustering of the self-organizing map using a clastering validity index based on inter-cluster and intra cluster density*. Pattern Recognition. **37** (2004), 175–188.

38. A. V. Zelensky and A. V. Zelensky: *Modeling of the training process in biathlon using artificial intelligence*. Scientific notes of Lesgaft University. **2**(144) (2007).