Journal of Data Science 0 (0), 1–19
??? 2026
DOI: 10.6339/26-JDS1226
Computing in Data Science

Enhance Supervised Self-Organization Clustering by Utilizing Unsupervised Learning Embeddings on Discrete Data

Qiang Fu[1] and Yuefeng Li[1,*]

[1]*School of Computer Science, Queensland University of Technology, Brisbane, Australia*

Abstract

The self-organizing map (SOM) is an unsupervised, competitive learning neural network that projects high-dimensional data onto a low-dimensional grid, effectively showcasing the topological relationships within the original dataset. However, the conventional SOM training algorithm is restricted to numeric data. Categorical data typically needs to be converted into binary format before SOM training, which can lead to the loss of crucial similarity information between categorical values. As a result, the trained SOM may not accurately reflect the true topological order. While a training data splitting method (TDSM) can help identify perfect representative neurons and enhance clustering outcomes, the training data itself often lacks sufficient information, such as data distribution, and can be uncertain and ambiguous. Even when perfect neurons are identified, further improvements in clustering results become challenging. This paper investigates the possibility of improving the performance of supervised TDSM SOM clustering by utilizing unsupervised self-organization granule encoding for discrete data. This approach to unsupervised learning is advantageous for uncovering uncertain and ambiguous information within discrete data, leading to a more effective topological representation of the training data.

Keywords *fuzzy set; granular computing; noncontinuous data clustering; self-organized granular encoding; training data splitting method*

1 Introduction

Human and animal brains are intricate, nonlinear, and parallel systems that comprise billions of neurons interconnected within numerous neural networks. Neurons are organized into networks that perform specific functions, such as sensory processing, motor control, and cognitive functions. These networks can adapt and reorganize based on experience and learning, a property known as neuroplasticity (Khacef et al., 2020). Biological neural networks serve as natural intelligent information processors. The artificial neural network (ANN), or simply neural network, is a machine learning method that evolved from simulating the human brain. ANNs employ a computational approach that depends on a large collection of artificial neurons, which are simplified versions of biological neurons. These networks comprise interconnected simple processing units, referred to as artificial neurons (or units/nodes), which aim to emulate the functions of biological neurons.

The self-organizing map (SOM) (Kohonen, 1990), a specific type of neural network paradigm, draws its inspiration from self-organizing and biological systems. Self-organizing systems are those that can adapt their internal structure and function in response to external conditions

*Corresponding author. Email: y2.li@qut.edu.au.

Received May 22, 2025; Accepted March 3, 2026

and stimuli (Miljković, 2017). SOM can perform unsupervised learning using training data, without requiring specified input/output pairs. It has been widely applied across various fields. As both a visualization tool and a data pattern classifier, SOM and its variants have found extensive applications in areas such as pattern recognition (Neagoe and Ropot, 2002; Wickramasinghe et al., 2019), medical diagnosis (Chushig-Muzo et al., 2020; Melin et al., 2020), anomaly detection (Licen et al., 2020; Li et al., 2021), and the detection of virus or worm attacks (Holloway, 2019; Qu et al., 2021).

The SOM training algorithm is primarily designed to work with numeric data. This limitation poses challenges when dealing with categorical data (Hsu, 2006). Before training a SOM using categorical data, it is common to convert these categories into a binary format, such as one-hot encoding (Ni et al., 2021) or similar techniques. Although this conversion enables the SOM to handle categorical data, a straightforward transformation approach can result in the loss of important similarity information among categorical values. For example, if categories are transformed without acknowledging their inherent relationships—such as treating "red," "blue," and "green" as equally distant—the resulting binary representation fails to effectively capture the nuances of their similarities and differences. This loss can hinder the SOM's ability to accurately reflect the relationships and topological order of the data during the training process. Even with the application of non-Euclidean distance measures (e.g., Hamming distance, Value Difference Metric) in SOM, effectively capturing the categorical structure remains a significant challenge. Some experiments are shown in Appendix Figure A1. The primary limitation of Hamming distance is that it treats all attribute values equally, without accounting for the statistical differences or distinct properties of the values (Jia et al., 2015). And the Value Difference Metric relies on a significant assumption: that all attributes are completely independent of one another, commonly referred to as the attribute independence assumption (Li et al., 2017).

Utilizing labeled data can significantly enhance the clustering results of a SOM by transitioning from unsupervised to supervised learning, as noted in (Riese et al., 2019; Lyu et al., 2024). Labeled data allows the SOM to refine its clustering by aligning the output with known categories. This means that the SOM can adjust its weights based on the labels, improving its ability to form meaningful clusters. Particularly, when employing an effective training data-splitting method (TDSM) (Fu et al., 2023) to identify the optimal neuron for each cluster, the original clustering performance can be significantly enhanced. However, since the validation metrics—such as purity, normalized mutual information (NMI), and adjusted Rand index (ARI)—all yield a value of 1 for each neuron-based cluster in the TDSM SOM, it becomes difficult to enhance the TDSM SOM clustering results without obtaining additional information from the training dataset. One key reason that SOM or TDSM-SOM cannot further improve clustering results for categorical data is the inherent challenges they face due to significant feature ambiguity and limited sample sizes. Specifically, this ambiguity occurs because the boundaries between ordinal and nominal categories are not always clearly defined. (Yang et al., 2020).

Fuzzy set theory has proven to be an effective approach for describing scenarios where the data is imprecise or ambiguous. It addresses these situations by assigning a degree of membership to indicate how much a particular object belongs to a set. From the perspective of granule computing (Bargiela and Pedrycz, 2022), once the degree of membership of each feature to the clusters has been identified, it indicates that the granularity of each feature has also been established. Consequently, the information contained within the attributes, as well as the relationships or influences between them, will be measured (Dai et al., 2024).

This paper enhances the performance of TDSM SOM clustering for discrete or categorical data by first revealing the mapping distribution relationships of each feature with the neurons,

and then identifying the perfect neurons for the transformed discrete or categorical data. The contributions of this paper are: 1) uncovering the fuzzy mapping between each discrete feature value and the self-organizing map neurons, and 2) improving the supervised TDSM SOM learning outcomes by leveraging unsupervised self-organizing granule encoding for discontinuous data.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of the necessary preliminaries. In Section 3, we introduce a novel fuzzy membership function designed to convert discrete data into a new embedding. Section 4 illustrates several numerical experiments. Finally, Section 5 presents a concise conclusion.

2 Related Studies

This section seeks to offer valuable insights into several key concepts originating from various areas of computational and artificial intelligence research. Notably, although these concepts arise from different fields, they share a unified goal: to enhance our understanding and representation of discrete data or categorical data while improving supervised TDSM SOM clustering through unsupervised learning techniques. By examining these concepts, we hope to foster constructive discussions and inspire innovative ideas within the realm of discrete data or categorical data representation.

2.1 SOM

The SOM is a type of ANN that employs unsupervised learning for clustering. It generates a low-dimensional, discretized representation of the input space from training samples, known as a map. This map is primarily utilized for data visualization, pattern detection (Nagar et al., 2022), data clustering (Bigdeli et al., 2022), and dimensionality reduction (Hidalgo et al., 2021). Simply put, SOM models are connected to the nodes of a regular grid, typically two-dimensional, with c rows and d columns. They are usually initialized with a $c \times d$ matrix, also known as the weight matrix (W), where each element of the matrix represents a node or neuron, denoted by a vector $N_{ij}, i \in [1, c],\ j \in [1, d]$. In other words, the SOM consists of $c \times d$ neurons, represented by W. Each neuron is characterized as a $1 \times f$ vector N_{ij}, with f indicating the number of features in the training data, as illustrated in Figure 1(a).

