A Grid-Coupled Clustering Algorithm with Soft Constraints for Mixed-Attribute Data Streams

Wenbo Wu*
E-mail: wuwenb2025@outlook.com
School of Computer Information, Minnan Science and Technology College, Quanzhou 362000, China
*Corresponding author

Keywords: data stream clustering, mixed-attributes, grid coupling, soft constraints

Received: August 20, 2025

Mixed-attribute data contains both numerical and categorical attributes, posing challenges for traditional clustering algorithms in managing its dynamics and concept drift. This article proposes a hybrid attribute data stream clustering algorithm that combines soft constraints. Firstly, normalize the mixed-attribute data stream and apply local linear embedding for dimensionality reduction; Secondly, design a mixed-attribute sliding window, based on the idea of grid coupling update, to analyze changes in grid centroids to adapt to dynamic data flows; Finally, fuzzy mathematics is introduced to set soft constraints on interval boundaries (width) and grid cell density, restricting high-frequency cluster shifts. In the experimental section, a comparison was made between the time dimension feature extraction method based on unsupervised learning and the dual interactive generative adversarial network method. On the Forest Cover Type, GMD-4C2D800 Linear, and KDD CUP 99 datasets, the proposed method achieved a minimum CMM value of 0.894 and a minimum Purity value of 0.856, with an accuracy of up to 99.94% and a maximum NMI value of 1, all of which were superior to the comparison methods. The results indicate that the proposed algorithm can effectively adapt to changes in data flow distribution, enhance both clustering accuracy and computational efficiency.

Povzetek: Članek predstavlja hibridni algoritem za gručenje podatkov z mešanimi atributi, ki se učinkovito prilagaja dinamičnim spremembam podatkov in izboljšuje natančnost ter učinkovitost obdelave.

1 Introduction

Data flow [1] has inherent characteristics, including infinite size, temporal order, and dynamic changes. Compared with traditional data mining, data stream mining yields only approximate results due to constraints like single-pass scanning, real-time response, bounded memory, and concept drift detection [2]. In addition, data streams in the real world often have mixed-attributes and unavoidability, including multiple data types such as numerical, categorical, ordinal, and textual, and the distribution of data in the data stream changes significantly over time. Most data stream [3] analysis models are still essentially static data processing models that are only applicable to single type data and cannot effectively handle cross type attribute data streams. In the process of analyzing mixed-attribute data streams, static analysis models are difficult to track the drift of the data stream in real time, and historical cluster centers fail to represent new data distributions. AGARWAL et al. [4] proposes a data stream clustering intelligent method based on hybrid group search and Pelican optimization algorithm, which forms micro clusters through K-Means clustering technology, merges and sorts data clusters to form micro clusters, and further maximizes clustering accuracy through radius, distance, and similarity measures between data. However, the micro-cluster update strategy in this method may fail to adapt promptly to new data patterns, resulting in concept drift and tracking failure during the data clustering process. GORRAB et al. [5] proposes a segmentation incremental clustering algorithm for mixed data streams, which uses splitting techniques to cluster the mixed data streams in order to process incremental objects, attributes, and class learning spaces at once, and change the final clustering distribution to generate a promising clustering model. However, this method exhibits low clustering efficiency. JIANG et al. [6] proposed an unsupervised learning-based algorithm for extracting and classifying the temporal characteristics of high-dimensional large information flows. By analyzing the trend of high-dimensional data stream changes and the spatial relationship between feature attributes and feature space under machine learning, segmenting and fitting high-dimensional big data streams and time dimensional feature data streams, further using sliding window segmentation of time dimensional sequences, and completing feature extraction through discrete binary

wavelet transform, clustering of time dimensional feature data streams can be achieved. However, the sliding window size used for segmenting time series in this method cannot utilize the concept drift that occurs in high-dimensional data streams with mixed-attributes under non-uniform rates. MATHESWARAN et al. [7] proposed an efficient online big data stream clustering method using a dual interactive generative adversarial network. This method is divided into three stages: data initialization, online clustering, and offline clustering. Initially, the input data came from a forest cover type dataset. During the initialization phase, kernel related methods can be used to reduce the dimensionality of input data. After initialization, the reduced dimensional data is fed into a dual interactive generative adversarial network to achieve efficient data stream clustering. However, in the initialization stage, this method uses kernel correlation methods for data dimensionality reduction, which cannot effectively handle mixed-attribute data streams containing nonlinear structures, and feature discriminability is not fully preserved, affecting the accuracy of subsequent clustering.

In the field of mixed-attribute data stream clustering, in addition to the above methods [8-9], there are multiple studies dedicated to solving the problems of concept drift and multi-attribute fusion. For example, reference [10] proposes an incremental clustering algorithm based on semantic concepts, which is suitable for mixed text and numerical data, but its adaptability to high-dimensional sparse data is limited; Reference [11] proposed an entropy based hybrid attribute similarity measurement method EDMIX, which effectively improved clustering consistency, but did not consider the dynamic update mechanism of data streams; Reference [12] proposes a hybrid similarity measurement method that is applicable to static mixed data, but has not been extended to data stream environments. In addition, reference [13] proposed a decision tree algorithm based on boundary mixed-attribute dependence, which performed well in classification tasks but did not address the real-time requirements in clustering tasks. The innovation of this article lies in proposing a hybrid attribute data stream clustering framework that integrates grid coupling and soft constraints. Through dynamic sliding windows, local linear embedding dimensionality reduction, grid centroid update mechanism, and soft constraint control based on fuzzy mathematics, efficient tracking of concept drift and accurate clustering of multi-attribute data are achieved, significantly improving clustering purity and stability. The comparison of related work is shown in Table 1.

