Nonlinear Analysis: Modelling and Control, Vol. 30, No. 6, 1163–1185
https://doi.org/10.15388/namc.2025.30.43803

Vilnius University Press

An advanced visualization of self-organizing maps by determining data clusters

Pavel Stefanovič[a], **Olga Kurasova**[b]

[a] Department of Information Systems,
Vilnius Gediminas Technical University
Saulėtekio ave. 11, LT-10223, Vilnius, Lithuania
pavel.stefanovic@vilniustech.lt

[b] Institute of Data Science and Digital Technologies,
Vilnius University
Akademijos str. 4, LT-08412, Vilnius, Lithuania
olga.kurasova@mif.vu.lt

Received: July 6, 2025 / **Revised:** September 7, 2025 / **Published online:** October 14, 2025

Abstract. This paper proposes a novel approach to improve the visualization capabilities of self-organizing maps and facilitate the identification of the resulting clusters. Unlike other clustering algorithms, self-organizing maps lack the feature to select a predefined number of clusters, and the boundaries of the clusters are not explicitly represented on the self-organizing maps. The main advantage of our proposed approach is that the option for selecting the desired number of clusters has been implemented. The experimental investigation was performed using four datasets with different characteristics. The improved visualization leverages various similarity distances to assess their impact on performance. The effectiveness of the novel approach to clustering results has been compared with those of the well-known k-means and hierarchical clustering methods, which allow for the selection of the desired number of clusters. Additionally, the visualization results, obtained by the proposed approach, were compared with those produced using the Orange Data Mining tool, where the u-matrix is applied to visualize a self-organizing map. The advantage of our approach compared to the u-matrix visualization has been highlighted in this paper. The performance of clustering algorithms has been measured by calculating the ratio of data items correctly assigned to clusters in the case when the clusters are predefined in the analyzed dataset. The results obtained showed that the most effective similarity distances are the cosine and correlation distances, which help to detect the correctly predefined clusters in the visualization of self-organizing maps.

Keywords: self-organizing maps, u-matrix, similarity distances, visualization, data clustering, number of clusters.

1 Introduction

Over the past decades, the amount of structured and unstructured data in different forms, such as numbers, text, sounds, and images, has increased. The main reason is that new

technologies enable the use of these data for various tasks, especially in the field of artificial intelligence, where the key is the utilization of historical data. Depending on the type of data analyzed, data analysis can be performed using different data mining techniques. Today's numerous intelligent systems utilize machine learning models, primarily based on classification algorithms. In the context of supervised learning, the data used to train machine learning models must be labeled. Machine learning models can be applied to a wide range of tasks, including the detection of phishing email, grain yield analysis in agriculture, the development of recommendation systems, and sentiment analysis. The labeled data are not always available. Often, raw data are first collected from various sources and then labeled by experts. With unlabeled data, unsupervised learning algorithms can be applied to a wide range of problems. Data clustering is a typical data mining task that does not require supervised learning. The clustering algorithms can be used in various areas to group unlabeled data by identifying similarities between the analyzed data. When data are clustered, visualization techniques become essential for representing these clusters and intuitively comprehending the underlying data structures and patterns. Numerous data visualization techniques are widely used, as they facilitate the rapid interpretation of analyzed data, considering that visual representations are generally much quicker to understand than numerical estimations.

Data clustering is a complex task because, when using unsupervised learning algorithms, the data are usually unlabeled, making it more challenging to evaluate clustering results compared to classification tasks. The performance of classification models can be easily estimated using typical measures such as accuracy, precision, recall, and F1 score. To calculate these measures, it is essential to compare the predicted class with the actual class labels provided in the data. However, these measures cannot be applied to evaluate the clustering results because the data are usually unlabeled. One of the measures used to estimate the results of data clustering is the silhouette score. Many researchers have proposed various heuristic measures to assess the quality of clustering, which can be applied to different clustering algorithms. There are several well-known clustering algorithms, but many of them cannot visualize the obtained clustering results. Additionally, the desired number of clusters is often determined automatically, and the researcher cannot manually select it. This choice is crucial, as it can significantly impact the details of the clustering results, which in turn affect the interpretation and application of the findings to the study's objectives. One of the most well-known k-means algorithms offers the possibility to choose a desired number of clusters in data clustering; however, this algorithm does not provide a visualization of the obtained clustering results. Furthermore, scientific studies show [3] that the outcomes of clustering algorithms are significantly influenced by the initial centers of the clusters, which are often chosen at random during the algorithm's execution. A Self-Organizing Map (SOM) [15] serves not only as an algorithm for clustering but also as a technique for multidimensional data visualization (see Section 3 for more details). The advantage of a SOM lies in its ability to provide a visual representation of clustering results. However, it does not allow the choice of the desired number of clusters in advance. Additionally, the visualization of the SOM fails to clearly define the boundaries of each cluster.

The main contributions of the paper are as follows:

- A novel approach is proposed to enhance the visualization of SOMs. The approach includes an option to select a desired number of clusters and similarity distance-based determination of data cluster boundaries, addressing a significant limitation of conventional SOMs.
- The effectiveness of the proposed visualization approach has been evaluated on four different datasets (Iris, Glass, Mushrooms, and Elections), each with unique characteristics. This comprehensive evaluation demonstrates the versatility and validity of the approach in various data mining scenarios.
- The paper compares the proposed approach with well-known clustering algorithms, such as k-means and hierarchical clustering, and with the visualization capabilities of the SOM implemented in the Orange Data Mining tool [4]. This comparison highlights the superiority of the proposed approach in terms of clarity of cluster visualization and the ability to select a specific number of clusters, thus making a significant contribution to the field of data clustering and visualization.

2 Related work

The data mining area includes numerous methods and algorithms, from data retrieval to data analysis and knowledge extraction. The amount of structured and unstructured data grows every day, therefore, data mining can be applied in different areas, for example, to discover something unknown or to find various patterns in the data. In the research by [12], a comprehensive survey of data mining techniques and applications was performed. The analysis has shown that the data classification, clustering, outliers' detection, regression analysis, and association analysis tasks are usually performed. To solve these tasks, the right data mining method needs to be chosen. [12] highlights that statistical methods are still widely used in data mining. Nowadays, data classification is closely related to machine learning model training, as various classification algorithms are applied to these models. Many types of research can be found related to machine learning, for example, the application in manufacturing, the different diseases and illness analysis, or educational data analysis. In data classification tasks, only the labeled data are used. When the data are unlabeled, the clustering algorithms can be employed. The main aim of clustering algorithms is to group the data analyzed by their similarity. There are many clustering algorithms that can be applied to data analysis. Clustering algorithms belong to the type of unsupervised learning because unlabeled data are analyze.

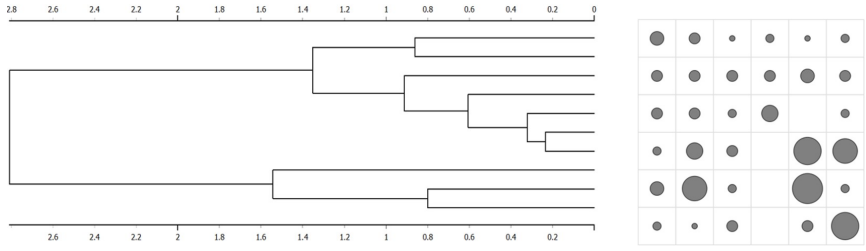
The literature analysis shows that the most used clustering algorithms are hierarchical clustering, density-based (DBSCAN), and k-means algorithms [8]. A lot of research demonstrates the application of these algorithms. These algorithms, developed over 60 years ago, continue to be useful in practice and are extensively employed in many modern applications. Over time, many modifications have emerged to improve these algorithms, either by combining them with other algorithms or by applying them to specific domains. However, the core principles of these algorithms have largely remained unchanged. In the research by [10], the k-DBSCAN algorithm was proposed to help in analyzing big data. The authors compared the results obtained by the classical DBSCAN,

HCA-DBCAN, density-grid algorithms, and the proposed K-DBSCAN. The main aim was to reduce the time required for clustering. In another research by [17], DBSCAN has been enhanced by incorporating triangle inequality, neighbor similarity, and a fast neighbor search algorithm. These improvements reduce the number of distance calculations needed during clustering, thereby increasing the efficiency of the DBSCAN algorithm. Compared to k-means and DBSCAN, the hierarchical clustering algorithm has its own advantage that provides not only the clusters, but also dendrograms, where clusters can be observed visually. [11] provided a deep review of k-means and hierarchical clustering in the context of air pollution data analysis. The most popular algorithm, k-means, has also undergone many improvements to handle big data analysis. The advantage of this algorithm is that it allows one to choose the desired number of clusters.