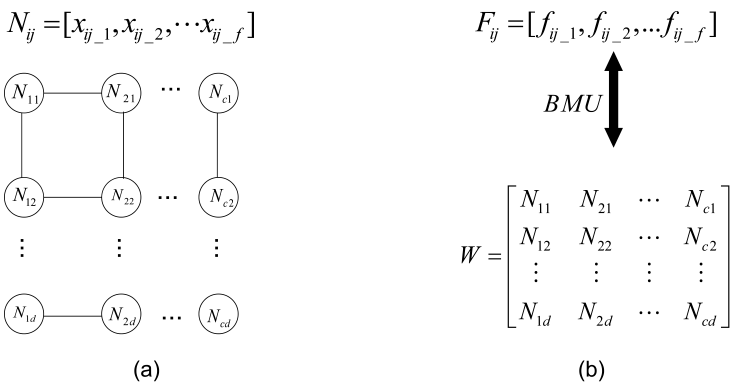

Figure 1: The fundamental framework of the self-organizing map involves the connection between the training sample F_{ij} and the weight matrix W.

The fundamental concept behind the training process of SOM is to adjust the weight matrix W so that it accurately represents the training dataset. This adjustment is particularly aimed at creating a meaningful topological representation of the input data (Lyu et al., 2024). To update the weight matrix W, each time a training sample is introduced into the training process, the matrix W will be modified by

$$W_{i+1} = W_i + G(u, v) \times l_r \times (X_i - W_i), \tag{1}$$

where i denotes number of iterations for the update, l_r is the learning rate, which controls how much W is adjusted during each iteration, and $G(u, v) = e^{d_{uv}/\sigma^2}$, which is the neighborhood function that defines the extent to which neighboring neurons are affected by the learning process. It helps maintain the input space's topological properties (Miljković, 2017). v represents the index of a neuron node within the network. u denotes the Best Match Unit (BMU) node. The BMU node refers to the neuron that exhibits the smallest Euclidean distance to the current input training sample F_{ij}, as illustrated in Figure 1 (b). The term d_{uv} refers to the Euclidean distance between the nodes u and v; σ is an optional parameter that indicates the magnitude of change applied to each weight; higher values will result in more aggressive updates to the weights, X_i denotes a $Y \times f$ matrix ($Y = c \times d$, the neuron number in the SOM), with each row containing the same data as the i-th training data sample, $i \in [1, n]$, (n is the number of data samples in the training data set T). The training process utilized competitive learning, enabling the trained map to adjust towards the winning unit (BMU) within the data space. In this context, competitive learning refers to a process where neurons compete to respond to a specific input. The neuron that most closely matches the input is activated (moving towards the input to the greatest extent), while other neurons are suppressed (moving towards the input to a lesser extent). As a result, the trained SOM retains the topological structure of the data (Qu et al., 2021), making it suitable for use in subsequent tasks, as illustrated in Figure 2. The blue blob represents the distribution of the training data, while the small white disc indicates the current training sample drawn from this distribution. Initially (left), the SOM nodes are randomly positioned within the data space. The node closest to the training sample (highlighted in yellow) is selected and moved toward the sample, along with its neighboring nodes on the grid, albeit to a lesser extent. After numerous iterations, the grid aligns with and approximates the data distribution (right). The algorithm for the SOM training process is presented in Appendix Algorithm A1.

The training algorithm for SOM is not designed to handle categorical data directly. Common approaches for measuring the distance between categorical values include 1-of-k coding

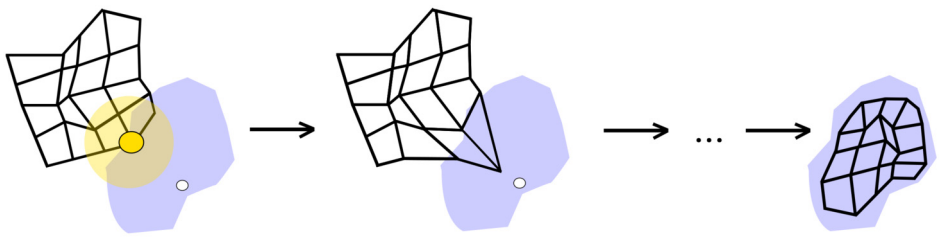

Figure 2: An illustration showing the training process of a self-organizing map, where the blue area represents the space of the training data. After training, the SOM captures and represents the topological structure of the training data.

(Wang et al., 2016) and the simple mismatch measure. The 1-of-k coding method transforms a categorical value into a set of binary values, while the simple mismatch measure assigns a distance of zero to identical values and a distance of one to different values. However, these methods fail to consider the inherent similarities within categorical data. Consequently, they do not accurately represent the structures of datasets that include categorical values on the map (Hsu and Lin, 2011).

Classic SOM algorithm struggles to cluster discrete data accurately for several reasons: 1) SOMs are effective at maintaining topological relationships between data points, meaning that similar data points in the input space should be close together on the SOM grid. However, this becomes challenging with categorical data, where the concepts of distance or topology may not be well-defined. 2) SOMs typically rely on distance metrics, such as Euclidean distance, to assess the similarity between data points. These metrics are tailored for continuous data and may not be appropriate for categorical data, where the distances between categories may lack significance (Fu and Li, 2025). 3) While SOMs are often employed for dimensionality reduction and visualization, this process is more complex for categorical data than for continuous data, as there is no inherent numerical scaling. 4) SOM neurons utilize weight vectors that are updated based on the input data during training. However, the rules for updating these weights are primarily designed for continuous data and may not perform effectively with discrete or categorical data, as concepts like "averaging" or "interpolation" do not apply.

2.2 TDSM-SOM

TDSM (Fu et al., 2023) SOM refers to a method that utilizes training data splitting to identify optimal representative neurons in SOM, thereby enhancing clustering outcomes in a distance-based neural network without altering the original network's internal algorithm or the quality of the training data. This approach enables a network with N neurons to expand into a new network comprising $m \times N$ neurons. Each of these neurons represents m subnetworks, with each subnetwork accurately reflecting a segment of the training set. In this scenario, the clustering quality indicators, such as purity, NMI, and ARI, achieve a value of 1. Specifically, whenever we apply SOM clustering to the training data, we obtain an updated weight matrix W_0. We then split the data into two parts: one part, referred to as the correct data I_c, consists of the dataset that can be perfectly represented by W_0. When we use W_0 to predict I_c, all clustering quality indicators equal 1. The other part is the incorrect data I_i, which cannot be represented by W_0. To identify I_i, we simply group all the data within each cluster that has a different label from the majority label in that cluster. Completing one iteration of SOM results in one split, yielding a correct subset of data, a weight matrix that accurately represents this correct dataset, and an incorrect subset of data. By following the same procedure, we can continuously split the incorrect subset until no incorrect subsets remain. Just as shown in Figure 3, for example, when $I_{i(n-1)} = I_{cn}$. The weight matrix W' is a combination of all the representations of the "represented data," serving as an ideal representation of the training set. This is the primary objective achieved through TDSM, which can then be utilized in the test data prediction phase.