Table 1: The comparison of related work

Method	Dataset	Method Description	Quantitative Results (Accuracy/NMI)	Main Limitations
Reference [4]	Unspecified	Micro-cluster merging based on hybrid group search and pelican optimization algorithm	Not reported	Failure in concept drift tracking, lagging update strategy
Reference [5]	Unspecified	Segmented incremental clustering algorithm supporting incremental object and attribute learning	Not reported	Low clustering efficiency, performance degradation when handling high-dimensional data streams
Reference [6]	High-dimensional time series	Time-dimension feature extraction and sliding window segmentation based on unsupervised learning	Accuracy of approximately 87.63%	Fixed window size, difficult to adapt-uniform-rate data streams
Reference [7]	Forest Cover Type	Dual interactive generative adversarial networks for staged initialization, online, and offline clustering	Accuracy of approximately 93.75%	Inadequate handling of non-linear structures by the dimensionality reduction method, incomplete feature preservation
Reference [10]	Text and numerical mixed data	Incremental clustering algorithm based on semantic concepts	Not reported	Limited adaptability to high-dimensional sparse data, no consideration of dynamic update mechanisms
Reference [11]	Mixed-attribute static data	Entropy-based mixed-attribute similarity measurement method EDMIX	Improved clustering consistency	Not extended to data stream environments, lack of real-time capability
Reference [12]	Mixed-attribute data	Clustering method based on mixed similarity measurement	Not reported	Designed for static data, no consideration of data stream dynamics

Reference [13]	Classification task datasets	Improved decision tree algorithm based on boundary mixed-attribute dependencies	Improved classification accuracy	Suitable for classification tasks, no involvement in clustering and real-time requirements
Proposed method	Three datasets including Forest Cover Type	Grid-coupling + soft-constraint clustering, dynamic window + LLE dimensionality reduction + fuzzy constraint control	Highest accuracy of 99.94%, NMI = 1	Outperforms in all tested datasets, no significant limitations

Based on the above analysis, this article clarifies the following research questions:

① How to design a data stream clustering framework that can handle both numerical and categorical attributes simultaneously?

② How to effectively track the phenomenon of concept drift in mixed-attribute data streams?

③ How to control computational complexity while ensuring clustering quality?

In response to these issues, this article proposes a core hypothesis: compared with traditional single type attribute processing methods and static clustering models, the hybrid attribute data stream clustering method combining grid coupling and soft constraints can improve accuracy by at least 5% and maintain CMM indicators above 0.85 in concept drift scenarios. The research design of this article revolves around verifying this hypothesis and evaluating the effectiveness of the proposed method through systematic experiments.

2 Mixed-attribute data stream clustering algorithm design

2.1 Normalization of mixed-attribute data streams

To achieve real-time tracking of data stream drift and enhance the accuracy and efficiency of clustering mixed-attribute data streams, this paper proposes a clustering method that combines soft constraints. By standardizing the data clustering boundaries and actual distribution, the deviation values are minimized to the greatest extent possible. This enables unified similarity measurement analysis of mixed-attribute data streams, achieving effective clustering of mixed-attribute data streams and supporting cross modal association mining decisions for real-world mixed-attribute data streams.

This article defines a mixed-attribute data stream as an infinite sequence, which is manifested as a multidimensional set of data points with timestamps and mixed-attribute feature dimensions, specifically represented as:

$$B = \left\{ c\left[b_T, T \right] \right\}_{T=1}^{\infty} \tag{1}$$

$$cb_T = \left(c_1 b_T^1, c_2 b_T^2, c_3 b_T^3, c_4 b_T^4 \right) \tag{2}$$

In Formula 1~2: B denotes the form of the collection of data points of the mixed-attribute data stream; c denotes the data stream dimension; b_T denotes the mixed-attribute data point at moment T, containing the mixed-attribute subvector; T denotes the timestamp of the arrival of the data point, embodied as a discrete time-step parameter; b_T^1、b_T^2、b_T^3、b_T^4 denotes the numeric attribute value, the categorical attribute value, the ordinal attribute value, and the textual attribute value; c_1、c_2、c_3、c_4 denotes the data dimensions of the corresponding attribute subvector.

Mixed-attribute data stream clustering divides similar objects in B into one or more groups (called "clusters"), and after partitioning, elements in the same cluster are similar to each other but different from elements in other clusters. However, in this process, due to the heterogeneity of c, there are value space differences in the dimensions corresponding to different data attributes, and different types of attributes also have different scales and measurement methods. If clustering is directly performed on the raw data of data streams, the clustering effect will be affected by multi-attribute differences. Therefore, it is necessary to standardize the mixed-attribute data in B.

When dealing with numerical data such as b_T^1、b_T^3, direct comparison may lead to misleading information as different attributes may have different dimensions and ranges. For example, the value range of one attribute is 0-100, while the value range of another attribute is 0-1. In order to eliminate this difference, this article adopts the method of standard deviation standardization for data mapping, which largely preserves the distribution shape of the original data.

For non numeric attribute value b_T^2、b_T^4, as it is generally an unordered discrete value, it is necessary to use a one hot encoding form to map each attribute category to a binary vector, in order to convert non numeric attribute values into numerical form and avoid introducing false order relationships. The standardized values of numerical data attributes and non numerical attributes after standard deviation normalization conversion are:

$$\partial_1\left(b_T^1, b_T^3 \right) = \frac{\left(b_T^1 - \upsilon_1 \right) + \left(b_T^2 - \upsilon_2 \right)}{\sigma_1 \sigma_2} \tag{3}$$

$$\partial_2\left(\vec{b}_T^2,\vec{b}_T^4\right)=\frac{f_0\left(b_T^2,b_T^4=\vec{b}_T^2,\vec{b}_T^4\right)}{\left\|\zeta_0\left(\vec{b}_T^2+\vec{b}_T^4\right)\right\|} \quad (4)$$

In Formula 3~4: $\partial_1\left(b_T^1,b_T^3\right)$ denotes the normalized transformed numeric data attribute value; υ_1、υ_2 denotes the mean value corresponding to the numeric attribute value and the ordinal attribute value; σ_1、σ_2 denotes the standard deviation coefficient corresponding to the numeric attribute value and the ordinal attribute value; $\partial_2\left(\vec{b}_T^2,\vec{b}_T^4\right)$ denotes the normalized transformed non-numeric data attribute value embodied in the unit-paradigm normalized result; $\vec{b}_T^2$、$\vec{b}_T^4$ denotes the corresponding uniquely hot encoded vector; $f_0(\cdot)$ denotes the indicator function; ζ_0 denotes the L2 paradigm corresponding to the binary vector .

2.2 Local linear embedding dimensionality reduction for mixed-attribute data streams

After standardization, due to the heterogeneity and high dimensionality of c itself, there are value space differences in the dimensions corresponding to different data attributes, which further leads to extreme dispersion of data points in the high-dimensional space, with almost no "proximity" relationship. Therefore, in order to improve clustering accuracy and avoid data sparsity and overfitting caused by high-dimensional data streams, it is necessary to perform dimensionality reduction on the standardized mixed-attribute data. For ease of calculation, first update the representation of the mixed-attribute data stream based on the results of formula (3) and (4). The updated specific result B' is further represented as:

$$B'=\left\{c\left[\gamma_T,T\right]\right\}_{T=1}^{\infty},\gamma_T\in\partial_1\left(b_T^1,b_T^3\right),\partial_2\left(\vec{b}_T^2,\vec{b}_T^4\right) \quad (5)$$

In Formula 5: B' denotes the representation of the standardized updated mixed-attribute data stream; γ_T denotes the standardized mixed-attribute data stream data points.