In cluster analysis, selecting the optimal number of clusters is crucial. Various clustering algorithms use different approaches to detect the optimal number of clusters. For example, the elbow method is used in the k-means algorithm. The main aim of the elbow method is to point out the curve of cluster variance with the number of clusters, where the smaller variance indicates the optimal number of clusters. In the case of hierarchical clustering, it often uses dendrograms to visualize and cut at a level that maximizes intra-cluster similarity while maintaining distinct inter-cluster separations. The optimal point in hierarchical clustering is generally more subjective. The self-organizing maps algorithm does not allow to choose of the desired number of clusters at all, the clusters are formed in the map and are evaluated by the researcher manually.

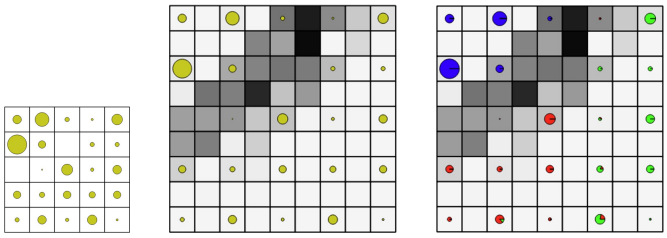
Related works showed that less popular, but also frequently used clustering algorithms are Gaussian mixtures, partition clustering, SOMs, and fuzzy clustering. No matter which clustering algorithms are used in data analysis, each of them has its strengths and weaknesses. Therefore, a distinct advantage of hierarchical clustering and SOMs over other algorithms lies in their capabilities to provide a visual representation of the clustering results. In the case of hierarchical clustering, the dendrogram can be formed in two ways: agglomerative and divisive. In the agglomeration method, each data item is first considered as a separate cluster, which is then merged with other data items by forming larger clusters based on the calculated similarity distances. Conversely, the divisive method starts with all data encompassed in a single cluster, which is then iteratively divided into smaller clusters [21].

As we can see in Fig. 1(a), the dendrogram represents all the similarities between the analyzed data. Obviously, identifying all clusters within the dendrogram poses a significant challenge and often requires manual intervention by researchers. Some suggestions on how to determine clusters according to the distance boundaries in the dendrogram can be found in the scientific literature [30]. Furthermore, according to many studies, hierarchical clustering is more suitable for small datasets, given that dendrograms tend to become overly complex and less interpretable with the increase in dataset size. When visualizing SOMs, the data are essentially presented in a simple table, where the most similar data are located in the same cell of the map, and the least similar data are positioned distantly. Orange Data Mining, a user-friendly tool, provides a set of approaches to data analysis and visualization, including the representation of SOMs. The tool was developed in the Python programming language.



(a) (b)

Figure 1. Example of visualization of data clusters using the iris dataset [7]: (a) the dendrogram, obtained by hierarchical clustering; (b) the visualization of a SOM



(a) (b) (c)

Figure 2. The example of visualization in the SOM using iris dataset: (a) without u-matrix visualization; (b) with u-matrix visualization; (c) with u-matrix visualization and labeled data colors

Figure 2 presents the iris data clustered and visualized by the SOM using the Orange Data Mining tool. The tool displays circles for each cell of the map (Fig. 2(a)), where the size of the circle represents the quantity of data items that have fallen into the same SOM cell. If the labeled data is analyzed, the pie chart can be displayed instead of a circle which helps to see the distribution of data items from different classes that fall in the same SOM cell (Fig. 2(c)). Like hierarchical clustering, analyzing unlabeled data makes it difficult to interpret the results obtained and to define the cluster boundaries in the SOM precisely (Fig. 2(a)). To address this problem, the visualization of the u-matrix [28] in SOM can be utilized (Figs. 2(b), 2(c)). As illustrated in Fig. 2(a), the size of the SOM is 5×5. However, in the u-matrix visualization, additional cells are inserted to signify the distances between neighboring SOM cells, as depicted in Figs. 2(b) and 2(c). In this way, the cells are colored by the values of the u-matrix elements. The light areas can be considered as clusters (denoting smaller distances between data items), whereas the dark areas act as cluster separators (signifying larger distances between data items). While greyscale visualization remains the most popular, there are also modifications employing diverse color schemes. In the example shown in Fig. 2, data from three classes of irises were analyzed. However, in Fig. 2(b), if the data classes are not colored in the visualization, it is difficult to ascertain the exact number of clusters within the original data. It is obvious that a researcher, lacking prior information about the data, can conclude that there are only two classes in Fig. 2(b) – one class located in the top-left corner and the other in

the bottom-right corner, as demarcated by the dark color of the cells. Conversely, if data labels were visually distinguished by colors (Fig. 2(c)), it would be possible to identify three distinct clusters: a blue cluster in the top-left corner, a predominantly red cluster in the center, and a green cluster on the right side of the SOM. While the u-matrix helps to identify potential clusters in the SOM, it does not permit the selection of a desired number of clusters, even with the knowledge that the analyzed data belong to three classes.

The analysis of research performed on SOMs reveals that several researchers have tried to improve either the visualization or the clustering performance of SOMs. Notably, all these studies were published over a decade ago. The first improvement in SOMs was proposed by [18] over 25 years ago, where the authors introduced alternative visualization methods. The main idea was to visualize the relationship between cells in SOMs to facilitate the detection of cluster boundaries without modifying the training process in the algorithm. This concept was further developed into Expanding SOM (ESOM) by [14]. During the ESOM learning process, the neural network expands, and the neuron corresponding to a distant data item gets a large expanding force. The authors stated that their experimental results demonstrated the superiority of the ESOM in producing better visualization results compared to the classical SOM while maintaining a similar execution time. [6] proposed an improvement of SOM clustering without changing the SOM algorithm. The main problem that the authors faced was that classical visualization techniques were not suitable for gene analysis. The proposed visualization was grounded in a graph network paradigm, where the similarity of input data was computed and moved from one graph to another. This methodology facilitated the formation of clusters comprising similar data items. A more recent approach proposed by [20] was designed to automatically detect the appropriate number of clusters in a SOM for the given data and to assign the data to the clusters. The main aim of the proposed method was to visualize both the probability of the different clusters and the number of data points contained within each neuron. Notably, this approach applies to both labeled and unlabeled data. Recently, [13] proposed a SOMTimeS approach for time series clustering based on a SOM algorithm. The efficiency of the proposed approach has been proven by performing experimental investigations using the healthcare data of patient-clinician serious illness conversations. The results obtained by the SOMTimeS were compared to the modified k-means algorithm. In the research by [16], another modification of the SOMs was proposed that authors called the dynamic time warping self-organizing map. This approach has also been applied in healthcare data analysis and for time series clustering and pattern recognition.

In summarizing the analysis of related work, it becomes evident that no existing approach definitely addresses the determination of clusters in SOMs. Several researchers have tackled this issue by developing methods for the automatic identification of clusters. Others tried to define the boundaries of the clusters in SOM. However, most of these modifications and improvements are suitable when applied to labeled data as this facilitates a clearer observation of the clustering results. Our study introduces a novel approach that focuses only on the SOM visualization using unlabeled data, without changing the clustering algorithm. This approach enables not only the definition of cluster boundaries but also the selection of the number of clusters or the automatic determination of the most suitable cluster count.

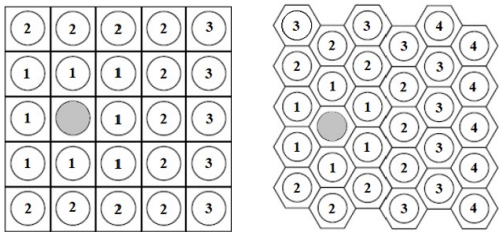
3 The background of the proposed approach

The main aim of the proposed approach is to detect data similarity in the self-organizing map in order to determine the number of clusters and the cluster boundaries. As with the u-matrix and other proposed approaches reviewed in related works, the similarity distance between cells of the SOM needs to be computed. The main difference with other methods is that our proposed approach does not use a labeled dataset, but the data clustering is performed on the basis of the detected centers of the clusters, and the clusters are expanded step by step depending on the similar cells. The SOM and the proposed approach have been described in this Section.