2.3 Granular Computing

Granular Computing (GrC) is still in its inception stage (Lin, 2023), but the concept of granular computing is not new. The fundamental ideas and principles of granular computing are intertwined with research in various fields, including belief functions, artificial intelligence, cluster

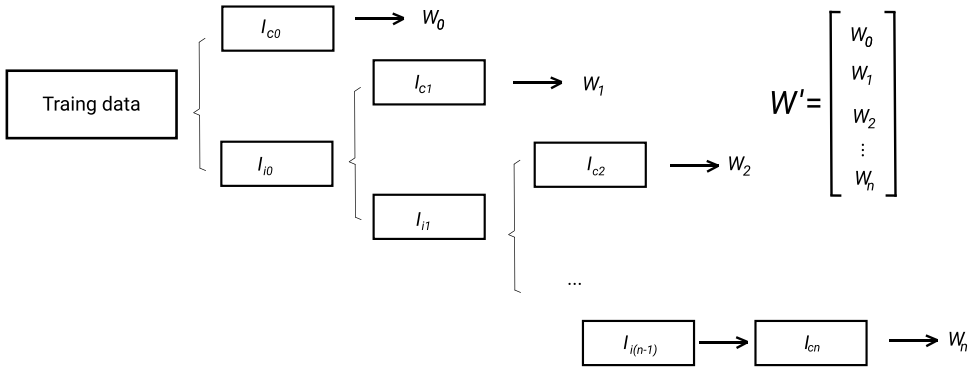

Figure 3: The process of continuously splitting the "unrepresented data" continues until no "unrepresented data," remains.

analysis, chunking, data compression, databases, decision trees, divide and conquer, fuzzy logic, interval computing, machine learning, software engineering, structured programming, quantization, quotient space theory, and rough set theory, to name just a few areas of study (Yao et al., 2013). Each granule signifies a group of objects that are similar, indistinguishable, or close to one another, allowing for data to be grouped at various levels of detail or abstraction. For example, granules such as "young," "middle-aged," and "old" represent age categories at a higher level of granularity compared to specific age numbers; similarly, small closed patterns can be seen as smaller granules. Both granules and clusters emphasize the significance of grouping data or objects based on shared characteristics or similarities. In many contexts, a granule is equivalent to a cluster and vice versa. In conclusion, while granules and clusters originate from different conceptual bases, they both concentrate on grouping and abstracting data according to similarities. These granules are composed of finer granules that are drawn together by distinguishability, similarity, and functionality. The process or operation of forming granules is referred to as granulation. Granulation appears to be an inherent methodology embedded in the human mind. In our daily lives, we routinely break down "objects" into smaller "sub-objects". For instance, the human body can be divided into parts like the head, neck, and so forth, while the Earth's surface can be categorized into features such as plateaus, hills, and plains. The boundaries of these sub-objects are inherently fuzzy, vague, and imprecise (Lin, 2009). Consequently, formalizing this concept flawlessly has proven to be a challenge. Clustering can be regarded as a technique for achieving granulation, and the outcomes of clustering can be applied within a granular computing framework.

In summary, both clustering and granular computing serve as methods for understanding and representing data at varying levels of abstraction. Granular computing is a computational problem-solving technique that involves segmenting large datasets into manageable "granules." This concept parallels clustering, which organizes data into groups based on similarity or proximity (Bargiela and Pedrycz, 2022). To enhance processing and reasoning across various levels, granular computing concentrates on deconstructing information into distinct levels of granularity, which denotes the extent of detail or the size of the components within a system (Song et al., 2023), as shown in Figure 4. In granular computing, the idea is to break down complex systems or data into smaller, more manageable components or granules, which can be analyzed, processed, and understood at different levels of abstraction or detail. Various features and relationships arise at different resolutions or levels of granularity, with each level exhibiting distinct

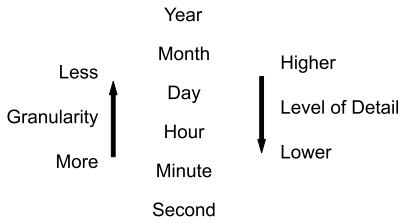

Figure 4: Increased detail corresponds to higher granularity, resulting in a greater number of data rows. Conversely, decreased detail leads to lower granularity, which means fewer data rows.

characteristics that set it apart from others (Guo et al., 2021). Granular computing seeks to leverage this aspect to create more efficient machine-learning and reasoning systems. Although clustering and granular computing are separate concepts, they share overlapping ideas that can enhance each other in numerous applications.

2.4 Fuzzy Set Theory

The inherently non-numerical nature of categorical attributes, compared to numerical attributes, along with the existence of incomplete and ambiguous categorical attributes in a dataset, heightens the uncertainty in decision-making (Aslam, 2023). Fuzzy Set computational theories and methodologies overlap with GrC (Ji et al., 2021), which provides a mathematical framework to handle uncertainty, vagueness, and imprecision, often present in real-world problems. It provides a more nuanced and adaptable approach than classical set theory, particularly for systems that must function under ambiguous conditions. The fundamental concepts of fuzzy set theory that have aided in the advancement of artificial intelligence were recently summarized in (Dzitac et al., 2017). In the realm of granular computing, fuzzy sets illustrate uncertainty or imprecision in data across various levels of granularity (Zadeh, 2023). The concepts of inclusion, union, intersection, complement, relation, convexity, and others are expanded to encompass such sets and various properties related to these concepts are defined within the framework of fuzzy sets (Pedrycz, 2021). Fuzzy sets are particularly advantageous in scenarios where data is inherently vague or uncertain. This is primarily due to their ability to represent and handle imprecise information effectively. Unlike traditional sets that require clear boundaries, fuzzy sets allow for degrees of membership, meaning an element can partially belong to a set rather than being strictly in or out.

A fuzzy set $\widetilde{A}$ in the universe of information I can be defined as a set of ordered pairs, and it can be represented mathematically by

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) \mid x \in I\}, \tag{2}$$

where $\mu_{\widetilde{A}}(x)$ denotes the membership function of $\widetilde{A}$, $\mu_{\widetilde{A}}(x) \in [0, 1]$, the membership function assigns a membership value (or degree of membership) to each element x in the universe of discourse I. This value indicates the extent to which the element belongs to the fuzzy set $\widetilde{A}$. The universe of discourse I (also known as the universal set) encompasses all potential elements or values that are pertinent to a specific problem or context. It establishes the domain within which the membership function operates. Fuzzy membership functions are crucial in fuzzy set theory (Deng and Deng, 2021), enabling the representation and manipulation of uncertain information. The most basic membership functions are created using straight lines. Due to

their straightforward formulas and computational efficiency, both triangular and trapezoidal membership functions are widely utilized, particularly in real-time applications (Ali et al., 2015).

In the SOM clustering, each neuron unit $N_{euron} = (w, M)$ possesses two characteristics: a synaptic weight vector $w = [n_{i_0}, n_{i_1}, \ldots, n_{i_{f-1}}]$, which matches the dimensionality of the input data ($x = [x_0, x_1, \ldots, x_{f-1}]$) and indicates the neuron's position within the data space. The term M_i refers to the training data samples represented by this neuron, known as its members M (Fu et al., 2023). Consequently, a neuron N_i can be viewed as a fuzzy set: $N_i = \{(x, \mu_{N_i}(x)) \mid x \in M_i\}$. In the context of TDSM-SOM, when N_i serves as a perfect representative neuron, it holds that $\mu_{N_i}(x) = 1$ ($x \in M_i$). This means that the membership degree is equal to 1.