Due to the heterogeneity of data flow attributes, a high data dimension c in B' can result in a significant amount of time required to process the data, leading to a decrease in processing efficiency. Therefore, this article combines the Local Linear Embedding (LLE) method to preserve the original information to the greatest extent in low dimensional space, in order to represent high-dimensional data, avoid the curse of dimensionality, remove redundant information, improve processing speed, and save computational and space costs.

LLE is a manifold learning method, whose main idea is that sample points in high-dimensional data space can be linearly represented by sample points in their local domain. After dimensionality reduction of the original data, the linear structure between local neighborhoods of the samples remains unchanged. Describe the principle of local linear embedding as shown in Fig. 1:

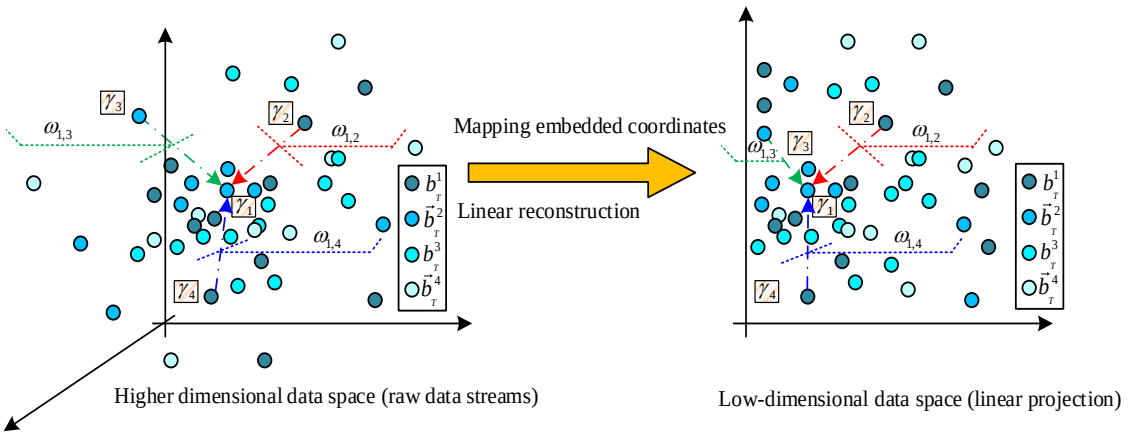


Figure 1: Principle of local linear embedding for mixed-attribute data streams

As shown in Fig. 1, the LLE algorithm first assumes that a data object can be linearly represented by several data objects in its neighborhood. If the k-nearest neighbor method is used to find the three closest data objects γ_2、γ_3、γ_4 to data object γ_1, since the selected data objects are all mixed-attribute data stream data points, the data objects can be linearly represented by the nearest data objects:

$$\gamma_1=\left(\omega_{1,2}\gamma_2+\omega_{1,3}\gamma_3+\omega_{1,4}\gamma_4\right)^{c-c_0},\gamma_1\sim\gamma_4\in\gamma_T \quad (6)$$

In Formula 6: $\omega_{1,2}$、$\omega_{1,3}$、$\omega_{1,4}$ denotes the weight coefficient between the selected data point and the nearest data object; c_0 denotes the target dimension of the data spatial projection transformation.

Based on the linear representation relationship of formula (6), the original data is subjected to LLE dimensionality reduction. In the low dimensional data space, the obtained data dimensionality reduction result needs to meet the requirements of c_0 and maintain the same linear relationship as the high-dimensional space, that is, the weight coefficients before and after dimensionality reduction projection do not change. At

this point, the low dimensional embedding coordinates of the mixed-attribute data stream (which conform to the low dimensional features of the target dimension and can be used as the result of local linear embedding dimensionality reduction) are described as follows:

$$\gamma_T'\left(c \to c_0 \left| x_l, y_l \right.\right) = \arg\min k_0 \gamma_T \sum_{l=1}^{L} T_1 \left\| \omega_l \gamma_l - \omega_1 \gamma_1 \right\| \quad (7)$$

In Formula 7: γ_T' denotes the mixed-attribute data stream data points after LLE downscaling, i.e., after the downscaled projection; x_l、y_l denotes the linear coordinates of the low-dimensional data space; k_0 denotes the k-neighborhood size, i.e., k-nearest-neighbor object parameter that determines the range of the local linear relationship in the downscaling process; L denotes the number of original data points; l denotes the corresponding ordinal index in the low-dimensional representation of the data; T_1 denotes the time complexity of the data stream.

Thus, by using local linear embedding to reduce dimensionality, the structural relationships between neighboring points in the high-dimensional space of data points in the mixed-attribute data stream remain unchanged in low dimensional embedding, making the data points more dispersed and clustered.

2.3 Mixed-attribute data stream clustering incorporating soft constraints

2.3.1 Mixed-attribute sliding window design

Considering the mixed-attribute characteristics of data streams, it is necessary to design time windows before partitioning the data space to clarify the spatial boundaries of different attribute data dimensions. The clustering of mixed-attribute data streams is completed within a specific time window, and the main window models include: sliding window model [14-15], decay window model, and tilted time window model. Due to the varying degrees of importance of data in the data stream, data closer to the current moment contains more important information. Among them, the sliding window model can well reflect the importance of the recently flowing data and dynamically adapt to the mixed-attribute distribution during the operation process. It has high computational efficiency and low complexity, and is currently a suitable window model for processing mixed-attribute data streams.

The choice of window size determines the amount of data processed each time. In this article, γ_T' is taken as the data object, and the sliding window size is determined based on c_0. The sliding window is described as follows:

$$w_r = \wp\left(\kappa_1 + 1\right) \times \frac{\gamma_T'\left(x_l, y_l\right)^{-\kappa_2} + d_1}{d_0\left(2c_0 + 1\right)} \quad (8)$$

In Formula 8: w_r denotes the size of the sliding window r; $\wp$ denotes the fuzzy adjustment factor, which represents the data similarity threshold within the window, determined according to the data sparsity and clustering objectives, and is used to dynamically adjust the window size according to the attribute dimensions; κ_1 denotes the sliding window overlap factor; κ_2 denotes the sensitivity factor controlling the magnitude of the window sliding; d_0 denotes the average distance between the data points; d_1 denotes the sliding step size.