3.1 Self-organizing maps

Many different clustering algorithms can be used, but our research aims to enhance the SOM visualization by identifying the appropriate number of clusters and delineating their boundaries. The SOM was introduced over 40 years ago by [15]. It is an unsupervised neural network trained using competitive learning. The advantage of this algorithm over other clustering algorithms lies in its dual capability. SOM is not just used to cluster data, but also to show results in a visual form, thereby simplifying interpretation of the results for researchers. The visual representation of SOMs may be presented in various formats [5]. The main aim of SOM is to preserve the topology of multidimensional data during their transformation into a lower-dimensional space, typically two-dimensional. SOMs are versatile, being employed for clustering, classification, and visualization of data from a wide range of domains and types [29]. For example, SOMs have been effectively utilized in text data analysis to cluster text documents [23]. In our previous research, SOM was used to adjust the data classes of the multi-label text dataset [26], which helps improve the quality of data labeling. In addition, the modified SOM has been demonstrated to be effective in detecting outliers within the analyzed data set [25]. In today's popular image analysis field, the SOM can also be applied as an additional layer within the architecture of object detection models or to help cluster image data [1]. In Breskuvienė et al.'s study [2], the capability of the modified SOM, so-called FID-SOM, has been used to tailor handling of imbalanced datasets common in fraud detection. The method stands out by creating a refined dataset composed of the Best-Matching Units from the trained SOM, which represent key attribute patterns. This approach ensures that the selected features carry the most informative content. FID-SOM has shown competitive performance compared to existing techniques and offers promising innovation in the field. The SOM often serves as a task-specific component within more complex solutions. In the research by [27], SOM was employed to identify similar users in a travel direction recommendation model.

The SOM is a set of nodes (cells). The connections between the inputs and the nodes are associated with specific weights. A set of weights corresponds to each node. The set of weights forms a vector M_{ij}, $i = 1, \ldots, k_a$, $j = 1, \ldots, k_b$, commonly referred to as a neuron or codebook vector, where k_a and k_b denote the number of rows and columns of the SOM, respectively. The learning process of the SOM algorithm starts with



(a) Rectangular topology (b) Hexagonal topology

Figure 3. Two-dimensional SOM of different topology.

the initialization of the components of the vectors M_{ij}, where they can be initialized at random, linear, or by the principal components. At each learning step (iteration), an input data item X_s, $s \in \{1, \ldots, N\}$, is passed to the SOM. Here N is the number of data items. The data item X_s is compared to all neurons M_{ij}. Usually, the Euclidean distance between this input data item X_s and each neuron M_{ij} is computed. The vector M_w with the minimal Euclidean distance to X_s is designated as a neuron winner. This winning neuron is commonly referred to as the Best Matching Unit (BMU). All neuron components are adapted according to the learning rule (Eq. 1):

$$M_{ij}(t+1) = M_{ij}(t) + h_{ij}^{w}\big(X_p - M_{ij}(t)\big), \tag{1}$$

where t is the number of the learning step, h_{ij}^{w} is the neighboring function ($h_{ij}^{w} \to 0$ as $t \to \infty$), w is a pair of indices of the neuron winner corresponding to the data item X_s, $s \in \{1, \ldots, N\}$. The learning process is iteratively repeated until the predefined maximum number of iterations is reached. When the training is completed, the winning neurons are determined for each data item from the set $X_1, X_2, \ldots, X_N$. Subsequently, every data item is allocated to a specific SOM cell $Cell_{i,j}$ ($i \in \{1, \ldots, k_a\}$, $j \in \{1, \ldots, k_b\}$), corresponding to its respective winning neuron.

Usually, in practice, two types of topologies of the SOMs are used: rectangular and hexagonal (Fig. 3). The main difference in the SOM topology is that the neighboring rank is determined differently. The neighboring rank concept plays a crucial role in the SOM training process. All the cells adjusted to a neuron can be defined as its neighbors of the first rank, then the cells adjacent to the first-rank neighbor, excluding those already considered, as neighbors of the second order, etc. In Fig. 3, the number at the center of the cell represents the neighboring rank compared to the grey cell $Cell_{3,2}$, i.e., first rank – 1, second rank – 2, etc. In the case of rectangular topology (Fig. 3(a)), the $Cell_{3,2}$ will have 8 neighboring cells of the first rank, and in the case of hexagonal topology – 6 neighboring cells (Fig. 3(b)).

3.2 Similarity distances

Various distance metrics can be found in the scientific literature, ranging from well-known metrics to heuristic distances. These measures find application across diverse

tasks, but their fundamental purpose is to quantify the pairwise similarity between data items. The literature review reveals that Euclidean and cosine distances are commonly utilized. For instance, in some research, the performance of various similarity distances has been investigated to determine the similarity between analyzed documents [19]. Furthermore, these similarity distances are often used in various classification or clustering algorithms, including but not limited to the k-nearest neighbor algorithm, hierarchical clustering, and SOM. In this paper, several similarity distances have been investigated to see how different distances influence the clusters resulting from the visualization of SOMs. Therefore, the Euclidean, cosine, correlation, Jaccard, and Spearman distances have been used in the proposed approach [33]. Euclidean distance is the distance between two data points in Euclidean space. In the context of data analysis, it is often used to find the dissimilarity or similarity between data items. Smaller values of the Euclidean distance indicate bigger similarity. Suppose we have two data items $R = (r_1, r_2, \ldots, r_q)$ and $P = (p_1, p_2, \ldots, p_q)$, where q is the dimensionality of the vector corresponding to each data item. The Euclidean distance between these data items is computed using Eq. 2

$$d_{\mathrm{Euc}}(R,P) = \sqrt{\sum_{k=1}^{q}(r_k - p_k)^2}. \tag{2}$$

Cosine similarity distance (Eq. 3) is useful in the analysis of high-dimensional data, such as in the detection of pairwise similarities in texts or images. The value of the cosine similarity indicates the cosine of the angle between two vectors in a multidimensional space.

$$d_{\mathrm{Cos}}(R,P) = 1 - \frac{\sum_{k=1}^{q} r_k p_k}{\sqrt{\sum_{k=1}^{q} r_k^2}\sqrt{\sum_{k=1}^{q} p_k^2}}. \tag{3}$$

In the context of similarity, correlation distance is often used to evaluate how well two points follow a linear trend (Eq. 4). In this case, a value close to 1 shows a high similarity.

$$d_{\mathrm{Corr}}(R,P) = 1 - \frac{\sum_{k=1}^{q}(r_k - \overline{r}_k)(p_k - \overline{p}_k)}{\sqrt{\sum_{k=1}^{q}(r_k - \overline{r}_k)^2 \sum_{k=1}^{q}(p_k - \overline{p}_k)^2}}. \tag{4}$$

Here $\overline{r}_k = (1/q)\sum_{k=1}^{q} r_k$ and $\overline{p}_k = (1/q)\sum_{k=1}^{q} p_k$.

The Jaccard distance (Eq. 5) is considered a measure of dissimilarity between two data points. This distance is often used for data clustering, document similarity detection, etc.

$$d_{\mathrm{Jacc}}(R,P) = 1 - \frac{\sum_{k=1}^{q} r_k p_k}{\sum_{k=1}^{q} r_k^2 + \sum_{k=1}^{q} p_k^2 - \sum_{k=1}^{q} r_k p_k}. \tag{5}$$

Spearman distance (Eq. 6) is based on Spearman's rank correlation coefficient. This similarity distance can be used in various applications such as clustering, classification, and recommendation systems applications. The Spearman distance helps to calculate the similarity or dissimilarity between two data points.

$$d_{\mathrm{Sp}}(R,P) = 1 - \frac{\sum_{k=1}^{q}(u_k - \overline{u}_k)(y_k - \overline{y}_k)}{\sqrt{\sum_{k=1}^{q}(u_k - \overline{u}_k)^2 \sum_{k=1}^{q}(y_k - \overline{y}_k)^2}}. \tag{6}$$

Here u_k and y_k are the rank of r_k and p_k taken over $r_1, r_2, \ldots, r_q$ and $p_1, p_2, \ldots, p_q$, respectively. $\overline{u}_k = (q+1)/2$, $\overline{y}_k = (q+1)/2$.