3 Proposed Self Organized Granular Encoding

In this section, we propose a SOG encoding method that represents the distribution of training samples, which can be utilized in TDSM-SOM clustering. After performing unsupervised SOM clustering, each neuron unit can be viewed as a granule, referred to as an N-granule. An N-granule is formed by grouping data points based on their similarity, represented as $g_N = \{N_1, N_2, \ldots N_m\}$, where N_i indicates a fuzzy set neuron. For each feature F, we can derive the F-granule by grouping based on its values, given by $g_F = \{F_1, F_2, \ldots F_n\}$, where n is the number of possible values for feature F. Data points in each F_x share the same feature value. In our context, F consists of discrete data values. In a feature column with a data index set $A = \{1, 2, \ldots, K\}$,, as illustrated in Figure 5, the data consists of discrete values and contains n unique feature values represented as $U = \{u_1, u_2, \ldots, u_n\}$. From these feature values in U, we can derive the feature granule g_F based on feature values in U. For a neuron N_x (considered a fuzzy set), it is straightforward to identify the indexes of its members (the data samples it represents) as M ($M = \{M_1, M_2, \ldots, M_m\}$). Utilizing both A and M, we can establish the mapping between A and M, which is the desired outcome depicted in Figure 5. The levels of fuzziness associated with each feature value in A are illustrated by the mapping presented in

$$E_{SOG}(A) = [e_1, e_2, \ldots e_m], \quad e_i = \frac{|A \cap M_i|}{|M_i|}, \quad N_i = (w_i, M_i), \tag{3}$$

which reflects its gradual transition among the various neurons. Subsequently, we can derive the encoding for the feature value u in U:

$$u \rightarrow [e_1(u), e_2(u), \ldots, e_m(u)], \tag{4}$$

where $e_i(u) = \frac{|E_i(u)|}{|M_i|}$ and $E_i(u) = \{\ y \in A \cap M_i | a_y = u\}$.

In this context, the symbol $\rightarrow$ represents the encoding operation, while a_y signifies the yth feature value in column A (as illustrated in Figure 5). Additionally, E_i refers to a specific intersection of A and M_i. Consequently, this encoder mapping function acts as the membership function for the newly defined fuzzy set. The encoder is derived from the membership function ($\mu_{\widetilde{N_A}}(u)$) of fuzzy set $\widetilde{N_A}$, where $\widetilde{N_A} = \{(u, \mu_{\widetilde{N_A}}(u)) \mid u \in U\}$, and $\mu_{\widetilde{N_A}}(u) = [e_1(u), e_2(u), \ldots, e_m(u)]$.

The shape of the membership function determines how membership values vary across the universe of discourse. Various shapes exist, such as triangular, trapezoidal, Gaussian, sigmoidal, and others Hanif et al. (2023). The selection of a membership function shape depends on the problem at hand and the linguistic terms used to define the fuzzy set. Different shapes represent varying degrees of fuzziness and uncertainty. Membership functions can also be adjusted or hybridized to refine their shape, ensuring they align with the input data and maximize accuracy.

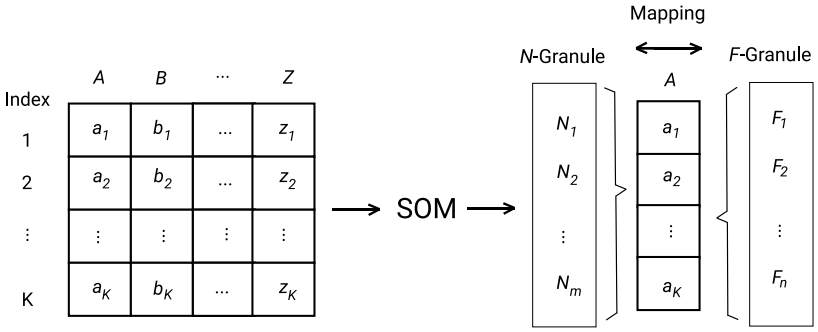

Figure 5: Each feature column's data can produce an F-granule, while all the neurons created by the SOM represent N-granule.

The choice of the membership function shape is typically guided by domain knowledge and experimental results, tailored to the specific application and data characteristics. In our case, the linguistic terms or classes are represented by the $\mu_{\widetilde{N_A}}$ for each discrete feature, whose membership function shape is defined as Singleton.

After encoding, the feature value is represented by a distribution of data indices rather than specific values. This approach minimizes the impact of distance metrics among feature values. The process of obtaining the indices set (F-granule) is independent of the actual feature values, regardless of whether the values are large or small, or whether their distances are significant or negligible. As long as the distribution remains unchanged, the F-granule remains the same. For example, if a feature's original value is 1 and it is changed to any unique value, such as 100 or 1000, the resulting A will remain identical. While this change might affect the results of N-granules, we do not really alter it here, as the goal is to highlight the theoretical advantages of SOG encoding.

3.1 Encoding Algorithm

The pseudo-algorithm is presented in Algorithm 1, while the detailed implementation process is outlined in the Appendix Algorithm A2. For a given discrete training dataset T, represented as a $K \times Z$ matrix, where K indicates the number of data samples and Z denotes the number of features. The algorithm detailed in A2 is crucial for obtaining the transferred encoding for unique feature values. This is represented by the dictionary D_{pfn}. Using the D_{pfn} generated by the algorithm, we can directly map each feature value u (the key in D_{pfn}) to its corresponding granular encoding $D_{pfn}[u]$.

Referring to the code from lines 6 to 13 in Algorithm A2, the computational complexity for each feature column is $O(K \times n \times m)$, where n represents the number of unique feature values in the input features, m denotes number of neurons used to generate the SOG encoding, consistent with the optimal neuron count from the TDSM process. Since both n and m are constants and do not depend on the number of input data samples (K), the complexity simplifies to $O(K)$, which is comparable to the traditional expression $O(n)$. Moreover, our experiments demonstrate that the number of unique feature values and neurons is relatively small compared to the size of the input data. However, the algorithm's scalability is hindered by its quadratic dependence on the number of samples, making it computationally intensive for large datasets. To improve efficiency, several strategies can be applied: batch processing allows data to be grouped and processed collectively,

Algorithm 1 Obtain the SOG encoding for an input feature column.

Input: A feature column's data
Output: The SOG encoding for each unique feature value

1: Group the indices of the input feature data into separate sets based on the unique values of each feature. Each set should contain the indices of the data points that share the same unique feature value.
2: The unique feature values are represented by a list of indices, denoted by A, which is a distribution of the sets of indices obtained from the previous operation.
3: Train the SOM on T to determine each neuron and its associated member indices, and M represents the list of each neuron's member indices set.
4: Both A and M are distributions of data indices, and we obtain the mapping between A and M (shown in Equation 3).
5: All unique feature values in the input features will be represented using the newly derived mapping representation, known as SOG encoding.

reducing the number of iterations; parallel computing distributes computations across multiple processors, speeding up nested loops; sparse representations minimize operations by focusing only on significant contributions from neurons or feature values; and approximation techniques use sampling methods to estimate membership probabilities, avoiding exhaustive calculations for all data points. These methods to enhance scalability will be the focus of our future research direction.