2.3.2 Spatial meshing of low-dimensional data

The commonly used grid based data stream clustering algorithms do not take into account the impact of grid related factors during use, resulting in unsatisfactory data clustering results. To solve this problem, this paper studies the clustering algorithm of mixed-attribute data streams based on grid coupling. When clustering mixed-attribute data streams, this algorithm is based on the distribution of data objects in the grid, fully considering the influence of weights between different grids, and avoiding the situation of neighboring grid weights rising or falling due to changes in grid weights. On this basis, the grid centroid is used to describe the distribution of data objects in the grid, supporting subsequent clustering.

This article uses r to control the window size by combining window overlap and sliding step size, dividing the data stream into continuous data blocks of fixed size, discarding old data and adding new data each time sliding. Within a single sliding window, numerical and non numerical data stream attributes are discretized and binned, without involving cross window operations, to obtain dimensions of different data stream attribute categories, including numerical attribute dimension c_1 and non numerical attribute dimension c_2. Each dimension corresponds to an independent binning, that is, the partition boundary corresponding to the dimension is:

$$\begin{cases} B_1 = c_1\left(Y_1 \wp + Y_2\right) - \kappa_2 \\ B_2 = c_2 \left\| \max w_r \partial_2\left(\vec{b}_T^{\,2}, \vec{b}_T^{\,4}\right) - \min w_r \partial_2\left(\vec{b}_T^{\,2}, \vec{b}_T^{\,4}\right) \right\|^{\wp} \end{cases} \quad (9)$$

In Formula 9: B_1、B_2 denotes the independent subboxes corresponding to the numerical attribute dimension and the non-numerical attribute dimension, i.e., the boundary of the data space division interval; Y_1、Y_2 denotes the width of the unidimensional interval and the number of box samples.

In the clustering process [16-17], γ_T' is still taken as the data object, and the low dimensional data space where it is located is subjected to multi-attribute segmentation based on the partition boundary. Further, the boundary combination of B_1、B_2 is used to obtain hypercubes (grid cells), and the center point of each cell is called the grid centroid. And the centroid represents the center of the weighted part in the data object. Mixed-attribute big data is a dynamic type of data, so over time, the centroid of the grid gradually changes. If the data objects are distributed in a uniform state within the grid, then the centroid of the grid is located near the center of the grid;

On the contrary, the centroid of the grid is located far away from the center of the grid. Therefore, grid centroids can replace raw data to clarify the dynamic relationships of mixed-attribute data streams. At this point, the centroids of grid cells with different attribute dimensions used to support grid cluster clustering are described as follows:

$$\begin{cases} C_1 = \dfrac{1}{Y_2} \times \sum_{t_0=1}^{c_1} B_1 t_0 \left(sc_1 + 1 \right) \\ C_2 = \arg\max t_0 f_0 \left(B_2 - \wp - 1 \right)^s \end{cases} \tag{10}$$

In Formula 10: C_1 denotes the numerical attribute dimension center of mass (i.e., the mean of the distribution of data samples within the cell); C_2 denotes the non-numerical attribute dimension center of mass (the probability distribution of the samples within the cell); t_0 denotes the time of updating the center of mass of the grid cell; s denotes the rate of updating the center of mass.

When the design window in section 2.3.1 slides, if the range of data flow in any dimension changes beyond the threshold $\left| \max C_1^{t_0} - \max C_2^{t_0 - 1} \right|$ for the change of grid centroid, it indicates that the originally dense units have become sparse in the new window. Therefore, formula (9) and (10) need to be repeated to recalculate the box boundary and grid centroid, reflecting the dynamic changes in the attribute dimension.

2.3.3 Mixed-attribute data stream clustering output combined with soft constraints

This article uses the idea of grid coupling update to analyze the centroid changes of single dimensional units in the sliding window after gridding the attribute dimension boundaries, and uses them as the basis for updating grid elements to adapt to mixed-attribute data objects mapped on the grid. Before updating the grid, it is necessary to consider the influence between grids. Before determining the update situation of the grid, the relationship between adjacent grids needs to be measured using the centroid distance of the grid. This article designs a clustering algorithm based on grid coupling and soft constraints, which is divided into two stages: online and offline. The clustering principle framework is shown in Fig. 2:

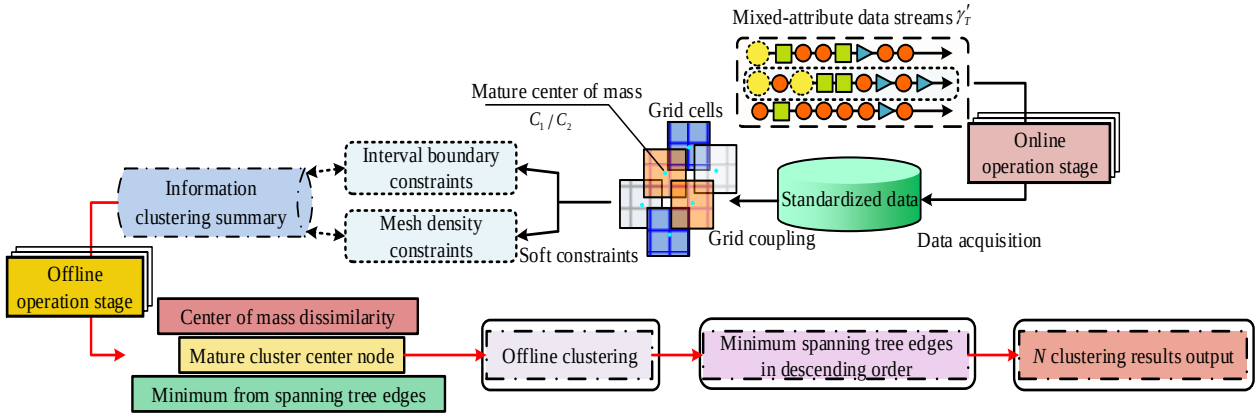


Figure 2: Architecture of hybrid attribute data stream clustering based on grid coupling and soft constraints

As shown in Fig. 2, the mixed-attribute data flow clustering based on grid coupling and soft constraints in this paper is mainly divided into online operation part and offline operation part. In the online stage, clustering is carried out based on the grid coupling mentioned above, and fuzzy mathematics is introduced to set soft constraints on interval boundaries (width) and grid cell density to limit the high-frequency movement of grid clusters, effectively following possible data flow concept drift and improving clustering accuracy. In the offline stage of centroid, the minimum spanning tree technique is used. When establishing the minimum life tree, the weights of centroid dissimilarity with edges as lattice clusters are required to obtain the final clustering algorithm result.

