STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–26.
Published online in International Academic Press (www.IAPress.org)

Optimizing Kohonen Classification of Mixed Data with Partial Distance and Referent Vector Initialization

Mouad Touarsi [1,*], Driss Gretete [1], Abdelmajid Elouadi [2]

[1] *Engineering Sciences Laboratory, Ibn Tofail University, Morocco*
[2] *Advanced Systems Engineering Laboratory, Ibn Tofail University, Morocco*

Abstract The success of neural network models in clustering problems is highly dependent on the quality and diversity of the data used. Self-organizing maps (SOM), a semi-supervised data learning tool introduced by Kohonen in the 1980s, have been widely used in various fields such as signal and text recognition, industrial data analysis, speech and image recognition, etc. SOM's competitive learning clustering method, where each node specializes in a specific subset of data, has proven to be a powerful technique.

In this paper, we propose a new SOM variant suitable for handling numerical, interval, and categorical attributes simultaneously. Instead of random initialization of weights, we utilize the ASAICC algorithm to select initial referent vectors.

Furthermore, we suggest representing one cluster using multiple referent vectors at once. The effectiveness of the proposed Kohonen variant is evaluated using well-known benchmark datasets, and the results are reported using reliable performance metrics. The simulation of the new algorithm is conducted using the R language, and the obtained results demonstrate the superiority of the proposed approach.

Keywords Mixed data classification, SOM maps, ASAICC algorithm, partial distances computation

AMS 2010 subject classifications 1916, 1916

DOI: 10.19139/soic-2310-5070-1916

1. Introduction

Neural networks are powerful computing systems that can model complex relationships between data inputs and outputs. They have applications in various fields such as image processing, speech recognition, machine translation, anomaly detection, medical diagnosis, and data clustering and visualization [11] [12] [14] [13].

One type of neural network that has gained popularity is the Kohonen network or Self-Organizing Map (SOM). Kohonen networks, developed by Kohonen in 1982, are inspired by the neuronal principles of the vertebrate brain. They use a self-adaptive map to represent a set of data, allowing for a multi-dimensional visualization of data relationships [1] [15].

However, one of the main challenges with Kohonen maps is that they were initially designed to deal only with numerical data. Later, other SOM variants were developed to handle non-categorical data. Nevertheless, Kohonen

*Correspondence to: Mouad Touarsi (mouad.touarsi@uit.ac.ma). Engineering Sciences Laboratory, Ibn Tofail University. Kenitra, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

maps have suffered from a lack of rigorous results on their convergence when dealing with heterogeneous data [16].

An important consideration when using Kohonen maps is the initialization of cluster centers. Different initialization methods can be classified into random and data analysis-based approaches [17].

One recent and powerful initialization method is the Adaptive Selection Algorithm for Initial Cluster Centers (ASAICC) proposed by Gao, Fan, Niu, and Wang in 2017. This method selects vectors with high density and low similarity as initial cluster centers by computing the SBNDF index. The ASAICC algorithm is suitable for high-dimensional problems and can also be used for outlier detection [53].

When dealing with data that contains categorical features, a modified version of the classical K-prototypes algorithm was suggested by. B. Kim. His approach consists in using partial distance computation to extend the K-prototypes algorithm. B. Kim proved that the proposed algorithm's CPU time increased at most linearly with the observations cardinality for the considered datasets, and that the computational performance was improved compared to the original K-prototypes algorithm [19].

In our paper, we propose a novel variant of the SOM map that utilizes partial distance computation and the ASAICC algorithm for referent vector initialization for mixed-data clustering. We introduce modifications to the ASAICC algorithm to suit mixed-data clustering and improve computational performance by reducing the number of relevant data axes. We present the formulation of the clustering problem for mixed data and a review of related works, describe our proposed SOM variant that employs partial distance computation, and demonstrate the suitability of the ASAICC algorithm as a referent vector initializer. Additionally, we describe the benchmark datasets and performance metrics used in our experiments, conduct experiments on well-known datasets, report the performance of our proposed SOM variant, and discuss the results and conclusion.

2. Clustering Mixed-Data: Formulation and Related Works

Clustering algorithms are widely used in various fields, such as marketing, biology, and social network analysis. Partitional clustering algorithms have been extensively studied and applied to datasets consisting of either numerical or categorical data. However, real-world datasets often include both numerical and categorical features, which pose a challenge for clustering algorithms.

In recent years, researchers have proposed various approaches to address this challenge, such as defining cluster centers that can handle both types of features, developing distance measures that can combine numerical and categorical features, and designing cost functions that can handle mixed data.

Several distance-based clustering algorithms for mixed data have been proposed in the literature, but the effectiveness of these algorithms depends on the specific characteristics of the datasets. Veldenet et al. [46] investigated five distance-based clustering algorithms for mixed data using three mixed data sets and concluded that no single algorithm worked well for all data sets.

Similarly, Fuss et al. [47] reviewed partitional clustering and model-based clustering algorithms for mixed data, while Balaji and Lavanya [48] presented a short review study on mixed data clustering.

Most partitional algorithms are designed for either numerical or categorical data (for example, K-means [50]) or pure categorical data (for example, K-modes [51]). The general idea of these algorithms can be extended to handle mixed data.

Specifically, partitional algorithms aim to optimize a cost function iteratively by defining a distance measure that can combine numerical and categorical features, and a cluster center that can represent both types of features.

The cost function is typically defined as the sum of distances between data points and their nearest cluster centers.

Combining the above ideas, most of the partitional clustering algorithms optimize the following cost function iteratively :

$$\sum_{i=1}^{n} \xi(X_i, C_i) \tag{1}$$

Where:

- ξ is a distance.
- n is the number of data-set points.
- C_i is the data-set center that is the nearest to the observation X_i.

An important reason for the early adoption and widespread adaptability of partitional algorithms is that they are linear in the number of data points, scales well to large datasets, and can be adapted to parallelization frameworks (for example, MapReduce) [66].

3. Mixed Data Techniques :

K-Prototypes: This is an extension of the K-means algorithm that combines the features of K-means and K-modes, allowing it to cluster data with mixed numerical and categorical types. K-prototypes uses a dissimilarity measure that handles both types of data by assigning weights to different variables, which helps in balancing the influence of each type during clustering [55, 56, 57] .

K-Prototypes is specifically designed for clustering mixed data types. It extends K-means by integrating a measure for categorical variables:

$$D(x, y) = \sum_{j \in \text{numerical}} (x_j - y_j)^2 + \gamma \sum_{j \in \text{categorical}} \delta(x_j, y_j) \tag{2}$$

This distance measure is key in evaluating the dissimilarity between data points, allowing the algorithm to effectively create clusters that consider both numerical and categorical differences.[55, 56, 57]

Gower's Distance: Gower's distance metric computes similarities between data points that could have both numerical and categorical data. This metric is particularly useful in clustering algorithms that require a distance matrix, such as hierarchical clustering [58, 59] .

Model-Based Clustering: Model-based approaches, such as those using Gaussian Mixture Models (GMMs), have been adapted for mixed data types by assuming different distributions for categorical and continuous variables. These methods estimate the parameters of these distributions within a probabilistic framework, thereby accommodating the inherent heterogeneity in mixed datasets.[55, 56, 57]

Fuzzy Clustering: Fuzzy clustering methods, like Fuzzy C-Means (FCM), can be extended to handle mixed data types by incorporating different membership functions for categorical and numerical data. This allows each point to belong to multiple clusters with varying degrees of membership, providing a soft partitioning of the data that can be particularly useful in ambiguous or overlapping cluster structures. [60]

Density-Based Clustering : Techniques like DBSCAN and OPTICS are extended for mixed data by defining a new notion of neighborhood that accounts for the density of both categorical and numerical points. This method

can identify arbitrarily shaped clusters based on density connectivity, offering robustness against noise and outlier points which are common in heterogeneous datasets [61].

Support Vector Machines (SVM):
Although primarily a classification tool, SVM can be adapted for clustering tasks through methods like Support Vector Clustering (SVC). For mixed data, using a custom kernel can be essential. A common approach is to define a kernel that combines the properties of both categorical and numerical data [62] :

$$K(x, y) = \exp(-\gamma_s \|x_{\text{num}} - y_{\text{num}}\|^2) + \exp(-\gamma_c \delta(x_{\text{cat}}, y_{\text{cat}})) \tag{3}$$

where x_{num} and y_{num} are numerical parts of the data vectors, x_{cat} and y_{cat} are categorical, γ_s and γ_c are parameters controlling the influence of each data type, and δ represents a matching function for categorical variables.

Neural networks: can be tailored for clustering mixed data by integrating feature extraction layers that handle different data types. Autoencoders, for example, can be designed with separate pathways for numerical and categorical inputs, merging these pathways before the bottleneck layer [65] :

$$z = f(W_n x_n + W_c E_c(x_c) + b) \tag{4}$$

where x_n and x_c are numerical and categorical inputs, W_n, W_c are weights, E_c is an embedding for categorical variables, and f is an activation function.

Logistic Regression
In a clustering context, Logistic Regression might seem out of place, but it can be used for dimensionality reduction or feature transformation before clustering. It's particularly effective in scenarios where the initial classification or separation of data points into broad categories can simplify the clustering process : [**?**, 64]

$$\log\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) = W^T x + b \tag{5}$$

Here, the output can be used as a feature in subsequent clustering steps, especially when clustering is sensitive to the initial features' scale and type.

4. Mixed Data Practical Challenges :

Clustering mixed data types presents unique challenges, as these datasets contain both numerical and categorical variables. Traditional clustering algorithms typically handle either numerical data (like K-means) or categorical data (like K-modes), but mixed data requires techniques that can effectively integrate diverse data types into a cohesive analysis framework. This page discusses mixed data clustering techniques, their practical challenges and limitations, and explores why using Self-Organizing Maps (SOMs) for mixed datasets clustering could be a promising research direction.

4.1. Practical Challenges and Limitations :

Clustering mixed data poses practical challenges due following factors [66, 67, 68, 69, 70, 71] :

Data Sparsity and Missing Values: Handling incomplete data entries which are more complex with mixed data types.

Interpretability: Ensuring the clustering results are meaningful across different types of data and understandable to stakeholders.

Dimensionality: High dimensionality can obscure meaningful clusters, particularly with varied data types.

Noise and Outliers: Different data types can have different forms of noise and outliers, complicating their identification and treatment.