4 The proposed approach

As previously mentioned, the SOM learning algorithm remains unchanged in the proposed approach, with enhancements being focused solely on the visualization aspect of SOMs. It is important to note that the results of the SOM are also influenced by the hyperparameters selected before training. However, this research does not delve into the effects of hyperparameter selection. Instead, the hyperparameters were chosen according to findings from our previous research [22, 24]. The basis of the proposed approach is the detection of similarity between data items in SOM cells. The different similarity distances, detailed in Subsection 3.2, have been explored, and their performance is presented in Section 5. Assume an unlabeled dataset $X = X_1, X_2, \ldots, X_N$ are analyzed, where N is the number of data items. The SOM of the size $k_a \times k_b$ has been trained using the dataset X. Once the neural network has been trained, the data items are assigned to the SOM cells according to the winning neurons. Additionally, we have selected a desired number of clusters C. The pseudocode of the proposed approach for cluster determination in the obtained SOM is outlined and described in Algorithm 1.

The proposed approach has been implemented in the MATLAB environment. To demonstrate the outcome of the proposed approach, the well-known iris dataset [7] was used to train the SOM. The dataset consists of 150 items, and each of the three classes has the same number of data items. As usual, in this dataset, the data items of the first class (Iris Setosa) differ from the second (Iris Versicolor) and third class (Iris Virginica) data items. Also, the data items of the second and third classes slightly overlap. Suppose that a SOM of size 6×6 is trained using the iris dataset. Figure 4(a) illustrates the cluster determination process using Algorithm 1, with the number of the desired clusters $C = 3$. Here, a correlation distance is used as a similarity measure. Each cell of the SOM presents three values: the number of data items that fall into the cell, the minimum and maximum values obtained in Step 4. The cell with the highest number of data items is considered as the center of the first cluster, denoted as $Center^1_{3,1}$. The values adjacent to the arrows show the correlation distance between two cells of the SOM. A black arrow indicates a distance within the range of the minimum and maximum values, while a red arrow signifies a distance outside this range. The first cluster (blue) is formed by expanding from the center $Center^1_{3,1}$ and moving to the top left corner of the SOM ($Center^1_{3,1} \rightarrow Cell_{2,1}, Cell_{2,2}, Cell_{2,1} \rightarrow Cell_{1,1}, Cell_{1,2}$). Not all distances are depicted; for cells already assigned to a cluster, alternative expansion routes are omitted. Once the first cluster is formed, the remaining two clusters are similarly identified, expanding from their respective centers ($Center^2_{2,4} \rightarrow Cell_{1,4}, Cell_{3,4}$; $Center^3_{3,6} \rightarrow Cell_{2,5}, Cell_{2,6}, Cell_{3,5}, Cell_{4,5}, Cell_{4,6} \rightarrow Cell_{5,5}, Cell_{5,6}$).

All cells of the SOM that were not previously assigned to a cluster from the centers during the expansion phase are allocated to the most similar center in Step 9 of Algorithm 1. The final result of this process is presented in Fig. 4(b). In this representation,

Algorithm 1. Determination of cluster in SOM.

Input: X – dataset analyzded, trained SOM, k_a – row number of the SOM, k_b – column number of the SOM, C – the desired number of the clusters (maximum clusters).

Step 1: Calculate pairwise similarities between data items in X that fall into the same SOM cell $Cell_{i,j}$, where $i = 1, \ldots, k_a$, $j = 1, \ldots, k_b$.

Step 2: Identify the minimum $Min_{i,j}$ and maximum $Max_{i,j}$ of pairwise similarity values for each cell $Cell_{i,j}$.

Step 3: Compute the overall average similarity $Avg^n_{i,j}$ between all data items in cell $Cell_{i,j}$ and its first-rank neighboring cells $Cell^n_{i,j}$, where $n = 1, \ldots, l$. In rectangular SOM topology, each cell has a maximum of 8 neighboring cells ($l = 8$), and in hexagonal topology, $l = 6$.

Step 4: Determine the first cluster center $Center^c_{i,j}$, where $c \in \{1, \ldots, C\}$, as the $Cell_{i,j}$ with the highest number of data items. If multiple cells have the same highest number of data items, select one randomly.

Step 5: Assign first-rank neighboring cells $Cell^n_{i,j}$ to cluster c based on similarity to the center $Center^c_{i,j}$ (in the case of rectangular SOM topology):

FOR $n = 1$ **TO** l

  IF $((Min_{i,j} \leqslant Avg^n_{i,j})$ **AND** $(Avg^n_{i,j} \leqslant Max_{i,j}))$

    $Cell^n_{i,j} \rightarrow$ assigned to the c cluster.

  ELSE

    $Cell^n_{i,j} \rightarrow$ marked as a free cell $Free_{i,j}$.

  END

ELSE

Step 6: Each cell $Cell^n_{i,j}$, assigned to cluster c, expands the cluster by trying to find similar first-rank neighboring cells. Repeat **Step 5** for cluster expansion. The process stops when no more cells can be assigned.

Step 7: Find the second cluster center $Center^c_{i,j}$ with the highest number of data items and furthest from the first cluster center. Select randomly if multiple cells meet the criteria.

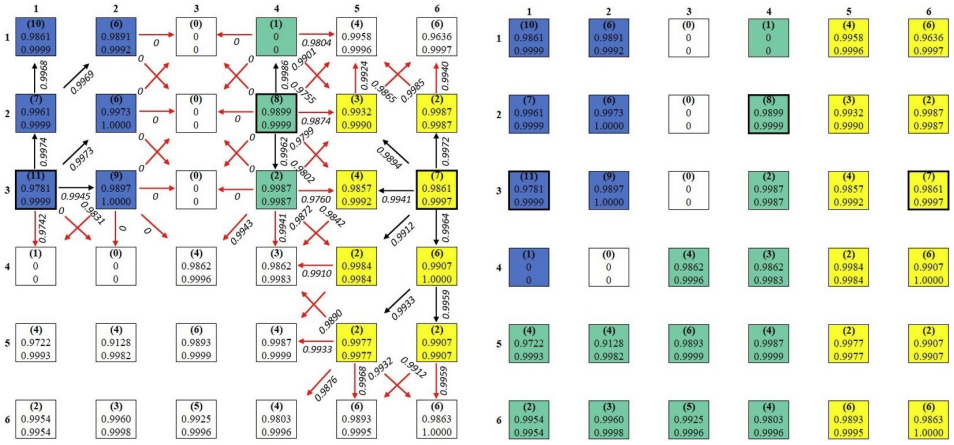
Step 8: Repeat **Steps 5–7** until all cells are assigned to a cluster or marked as free. In case, there are no free cells left and the desired number of clusters has not been found, the algorithm stops with the possible maximum number of clusters of the analyzed dataset.

Step 9: Compute the overall average similarity $Free^c_{AVG}$ between all data items in cell $Free_{i,j}$ and each center $Center^c_{i,j}$ of the clusters data items, where $c \in \{1, \ldots, C\}$.

Step 10: The cells $Free_{i,j}$ are assigned to the cluster c whose value of $Free^c_{AVG}$ is the smallest (most similar).

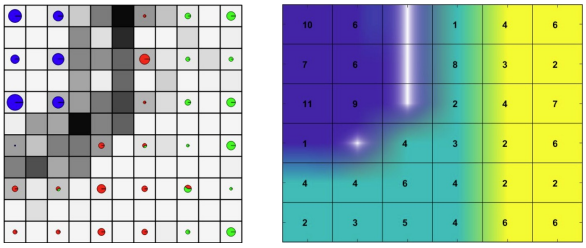
distinct colors correspond to different clusters, while the white color indicates cells with no data items. The clustering results obtained by our proposed approach have been compared to those obtained using the Orange Data Mining tool [4], where the SOM is implemented as well. To keep the same structure of the SOM produced by the Orange Data Mining tool, first, the SOM has been trained, and the weights of the SOM derived by the tool have been saved and used in our proposed approach. In Fig. 5, the SOM trained by using the Orange Data Mining tool is displayed. Figure 5(a) illustrates the SOM visualization within the tool, while Fig. 5(b) shows the SOM visualization as obtained by our proposed approach. To comprehend the clustering results from the Orange Data Mining tool, pie charts have been colored (see Fig. 5(a)).