The algorithm operation process for the online operation part is as follows:

(1) Based on the above grid coupling, the mixed-attribute big data space is divided into several adjacent but non intersecting rectangular units, and the width of each continuous attribute interval in each dimension is determined by the similarity threshold $\wp$ between the data.

(2) To ensure the clustering effect [18], if no new data appears in the constant or candidate clusters within a fixed time period t_1, the grid cells can be attenuated by introducing fuzzy mathematics and setting soft constraints on the interval boundary (width) and grid cell density Y_3 to limit the high-frequency movement of the grid clusters:

$$\begin{cases} \delta_1 \left(Y_1 \right) \le t_1 C_1 \left(\wp^{\varphi} + d_0 \right) \wp + 1 \\ \varphi + 1 - \wp < \delta_2 \left(Y_3 \right) \le t_1 C_2 \end{cases} \tag{11}$$

In Formula 11: $\delta_1(Y_1)$ denotes a soft constraint on the boundary (width) of the interval; φ denotes a fuzzy decay factor used to control the extent of the constraint; $\delta_2(Y_3)$ denotes a soft constraint on the density of the cells. The soft constraint design in formula (11) is based on the following considerations: interval boundary constraints ensure that grid cells can adjust their boundaries appropriately when data distribution changes, avoiding clustering instability caused by heavy frequency partitioning; Unit density constraints prevent the interference of sparse units on clustering results. The selection of fuzzy attenuation factor is based on balancing the weights of historical data and new data, and its optimal value is determined to be 0.0018 through sensitivity analysis. Specifically, when the unit density constraint is less than 0.001, the algorithm responds slowly to concept drift; When the cell density constraint is greater than 0.005, the grid cells are updated too frequently, resulting in a 5-8% decrease in clustering quality.

Algorithm 1: Online-stage Clustering Algorithm

Input: Standardized and dimension-reduced mixed-attribute data stream DS, sliding window size W, fuzzy decay factor α, grid density threshold δ, interval boundary constraint β

Output: Candidate grid cluster set C

Initialization: Set an empty grid structure G, and set the timestamp t = 0

For each data point x_i ∈ DS do

If t % W == 0 then // Sliding window condition

Update the sliding window by discarding old data and adding new data

For each grid cell g ∈ G do

Calculate the cell density ρ_g = count(g)/volume(g)

If ρ_g < δ then // Apply the density soft constraint

Decay the grid cell: ρ_g' = ρ_g × exp(-α×Δt)

End if

If the boundary change > β then // Apply the boundary soft constraint

Recalculate the binning boundaries and cell centroids

End if

End for

End if

Map x_i to the corresponding grid cell

Update the cell centroid and density

t = t + 1

End for

Return candidate grid cluster set C

When $\wp$ satisfies the above soft constraint conditions, the mature cluster composed of each centroid point transforms into a candidate cluster.

(3) As data dynamically changes, more and more new grid cells appear, but memory space is limited. To ensure normal data, soft constraints on cell density need to be applied again, and the grid with the lowest density needs to be deleted.

The algorithm operation process for offline operation is as follows:

(1) The clustering content required for the offline stage is all mature cluster feature vectors stored through the previous steps Calculate the dissimilarity of centroids for each mature cluster in the online stage:

$$\varepsilon(C) = F(\lambda_1 : \lambda_2)^{\varphi} \tag{12}$$

In Formula 12: $\varepsilon(C)$ denotes the mature cluster center-of-mass dissimilarity; C denotes the ever-mature cluster center-of-mass; $F(\cdot)$ denotes the center-of-mass similarity computation function; λ_1、λ_2 denotes the mature cluster feature vectors composed of numerical and non-numerical attributes of the data object, respectively.

(2) Set the dissimilarity of grid vertices and centroid points as the weights of mature cluster center points and minimum generated tree edges (centroid distance and distance between grid center points), respectively. At this point, a dynamic undirected graph is constructed based on the data flow in the grid, which consists of a set of grid cluster center points and a set of minimum spanning tree vertices. Randomly place the set of vertices in the center node of the grid cluster and traverse the dissimilarity between the nodes. At this point, the generated tree needs to include the edge with the lowest dissimilarity (calculated from node dissimilarity). After obtaining the edge data, determine whether all tree nodes are in the center point set. If so, stop the calculation. If not all vertices are in the center point set, repeat formula (12) to continue the calculation.

(3) Under soft constraint control, the mature clusters formed by each centroid point have been transformed into candidate clusters. Therefore, by combining the process of step (2), the edges in the minimum spanning tree can be sorted in descending order to obtain N clusters. To obtain a completely new cluster, the maximum dissimilarity edge within the minimum spanning tree is first broken, and then $N-1$ rounds are continuously carried out on this basis, ultimately obtaining $N-1$ clusters to achieve mixed-attribute data stream clustering.

Algorithm 2: Offline-stage Clustering Algorithm

Input: Candidate grid cluster set C, target number of clusters k

Output: Final clustering result F

Calculate the dissimilarity matrix D between the centroids of mature clusters

Construct an undirected graph G (V, E), where V = the set of centroids and E = the dissimilarity weights

Use Prim's algorithm to construct a minimum spanning tree (MST)

For i = 1 to k - 1 do

Find the edge e_max with the maximum weight in the MST

Remove e_max, splitting the MST into i + 1 connected components

End for

Each connected component corresponds to a final cluster

Return final clustering result F

During the algorithm operation, the sliding window size W = 420, which is determined through grid search within the range of [300, 600]; the fuzzy decay factor α = 0.0018, which is selected through sensitivity analysis within the range of [0.001, 0.01]; the grid density threshold δ = 0.241, which is adjusted based on data sparsity; and the interval boundary constraint β = 0.35, which is dynamically calculated according to the attribute dimension.