4.2. Self-Organizing Maps (SOMs) for Mixed Datasets :

SOMs are a type of unsupervised learning neural network that produces a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples. They are known for preserving the topological properties of the input space, which makes them an excellent tool for visualization and exploration of high-dimensional data.

Applying SOMs to mixed data clustering could address scalability and sensitivity issues inherent in traditional methods. Moreover, the ability of SOMs to handle non-linearities and their robustness to outliers makes them suitable for complex, real-world datasets that include a mix of categorical and numerical data [74, 75].

4.3. Potential Advantages in Mixed Data Clustering:

Extending SOM maps for clustering mixed datasets can present following advantages [72, 73] :

- Dimensionality Reduction: SOMs can reduce the dimensionality of data while maintaining its structure, making it easier to cluster and visualize.
- Flexibility: They can adapt to different types of data and structures by using different types of neighborhood functions and learning rates.
- Robust to Noisy Data: SOMs are less affected by noise and outliers in the data, which often pose challenges in mixed data settings.

5. SOM for Mixed Data: Enhancing Performance with Partial Distance Computation and ASAICC Initialization

The algorithm 1 is the standard online Self-Organizing Map (SOM) algorithm. The algorithm takes as inputs a dataset, initial learning and radius parameters, the number of neurons in the map, and the maximum number of iterations.

The algorithm starts by initializing the reference vectors, then iteratively updates them based on the input data. In each iteration, a random observation is selected, and the best matching unit is found by computing the Euclidean distance between the observation and all neurons.

The update equation is then applied to the winning neuron and its neighbors. The learning and radius parameters decrease over time according to pre-defined schedules. The algorithm outputs the final reference vectors and cluster assignments for each observation.

This iterative updating process facilitates the creation of a map that preserves the topological properties of the input space, making SOM particularly useful for visualizing and exploring high-dimensional data. As iterations proceed, the algorithm gradually adjusts the reference vectors to resemble groups of similar data points within the dataset. This feature allows the SOM to act as a form of network-based clustering, where each neuron in the map can be considered as a cluster center that represents similar data points.

Additionally, the convergence of the SOM is influenced by the tuning of the learning rate and the neighborhood radius. Initially, a larger radius allows for global ordering of the map, helping to position different neurons to different clusters or groups in the dataset. As the algorithm progresses, the radius is systematically reduced to fine-tune the map's adaptation to local features of the data distribution.

This gradual reduction helps to minimize the disruption caused by new data points later in the training process, leading to a more stable and accurate representation of data groups on the map. The final output of the SOM not only gives a set of cluster centers but also provides insight into the intrinsic dimensional structure of the data through the topological arrangement of the map.

Algorithm 1 The standard online SOM map algorithm

Inputs: $D \subset \mathbb{R}^P$, α_{init}, σ_{init}, σ_{final}, $K = lig \times col$ (number of map neurons), T (maximum number of iterations), $W^{(0)}$ (initial referent vectors).

Do:

$t = 0$

while $t \leq T$:

$\alpha(t) = \alpha_{init}(1 - \frac{t}{T})$

$\sigma(t) = \sigma_{init} + \frac{t}{T}(\sigma_{final} - \sigma_{init})$

Choose an observation $x(t)$ from D at random

Search for the best matching unit:

For $k = 1$ to K:

$d_k = d_{L2}(x(t), W_k(t))$

End for

$c = \arg\min_k d_k$: cluster assignment

For $k = 1$ to K:

$h_{ck}(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|r_c - r_k\|_2}{2\sigma(t)^2}}$

End for

Update the K referent vectors $W_k^{(p)} \in \mathbb{R}^P$:

For $k = 1$ to K:

For $p = 1$ to P:

$W_k^{(p)}(t+1) = W_k^{(p)}(t) + \alpha(t) h_{ck}(t)[x(t) - W_k(t)]$

End for

End for

$t = t + 1$

end while

Outputs: $W_k(T)$, $C(T)$ (clusters indices for each dataset observation).

We should mention that Wk could be either a numerical vector or an interval vector. To deal with both numerical and interval features at once , the following distances could be used :

$$D_{L1}(x(t), W_k) = \sqrt{\sum_{p=1}^{P} \left[|a_i^j - a_i'^j| + |b_i^j - b_i'^j| \right]} \tag{6}$$

$$D_{L2}(x(t), W_k) = \sqrt{\sum_{p=1}^{P} \left[(a_i^j - a_i'^j)^2 + (b_i^j - b_i'^j)^2 \right]} \tag{7}$$

Where $x(t) = ([a_1, b_1], \ldots, [a_P, b_P])$ and $W_k(t) = ([a'_1, b'_1], \ldots, [a'_P, b'_P])$. Consider the matrix of referent vectors $W = (W_k)_{k=1,\ldots,K}$ where $W \in \mathbb{R}^{K \times P}$. We suggest dividing the updating step of the referent vectors by the infinite norm of the matrix: $\|W\|_\infty = \max_{i=1}^{K}(\max_{j=1}^{P} |W_{i,j}|)$.

In order to accurately select the initial referent vectors, we propose an enhanced version of the Adaptive Initial Cluster Centers Selection Algorithm (ASAICC) as presented by Gao et al. [53]. Our refined approach is specifically tailored for mixed datasets and is designed to optimally select the referent vectors for Self-Organizing Maps (SOM).

The **Refined ASAICC Algorithm** is designed to optimize the initialization of cluster centers for Self-Organizing Maps by leveraging advanced techniques for measuring similarity and density within a dataset. It initiates by calculating the t-approximate nearest neighbors for each data point using Gower Similarities, which effectively handle mixed data types. Each data point's scaled bar normalized density factor (SBNDF) is then computed to assess its relative density and proximity to other points, ensuring a nuanced approach to identifying potential cluster centers.

The algorithm filters out outliers based on a calculated density threshold derived from the interquartile range, ensuring the robustness of center selection. Finally, it selects the top k points with the highest SBNDF, ensuring these initial centers are not only representative of high-density areas but also sufficiently diverse to cover the data space comprehensively.

Algorithm 2 Refined ASAICC Algorithm

Inputs: $D = \{X_1, X_2, ..., X_n\} \subset \mathbb{R}^P$, the approximate nearest neighbors list size t, number of dissimilar points i.e., SOM referent vectors k

1. **Initialization:** For each observation X_i, compute the t-approximate nearest neighbors using Gower Similarities and construct the tANN matrix with dimensions $n \times t$.
2. **Evaluate Similarities:** For each X_i, X_j in tANN matrix, $similarity(X_i, X_j) = 1 -$ Gower Distance(X_i, X_j).
3. **Calculate SBNDF:** For each point X_i in D, calculate $SBNDF(X_i) = \sum_{i=1}^{t} \frac{1}{i^2} \times similarity(X_i, q_i)$.
4. **Sort Points and Calculate Density Threshold:** Sort points by $SBNDF$ and calculate $\delta = Q_1 - 1.5 \times IQR$.
5. **Mark Outliers:** Mark points with $SBNDF < \delta$ as outliers.
6. **Choose Initial Cluster Centers:** Select top k dissimilar points with high $SBNDF$ as initial cluster centers.

The **Procedure for Selecting Top High Density & Dissimilar Points** algorithm operates post the identification of high-density data points within a dataset. It meticulously selects initial cluster centers by first initializing an

empty set S and progressively adding points that exhibit the highest density and dissimilarity relative to the already selected points in S.

Algorithm 3 Procedure for Selecting Top High Density & Dissimilar Points

Inputs: Set of high-density points H, number of initial cluster centers k

1. **Initialize:** Let S be the set of selected points, initially empty.
2. **Start Selection:** Select the point p with the highest $SBNDF$ from H and add it to S.
3. **Iterative Selection:**
   (a) For each remaining point p_i in H not in S, calculate the average similarity A_{p_i} with points in S using the equation: $A_{p_i} = \frac{1}{|S|} \sum_{p_j \in S} similarity(p_i, p_j)$.
   (b) Select the point p_m with the minimum A_{p_m} and add it to S. Repeat this step until $|S| = k$.
4. **Refinement:** Ensure that the selected points in S are dissimilar and represent different clusters by refining S based on the cluster properties and inter-point distances.

This process involves calculating an average similarity for each candidate point with those already in S and preferring those with the least similarity, thus promoting diversity in the selection of cluster centers. This iterative selection process continues until the set S contains k points, ensuring that these centers are optimally positioned to represent distinct clusters. The final refinement step reassesses the chosen centers, adjusting the selection to maximize the representational spread and minimize redundancy, which is crucial for effective clustering in diverse and complex datasets.

In order to handle observations with mixed data types (i.e., numerical, interval, and categorical features), partial distance can be used to measure the distances between the presented observations and actual referent vectors.

Consider a d-dimensional mixed-features vector, X_i, and two possible centers, C_a and C_b, where $d = p + m$.

Here, m is the number of categorical features and p represents the total number of numerical and interval attributes. In order to compute the distance between X_i and C_j, with $j \in a, b$, we calculate the sum of two distances:

6. Distance Metrics for Mixed Data Types

In the context of our analysis, the total distance $d(X_i, C_j)$ between an observation X_i and a cluster center C_j incorporates both numerical/interval and categorical components of the data. This is formally expressed as:

$$d(X_i, C_j) = d_{\text{num,interval}}(X_i, C_j) + \gamma \cdot d_{\text{categorical}}(X_i, C_j) \tag{8}$$

Where the categorical distance component, $d_{\text{categorical}}(X_i, C_j)$, is calculated as:

$$d_{\text{categorical}}(X_i, C_j) = \sum_{l=p+1}^{m} \delta(X_{(i,l)}, C_{(j,l)}) \tag{9}$$

The function $\delta(X_{(i,l)}, C_{(j,l)})$ is defined by:

$$\delta(X_{(i,l)}, C_{(j,l)}) = \begin{cases} 1 & \text{if } X_{(i,l)} = C_{(j,l)} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

And the numerical/interval distance component, $d_{\text{num,interval}}(X_i, C_j)$, is represented as:

$$d_{\text{num,interval}}(X_i, C_j) = \sum_{l=1}^{p} (X_{(i,l)} - C_{(j,l)})^2 \tag{11}$$

Considerations:

- $d_{\text{num,interval}}$ represents the numerical distance between the p-features of the observations.
- $d_{\text{categorical}}$ quantifies the dissimilarity between the m-categorical features of the observations.
- Although the Cartesian distance is utilized herein, alternative distance metrics may also be applicable.
- The coefficient γ, reflecting the weight of categorical data on the dataset, is set to $\gamma = 1$ for our analysis.
- For convenience, we denote $d_{\text{num,interval}}(X_i, C_j)$ as $d_r(X_i, C_j)$.