3.2 Comparison with Entity Embedding

We conducted a comparison between the newer categorical data embedding method, entity embedding, and SOG encoding within the context of SOM clustering, using datasets sourced from Kaggle. As presented in Table 1, the findings reveal that SOG encoding performs on par with entity embedding. The results of the T-test indicate that it is not possible to determine a clear winner between the two methods. Across the seven datasets, entity embedding outperforms SOG encoding in the first three datasets, while SOG encoding achieves better results in the last three datasets. For one dataset, both methods produce identical outcomes. However, SOG encoding demonstrates a significant advantage in training time, as one epoch of entity embedding training requires as much time as SOG encoding with 100 neurons (refer to Table 2). In practice, using more than 100 neurons for SOM clustering is rare, as the number of neurons generally corresponds to the number of clusters.

4 Experiment

We employed the supervised TDSM SOM model as our baseline and for the proposed model, we will first apply unsupervised self-organized granular (SOG) encoding to both the training and test sets before reapplying the TDSM SOM, as shown in Figure 6. Although normalization can enhance the performance of models that rely on distance metrics by mitigating the dominance of features with larger scales, our proposed SOG encoding focuses on capturing the distributional information of features rather than their distances. Applying normalization would alter the original distribution of the training data, which could lead to suboptimal results. To ensure consistency, we avoid using standardization in both the baseline model and the proposed

Table 1: T-test results for the datasets present a comparison of entity embedding and SOG encoding applied during SOM clustering. The p-values are calculated under the assumption that SOG encoding performs better than entity embeddings.

Dataset	ScoreType	Entity Embedding	SOG Encoding	T-test p-value
Netflix Userbase Dataset	Purity	0.4321	0.4255	0.8998
	NMI	0.0179	0.0148	0.8107
	ARI	0.0212	0.0186	0.8488
Customer Segmentation	Purity	0.6384	**0.6365**	0.9932
	NMI	0.0133	0.0002	0.9946
	ARI	0.0203	0.0004	0.9872
Austisum Screening data for toddlers	Purity	0.9287	0.7740	0.9999
	NMI	0.6448	0.1495	0.9999
	ARI	0.7384	0.2443	0.9999
Airline Passenger Satisfaction	Purity	0.5610	0.5610	0.5000
	NMI	0.0000	0.0000	0.5000
	ARI	0.0000	0.0000	0.5000
Cardiovascular Diseases Risk Prediction Dataset	Purity	0.3810	**0.3812**	0.4251
	NMI	0.0187	**0.0188**	0.4452
	ARI	0.0168	**0.0169**	0.4141
Hotel Reservations Dataset	Purity	0.7558	0.7558	0.4199
	NMI	0.1450	**0.1458**	0.4461
	ARI	0.2278	**0.2283**	0.4934
Car Price	Purity	0.6339	**0.6482**	0.3103
	NMI	0.5417	**0.5475**	0.4368
	ARI	0.2910	**0.2976**	0.4337

methods. The primary reason for deploying SOG encoding prior to performing TDSM-SOM clustering is our belief that SOG encoding can effectively identify uncertainty, vagueness, and imprecise information within the training set, as discussed in Section 2.4. For the test datasets, the number of features considered remained unchanged. In our experiment, to capture a broader range of challenges in categorical data clustering, we have included standard benchmark datasets widely used in categorical clustering research, such as those from UCI, along with several new datasets sourced from Kaggle. We classified features with fewer than 20 unique values as discrete, including categorical data. In our proposed model, after preprocessing, the discrete features are converted into SOG-encoded features. To enhance the reliability of performance evaluation and eliminate bias caused by data splitting, we implement k-fold cross-validation to generate training and testing datasets. To evaluate the accuracy of clustering results, we utilize three commonly used metrics: purity, NMI, and ARI. Table 3 presents a comprehensive overview of the preprocessing methodology used for the experimental datasets. The "Class Label" column indicates the feature used as the class label or target and serves as a reference for validating and interpreting the results. And all the datasets have no empty, blank, or missing values, so the sparsity calculated here means the entries' values are 0.

Table 2: SOG encoding and entity embedding time comparison for Netflix userbase dataset.

SOG Encoding		Entity Embedding	
Neuron	Time	Epochs	Time
5	0.1999	1	0.9023
10	0.2546	2	1.2274
15	0.2497	3	1.0543
20	0.2757	4	1.0654
25	0.3808	5	1.402
30	0.3017	6	1.119
35	0.4371	7	1.2175
40	0.5453	8	1.4649
45	0.5474	9	1.3009
50	0.5763	10	1.1985
55	0.5977	100	4.9734
60	0.6437	200	8.4563
65	0.6066		
70	0.6729		
75	0.7656		
80	0.8313		
85	0.7542		
90	0.8516		
95	0.7756		
100	0.9637		

4.1 Experiment Result

The proposed method consistently outperforms the baseline across all datasets, as demonstrated in Table 4. This table highlights that achieving better clustering results than the baseline is feasible, as shown by the values in two specific columns. The column“Mean Score in Baselin” represents the average result obtained using k-fold cross-validation for the baseline model, whereas “Mean Score in Proposed Method” reflects the average validation scores achieved by applying the same cross-validation approach to the proposed method. The “Confidence Interval” column indicates the range within which the true performance metric (e.g., Purity, ARI, NMI) is likely to fall with 95% confidence. The Mean Score Increased Percentage column shows the percentage improvement in the proposed method’s mean result compared to the baseline’s mean outcome. The unit of measurement for the time spent on both the baseline and proposed models is seconds. This represents the duration required for one fold in the k-fold cross-validation process. Additionally, the Neuron Number refers to the number of neurons obtained during the TDSM process, which is utilized to generate the SOG encoding for the training data.

The proposed method demonstrates significant potential for enhancing clustering performance, as evidenced by the increased mean values of clustering validation metrics. Specifically, the improvement in mean Purity ranges from 0.09% to 21.03%, the ARI improvement spans from 0.60% to 157.99%, and the NMI improvement varies from 1.02% to 146.02%. The time re-

Table 3: Dataset description and data pre-processing.

Datase	Source	Features No.	Discrete Features No.	Training Data Samples No.	Class Label	Class Distributions	Sparsity (%)
Nusery	UCI	4	4	12960	class	not_recom: 4320 priority: 4266 recommend: 2 spec_prior: 4044 very_recom: 328	31.66
BalanceScale	UCI	4	4	625	class	B: 49 L: 288 R: 288	20.00
Mushroom	UCI	22	22	1728	poisonous	edible: 4208 poisonous: 3916	25.94
Car Evaluation	UCI	6	6	1728	class	acceptable: 384 good: 69 unacceptable: 1210 very good: 65	29.02
Average Time Spent By A User On Social Media Dataset	Kaggle	11	9	1000	time_spent	1 hour: 99 2 hours: 112 3 hours: 107 4 hours: 120 5 hours: 125 6 hours: 110 7 hours: 107 8 hours: 109 9 hours: 111	29.16
Hair Health Prediction Dataset	Kaggle	11	10	999	Hair Loss	absence of baldness: 502 presence of baldness: 497	34.22
BI intro to Data Cleaning Eda And Machine Learning	Kaggle	10	4	77	prevEducation	Bachelors: 24 Barrrchelors: 1 Diploma: 12 Doctorate: 5 High School: 19 Masters: 16	7.20
Hotel Reservations	Kaggle	17	13	36275	booking_status	Not_Canceled: 11885 Canceled: 24390	44.56

quired for the proposed methods is longer compared to the baseline model. This additional time is spent generating the SOG encoding for both the training and testing data, which significantly increases the dimensionality of the input data, thereby causing the time increase. However, this is acceptable for a single fold in the k-fold cross-validation process. Nonetheless, as the value of k increases, the total time spent will scale proportionally, increasing by a factor of k.