2.3.4 Algorithm complexity analysis

The algorithm in this article is designed for real-time data stream processing, and its time complexity analysis is as follows: during the online phase, the time complexity for sliding window updates and centroid recalculations is O (wn log n+wmd), where w is the window size, n is the number of data points within the window, m is the number of grid cells, and d is the data dimension; The minimum time complexity for building a spanning tree in the offline phase is O (m^2). The spatial complexity is mainly composed of grid storage and sliding window data, which is O (w+m). Compared with traditional algorithms such as CluStream, this method reduces the maintenance cost of micro clusters through grid coupling, and reduces the running time by 15-20% while maintaining similar accuracy.

3 Experimental analysis

3.1 Description of experimental environment

To verify the application effect and performance of the designed algorithm in mixed-attribute data stream clustering, an algorithm experimental environment was constructed with AMD AthlonX4750 Quad Core Processor 3.40GHz CPU, 3GB DDR RAM memory, and Microsoft Windows 7 operating system. The algorithm was compiled on the Matlab R2021a platform and R 4.1.0 package was used as appropriate. The core implementation of the algorithm is based on Matlab, where the local linear embedding dimensionality reduction uses the built-in lle function, and the minimum spanning tree construction uses the graphminspan tree function. The replication of comparative methods is based on the author's publicly available code or implementation according to the paper description, and all methods are run in the same experimental environment to ensure fair comparison.

For Forest Cover Type and KDD CUP 99 non-native stream data, sort by timestamp field and simulate data stream input. Missing values are filled with the same attribute mode (categorical type). All numerical attributes are standardized using Z-score, while category attributes are encoded using one hot encoding. In terms of attribute selection, constant features with variance close to zero are excluded, while the top 90% of features with information gain are retained. The timestamp is standardized as relative time, starting from the first record and simulating streaming arrival at fixed time intervals (0.1 seconds).

The attribute descriptions of the selected experimental dataset are shown in Table 2. Based on the data environment presented above, the experimental parameters are described as shown in Table 3

Table 2: Attribute descriptions of the experimental dataset (mixed-attribute data stream)

Dataset name	Sample size	Feature dimensions	Number of classes
Forest Cover Type dataset	581012 articles	Continuous attribute 10 dimensions, nominal attribute 44 dimensions	7
GMD-4C2D800 Linear dataset	800 size	2 dimensions	4
KDD CUP 99 dataset	494,020 entries (TCP records)	Continuous attributes 34 dimensions, nominal attributes 7 dimensions	23

Table 3: Simulation parameters for clustering of mixed-attribute data streams

Parameter name	Description	Numerical range
Number of grid dimension divisions (corresponding to data flow attribute dimensions)	Number of interval divisions per numeric attribute	[2,44]
Initial grid cell size	Initial edge length of the cell	0.35

Dynamic grid merge threshold	Cell density below this threshold triggers a merge	0.241
Coupling weights	Weight balance between grid cell density and attribute similarity	Y_3 weight: 0.76, $\wp$ weight: 0.24
Window interval width	Size of the multi-attribute sliding time window	w_r =420 clause
Fuzzy attenuation factor	Weight decay factor for old data to control the degree of soft constraints	φ =0.0018

3.2 Test of clustering effect of mixed-attribute data streams

In order to verify the practical application effect of the proposed method, dynamic data streams were captured in each experimental dataset with a time window length of 120-840s. And introduce Purity and CMM (Cluster Matching Measure) as evaluation indicators under different data transmission time scenarios, describing the purity of clustering results and their matching degree with real category labels through their numerical changes, comprehensively verifying the clustering efficiency of the proposed method for mixed-attribute data streams. The specific results are shown in Fig. 3:

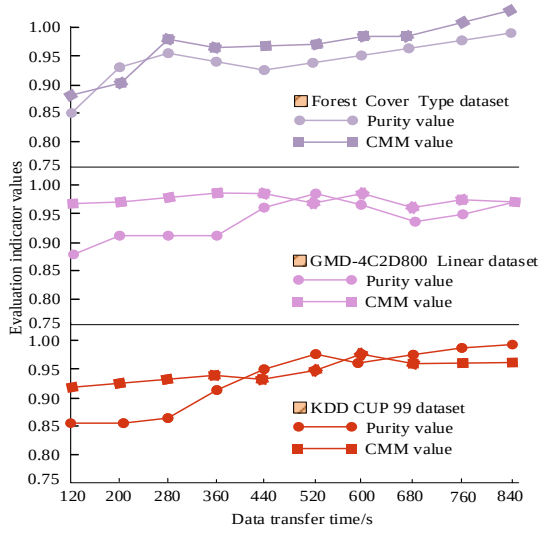


Figure 3: Performance test of data stream clustering under time transformation

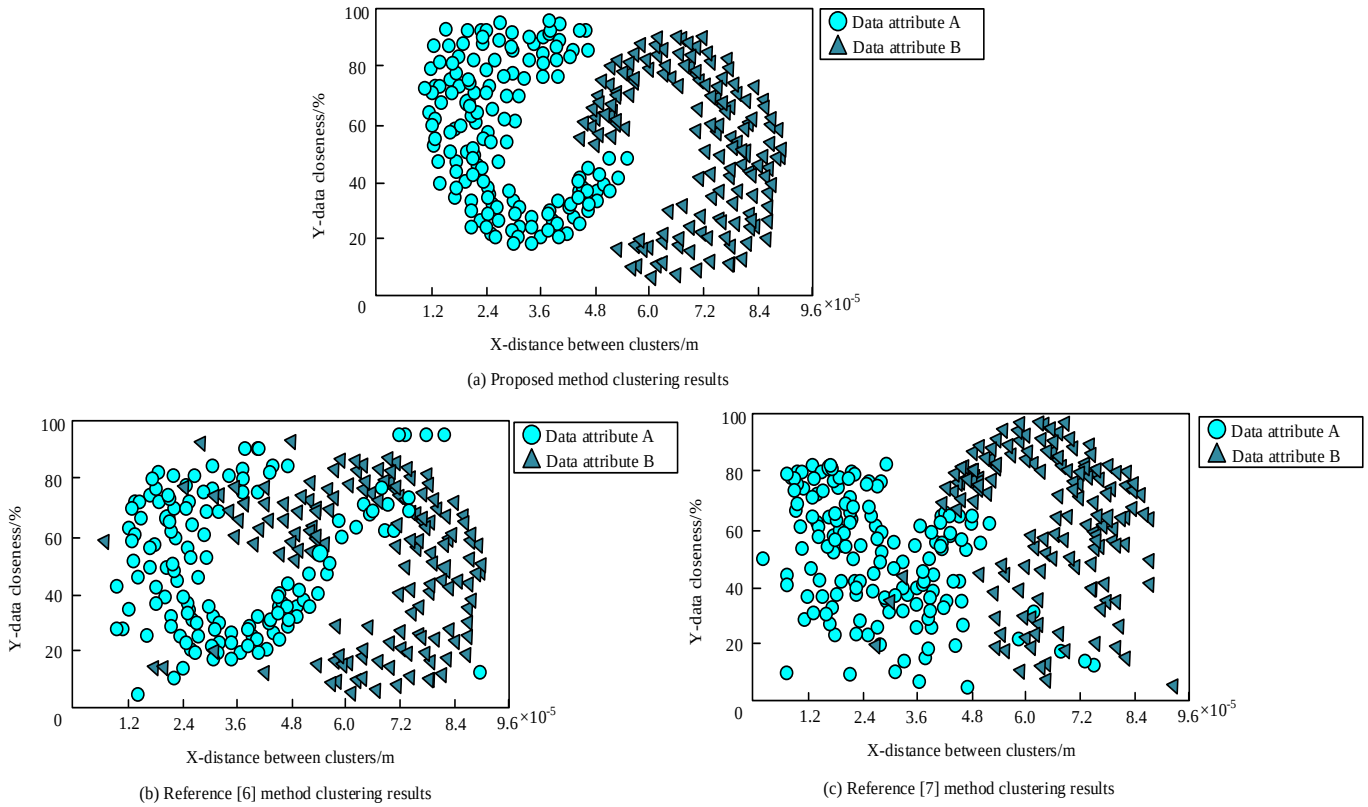
From Fig. 3, it can be seen that among the three different data streams, as the data transmission time increases, the proposed algorithm shows an overall increasing trend in both CMM and Purity metrics. However, due to the dynamic changes in the data stream, temporary performance degradation may occur (i.e. the underlying distribution of the data stream may undergo sudden changes over time, the algorithm needs time to adapt to the new distribution, and may erroneously merge new and old clusters during the transition period), resulting in a slight decrease in the numerical evaluation results obtained in some time intervals. Further analysis of the relationship between performance changes and data features reveals that in the Forest Cover Type dataset, due to the inclusion of 10 continuous attributes and 44 nominal attributes, as well as a total of 7 categories, the data distribution exhibits significant imbalance. This imbalance leads to a brief decrease in CMM metrics during time window switching, but as the soft constraint mechanism is adjusted, the algorithm quickly returns to stability. In contrast, although the GMD-4C2D800 dataset has fewer categories (4 categories), the data distribution is uniform, so the Purity index always remains above 0.88. The KDD CUP 99 dataset contains changes in network attack patterns and significant concept drift, but the proposed method can still maintain CMM values above 0.92 through dynamic window adjustment, demonstrating good adaptability.

To quantitatively demonstrate the improvement of the proposed method compared to the comparative method, Table 4 lists the performance improvements on each datase

Table 4: Performance improvement of the proposed method compared to the comparative method

Dataset	Comparative indicators	Reference [6] method	Reference [7] method	Proposed method	Increase amplitude (vs [6])	Increase amplitude (vs [7])
Forest Cover Type	CMM	0.8329	0.9057	0.9644	0.158	0.065
	Purity	0.8763	0.9375	0.9752	0.113	0.040
GMD-4C2D800 Linear	CMM	0.7992	0.8873	1.0000	0.251	0.127
	Purity	0.8601	0.8909	0.9994	0.162	0.122
KDD CUP 99	CMM	0.8970	0.9113	0.9921	0.106	0.089
	Purity	0.9203	0.9462	0.9985	0.085	0.055

To further validate the clustering effect of the proposed method on mixed-attribute data streams, the Forest Cover Type dataset, GMD-4C2D800 Linear dataset, and KDD CUP 99 dataset are simplified and described as test datasets 1-3. Based on this, the methods of reference [6] and reference [7] are introduced as comparative methods, and the data flow distribution after clustering by each method is projected onto a two-dimensional space through principal component analysis (PCA) for visualization display to verify the clustering effect. All visualized images were generated at a resolution of 600 dpi, as shown in Fig. 4-6:



(a) Proposed method clustering results

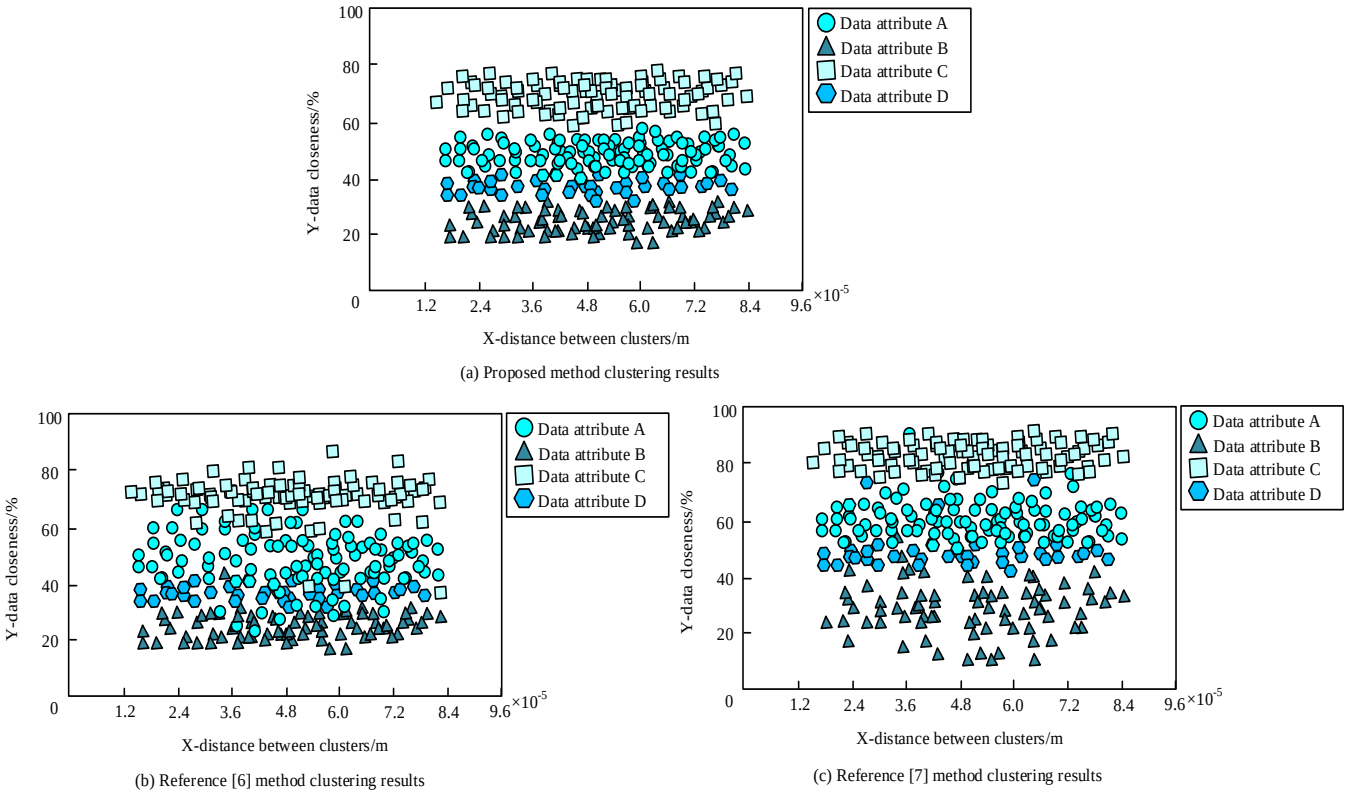
(b) Reference [6] method clustering results