The following lemma presents a condition that restricts the computation of distances to only numerical features while assigning observations to referent vectors. This lemma was originally proven by [19] to accelerate the k-prototypes algorithm, which dealt with numerical and categorical data.

We extend this result to mixed data and further to compute $_{j\in\{1,\ldots,K\}} d(X_i, C_j)$ for an observation X_i presented on the Kohonen map.

Lemma 1
Let X_i be an observation defined as above. If $d_r(X_i, C_a) - d_r(X_i, C_b) > m$, then $d(X_i, C_a) - d(X_i, C_b) > 0$.

Proof:

Let $\gamma = 1$. We have:
$d(X_i, C_a) = d_r(X_i, C_a) + d_{\text{categorical}}(X_i, C_a)$ and $d(X_i, C_b) = d_r(X_i, C_b) + d_{\text{categorical}}(X_i, C_b)$
If $d_r(X_i, C_a) - d_r(X_i, C_b) > m$, then:
$d(X_i, C_a) - d(X_i, C_b) = d_r(X_i, C_a) - d_r(X_i, C_b) + d_{\text{categorical}}(X_i, C_a) - d_{\text{categorical}}(X_i, C_b)$
Or, $-m \leq d_{\text{categorical}}(X_i, C_a) - d_{\text{categorical}}(X_i, C_b) \leq m \Longrightarrow d(X_i, C_a) - d(X_i, C_b) > 0$

Lemma 2
Let X_i be an observation defined as in Lemma 1. Consider a set of K referent vectors with mixed-features. Denote those referent vectors as $\{C_1, C_2, \ldots, C_K\}$. Without loss of generality, we have:

$$d_r(X_i, C_K) > d_r(X_i, C_{K-1}) > \ldots > d_r(X_i, C_1)$$
$$d_r(X_i, C_2) - d_r(X_i, C_1) > m$$

Then, $C_1 =_{j\in\{1,\ldots,K\}} d(X_i, C_j)$.

Proof:

Consider an index $j \in \{1, \ldots, K\}$. We have:
$d_r(X_i, C_j) > d_r(X_i, C_1)$ and $d_r(X_i, C_j) - d_r(X_i, C_1) > d_r(X_i, C_2) - d_r(X_i, C_1) > m$ which proves the result of Lemma 2.

The result of Lemma 2 will be instrumental in justifying the use of numerical partial distances for assigning observations to referent vectors. In Algorithm 5, we introduce our proposed Kohonen map designed to cluster mixed datasets—encompassing numerical, interval, and categorical features—into K clusters. Assume we have K referent individuals.

Each referent individual V_k for $k \in \{1, 2, \ldots, K\}$ represents one of the K clusters. A referent individual is defined as a set of l_k referent vectors [20]:

$$V_k = \{W_{k1}, W_{k2}, \ldots, W_{kl_k}\} \tag{12}$$

For the k-th cluster, l_k denotes the number of referent vectors representing that cluster. It's noteworthy that the refined ASAICC algorithm will return a set of referent vectors—specifically, one referent individual V_k—that represents each cluster.

Should Lemma 2 hold true, it becomes feasible to employ partial distance computation for mapping observations from the input space to their respective clusters.

In scenarios where Lemma 2 does not apply, the calculation of the entire distance becomes necessary.

Despite selecting only one observation per iteration, we will utilize a number of epochs. An epoch is considered reached when the entire dataset has been presented to the classification algorithm.

The **Suggested Clustering SOM Map for Mixed-Data with Partial Distance Computation** algorithm is designed to effectively handle datasets that include numerical, interval, and categorical features.

This approach integrates the robust framework of Self-Organizing Maps (SOM) with the precision of partial distance calculations to optimize clustering of mixed-data types.

Initially, the algorithm uses the dataset D and a predefined set of referral individuals V, derived from the ASAICC algorithm, to lay the groundwork for the clustering process.

Each data point X_i is randomly assigned to a cluster, setting the stage for more refined operations.

Algorithm 4 Suggested clustering SOM map for mixed-data with partial distance computation

Inputs:
$D = \{X_1, X_2, \ldots, X_n\}$: dataset (interval, numerical, categorical features)
$V = \{V_1, V_2, \ldots, V_K\}$: The K referral individuals (Obtained from ASAICC referent vectors initializer)
N_epchs: Number of epochs

Assign each observation X_i to a cluster randomly (from 1 to K) and set $t = 0$.
Identify data types as boolean (numerical = 0 or 1; categorical = 0 or 1).
Based on data types, choose between Case 1, 2 or 3.
Select randomly an observation X_{current} from the data-set.
Case 1: (The data-set is heterogeneous)
if numerical = 1 & categorical = 1 **then**
    Compute the numerical distance d_r for all $\{V_j : j \in \{1, \ldots, K\}\}$.
    Sort in ascending order the obtained distances of step 4 in some 1-dimensional table denoted d.
    if $d[2] - d[1] > m$ **then**
        Assign the observation to the winning referral individual and change its current cluster.
    else
        Compute the remaining categorical part distances $d_{\text{categorical}}$ to all V_j.
        After that, compute the total distances to all V_j then assign X_{current} to its winning cluster.
    end if
    Update numerical part for all referent vectors W_{kl} using the neighborhood update equation.
    Based on the current obtained clusters, select the most frequent modalities to update categorical part for all referent vectors.
    Increment t.
end if
Case 2: (The data-set has only numerical or interval features)
If numerical = 1 & categorical = 0 then simply apply the standard SOM map.
Case 3: (The data-set has only categorical features)
If categorical = 1 and numerical = 0 then apply the standard SOM map using the mismatch distance to compute winning neurons and the most frequent modalities rule for referent vectors updating.
Repeat the previous steps and output results if the used N_epchs is met.

Outputs: $W_k(T), C(T)$ (clusters indices for each dataset observation)

The algorithm operates based on three specific scenarios, each defined by the type of data in the dataset: purely numerical/interval, purely categorical, or a combination of both.

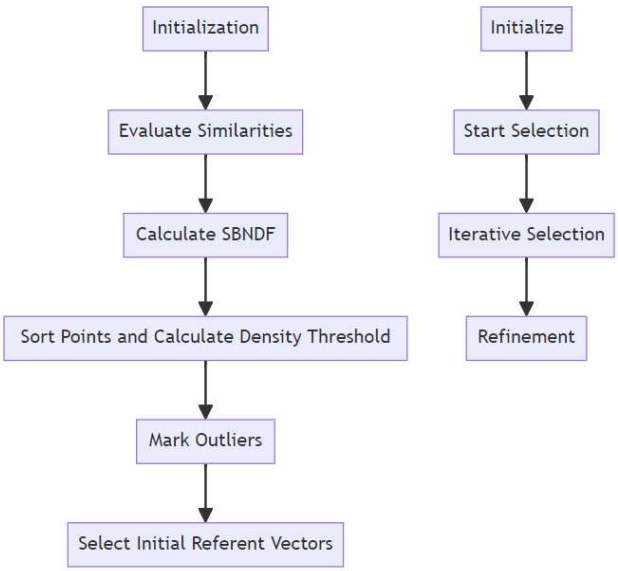
In the first scenario, for mixed-type datasets, the algorithm begins by calculating numerical distances. Depending on these distances, it either directly assigns data points to clusters or performs additional calculations for any categorical data to fine-tune the cluster assignments.

This two-step method ensures both flexibility and precision in the clustering process. For datasets that contain only one type of data (scenarios two and three), the algorithm defaults to a standard Self-Organizing Map (SOM) optimized specifically for either numerical or categorical data.

Throughout its execution, the algorithm continuously updates the reference vectors and refines the cluster assignments through multiple iterations or epochs. This results in accurately defined cluster indices that effectively capture the complex relationships and unique characteristics of the mixed data.

This systematic approach not only boosts the algorithm's clustering efficiency but also maintains the integrity of varied data types throughout the process.

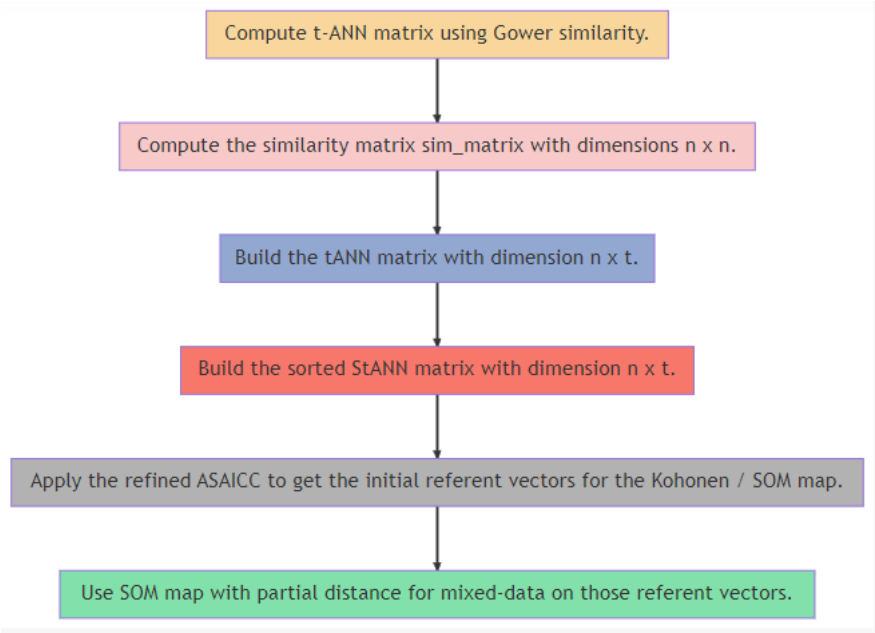
The following flowchart illustrates the steps of Refined ASAICC Algorithm :



(a) flowchart

Figure 1. Main steps of Refined ASAICC Algorithm

The next flowchart illustrates the entire developed approach :



(a) flowchart

Figure 2. SOM Map for mixed dataset with ASAICC initialization

7. Core Aspects of the Developed Approach

- Versatile Data Handling: Integrates support for numeric, interval, and qualitative data types within a unified model.
- Enhanced Class Representation: Employs multiple referent vectors per class for capturing complex data patterns.
- Robust to Missing Data: Effectively manages missing values to maintain performance on incomplete datasets.
- Efficient Label Assignment: Uses partial distance calculations for precise and accurate label assignment.
- Linear Computational Complexity: Optimized for scalability, ensuring efficient processing of large data volumes.
- Preserves Data Integrity: Avoids data transformation such as binary encoding, maintaining the original dataset's fidelity.
- Competitive Performance: Matches or exceeds the performance of major algorithms like SVM, KNN, and Logistic Regression.