5 Conclusion

The SOM is a type of unsupervised, competitive learning neural network that plays a crucial role in data visualization and clustering. Its primary function is to map high-dimensional data onto a low-dimensional grid, typically a two-dimensional space. This mapping process allows for an intuitive representation of complex datasets, making it easier to analyze and interpret the underlying patterns and structures. TDSM SOM is a supervised clustering method designed to enhance the performance of traditional SOM clustering. However, when dealing with discrete data, particularly categorical data, it becomes challenging to achieve significant improvements

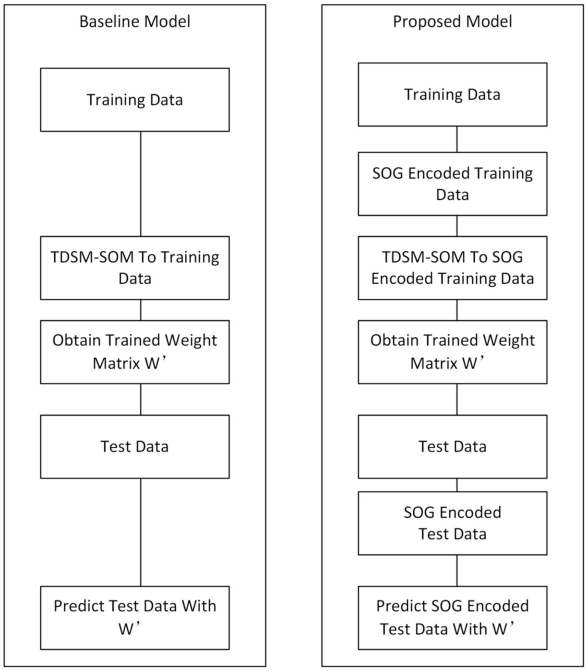

Figure 6: The comparison between the baseline model and the proposed model in the experimental procedure.

Table 4: Comparison of clustering outcomes: baseline vs. proposed method.

Dataset	K value in K-folder Cross Validation	Score Type	Mean Score in Baseline	Mean Score in Proposed Method	95% Confidence Interval	Mean Score Increased Percentage	Time Spent on Baseline Model	Time Spend on Proposed Model	Neuron Number
Nursery	20	Purity	0.8621	**0.8970**	(0.8470, 0.9470)	**4.05%**	6.1648	14.8986	5
		ARI	0.6582	**0.7652**	(0.6664, 0.8640)	**16.26%**			
		NMI	0.7611	**0.8127**	(0.7383, 0.8871)	**6.78%**			
BalanceScale	20	Purity	0.8272	**0.8847**	(0.8606, 0.9087)	**6.95%**	0.1297	0.1407	3
		ARI	0.5128	**0.5838**	(0.4888, 0.6788)	**13.85%**			
		NMI	0.5165	**0.5614**	(0.4696, 0.6531)	**8.69%**			
Mushroom	15	Purity	0.9988	**0.9997**	(0.9992, 1.000)	**0.09%**	5.2292	11.5380	3
		ARI	0.9928	**0.9988**	(0.9966, 1.001)	**0.60%**			
		NMI	0.9870	**0.9971**	(0.9915, 1.002)	**1.02%**			
Car Evaluation	20	Purity	0.7941	**0.9611**	(0.9346, 0.9877)	**21.03%**	0.4515	0.5713	4
		ARI	0.2973	**0.7670**	(0.6295, 0.9046)	**157.99%**			
		NMI	0.3025	**0.7442**	(0.6057, 0.8827)	**146.02%**			
Average Time Spent By A User On Social MediaDataset	50	Purity	0.4980	**0.5080**	(0.4897, 0.5262)	**2.01%**	0.8875	1.9536	9
		ARI	−0.0096	**−0.0018**	(−0.0228, 0.0192)	**81.25%**			
		NMI	0.5413	**0.5498**	(0.5327, 0.5669)	**1.57%**			
Hair Health Prediction Dataset	25	Purity	0.5896	**0.5966**	(0.5770, 0.6163)	**1.19%**	0.2853	0.3821	2
		ARI	0.0026	**0.0058**	(−0.0074, 0.0191)	**123.08%**			
		NMI	0.0212	**0.0220**	(0.0126, 0.0315)	**3.77%**			
BI intro to data cleaning eda and machine learning	40	Purity	0.9375	**0.9500**	(0.9029, 0.9970)	**1.33%**	0.0415	0.1515	10
		ARI	0.5500	**0.6500**	(0.5003, 0.7996)	**18.18%**			
		NMI	0.5500	**0.6500**	(0.5003 0.7996)	**18.18%**			
Hotel Reservations Dataset	20	Purity	0.8200	**0.8400**	(0.8368, 0.8432)	**2.44%**	59.8411	113.1523	2
		ARI	0.4007	**0.4542**	(0.4455, 0.4629)	**13.35%**			
		NMI	0.2762	**0.3237**	(0.3155, 0.3319)	**17.20%**			

in clustering performance. Achieving significant improvements in clustering performance can be quite challenging due to the inherent characteristics of discrete variables, which often lack the continuous nature that SOMs typically exploit. This paper employs supervised learning representation (SOG encoding) to enhance the clustering performance of TDSM SOM. The contributions of this paper are as follows: first, it improves supervised learning outcomes by leveraging unsupervised learning techniques. Second, it introduces an innovative embedding method for discrete data that leverages granular computing and fuzzy set theory, facilitating the discovery of data distribution among discrete and categorical data, which aids in revealing ambiguous, vague, uncertain, and imprecise information contained within the discrete data.

Supplementary Material

All data and code associated with the data are in the GitHub repository https://github.com/foolishfool/TDSMSOG.

A Appendix

This appendix provides supplementary materials that support the methodology presented in the main paper. Specifically, it includes the detailed procedures for the SOM training process and the algorithm used to identify new embedding vectors for feature values within an F-granule. In addition, we present a figure comparing the clustering results produced by SOM when using different distance metrics.

Algorithm A1 describes the training process of the SOM, which learns the weight matrix used for clustering and representation learning. Algorithm A2 illustrates how feature values are mapped to embedding vectors using the neuron membership information produced by the trained SOM. Figure A1 shows a comparison of SOM clustering outcomes using several commonly used distance metrics.

Algorithm A1 Self-organization mapping training process.

Input: Training data set T.

Output: A weight matrix W, which can be utilized in the prediction process.

Initialize a $2D$ grid with c rows and d columns. Then define the weight matrix $W_{initial}$ with dimensions $n \times m$, where $n = c \times d$ and m represents the number of features in the training data.
for each data sample x in T **do**
    // Identify the best matching unit index B_i in $W_{initial}$.
    Let u be ${B_i}^{th}$ vector that has the minimum distance to x.
    for each vector v in $W_{initial}$ **do**
        Calculate neighborhood function $G(u, v) = e^{d_{uv}/\sigma^2}$
        Update $W_{initial}$ using the equation provided in Equation (1).
    end for
end for
return W, where $W = W_{initial}$

Algorithm A2 Identify a new embedding vector for a feature value within an F-granule.