It is relevant to note, however, that the data labels in the SOM training process are not used and do not influence the clustering results, no matter Orange Data Mining tool or our proposed approach is used. The labels are only used for data visualization in the obtained



(a) Steps 1–8 (b) Steps 9–10

Figure 4. The 6×6 SOM: the cluster determination using the proposed approach.

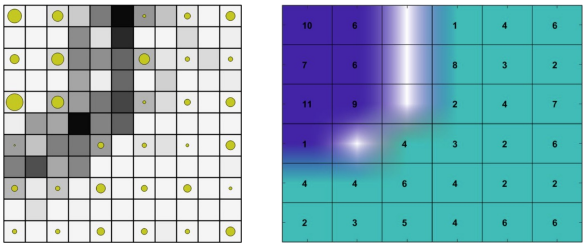


(a) (b)

Figure 5. The 6×6 SOMs obtained using the iris dataset: (a) visualized by u-matrix (using label colors) in Orange Data Mining tool; (b) visualized by the proposed approach with correlation distance and choosing three clusters.

SOM. We can see in Fig. 5 a that the pie charts represent data items. The size of each pie chart corresponds to the number of data items that fall in the same cell. Different colors in these charts signify different clusters. Considering u-matrix, we can conclude that the red and green cells indicate one cluster, while the blue cells denote a separate cluster.

In contrast, our proposed approach specifies that the data be assigned to three clusters (see Fig. 5(b)), where the numbers in each SOM cell represent the number of data items that fall into the cell. In this visualization of the SOM, we use linear color interpolation to differentiate the cluster smoothly. The main reason for choosing this visualization technique is to present a smooth transition from one cluster to another. As mentioned before, the white color corresponds to the absence of data items in the cells. This indicates the largest difference between clusters. A comparison of the SOMs presented in Fig. 5 reveals that by using our proposed approach, some data items at the bottom of the SOM have been assigned to the same group, although this is not the case. At the bottom of Fig. 5(a), there are some cells of the SOM where data items colored red and green fall



(a) (b)

Figure 6. The 6×6 SOMs obtained using the iris dataset: (a) visualized by u-matrix (without label colors) in Orange Data Mining tool; (b) visualized by the proposed approach with correlation distance and choosing two clusters.

into the same cell. By using our proposed approach (Fig. 5(b)), these data items are assigned to the same cluster. The main reason is that the proposed approach tries to form the desired number of clusters.

Suppose we analyze the iris dataset without labels. In this case (Fig. 6(a)), we see only one-color circles. Based on the u-matrix visualization, it could determine two clusters: one in the top-left corner and the other comprising the remaining data, separated from the former by dark cells. The results obtained by our proposed approach are shown in Fig. 6(b), with a choice of two clusters. This visualization technique reveals a predefined number of clusters, regardless of whether labeled or unlabeled data are analyzed.

5 Experimental investigation

To assess the performance of the proposed approach, labeled datasets with different characteristics were chosen (see Subsection 5.1). To evaluate the clustering results, the obtained clusters were compared with class labels of the original data. During the experimental investigation, the different similarity distances, described in Subsection 3.2, were used in the proposed approach to obtain the clusters. Furthermore, the results obtained by the proposed approach were compared with those obtained by k-means and hierarchical clustering algorithms (see Subsection 5.2). In this investigation, the influence of the k-means and the hierarchical clustering parameters has not been analyzed. In the k-means algorithm, the random center initialization is used. In the case of hierarchical clustering, the Euclidean distance and average linkage parameters are used.

5.1 Datasets analyzed

The experimental investigation was carried out using four labeled datasets. These datasets were selected to cover a range of different characteristics, including different numbers of classes, numbers of attributes, sizes of datasets, and types of attributes (such as real numbers, integers, categorical, and text).

It is important to remember that data labels are not used in clustering algorithms. Our experimental investigation assumes that the data items assigned to the same class have

Table 1. Characteristics of the datasets analyzed

Number of data: Items; attributes; classes	Description
Dataset: Iris [7]	
150; 4; 3	The iris dataset, as the most popular dataset from the UC Irvine Machine Learning Repository, is widely used to illustrate various approaches. The dataset size is small, the attributes are real numbers, and the dataset is class-balanced. Many studies have shown that one class of the dataset is always separated from the others. Using this dataset in a classification task yields an accuracy of approximately 98–100%.
Dataset: Glass [9]	
219; 9; 6	The glass dataset is unbalanced with real attributes. Using the dataset in a classification task, the accuracy is around 72–80%, depending on the classification model. This variability in accuracy indicates that the dataset has a significant overlap between the different classes, suggesting that the clustering results obtained may also be ambiguous.
Dataset: Mushrooms [32]	
8416; 22; 2	The mushroom dataset, the largest selected for our research, is distinct due to it having categorical attributes. These attributes were first converted into numerical values using a label encoding method. The dataset differs from the iris and glass datasets in that the attributes usually acquire just a few values that are not significantly different from each other. When using this dataset to solve a classification task, the classification accuracy is approximately 93–100%.
Dataset: Elections [31]	
6444; 382; 2	The election dataset consists of short text data collected from Twitter during the 2016 US presidential election. This textual dataset was converted into document vectors using the multilanguage BERT transformers model, resulting in 384 variables per document. Compared to the other datasets analyzed, this dataset is larger in terms of both the number of data items and the number of attributes. When using this dataset for classification tasks, an accuracy of around 80–90% is typically achieved.

to form a cluster. In this case, the data labels are only used to calculate how many data items are correctly assigned to the right clusters. As mentioned before, the classification tasks are not suitable when the unlabeled datasets are analyzed, therefore, in this case, the data clustering is performed. Table 1 provides the characteristics and descriptions of the selected datasets.

5.2 Validation of experimental results

Each analyzed dataset was clustered using three algorithms into as many clusters as the number of classes in the original dataset. In an experimental investigation, the results of the proposed approach were compared to those of the k-means and hierarchical clustering algorithms, as these algorithms allow the selection of the desired number of clusters. To evaluate the performance of each algorithm, the ratio was calculated for each class by dividing the number of data items allocated to the respective cluster by the total number of data items in that class. The main aim of the evaluation was to find whether the obtained clusters using the algorithms correspond to the true classes of the data analyzed. To explain the validation of the experimental results, a simple example is presented. Suppose we have a subset of the iris dataset, where five data items of each different classes, Iris

Figure 7. The 6×6 SOMs obtained using a subset of the iris dataset: (a) visualized (using label colors) in the Orange Data Mining tool; (b) visualized by the proposed approach with correlation distance and choosing three clusters.

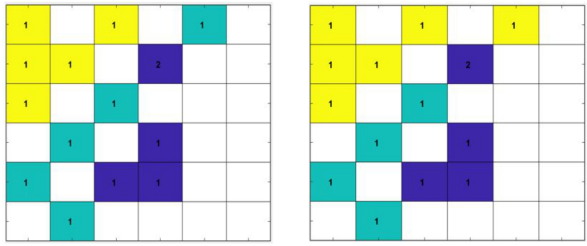
Table 2. Evaluation of the results of clustering the subset of the iris dataset obtained using the proposed approach.

Similarity distance	Ratio for Class 1	Ratio for Class 2	Ratio for Class 3	Overall ratio
Correlation distance	5/5 (1)	4/5 (0.8)	5/5 (1)	14/15 (0.93)
Cosine distance	5/5 (1)	5/5 (1)	5/5 (1)	**15/15 (1)**
Euclidean distance	5/5 (1)	4/5 (0.8)	5/5 (1)	14/15 (0.93)
Jaccard distance	5/5 (1)	2/5 (0.4)	3/5 (0.6)	10/15 (0.67)
Spearman distance	5/5 (1)	5/5 (1)	1/5 (0.2)	11/15 (0.73)

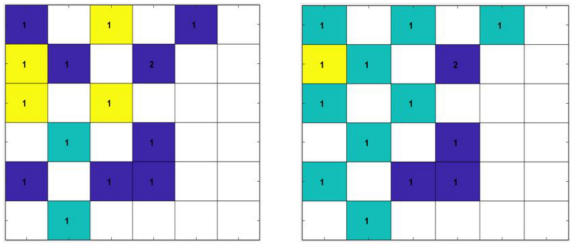
Setosa (Class 1), Iris Versicolar (Class 2), and Iris Virginica (Class 3), are randomly selected. First, the SOM has been trained and visualized using the Orange Data Mining tool (see Fig. 7(a)). The obtained weights of the SOM have been saved and used in our proposed approach to determine cluster boundaries. To evaluate the quality of the proposed approach, the desired number of clusters C has to be chosen as 3 since data from three classes are analyzed. To assess the quality of the proposed method, the desired number of clusters has to be chosen as 3, since data from three classes are analyzed. The SOM visualized by the proposed approach using the correlation distance when the clusters are formed is presented in Fig. 7(b).