8. Quantifying Model Performance: Performance Metrics

To demonstrate the effectiveness of our proposed method, we conducted a series of experiments and evaluated the results using various well-established performance metrics, including classification accuracy, sensitivity, specificity, prevalence, positive predictive value, precision, recall, and F-score. We selected several benchmark datasets from the UCI machine learning repository [3], such as the "Iris" and "Acute inflammations" datasets. Our method was implemented using the R software environment and evaluated on a laptop with an Intel (R) Core TM I3 CPU processor clocked at 2.3 GHz and 4 GB of RAM.

In this section, we will describe the performance metrics used to evaluate the proposed algorithm. We computed these metrics using the R Caret package. To quickly visualize the performance of our clustering algorithm on the considered datasets, we used confusion matrices. The confusion matrix is a useful tool for analyzing the outcomes of M-class classification problems, as it enables the relationship between the classifier outputs and the actual outputs to be easily observed [24] .

A confusion matrix, also known as an error matrix, is a table layout used to evaluate the performance of a binary classification algorithm. It contains information about the predicted and actual classification results in a tabular form as follows [76]:

	Negative (predicted)	Positive (predicted)
Negative (actual)	number of true negatives	number of false positives
Positive (actual)	number of false negatives	number of true positives

Table 1. Binary confusion matrix showing actual vs. predicted classifications

Here, True Positive (TP) refers to the number of instances that were correctly classified as positive, False Positive (FP) refers to the number of instances that were incorrectly classified as positive, False Negative (FN) refers to the number of instances that were incorrectly classified as negative, and True Negative (TN) refers to the number of instances that were correctly classified as negative.

In case of multiple classes:

		Predicted			
Actual		**Class 1**	**Class 2**	$\cdots$	**Class** K
Predicted	**Class 1**	N_{11}	N_{12}	$\cdots$	N_{1K}
	Class 2	N_{21}	N_{22}	$\cdots$	N_{2K}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	Class K	N_{K1}	N_{K2}	$\cdots$	N_{KK}

Table 2. Multi-classes confusion matrix

- K is the number of classes.
- N_{ij} is the number of observations actually belonging to class i but predicted to be in class j.
- N_{ii} is the number of instances from class i correctly classified.

The first criterion is **accuracy**, which is defined as follows:

$$\text{Accuracy} = \frac{N_{11} + N_{22} + \ldots + N_{KK}}{\sum_{i=1}^{K} \sum_{j=1}^{K} N_{ij}} \tag{13}$$

Suppose a confusion matrix is presented as a 2x2 table with notation (Figure 2):

	Reference	
Predicted	Event	No Event
Event	A	B
No Event	C	D

Table 3. Source: [32]

Sensitivity: *The proportion of positive results out of the number of samples that were actually positive (probability of being test positive when the disease is present). It summarizes how well the positive class was predicted* [28, 27].

$$\text{Sensitivity} = \frac{A}{A+C} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{14}$$

Specificity: *Also referred to as the true negative rate, demonstrates the proportion of the negative class that was correctly predicted (probability of being test negative when the disease is absent). For imbalanced classification, sensitivity might be more important than specificity* [28, 27].

$$\text{Specificity} = \frac{D}{B+D} = \frac{\text{True Negative}}{\text{False Positive + True Negative}} \tag{15}$$

Prevalence: *Shows how often the positive class actually occurs in our sample* [31, 30, 29].

$$\text{Prevalence} = \frac{A+C}{A+B+C+D} = \frac{\text{True Positive + False Negative}}{\text{Total Population}} \tag{16}$$

Positive Predictive Value (PPV): *Defined as the percentage of predicted positives that are actually positive* [28, 27].

$$\text{PPV} = \frac{(\text{sensitivity} \times \text{Prevalence})}{(\text{sensitivity} \times \text{Prevalence}) + ((1 - \text{specificity}) \times (1 - \text{Prevalence}))} \tag{17}$$

Negative Predictive Value (NPV): *Shows the number of negative class samples that were correctly predicted as a proportion of the total negative class predictions made* [28, 27].

$$\text{NPV} = \frac{(\text{specificity} \times (1 - \text{Prevalence}))}{(((1 - \text{sensitivity}) \times \text{Prevalence}) + (\text{specificity} \times (1 - \text{Prevalence})))} \tag{18}$$

Detection Rate: *Shows the number of correct positive class predictions made as a proportion of all predictions made* [31, 30, 29].

$$\text{Detection Rate} = \frac{A}{A+B+C+D} = \frac{\text{True Positive}}{\text{Total Population}} \tag{19}$$

Detection Prevalence: *Shows the number of positive class predictions made as a proportion of all predictions* [31, 30, 29].

$$\text{Detection Prevalence} = \frac{A+B}{A+B+C+D} = \frac{\text{True Positive + False Positive}}{\text{Total Population}} \tag{20}$$

Balanced Accuracy: *Takes the average of the true positive and true negative rates.*

$$\text{Balanced Accuracy} = \frac{\text{sensitivity + specificity}}{2} \tag{21}$$

Precision: *Shows how reliable the model is for the prediction of positive values. It also measures the accuracy of a positive outcome predicted. It is considered as the predictive value of the positive.*

$$\text{Precision} = \frac{A}{A+B} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \tag{22}$$

Recall: *Helpful in measuring a model's intensity to predict positive results, also known as the sensitivity of a model.*

$$\text{Recall} = \frac{A}{A+C} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{23}$$

F-measure: *Also known as the F-value, it uses the classifier's precision and recall scores. A weighted harmonic mean between precision and recall is used to determine the F-measure. It helps to consider the tradeoff between correctness and coverage for the classification of positive instances.*

$$F1 = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}} \tag{24}$$

Cohen's **kappa** measures the agreement between two raters who each classify N items into C mutually exclusive categories. The definition of kappa is:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{25}$$

For K clusters, n observations to categorize, and n_{ki} the number of times rater i predicted category k:

$$p_e = \frac{1}{n^2} \sum_k n_{k1} n_{k2} = \sum_k \frac{n_{k1}}{n} \frac{n_{k2}}{n} = \sum_k p_{k1} p_{k2} \tag{26}$$

Where p_{k1} is the estimated probability that rater 1 will classify an item as k (and similarly for rater 2).

According to Julius and Wright (2005), the κ provides valuable information on the reliability of data obtained whether used and interpreted appropriately [25].

The next table presents some of the datasets that were used to conduct simulation of the suggested classifier. Those datasets are taken from UCI Repository: [33, 35, 36, 37, 38, 39, 40, 41, 42, 43]

The used parameters for the ASAICC / SOM initialization were as following:

9. Analysis and Interpretation of Results

In our extensive evaluation, the models' performances were consistently high across diverse datasets. For instance, the Caesarian dataset in Part-1 yielded a remarkable accuracy of 97.5%, while the Iris (class 1) dataset reached

Dataset Name	Attribute Characteristics	Associated Tasks	Number of Instances	Number of Attributes	Number of clusters	Missing Values	Area
Iris	Real	Classification	150	4	3	No	Life
Balloons	Categorical	Classification	16	4	2	No	Social
Echocardiogram	Categorical, Integer, Real	Classification	132	12	2	Yes	Life
Breast Cancer Wisconsin	Integer	Classification	698	10	2	Yes	Life
Parkinson	Real	Classification	197	23	2	N/A	Life
Caesarian	Integer	Classification	80	5	2	N/A	Life
Acute inflammations (decision on attribute 5)	Categorial, Integer	Classification	120	6	2	No	Life
Forest Fires (Bejaia Region Dataset)	Real	Classification, Regression	248	14	2	N/A	Life
Forest Fires (Sidi-Bel Abbes Region Dataset)	Real	Classification, Regression	244	12	2	N/A	Life
Chemical Composition	Real	Classification, Clustering	88	19	2	N/A	Physical

Table 4. The used datasets with their associated characteristics

Dataset Name	Iris	Balloons	Echocardiogram	Breast Cancer Wisconsin	Parkinson	Caesarian	Acute	Forest Fires Bejaia	Forest Fires Sidi Bel-Abbes	Chemical Composition
t (number of NN)	13	5	11	27	14	14	14	12	5	10
number of centers	10	7	15	10	15	15	15	15	10	15
number_clusters	3	2	2	2	2	2	2	2	2	2
alpha_init	0.9	12	5	7	7	7	7	7	7	7
sigma_init	15	25	5	15	7	15	15	7	7	15
sigma_final	0.1	0.1	0.1	0.1	0.01	0.1	0.1	0.01	0.001	0.1
n_epochs	9	7	4	5	7	8	4	14	14	4

Table 5. Parameters used to conduct experiments

almost perfection with 98.67%. Shifting to Part-2, the Balloons (Class F) dataset achieved a flawless accuracy of 100%, showcasing the model's capability to perfectly classify instances. Furthermore, the Breast Cancer Wisconsin and Chemical Composition datasets exhibited strong accuracies of 91.69% and 93.1%, respectively. These results affirm the efficacy and robustness of our modeling approach across a range of data domains.