Input: A column feature data index set A from T, $A = [1, 2, \ldots, K]$.

Output: A dictionary whose key is the specific unique feature value in U (unique feature value set) and the dictionary's value is a $m \times 1$ vector, where m denotes the number of neurons in the SOM. The value of the dictionary is the proposed encoding of the key.

Extract the unique feature value set U from the currently processed feature column.
Group A using U and generate F, $F = [\ F_1, F_2, \ldots, F_n]$. Each F_i represents a list of data indexes in T, corresponding to data points that share the same feature value u.
Train T with the SOM to obtain each neuron along with its member indexes M, M_i is a list of indexes from the data that the ith neuron can represent.
Create a dictionary D_{pfn} (pfn represents the **p**robability of the **f**eature values occurring in a specific **n**euron, with keys representing u and value as the membership grade list of u in M. Initialize the dictionary with empty lists as values.)
Create a dictionary D_{ffn} (ffn represents the **f**requency of the **f**eature value appearing in a **n**euron., the key is the index of M_i (value m in Equation (4)) and value is the frequency of data index in F_i but also shown in M_i, initial value is 0)
for each list f in F_x **do** ($x \in [1, n]$)
    for each data index a in f **do**
        for each member index list M_y in M **do** ($y \in [1, m]$)
            if a in M_y **then** $D_{ffn}[y] += 1$
            end if
        end for
    end for
end for
for key, value in D_{ffn} **do** (
    p[key] = value/len(M_i) (p is a list)
    $D_{pfn}[u] = p$)
end for
return D_{pfn}

Funding

This article was partially supported by the Grant DP220101360 from the Australian Research Council.

References

Ali OAM, Ali AY, Sumait BS (2015). Comparison between the effects of different types of membership functions on fuzzy logic controller performance. *International Journal*, 76: 76–83.

Aslam M (2023). Cochran's Q test for analyzing categorical data under uncertainty. *Journal of Big Data*, 10(1): 147. https://doi.org/10.1186/s40537-023-00823-3

Bargiela A, Pedrycz W (2022). Granular computing. In: *Handbook on Computer Learning and Intelligence: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation* (PP Angelov, ed.), 97–132. World Scientific.

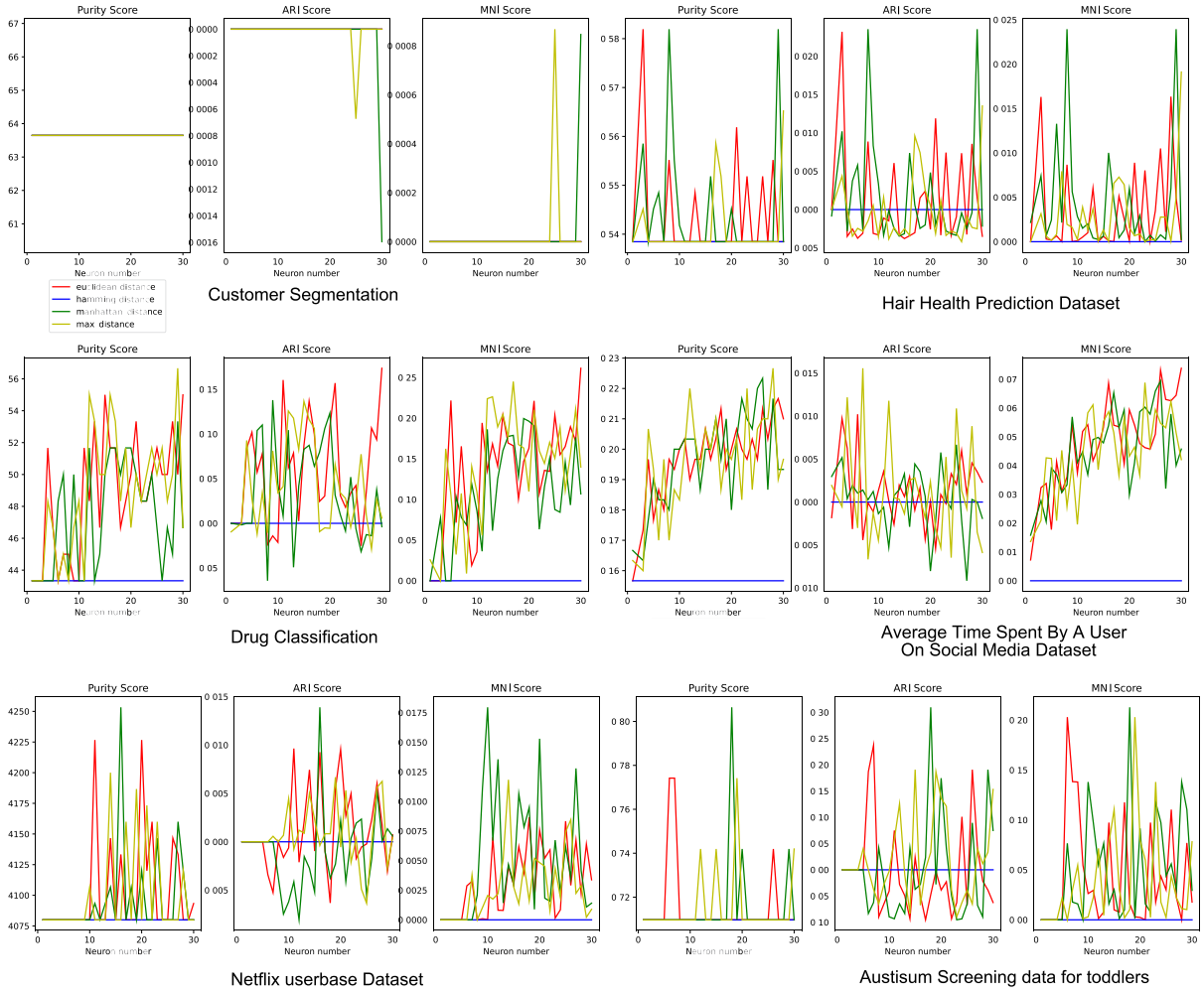

Figure A1: Comparison of SOM clustering results using various distance metrics: Euclidean, Hamming, Manhattan, and maximum distance. Data sets come from Kaggle.

Bigdeli A, Maghsoudi A, Ghezelbash R (2022). Application of self-organizing map (SOM) and K-means clustering algorithms for portraying geochemical anomaly patterns in Moalleman district, NE Iran. *Journal of Geochemical Exploration*, 233: 106923. https://doi.org/10.1016/j.gexplo.2021.106923

Chushig-Muzo D, Soguero-Ruiz C, Engelbrecht AP, Bohoyo PDM, Mora-Jiménez I (2020). Data-driven visual characterization of patient health-status using electronic health records and self-organizing maps. *IEEE Access*, 8: 137019–137031. https://doi.org/10.1109/ACCESS.2020.3012082

Dai J, Zhu Z, Zou X (2024). Fuzzy rough attribute reduction based on fuzzy implication granularity information. *IEEE Transactions on Fuzzy Systems*, 32, 3741–3752. https://doi.org/10.1109/TFUZZ.2024.3381993

Deng J, Deng Y (2021). Information volume of fuzzy membership function. *International Journal of Computers Communications & Control*, 16(1).