We can see that there is only one data item mismatch in the obtained cluster compared to the original dataset class. Fig. 7(a) shows a red circle in the top right corner of the SOM, while Fig. 7(b) shows a yellow cell in the same place, indicating a different cluster from the one in Fig. 7(a). The strength of the proposed approach lies in its flexibility to employ different similarity distances in cluster formation, thereby enabling the selection of the most appropriate one for the problem to be solved. The clustering results using various similarity distances are presented in Fig. 8. We can see that, using the cosine similarity distance, the clusters obtained by the proposed approach correspond to all the true classes of the dataset, as presented in Fig. 7(a). Additionally, the ratio for each class is calculated and presented in Table 2. The results show that the usage of the cosine distance provides a perfect match of clusters to classes.

The evaluation of clustering results using the whole iris dataset is summarized in Table 3. As we can see, almost every clustering algorithm accurately detects the cluster



(a) Cosine distance (b) Euclidean distance



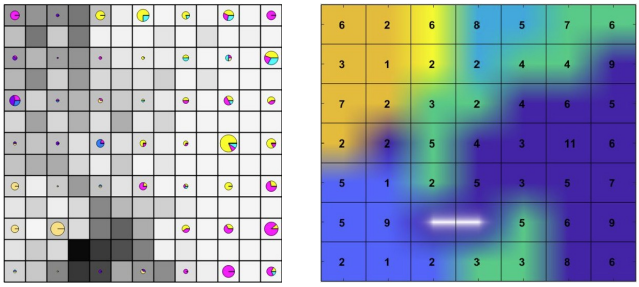
(c) Jaccard distance (d) Spearman distance

Figure 8. The 6×6 SOMs obtained using the subset of the iris dataset, visualized by the proposed approach with different similarity distances and choosing three clusters.

Table 3. Evaluation of the results of clustering the iris dataset.

Algorithm/Distance	Ratio for Class 1	Ratio for Class 2	Ratio for Class 3	Overall ratio
k-means clustering	0/50 (1)	48/50 (0.96)	36/50 (0.72)	134/150 (0.89)
Hierarchical clustering	0/50 (1)	0/50 (1)	2/50 (0.04)	102/150 (0.68)
Approach proposed (Euclidean)	34/50 (0.68)	0/50 (0)	50/50 (1)	84/150 (0.56)
Approach proposed (Cosine)	0/50 (1)	29/50 (0.58)	49/50 (0.98)	128/150 (0.85)
Approach proposed (Correlation)	0/50 (1)	48/50 (0.96)	48/50 (0.96)	**146/150 (0.97)**
Approach proposed (Jaccard)	0/50 (1)	9/50 (0.18)	6/50 (0.12)	65/150 (0.43)
Approach proposed (Spearman)	Three clusters have not been detected ($c = 2$)			

corresponding to the first class, while other clusters are detected differently. In the case of the proposed approach, the best results are obtained using the correlation distance as the similarity measure between SOM cells with an overall ratio of 0.97, and only four cluster and class mismatches were obtained. The worst results are obtained when employing the Euclidean distance. The results of the overall ratio obtained using the cosine distance are also encouraging and equal to 0.85. When using Spearman distance, the proposed approach found only two clusters, so ratios were not calculated. Clustering using the k-means algorithm also resulted in a high ratio of 0.89. The hierarchical clustering has an overall ratio of 0.68, and the main reason why the third class was not correctly detected is that it was similar to the second class. The SOM visualization using the correlation distance as a similarity distance is depicted in Fig. 5. As we can see, the clusters obtained by our proposed approach correspond to the data classes (see Fig. 5(b)) marked in different colors in the SOM visualized by the Orange Data Mining tool (see Fig. 5(a)). It is



(a) (b)

Figure 9. The 7×7 SOMs obtained using the glass dataset: (a) Orange Data Mining tool, SOM visualized by u-matrix; (b) visualized by the proposed approach with cosine distance and choosing six clusters.

Table 4. Evaluation of the results of clustering the glass dataset.

Algorithm/Distance	Ratio for Class 1	Ratio for Class 2	Ratio for Class 3	Ratio for Class 4	Ratio for Class 5	Ratio for Class 6	Overall ratio
k-means clustering	48/70 (0.69)	59/76 (0.78)	3/17 (0.18)	7/13 (0.53)	0/9 (0)	27/29 (0.93)	**144/219 (0.67)**
Hierarchical clustering	70/70 (1)	1/76 (0.01)	0/17 (0)	2/13 (0.15)	1/9 (0.11)	2/29 (0.06)	76/219 (0.35)
Approach proposed (Euclidean)	Six clusters have not been detected ($c = 4$)						
Approach proposed (Cosine)	43/70 (0.61)	16/76 (0.21)	3/17 (0.18)	8/13 (0.62)	0/9 (0)	22/29 (0.76)	92/219 (0.43)
Approach proposed (Correlation)	43/70 (0.61)	16/76 (0.21)	3/17 (0.18)	8/13 (0.62)	0/9 (0)	22/29 (0.76)	92/219 (0.43)
Approach proposed (Jaccard)	Six clusters have not been detected ($c = 5$)						
Approach proposed (Spearman)	32/70 (0.46)	7/76 (0.09)	3/17 (0.18)	8/13 (0.62)	0/9 (0)	22/29 (0.76)	72/219 (0.34)

necessary to mention here again that our algorithm does not require labeled data, while the SOM visualization using the Orange Data Mining tool needs to provide labeled data.

When analyzing the glass dataset, the highest clustering performance was achieved using the k-means algorithm, with an overall ratio of 0.67 (see Table 4). As mentioned, this dataset is unbalanced, and even for classification tasks the accuracy is usually not very high. Using the SOM with Jaccard and Euclidean distances did not detect the desired number of clusters. Specifically, the Jaccard distance resulted in the dataset being divided into five clusters, whereas the utilization of the Euclidean distance led to its division into four clusters. The lowest rates were obtained using hierarchical clustering and the proposed approach with Spearman distance, each yielding 0.35 and 0.34, respectively. Both cosine distance and correlation distance produced identical cluster detection ratios of 0.43.

The SOM visualization obtained by applying the proposed approach to the glass dataset is illustrated in Fig. 9. As we can see in the visualization obtained with the Orange Data Mining tool (see Fig. 9(a)), the data items of different classes overlap and are mostly spread over the whole SOM. It is difficult to unambiguously detect clusters and their boundaries. The majority of data items of the same class are placed in the bottom left

Table 5. Evaluation of the results of clustering the mushroom dataset.

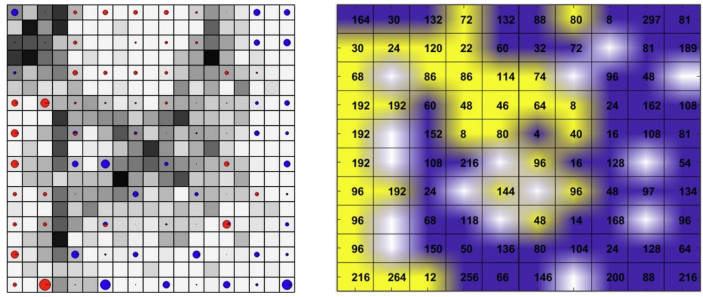
Algorithm/Distance	Ratio for Class	Ratio for Class 2	Overall ratio
k-means clustering	4264/4488 (0.95)	1757/3928 (0.45)	6021/8416 (0.71)
Hierarchical clustering	4488/4488 (1)	29/3928 (0)	4517/8416 (0.53)
Approach proposed (Euclidean)	4216/4488 (0.94)	2006/3928 (0.51)	6222/8416 (0.73)
Approach proposed (Cosine)	2279/4488 (0.51)	3242/3928 (0.82)	5521/8416 (0.65)
Approach proposed (Correlation)	2231/4488 (0.49)	3274/3928 (0.83)	5505/8416 (0.65)
Approach proposed (Jaccard)	3896/4488 (0.86)	2602/3928 (0.66)	**6498/8416 (0.77)**
Approach proposed (Spearman)	2983/4488 (0.66)	2960/3928 (0.75)	5943/8416 (0.70)

corner of the SOM. Also, there are many data items whose classes are represented by the purple color and are placed in the bottom right corner. Six clusters are obtained using the proposed approach (Fig. 9(b)), where the biggest cluster starts (dark blue color) from the middle right side (cell with the number 11) of the SOM and is arranged from the bottom right corner to the top right corner of the SOM. One of the big clusters (blue color) is placed in the bottom left corner of the SOM, and another, in the top left side (orange color). The cluster with a smaller number of data items is placed in the top middle side of the SOM (light blue color). The last cluster is split into two parts, and the majority side of the cluster is arranged in the middle of the SOM.