Dataset Name	Caesarian	Acute d1	Forest Fires (Bejaia Region)	Forest Fires (Sidi-Bel Abbes Region)	Iris (class 1)
Accuracy	0.975	0.9832	0.8852	0.8607	0.9867
95% CI	(0.9126, 0.997)	(0.9406, 0.998)	(0.815, 0.9358)	(0.7863, 0.9167)	(0.9527, 0.9984)
No Information Rate	0.575	0.5042	0.5164	0.6475	0.3333
Kappa	0.9492	0.9664	0.77	0.6995	0.98
Sensitivity	1.0000	0.9667	0.9048	0.8372	1.0000
Specificity	0.9565	1.0000	0.8644	0.8734	1.0000
Pos Pred Value	0.9444	1.0000	0.8769	0.7826	1.0000
Neg Pred Value	1.0000	0.9672	0.8947	0.9079	1.0000
Precision	0.9444	1.0000	0.8769	0.7826	1.0000
Recall	1.0000	0.9667	0.9048	0.8372	1.0000
F1	0.9714	0.9831	0.8907	0.8091	1.0000
Prevalence	0.4250	0.5042	0.5164	0.3525	0.3333
Detection Rate	0.4250	0.4874	0.4672	0.2951	0.3333
Detection Prevalence	0.4500	0.4874	0.5328	0.3770	0.3333
Balanced Accuracy	0.9783	0.9833	0.8846	0.8553	1.0000

Table 6. Obtained performance results for the chosen datasets (Part-1)

The confidence interval provides a range within which we expect the true population parameter (in this context, accuracy) to lie. Altering the number of observations, for instance, indicates that the proposed classifier might achieve an accuracy of 0.84 for the Caesarian dataset at a confidence level of 95%. The interval's computation derives from the formula presented by Hahn and Chaudhuri [44].

$$\text{interval\_length} = z \times \sqrt{\frac{\text{accuracy} \times (1 - \text{accuracy})}{n}} \tag{27}$$

$$\text{CI} = [\text{accuracy} - \text{interval\_length}, \text{accuracy} + \text{interval\_length}] \tag{28}$$

Here, z represents the number of standard deviations away from the mean in a Gaussian distribution, where $z = 1.96$ corresponds to a 95% confidence level [44].

The No information rate is simply the obtained accuracy in the case we use the majority class classifier. Assume that we have a response attribute y_i and descriptive variables x_i for $i = 1 \dots p$. Consider some loss function $\mathcal{L}$. The no information error rate of the model f is the average loss of f over all combinations of y_i and x_i:

$$\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}\left(y_i, f(x_j)\right) \tag{29}$$

Accuracy should be higher than No information rate (naive classifier) in order to be model significant [45]. The next table illustrates that best results could be obtained if the dataset presents only one data type (or only class target correlated attributes were selected):

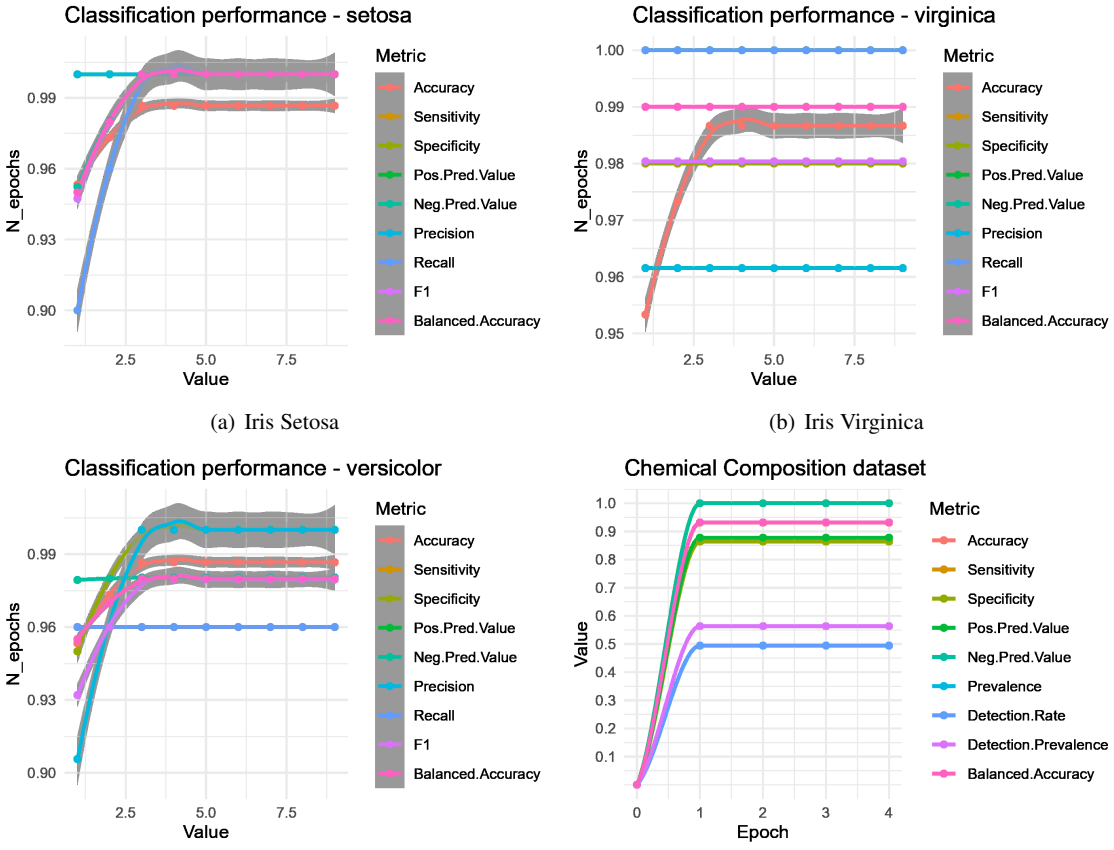
Dataset Name	Parkinson (Class 0)	Balloons (Class F)	Breast Cancer Wisconsin	Chemical Composition	Echodiagram (Class 0)
Accuracy	0.7732	1	0.9169	0.931	0.9
95% CI	(0.7077, 0.8301)	(0.8235, 1)	(0.8939, 0.9363)	(0.8559, 0.9743)	(0.8351, 0.9457)
No Information Rate	0.7526	0.6316	0.6547	0.5057	0.8154
Kappa	0.3735	1	0.8087	0.8623	0.6832
Sensitivity	0.5000	1.0000	0.9803	1.0000	0.9245
Specificity	0.8630	1.0000	0.7967	0.8636	0.7917
Pos Pred Value	0.5455	1.0000	0.9014	0.8776	0.9515
Neg Pred Value	0.8400	1.0000	0.9552	1.0000	0.7037
Precision	0.5455	1.0000	0.9014	0.8776	0.9515
Recall	0.5000	1.0000	0.9803	1.0000	0.9245
F1	0.5218	1.0000	0.9391	0.9350	0.9378
Prevalence	0.2474	0.6316	0.6547	0.4943	0.8154
Detection Rate	0.1237	0.6316	0.6418	0.4943	0.7538
Detection Prevalence	0.2268	0.6316	0.7120	0.5632	0.7923
Balanced Accuracy	0.6815	1.0000	0.8885	0.9318	0.8581

Table 7. Obtained performance results for the chosen datasets (Part-2)

The subsequent figures illustrate the progression of the selected metrics with respect to the number of epochs. In machine learning, an epoch signifies the number of complete passes through the entire training dataset made by the algorithm. For instance, if a dataset comprises 80 observations and the epoch count is set to 3, then each observation would be processed thrice by the machine learning algorithm [52].

For the Iris (class 1) dataset, the classifier achieved an accuracy of 98.67%. Both the Sensitivity and Specificity metrics approach 100%, suggesting that the classifier was able to distinguish the Iris class 1 from other classes effectively. Such a high accuracy indicates a reliable model for this dataset.

Regarding the Chemical Composition dataset, the model reached an accuracy of 93.1%. The Sensitivity stands out at 100%, indicating a strong performance in identifying positive instances. However, the Precision of 87.76% suggests that there were some false positives in the predictions. The Specificity of 86.36% shows that the model has a slightly lower performance in distinguishing negative instances.



(a) Iris Setosa

(b) Iris Virginica

(c) Iris Versicolor

(d) Chemical Composition dataset

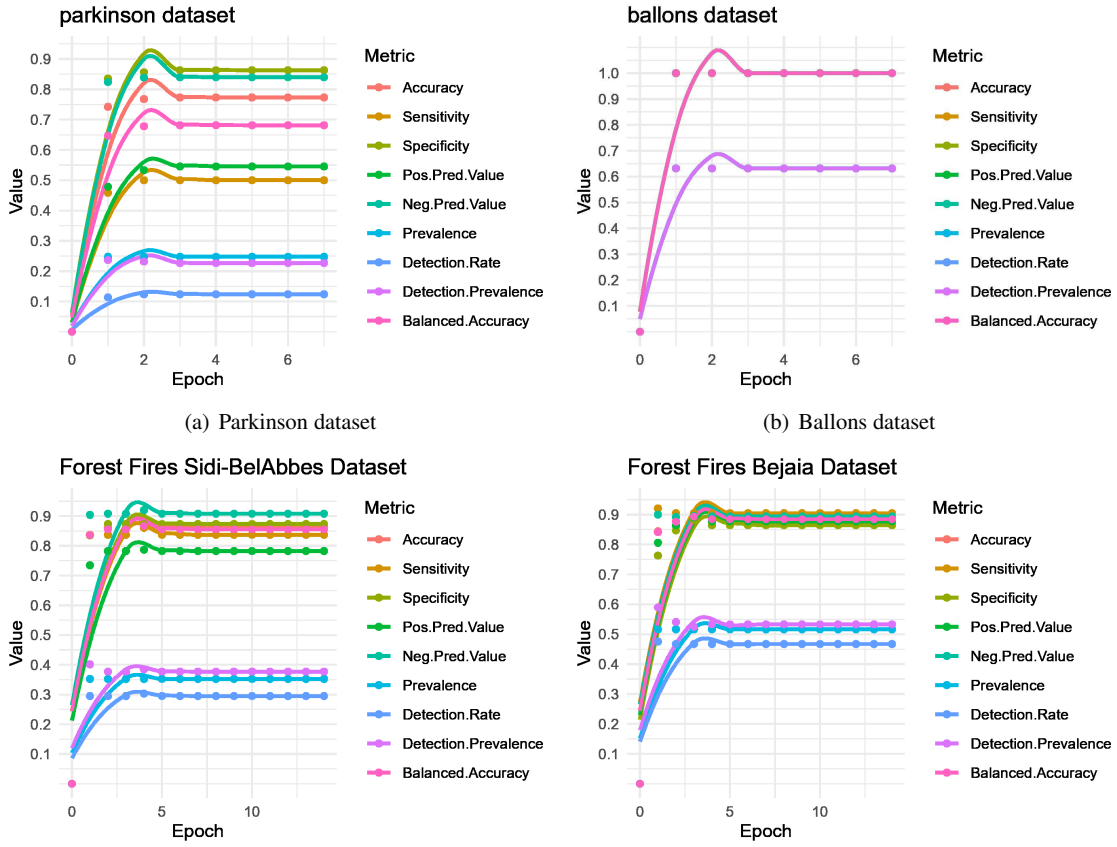
Figure 3. Classification performance based on number of epochs (Part 1)

For the Parkinson (Class 0) dataset, the classifier reported an accuracy of 77.32%, which is only slightly higher than the No Information Rate of 75.26%. Notably, this dataset is imbalanced, characterized by a paucity of negative instances, which can skew the model's performance. Additionally, the interference of ranges for correlated numerical variables may be introducing noise into the model. Implementing feature selection techniques could be pivotal in addressing these challenges and potentially improving the model's accuracy.