Dzitac I, Filip FG, Manolescu MJ (2017). Fuzzy logic is not fuzzy: World-renowned computer

scientist Lotfi A. Zadeh. *International Journal of Computers Communications & Control*, 12(6): 748–789. https://doi.org/10.15837/ijccc.2017.6.3111

Fu Q, Li Y (2025). Automated contrastive optimization of class-based feature distribution for noncontinuous dataset. *Knowledge and Information Systems*, 1–40. https://doi.org/10.1007/s10115-025-02576-2

Fu Q, Li Y, Albathan M (2023). A supervised method to enhance distance-based neural network clustering performance by discovering perfect representative neurons. *Granular Computing*, 8(5): 1051–1065. https://doi.org/10.1007/s41066-023-00370-5

Guo S, Zhao H, Yang W (2021). Hierarchical feature selection with multi-granularity clustering structure. *Information Sciences*, 568: 448–462. https://doi.org/10.1016/j.ins.2021.04.046

Hanif R, Mustafa S, Iqbal S, Piracha S (2023). A study of time series forecasting enrollments using fuzzy interval partitioning method. *Journal of Computational and Cognitive Engineering*, 2(2): 143–149. https://doi.org/10.47852/bonviewJCCE2202159

Hidalgo DR, Cortés BB, Bravo EC (2021). Dimensionality reduction of hyperspectral images of vegetation and crops based on self-organized maps. *Information Processing in Agriculture*, 8(2): 310–327. https://doi.org/10.1016/j.inpa.2020.07.002

Holloway EM (2019). Self organized multi agent swarms (SOMAS) for network security control.

Hsu CC (2006). Generalizing self-organizing map for categorical data. *IEEE Transactions on Neural Networks*, 17(2): 294–304. https://doi.org/10.1109/TNN.2005.863415

Hsu CC, Lin SH (2011). Visualized analysis of mixed numeric and categorical data via extended self-organizing map. *IEEE Transactions on Neural Networks and Learning Systems*, 23(1): 72–86.

Ji W, Pang Y, Jia X, Wang Z, Hou F, ..., Wang R (2021). Fuzzy rough sets and fuzzy rough neural networks for feature selection: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(3): e1402.

Jia H, Cheung Ym, Liu J (2015). A new distance metric for unsupervised learning of categorical data. *IEEE Transactions on Neural Networks and Learning Systems*, 27(5): 1065–1079. https://doi.org/10.1109/TNNLS.2015.2436432

Khacef L, Rodriguez L, Miramond B (2020). Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning. *Electronics*, 9(10): 1605. https://doi.org/10.3390/electronics9101605

Kohonen T (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9): 1464–1480. https://doi.org/10.1109/5.58325

Li C, Jiang L, Li H, Wu J, Zhang P (2017). Toward value difference metric with attribute weighting. *Knowledge and Information Systems*, 50(3): 795–825. https://doi.org/10.1007/s10115-016-0960-x

Li N, Jiang K, Ma Z, Wei X, Hong X, Gong Y (2021). Anomaly detection via self-organizing map. In: *2021 IEEE International Conference on Image Processing (ICIP)* (Organizing Committee of IEEE ICIP 2021, ed.), 974–978. IEEE.

Licen S, Di Gilio A, Palmisani J, Petraccone S, de Gennaro G, Barbieri P (2020). Pattern recognition and anomaly detection by self-organizing maps in a multi month e-nose survey at an industrial site. *Sensors*, 20(7): 1887. https://doi.org/10.3390/s20071887

Lin TY (2009). Granular computing I: The concept of granulation and its formal model. *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, 1(1): 21–42. https://doi.org/10.1504/IJGCRSIS.2009.026723

Lin TY (2023). Granular computing: Practices, theories, and future directions. In: *Granular,*

Fuzzy, and Soft Computing (T-Y Lin, C-J Liau, J Kacprzyk, eds.), 199–219. Springer.

Lyu Z, Ororbia A, Li R, Desell T (2024). Minimally supervised learning using topological projections in self-organizing maps. arXiv preprint: https://arxiv.org/abs/2401.06923

Melin P, Monica JC, Sanchez D, Castillo O (2020). Analysis of spatial spread relationships of coronavirus (COVID-19) pandemic in the world using self organizing maps. *Chaos, Solitons & Fractals*, 138: 109917.

Miljković D (2017). Brief review of self-organizing maps. In: *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (MIPRO Technical Program Committee, ed.), 1061–1066. IEEE.

Nagar D, Pannerselvam K, Ramu P (2022). A novel data-driven visualization of n-dimensional feasible region using interpretable self-organizing maps (iSOM). *Neural Networks*, 155: 398–412. https://doi.org/10.1016/j.neunet.2022.08.019

Neagoe VE, Ropot AD (2002). Concurrent self-organizing maps for pattern classification. In: *Proceedings First IEEE International Conference on Cognitive Informatics*, 304–312. IEEE.

Ni M, Cheng H, Lai J (2021). Gan–som: A clustering framework with SOM-similar network based on deep learning. *The Journal of Supercomputing*, 77: 4871–4886. https://doi.org/10.1007/s11227-020-03464-y

Pedrycz W (2021). *An Introduction to Computing with Fuzzy Sets: Analysis, Design, and Applications*. Springer, Intelligent Systems Reference Library (Vol. 190).

Qu X, Yang L, Guo K, Ma L, Sun M, ..., Li M (2021). A survey on the development of self-organizing maps for unsupervised intrusion detection. *Mobile Networks and Applications*, 26: 808–829. https://doi.org/10.1007/s11036-019-01353-0

Riese FM, Keller S, Hinz S (2019). Supervised and semi-supervised self-organizing maps for regression and classification focusing on hyperspectral data. *Remote Sensing*, 12(1): 7. https://doi.org/10.3390/rs12010007

Song M, Hu L, Feng S, Wang Y (2023). Feature ranking based on an improved granular neural network. *Granular Computing*, 8(1): 209–222. https://doi.org/10.1007/s41066-022-00324-3

Wang F, Franco H, Pugh J, Ross RJ (2016). Empirical comparative analysis of 1-of-K coding and K-prototypes in categorical clustering. *CEUR Workshop Proceedings*, 1751, 248–259.

Wickramasinghe CS, Amarasinghe K, Manic M (2019). Deep self-organizing maps for unsupervised image classification. *IEEE Transactions on Industrial Informatics*, 15(11): 5837–5845. https://doi.org/10.1109/TII.2019.2906083

Yang X, Dong M, Guo Y, Xue JH (2020). Metric learning for categorical and ambiguous features: An adversarial method. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (F Hutter, K Kersting, J Lijffijt, I Valera, eds.), 223–238. Springer.

Yao JT, Vasilakos AV, Pedrycz W (2013). Granular computing: Perspectives and challenges. *IEEE Transactions on Cybernetics*, 43(6): 1977–1989. https://doi.org/10.1109/TSMCC.2012.2236648

Zadeh LA (2023). Fuzzy logic. In: *Granular, Fuzzy, and Soft Computing* (TY Lin, CJ Liau, J Kacprzyk, eds.), 19–49. Springer US, New York, NY.