In the analysis of the mushroom dataset (see Table 5), the highest clustering performance with a ratio of 0.77 was obtained by applying the Jaccard distance in the proposed approach. Slightly lower ratios of 0.73 were obtained using the Euclidean distance and of 0.70 using the Spearman distance. The clustering outcomes from the k-means algorithm were also comparable, achieving a ratio of 0.71. The lowest ratios of 0.65 were obtained using the cosine distance and the correlation distance, but they were not too different from the highest ratio obtained. In the cases of SOM using Euclidean distance and k-means clustering, we can see that almost all data items in the first cluster correspond to the original class, while in the second cluster just around half of the data items of the second class. This means that many data items from the second class were assigned to the first cluster. Not only the ratio, but also the distribution of the class ratio needs to be observed, the lowest ratio was obtained with hierarchical clustering (0.53). The majority of data items formed one cluster, and only 29 items formed a second cluster.

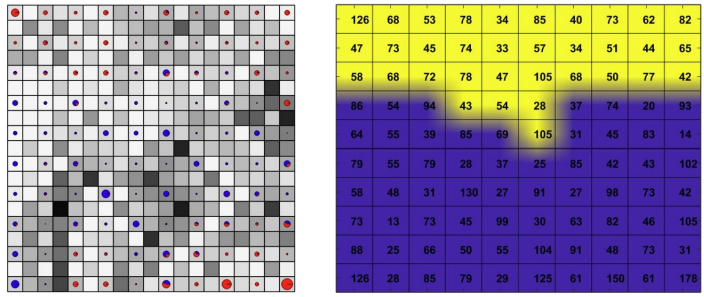
The visualization results in Fig. 10(a) show that the data items from one class (represented in blue) are more distributed on the right side of the SOM, while the items from the second class (shown in red) are scattered from the top to the bottom of the SOM. Notably, a subset of data items from the first class can be found in the top left corner of the SOM. A comparison of the visualization results derived from the Orange Data Mining tool (Fig. 10(a)) and those produced by the proposed approach shows (Fig. 10(b)) that the obtained clusters correspond to the data classes. The data items in red in the top middle side of the SOM in Fig. 10(a), have been assigned to a different cluster than their original class using the proposed approach. The same applies to some of the blue-colored data items in the middle of the SOM that have fallen in opposite cluster.

The analysis of the election dataset shows (see Table 6) that almost all similarity measures used in the proposed approach give identical results. The ratios for the first and



(a) (b)

Figure 10. The 10×10 SOMs obtained using the mushrooms dataset: (a) Orange Data Mining tool, SOM visualized by u-matrix; (b) visualized by the proposed approach with Jaccard distance and choosing two clusters.



(a) (b)

Figure 11. The 10×10 SOMs obtained using the election dataset: (a) Orange Data Mining tool, SOM visualized by u-matrix; (b) visualized by the proposed approach with cosine distance and choosing two clusters.

second classes are the same, and the overall ratio is 0.64, except for the use of the Jaccard similarity distance (0.55). The main reason for this may be that the transformation of the text data into vectors using the multilanguage BERT transformer model results in similar values for the acquired attributes. Therefore, the calculated similarity distances become almost the same and do not have a significant influence on the proposed approach. The k-means clustering yielded a lower value of the overall ratio compared to the SOM approaches and is equal to 0.44. The slightly higher value of the overall ratio compared to the k-means results was obtained using hierarchical clustering at 0.50, but in this case, the majority of the data was grouped into one large cluster, with only one item forming a separate cluster. So, the ratio indicates that the results of clustering are worst. The visual results of the SOM are presented in Fig. 11. The majority of the first-class (blue) data items are placed in the middle of the SOM. The second-class data items are distributed at the top and bottom of the SOM. Even if the dataset labels are presented in the SOM (Fig. 11(a)) it is hard to say exactly where the boundaries of each cluster are. Our proposed approach split data into two parts (Fig. 11(b)) by taking the top data in the SOM into one cluster (yellow) and going from the middle to bottom another cluster was formed (blue).

Table 6. Evaluation of the results of clustering the election dataset.

Algorithm/Distance	Ratio for Class	Ratio for Class 2	Overall ratio
k-means clustering	1828/3226 (0.57)	1039/3218 (0.32)	2867/6444 (0.44)
Hierarchical clustering	3226/3226 (1)	1/3218 (0)	3227/6444 (0.50)
Approach proposed (Euclidean)	2631/3226 (0.82)	1524/3218 (0.47)	**3227/6444 (0.64)**
Approach proposed (Cosine)	2631/3226 (0.82)	1524/3218 (0.47)	**3227/6444 (0.64)**
Approach proposed (Correlation)	2631/3226 (0.82)	1524/3218 (0.47)	**3227/6444 (0.64)**
Approach proposed (Jaccard)	1545/3226 (0.48)	2019/3218 (0.63)	3573/6444 (0.55)
Approach proposed (Spearman)	2631/3226 (0.82)	1524/3218 (0.47)	**227/6444 (0.64)**

6 Discussion

The comparative experimental investigation has been performed using four datasets, and the usability of the proposed approach to determine clusters in the visualization of the SOM has been experimentally proved. We must admit that the results can vary using the different datasets, learning parameters of k-means, hierarchical clustering, and SOM. It is not possible to take into account all possible options. The datasets have been chosen with different characteristics, but in the future, more datasets could be investigated. Also, more attention could be paid to the influence of learning parameters on clustering results. Our obtained results cannot be compared to other approaches reviewed in related works, because implementations of all the proposed approaches are not publicly available. In addition, many of them use the labeled data, and clusters are formed according to the known labels. The main problem with evaluating the clustering results is that there are not many well-known measures to evaluate the quality of the clusters, and usually, the clustering results are evaluated manually by the researcher. In this way, SOMs have the advantage compared to other clustering algorithms. During the experimental investigation, it was observed that the similarity measures used in the proposed approach have an influence on the clustering results.

7 Conclusion

Clustering unlabeled data is a complex task because it is difficult to detect groups unambiguously without knowing the context and output of the analyzed data. There exist many clustering algorithms, so it is also a dilemma of which algorithm needs to be selected and whether the results obtained are trustworthy. Unlike classification tasks, data clustering presents more challenges in many aspects. Nevertheless, clustering algorithms are highly used in the scientific area, and continuous improvements in data clustering have been developed. The main aim of our proposed approach is to help determine the desired number of clusters in the visualization of SOMs, which has not been implemented in related work. In addition, the boundaries of the clusters are determined and presented in the SOM.

The performance of the proposed approach to determine the clusters in the visualization of SOMs has been experimentally investigated. The proposed approach is based on the calculation of the similarity distance between the data in cells of the SOM, so

various combinations of distances and the SOM have been investigated. The effectiveness of the approach is hard to estimate because there are no measures that unequivocally could determine whether the clusters are formed well or not. Therefore, the number of correctly assigned labeled data to the different clusters has been calculated manually. The results have been validated using four datasets with different characteristics. Comparative analysis has shown that using the cosine distance and correlation distance the performance is the highest because, in all cases of analyzed datasets, the desired number of clusters has been found. In some cases, the value of the ratio was lower compared to the results of other similarity measures, but at the same time, the ratio of each class was distributed, and there was no dominant cluster. Visualizing the SOM using these distances showed that the visualization results of the Orange Data Mining tool using the u-matrix have been improved, and the boundaries of the clusters are formed more clearly with the desired number of clusters. Comparing results of hierarchical and k-means clustering has shown that in many cases, only k-means clustering forms the clusters in the high ratio, but this algorithm does not have visualization to observe the obtained clusters visually.