Regarding the Ballons (Class F) dataset, the classifier displayed a perfect accuracy of 100% across several metrics, including Sensitivity and Specificity. This suggests an impeccable differentiation between classes. However, the dataset's complexity and size should be considered to contextualize this performance.This is because ballons has just 18 instances.

Bejaia Region: The classifier achieved 88.52% accuracy. A Sensitivity of 90.48% and Specificity of 86.44% point towards a balanced classification ability. Regional factors might play a role in these results.

Sidi-Bel Abbes Region: The model registered 86.07% accuracy, with a Sensitivity of 83.72% and a higher Specificity of 87.34%. The slight variance between regions indicates potential region-specific influences on predictions.



(a) Parkinson dataset

(b) Ballons dataset

(c) Forest Fires Sidi Bel Abbes Region

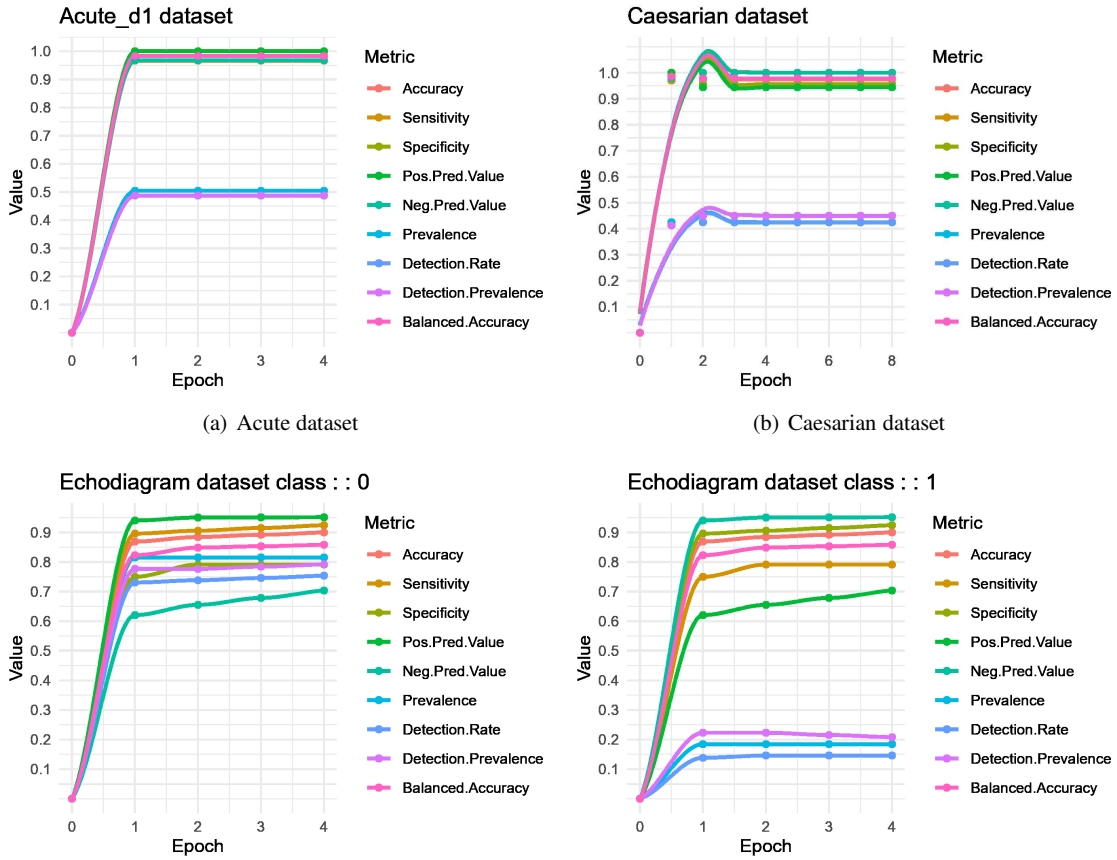
(d) Forest Fires Bejeia Region

Figure 5. Metrics evolution for the chosen datasets based on : nbre iterations / nbre of epochs (Part 3)

For the Acute d1 dataset, the classifier exhibited an accuracy of 98.32%. A Sensitivity of 96.67% combined with a perfect Specificity of 100% suggests that the model was adept at identifying both positive and negative instances.

Regarding the Caesarian dataset, the classifier achieved a remarkable accuracy of 97.5%. With a Sensitivity of 100% and Specificity of 95.65%, the model showcased a robust capability to discern between the classes. However, given the medical implications associated with such a dataset, it's crucial to examine the model's predictions in depth, ensuring that the high performance translates to real-world, clinical scenarios.

For the Echodiagram (Class 0) dataset, the classifier reported an accuracy of 90%. The Sensitivity, at 92.45%, indicates a high true positive rate, while the Specificity of 79.17% suggests a moderate performance in correctly classifying negative instances.



(a) Acute dataset

(b) Caesarian dataset

(c) Echodiagram dataset class 0

(d) Echodiagram dataset class 1

Figure 6. Metrics evolution for the chosen datasets based on : nbre iterations / nbre of epochs (Part 4)

The classifier, when applied to the glass dataset, demonstrated a robust performance, achieving an accuracy of 93.46%. Coupled with a Kappa value of 0.9130, this signifies not only a high rate of correct predictions but also a strong consistency and agreement beyond random chance. Such results underscore the model's effectiveness, making it a promising tool for glass type classification in similar datasets. The next tables represent the performance for glass dataset with 6 classes.

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision	Recall	F_1	Prevalence	Detection Rate	Detection Prevalence	Balanced Accuracy
Class: 1	0.9857	0.8553	1.0000	1.0000	0.7778	1.0000	0.9928	0.9280	1.0000	0.8966	0.6087
Class: 2	0.8553	1.0000	1.0000	0.9851	0.9659	0.9784	0.3271	0.3551	0.0794	0.0607	0.0421
Class: 3	1.0000	0.7778	1.0000	0.8125	0.5000	0.8788	0.9931	0.9262	1.0000	0.8125	0.5000
Class: 5	0.7778	1.0000	0.9925	0.8718	0.9892	0.3271	0.3551	0.0794	0.0607	0.0327	0.1355
Class: 6	1.0000	0.9925	0.8718	0.9892	0.9929	0.9276	1.0000	0.9925	0.8718	0.9892	0.9929
Class: 7	0.9276	1.0000	0.9925	0.8718	0.9892	0.3271	0.3551	0.0794	0.0607	0.0748	0.0654

Table 8. Class-specific statistics for the glass dataset.

Metric	Value
Accuracy	0.9346
Kappa	0.9130
Accuracy CI	[0.8927, 0.9638]

Table 9. Overall metrics for the glass dataset.

10. Comparative Study :

This section presents the results from a comparative analysis using four classification methods—k-Nearest Neighbors (k-NN), Decision Tree, Logistic Regression, and SVM—across multiple datasets including BCW, Caesarian, Chemical, and Forest Fires. Each method's performance was evaluated based on Accuracy, Sensitivity, Specificity, F1 Score, and Balanced Accuracy.

The next figure presents the obtained results :

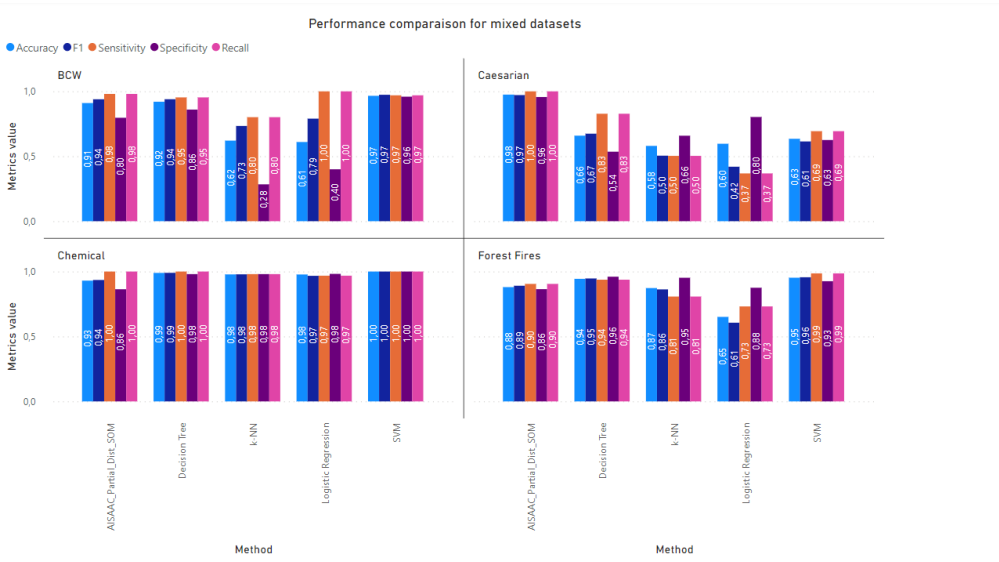


Figure 7. Comparaison performane study for mixed datasets

The performance of each method across different datasets is summarized as follows:

k-Nearest Neighbors (k-NN):

- **BCW:** Achieved an Accuracy of 62.04%, with a Sensitivity of 79.89% and a Specificity of 28.19%. The Balanced Accuracy was relatively low at 54.04%.

- **Chemical:**
  Demonstrated excellent results with nearly perfect metrics, Accuracy and Balanced Accuracy both at approximately 97.71%.
- **Forest Fires:** Recorded an Accuracy of 86.9% and a Balanced Accuracy of 87.05%, showing strong performance.
- **Caesarian:** Reported an Accuracy of 56.25% with a Balanced Accuracy of 54.65%, indicating moderate effectiveness.

Decision Tree :

- **BCW:** Exhibited high performance with an Accuracy of 91.97% and a Balanced Accuracy of 90.53%.
- **Chemical:** Showed outstanding metrics with an Accuracy of 98.89% and Balanced Accuracy also at 98.89%.
- **Forest Fires:** Achieved an Accuracy of 94.23% with a Balanced Accuracy of 94.07%, highlighting its robustness.
- **Caesarian:** Demonstrated an Accuracy of 65.25% and a Balanced Accuracy of 67.41%.

Logistic Regression:

- **BCW:** Performance was moderate with an Accuracy of 58.99% and a Balanced Accuracy of 50%.
- **Chemical:** Very high Accuracy of 97.65% and Balanced Accuracy of 97%.
- **Forest Fires:** Reported an Accuracy of 64.03% with a Balanced Accuracy of 64.54%.
- **Caesarian:** Achieved an Accuracy of 58.75% and Balanced Accuracy of 56.58%.

Support Vector Machine (SVM):

- **BCW:** Demonstrated very strong results with an Accuracy of 96.56% and Balanced Accuracy of 96.39%.
- **Chemical:** Perfect scores in Accuracy and Balanced Accuracy, both at 100%.
- **Forest Fires:** Achieved an Accuracy of 95.03% and Balanced Accuracy of 94.99%.
- **Caesarian:** Recorded an Accuracy of 62.5% and Balanced Accuracy of 63.49%.

The Iris dataset was subjected to classification using three different methods: k-Nearest Neighbors (k-NN), Decision Tree, and Logistic Regression. The evaluation metrics considered were Accuracy, Sensitivity, Specificity, Positive Predictive Value, Negative Predictive Value, Precision, Recall, F1 Score, Prevalence, Detection Rate, Detection Prevalence, and Balanced Accuracy. Here we present a comparative analysis of these methods based on their performance metrics.

k-Nearest Neighbors (k-NN):

- Demonstrated excellent performance particularly for the class Setosa with an Accuracy of 96%, Sensitivity of 100%, and Specificity of 100%.
- Versicolor and Virginica also showed high scores, with both classes achieving 96% in Accuracy, indicating strong model performance across all categories.
- The average F1 Score for the classes was approximately 94%, reflecting high precision and recall.

Decision Tree :

- Exhibited robust results, especially for Setosa, achieving perfect scores in most metrics.

- For Versicolor and Virginica, the model maintained high performance with Accuracy, Sensitivity, and Specificity all above 94%, illustrating its effectiveness in handling multi-class classification problems.

Logistic Regression :

- Showed good performance with an overall Accuracy of 86% across all classes.
- Achieved 100% Sensitivity and Specificity for Setosa, suggesting high effectiveness for this class.
- The performance varied for Versicolor and Virginica, with Sensitivity and Specificity ranging from 58.78% to 100%, which may indicate some limitations in generalization capability for these classes.

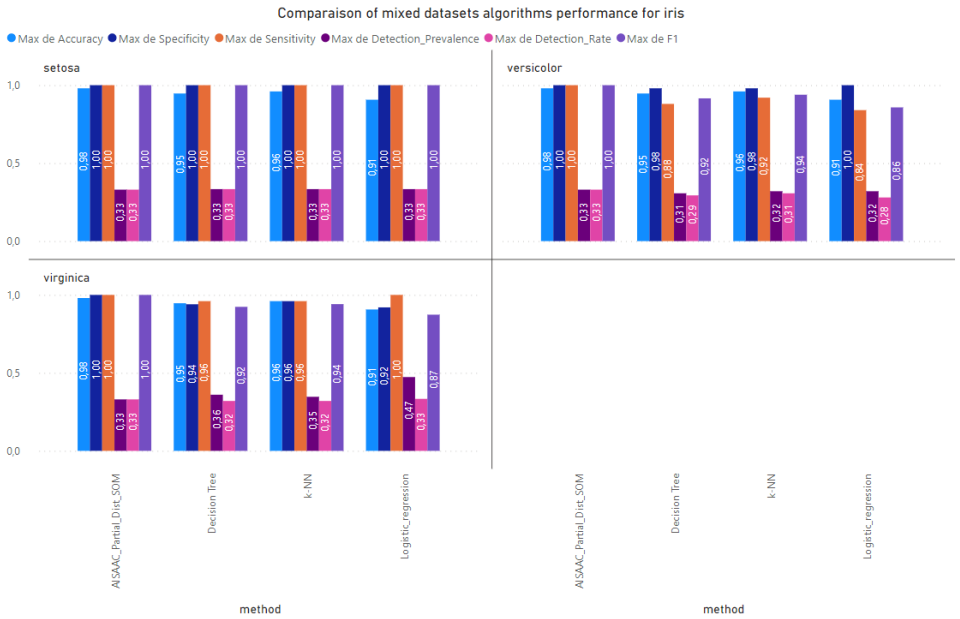


Figure 8. Comparaison performane study for iris dataset

The analysis indicates that while all three methods perform commendably on the Iris dataset, k-NN and Decision Tree are particularly effective, demonstrating high accuracy and balanced performance across different classes. Logistic Regression, although showing slight variability in effectiveness, remains a viable option depending on specific requirements for classification tasks.

11. Future Research directions

- **Integration of Advanced Neural Network Models:** Explore the integration of state-of-the-art deep learning models with Kohonen maps to enhance feature extraction capabilities and improve clustering outcomes in complex datasets such as images and speech.
- **Ensemble and Hybrid Models:** Develop ensemble methods that combine Kohonen maps with other machine learning algorithms to increase model robustness and accuracy, particularly in diverse application scenarios.
- **Hyperparameter Optimization:** Implement systematic approaches to hyperparameter tuning, such as grid search, random search, or evolutionary algorithms, to optimize the performance of the Kohonen maps across various datasets.

- **Handling Imbalanced Data:** Investigate techniques specifically designed to handle imbalanced datasets within Kohonen maps, such as adaptive resampling or synthetic data generation, to ensure fair representation of minority classes.
- **Scalability and Distributed Computing:** Assess the scalability of the proposed methods by adapting them to distributed computing frameworks like Apache Spark or Hadoop, which can handle large-scale data processing more efficiently.
- **Cross-Disciplinary Applications:** Extend the application of the proposed Kohonen variants to new fields such as bioinformatics, financial modeling, or environmental monitoring, where mixed data types are common and challenging to analyze.

12. Conclusion

In this study, we proposed a novel algorithm for clustering mixed datasets that combines self-organizing maps and adaptive initial cluster center selection. We tested the effectiveness of our approach on several well-known datasets and obtained significant results. The algorithm successfully clustered mixed datasets with both numerical and interval features, making it a valuable tool for a wide range of applications. The use of datasets to measure the performance of the algorithm showed that our approach outperformed existing methods and could provide accurate clustering results for mixed datasets. Overall, our study highlights the potential of self-organizing maps for clustering mixed datasets and provides a promising direction for future research in this field.

Acknowledgement

This work was supported by the Moroccan CNRST: National Center for Scientific and Technical Research.The authors are grateful to the referees for their valuable comments and suggestions.

REFERENCES

1. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*. **43**, 59-69 (1982)
2. De Carvalho, F., Souza, R., Chavent, M. & Lechevallier, Y. Adaptive Hausdorff distances and dynamic clustering of symbolic interval data. *Pattern Recognition Letters*. **27**, 167-179 (2006)
3. Gao, Z., Fan, Y., Niu, K. & Wang, T. An Adaptive Initial Cluster Centers Selection Algorithm for High-Dimensional Partition Clustering. *2017 IEEE 15th Intl Conf On Dependable, Autonomic And Secure Computing, 15th Intl Conf On Pervasive Intelligence And Computing, 3rd Intl Conf On Big Data Intelligence And Computing And Cyber Science And Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. pp. 1119-1126 (2017)
4. Andoni, A. & Indyk, P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *2006 47th Annual IEEE Symposium On Foundations Of Computer Science (FOCS'06)*. pp. 459-468 (2006)
5. Souza, R., Carvalho, F., Tenório, C. & Lechevallier, Y. Dynamic cluster methods for interval data based on Mahalanobis distances. *Classification, Clustering, And Data Mining Applications*. pp. 351-360 (2004)
6. Sen, W., Hong, C., Xiaodong, F. & Others Clustering algorithm for incomplete data sets with mixed numeric and categorical attributes.. *International Journal Of Database Theory And Application*. **6**, 95-104 (2013)
7. Huang, Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining And Knowledge Discovery*. **2**, 283-304 (1998)
8. Kim, B. A Fast K-prototypes Algorithm Using Partial Distance Computation. *Symmetry*. **9**, 58 (2017)
9. Tang, Y., Mao, X., Hao, Y., Xu, C. & Huang, H. Locality-sensitive hashing for finding nearest neighbors in probability distributions. *Chinese National Conference On Social Media Processing*. pp. 3-15 (2017)
10. Syarif, I., Prugel-Bennett, A. & Wills, G. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika*. **14**, 1502 (2016)
11. Dreyfus, G., Martinez, J., Samuelides, M., Gordon, M., Badran, F., Thiria, S. & Hérault, L. Réseaux de neurones-Méthodologie et applications. (Eyrolles,2002)
12. Ettaouil, M., Ghanou, Y., El Moutaouakil, K. & Lazaar, M. Image medical compression by a new architecture optimization model for the Kohonen networks. *International Journal Of Computer Theory And Engineering*. **3**, 204 (2011)
13. Ganesan, N., Venkatesh, K., Rama, M. & Palani, A. Application of neural networks in diagnosing cancer disease using demographic data. *International Journal Of Computer Applications*. **1**, 76-85 (2010)
14. Solaiman, B. & Richard, L. Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur. (2003)
15. Ritter, H., Martinetz, T., Schulten, K., Barsky, D., Tesch, M. & Kates, R. Neural computation and self-organizing maps: an introduction. (Addison-Wesley Reading, MA,1992)
16. Cottrell, M., Olteanu, M., Rossi, F. & Villa-Vialaneix, N. Self-OrganizingMaps, theory and applications. (2018)