In the future, deeper research is needed using more datasets to prove the performance of the proposed approach compared to other clustering algorithms. In this paper, we used well-known datasets to show the possibilities of the proposed approach, but the experiments could be carried out with more significant data in today's data analysis. Also, the influence of the training parameters on the results of the proposed approach could be analyzed, for example, focusing on the size of the SOM and different topologies of the SOM. Additionally, training parameters could also improve the results.

Author contributions. All authors (P.S. and O.K.) have contributed as follows: methodology, P.S., O.K.; formal analysis, P.S., O.K.; validation, P.S., O.K.; writing – original draft preparation, P.S., O.K.; writing – review and editing, P.S., O.K. All authors have read and approved the published version of the manuscript.

Conflicts of interest. The authors declare no conflicts of interest.

References

1. S. Aly, S. Almotairi, Deep convolutional self-organizing map network for robust handwritten digit recognition, *IEEE Access*, **8**:107035–107045, 2020, https://doi.org/10.1109/ACCESS.2020.3000829.
2. D. Breskuvienė, G. Dzemyda, Enhancing credit card fraud detection: Highly imbalanced data case, *J. Big Data*, **11**(1):182, 2024, https://doi.org/10.1186/s40537-024-01059-5.
3. K. Chowdhury, D. Chaudhuri, A.K. Pal, An entropy-based initialization method of K-means clustering on the optimal number of clusters, *Neural Comput. Appl.*, **33**:6965–6982, 2021, https://doi.org/10.1007/s00521-020-05471-9.
4. J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, B. Zupan, Orange: Data mining toolbox in Python, *J. Mach. Learn. Res.*, **14**(1):2349–2353, 2013.

5. G. Dzemyda, O. Kurasova, Comparative analysis of the graphical result presentation in the SOM software, *Informatica*, **13**(3):275–286, 2002, https://doi.org/10.3233/INF-2002-13302.

6. E.A. Fernandez, M. Balzarini, Improving cluster visualization in self-organizing maps: Application in gene expression data analysis, *Comput. Biol. Med.*, **37**(12):1677–1689, 2007, https://doi.org/10.1016/j.compbiomed.2007.04.003.

7. R.A. Fisher, The use of multiple measurements in taxonomic problems, *Annals Eugenics*, **7**(2): 179–188, 1936, https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

8. K. Gao, G. Mei, F. Piccialli, S. Cuomo, J. Tu, Z. Huo, Julia language in machine learning: Algorithms, applications, and open issues, *Comput. Sci. Rev.*, **37**:100254, 2020, https://doi.org/10.1016/j.cosrev.2020.100254.

9. B. German, Glass identification, UCI Machine Learning Repository, 1987, https://doi.org/10.24432/C5WW2P.

10. N. Gholizadeh, H. Saadatfar, N. Hanafi, K-DBSCAN: An improved DBSCAN algorithm for big data, *J. Supercomput.*, **77**:6214–6235, 2021, https://doi.org/10.1007/s11227-020-03524-3.

11. P. Govender, V. Sivakumar, Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019), *Atmos. Pollut. Res.*, **11**(1):40–56, 2020, https://doi.org/10.1016/j.apr.2019.09.009.

12. M.K. Gupta, P. Chandra, A comprehensive survey of data mining, *Int. J. Inf. Technol.*, **12**(4):1243–1257, 2020, https://doi.org/10.1007/s41870-020-00427-7.

13. A. Javed, D.M. Rizzo, B.S. Lee, R. Gramling, Somtimes: Self organizing maps for time series clustering and its application to serious illness conversations, *Data Min. Knowl. Discovery*, pp. 1–27, 2023, https://doi.org/10.1007/s10618-023-00979-9.

14. H. Jin, W.-H. Shum, K.-S. Leung, M.-L. Wong, Expanding self-organizing map for data visualization and cluster analysis, *Inf. Sci.*, **163**(1–3):157–173, 2004, https://doi.org/10.1016/j.ins.2003.03.020.

15. T. Kohonen, *Self-Organization and Associative Memory*, Springer Ser. Inf. Sci., Vol. 8, Springer, Berlin, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-88163-3.

16. K. Li, K. Sward, H. Deng, J. Morrison, R. Habre, M. Franklin, Y.-Y. Chiang, J.L. Ambite, J.P. Wilson, S.P. Eckel, Using dynamic time warping self-organizing maps to characterize diurnal patterns in environmental exposures, *Sci. Rep.*, **11**(1):24052, 2021, https://doi.org/10.1038/s41598-021-03515-1.

17. S.-S. Li, An improved dbscan algorithm based on the neighbor similarity and fast nearest neighbor query, *IEEE Access*, **8**:47468–47476, 2020, https://doi.org/10.1109/ACCESS.2020.2972034.

18. D. Merkl, A. Rauber, Alternative ways for cluster visualization in self-organizing maps, in *Proceedings of the Workshop on Self-Organizing Maps (WSOM'97)*, Technical University of Vienna, 1997, pp. 106–111.

19. Ş Öztürk, Comparison of pairwise similarity distance methods for effective hashing, *IOP Conf. Ser.: Mater. Sci. Eng.*, **1099**:012072, 2021, https://doi.org/10.1088/1757-899X/1099/1/012072.

20. S.-L. Shieh, I.-En. Liao, A new approach for data clustering and visualization using self-organizing maps, *Expert Syst. Appl.*, **39**(15):11924–11933, 2012, https://doi.org/10.1016/j.eswa.2012.02.181.

21. K.S. Sreedhar, M. Madheswaran, B.A. Vinutha, Manjunatha S.H., K.V. Charan, A brief survey of unsupervised agglomerative hierarchical clustering schemes, *Int. J. Eng. Technol.*, **8**(1):29–37, 2019.

22. P. Stefanovič, O. Kurasova, Influence of learning rates and neighboring functions on self-organizing maps, in J. Laaksonen, T. Honkela (Eds.), *Advances in Self-Organizing Maps. 8th International Workshop, WSOM 2011, Espoo, Finland, June 13–15, 2011. Proceedings*, Lect. Notes Comput. Sci., Vol. 6731, Springer, Berlin, Heidelberg, 2011, pp. 141–150, https://doi.org/10.1007/978-3-642-21566-7_14.

23. P. Stefanovič, O. Kurasova, Creation of text document matrices and visualization by self-organizing map, *Inf. Technol. Control*, **43**(1):37–46, 2014.

24. P. Stefanovic, O. Kurasova, Investigation on learning parameters of self-organizing maps, *Balt. J. Mod. Comput.*, **2**(2):45, 2014.

25. P. Stefanovič, O. Kurasova, Outlier detection in self-organizing maps and their quality estimation, *Neural Network World*, **28**(2):105–117, 2018, https://doi.org/10.14311/NNW.2018.28.006.

26. P. Stefanovič, O. Kurasova, Approach for multi-label text data class verification and adjustment based on self-organizing map and latent semantic analysis, *Informatica*, **33**(1):109–130, 2022, https://doi.org/10.15388/22-INFOR473.

27. P. Stefanovič, S. Ramanauskaitė, Travel direction recommendation model based on photos of user social network profile, *IEEE Access*, **11**:28252–28262, 2023, https://doi.org/10.1109/ACCESS.2023.3260103.

28. A. Ultsch, H.P. Siemon, *Exploratory Data Analysis: Using Kohonen Networks on Transputers*, Dekanat Informatik, Universität Dortmund, 1989.

29. K. Yoshioka, H. Dozono, The classification of the documents based on Word2Vec and 2-layer self organizing maps, *Int. J. Mach. Learn. Comput.*, **8**(3):252–255, 2018, https://doi.org/10.18178/ijmlc.2018.8.3.695.

30. S. Zhou, Z. Xu, F. Liu, Method for determining the optimal number of clusters based on agglomerative hierarchical clustering, *IEEE Trans. Neural Networks Learn. Syst.*, **28**(12):3007–3017, 2016, https://doi.org/10.1109/TNNLS.2016.2608001.

31. Election tweet, Kaggle, 2016. https://www.kaggle.com/datasets/benhamner/clinton-trump-tweets

32. Mushroom, UCI Machine Learning Repository, 1987, https://doi.org/10.24432/C5959T.

33. Pairwise distance between pairs of observations, MATLAB version: 9.13.0 (R2022b), The MathWorks, 2022, https://se.mathworks.com/help/stats/pdist.html.