17. Akinduko, A. & Mirkes, E. Initialization of self-organizing maps: principal components versus random initialization. A case study. *ArXiv Preprint ArXiv:1210.5873*. (2012)
18. Gao, Z., Fan, Y., Niu, K. & Wang, T. An Adaptive Initial Cluster Centers Selection Algorithm for High-Dimensional Partition Clustering. *2017 IEEE 15th Intl Conf On Dependable, Autonomic And Secure Computing, 15th Intl Conf On Pervasive Intelligence And Computing, 3rd Intl Conf On Big Data Intelligence And Computing And Cyber Science And Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. pp. 1119-1126 (2017)
19. Byoungwook, K. A Fast K-prototypes Algorithm Using Partial Distance Computation. (Preprints,2017)
20. Hajjar, C. Cartes auto-organisatrices pour la classification de données symboliques mixtes, de données de type intervalle et de données discrétisées.. (Supélec,2014)
21. Andoni, A. & Indyk, P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *2006 47th Annual IEEE Symposium On Foundations Of Computer Science (FOCS'06)*. pp. 459-468 (2006)
22. Kulkarni, A., Chong, D. & Batarseh, F. Foundations of data imbalance and solutions for a data democracy. *Data Democracy*. pp. 83-106 (2020)
23. Fernández, A., Garcıa, S., Galar, M., Prati, R., Krawczyk, B. & Herrera, F. Learning from imbalanced data sets. (Springer,2018)
24. Diez, P. Smart Wheelchairs and Brain-computer Interfaces: Mobile Assistive Technologies. (Academic Press,2018)
25. Sim, J. & Wright, C. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical Therapy*. **85**, 257-268 (2005)
26. Cohen, J. A coefficient of agreement for nominal scales. *Educational And Psychological Measurement*. **20**, 37-46 (1960)
27. Tharwat, A. Classification assessment methods. *Applied Computing And Informatics*. (2020)
28. Parikh, R., Mathai, A., Parikh, S., Sekhar, G. & Thomas, R. Understanding and using sensitivity, specificity and predictive values. *Indian Journal Of Ophthalmology*. **56**, 45 (2008)
29. Hamed, T., Dara, R. & Kremer, S. Intrusion detection in contemporary environments. *Computer And Information Security Handbook*. pp. 109-130 (2017)
30. Hossin, M. & Sulaiman, M. A review on evaluation metrics for data classification evaluations. *International Journal Of Data Mining Knowledge Management Process*. **5**, 1 (2015)
31. Kuhn, M. The caret package. *Journal Of Statistical Software*. **28** (2009)
32. Max Kuhn Package caret. (2020,3,20), https://cran.r-project.org/web/packages/caret/caret.pdf
33. Dua, D. & Graff, C. UCI Machine Learning Repository. (University of California, Irvine, School of Information,2017), http://archive.ics.uci.edu/ml
34. Tung, A., Xu, X. & Ooi, B. Curler: finding and visualizing nonlinear correlation clusters. *Proceedings Of The 2005 ACM SIGMOD International Conference On Management Of Data*. pp. 467-478 (2005)
35. Pazzani, M. Influence of prior knowledge on concept acquisition: Experimental and computational results.. *Journal Of Experimental Psychology: Learning, Memory, And Cognition*. **17**, 416 (1991)
36. Michalski, R., Mozetic, I., Hong, J. & Lavrac, N. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proc. AAAI*. **1986** pp. 1-041 (1986)
37. Salzberg, S. Exemplar-based learning: Theory and implementation. (Harvard University, Center for Research in Computing Technology, Aiken . . . ,1988)
38. Wolberg, W. & Mangasarian, O. Multisurface method of pattern separation for medical diagnosis applied to breast cytology.. *Proceedings Of The National Academy Of Sciences*. **87**, 9193-9196 (1990)
39. Little, M., McSharry, P., Roberts, S., Costello, D. & Moroz, I. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Nature Precedings*. pp. 1-1 (2007)
40. Amin, M. & Ali, A. Performance evaluation of supervised machine learning classifiers for predicting healthcare operational decisions. *Wavy AI Research Foundation: Lahore, Pakistan*. (2018)
41. Abid, F. & Izeboudjen, N. Predicting Forest Fire in Algeria Using Data Mining Techniques: Case Study of the Decision Tree Algorithm. *International Conference On Advanced Intelligent Systems For Sustainable Development*. pp. 363-370 (2019)
42. He, Z., Zhang, M. & Zhang, H. Data-driven research on chemical features of Jingdezhen and Longquan celadon by energy dispersive X-ray fluorescence. *Ceramics International*. **42**, 5123-5129 (2016)
43. Woolery, L., Grzymala-Busse, J., Summers, S. & Budihardjo, A. The use of machine learning program LERS-LB 2.5 in knowledge acquisition for expert system development in nursing.. *Computers In Nursing*. **9**, 227-234 (1991)
44. Hahn, G. & Meeker, W. Statistical intervals: a guide for practitioners. (John Wiley Sons,2011)
45. Friedman, J., Hastie, T., Tibshirani, R. & Others The elements of statistical learning. (Springer series in statistics New York,2001)
46. Velden, M., Iodice D'Enza, A. & Markos, A. Distance-based clustering of mixed data. *Wiley Interdisciplinary Reviews: Computational Statistics*. **11**, e1456 (2019)
47. Foss, A., Markatou, M. & Ray, B. Distance metrics and clustering methods for mixed-type data. *International Statistical Review*. **87**, 80-109 (2019)
48. Balaji, K. & Lavanya, K. Clustering algorithms for mixed datasets: a review. *International Journal Of Pure And Applied Mathematics*. **18**, 547-556 (2018)
49. Ahmad, A. & Khan, S. Survey of state-of-the-art mixed data clustering algorithms. *Ieee Access*. **7** pp. 31883-31902 (2019)
50. MacQueen, J. & Others Some methods for classification and analysis of multivariate observations. *Proceedings Of The Fifth Berkeley Symposium On Mathematical Statistics And Probability*. **1**, 281-297 (1967)
51. Khan, S. & Kant, S. Computation of Initial Modes for K-modes Clustering Algorithm Using Evidence Accumulation.. *IJCAI*. **7** pp. 2784-2789 (2007)
52. Brownlee, J. What is the Difference Between a Batch and an Epoch in a Neural Network?. *Machine Learning Mastery*. **20** (2018)
53. Gao, Z., Fan, Y., Niu, K. & Wang, T. An Adaptive Initial Cluster Centers Selection Algorithm for High-Dimensional Partition Clustering. *2017 IEEE 15th Intl Conf On Dependable, Autonomic And Secure Computing, 15th Intl Conf On Pervasive Intelligence And Computing, 3rd Intl Conf On Big Data Intelligence And Computing And Cyber Science And Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. pp. 1119-1126 (2017)

54. Perišić, A. & Pahor, M. Clustering mixed-type player behavior data for churn prediction in mobile games. *Central European Journal Of Operations Research*. **31**, 165-190 (2023)
55. Ganapathy, A., Hannah, D. & Agarwal, A. Improved estimation of extreme floods with data pooling and mixed probability distribution. *Journal Of Hydrology*. **629** pp. 130633 (2024)
56. Soemitro, D. & Neto, J. Spectral Clustering of Categorical and Mixed-type Data via Extra Graph Nodes. *ArXiv Preprint ArXiv:2403.05669*. (2024)
57. Veronesi, V. & Others On the Effects of Measurement Error and Misclassification on Clustering Algorithms for Mixed-Type Data. (Università degli Studi di Milano-Bicocca,2024)
58. Liu, P., Yuan, H., Liu, N. & Peres, M. A Modified Gower Distance-Based Clustering Analysis for Mixed-Type Data. *Available At SSRN 4779022*.
59. D'Orazio, M. Gower's similarity coefficients with automatic weight selection. *ArXiv Preprint ArXiv:2401.17041*. (2024)
60. Wen, H., Zhao, S. & Liang, M. Unsupervised attribute reduction algorithm for mixed data based on fuzzy optimal approximation set. *Mathematics*. **11**, 3452 (2023)
61. Munguıa Mondragón, J., Rendón Lara, E., Alejo Eleuterio, R., Granda Gutirrez, E. & Del Razo López, F. Density-Based Clustering to Deal with Highly Imbalanced Data in Multi-Class Problems. *Mathematics*. **11**, 4008 (2023)
62. Jung, T. & Kim, J. A new support vector machine for categorical features. *Expert Systems With Applications*. **229** pp. 120449 (2023)
63. Tang, W., He, H. & Tu, X. Applied categorical and count data analysis. (Chapman,2023)
64. Das, A. Logistic regression. *Encyclopedia Of Quality Of Life And Well-Being Research*. pp. 3985-3986 (2024)
65. Balaji, K., Lavanya, K. & Mary, A. Clustering of mixed datasets using deep learning algorithm. *Chemometrics And Intelligent Laboratory Systems*. **204** pp. 104123 (2020), https://www.sciencedirect.com/science/article/pii/S0169743920303695
66. Ahmad, A. & Khan, S. Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access*. **7** pp. 31883-31902 (2019)
67. Ezugwu, A., Ikotun, A., Oyelade, O., Abualigah, L., Agushaka, J., Eke, C. & Akinyelu, A. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications Of Artificial Intelligence*. **110** pp. 104743 (2022)
68. Lévy, L., Bosom, J., Guerard, G., Amor, S., Bui, M. & Tran, H. Application of Pretopological Hierarchical Clustering for Buildings Portfolio. *SMARTGREENS*. pp. 228-235 (2021)
69. Lévy, L., Bosom, J., Guerard, G., Amor, S., Bui, M. & Tran, H. DevOps Model Approach for Monitoring Smart Energy Systems. *Energies*. **15**, 5516 (2022)
70. Amor, S., Choufa, M., Cornet, C., Djebali, S., Guerard, G., Lévy, L. & Tran, H. Pretopology-based Clustering for Mixed Data. *ROADEF2023*. (2023)
71. Amor, S., Choufa, M., Cornet, C., Djebali, S., Guerard, G., Lévy, L. & Tran, H. Clustering Mixed Data Comprising Time Series. *SOICT2023*. (2023)
72. Hsu, C., Lin, S. & Tai, W. Apply extended self-organizing map to cluster and classify mixed-type data. *Neurocomputing*. **74**, 3832-3842 (2011)
73. Kuo, R., Wu, C. & Kuo, T. An ensemble method with a hybrid of genetic algorithm and K-prototypes algorithm for mixed data classification. *Computers Industrial Engineering*. **190** pp. 110066 (2024)
74. Purbasari, I., Puspaningrum, E. & Putra, A. Using Self-Organizing Map (SOM) for Clustering and Visualization of New Students based on Grades. *Journal Of Physics: Conference Series*. **1569**, 022037 (2020,7), https://dx.doi.org/10.1088/1742-6596/1569/2/022037
75. Bowen, F. & Siegler, J. Self-organizing maps: a novel approach to identify and map business clusters. *Journal Of Management Analytics*. pp. 1-19 (2024)
76. Markoulidakis, I., Kopsiaftis, G., Rallis, I. & Georgoulas, I. Multi-class confusion matrix reduction method and its application on net promoter score classification problem. *Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference*. pp. 412–419 (2021)