(c) Reference [7] method clustering results

Figure 4: Clustering results for test dataset 1

In Figures 4-6, different colors represent different clustering clusters, and shapes represent real categories. As shown in Fig. 4~6, with the continuous increase of data attributes, there are significant differences in the clustering performance of various methods for mixed-attribute data streams. Among them, in the process of clustering various types of data attributes using the proposed method, the clustering results exhibit high accuracy, and the inter-cluster distances and data density distributions are generally well-structured; However, the other two methods may result in incorrect clustering during the clustering process, especially when the mixed-attributes continue to increase. The accuracy of the clustering results of the other two methods is significantly lower. It can be seen that using the proposed method can achieve higher clustering accuracy, as reflected in improved NMI and purity scores for mixed-attribute data streams.

In order to effectively distinguish the actual state of each method in the data stream clustering process, that is, whether it has reached the ideal state, accuracy and NMI (Normalized Mutual Information) values are introduced as evaluation indicators to test the clustering partitioning effect of different methods on each dataset. The specific results are shown in Table 5.

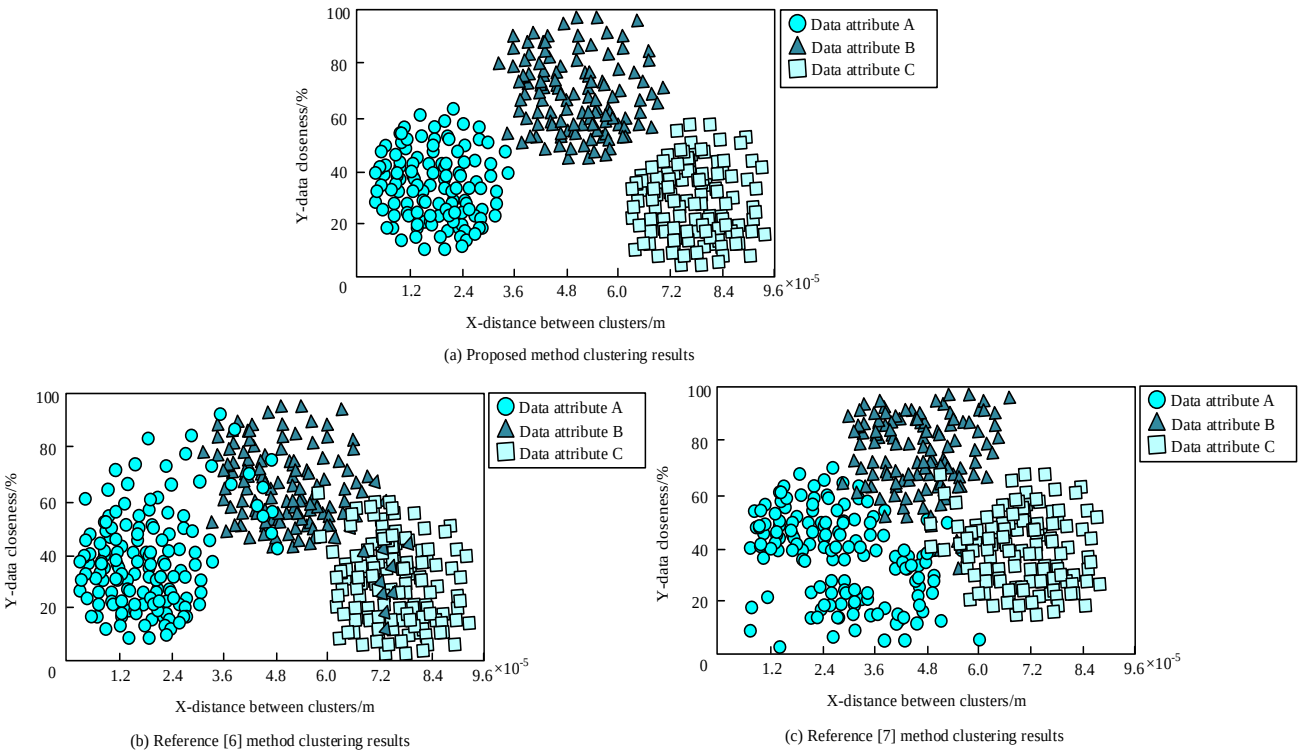


(a) Proposed method clustering results

(b) Reference [6] method clustering results

(c) Reference [7] method clustering results

Figure 5: Clustering results for test dataset 2



(a) Proposed method clustering results

(b) Reference [6] method clustering results

(c) Reference [7] method clustering results

Figure 6: Clustering results for test dataset 3

Table 5: Data stream clustering division state test results

Dataset name	Clustering algorithm	NMI value	Accuracy/%	Running time/s
Forest Cover Type dataset	Proposed method	0.9644	97.52	285.3
	Reference [6] method	0.8329	87.63	320.7
	Reference [7] method	0.9057	93.75	412.5
	DenStream	0.8215	85.92	378.9
	CluStream	0.8453	88.47	295.6
	StreamKM++	0.7986	83.15	265.8
GMD-4C2D800 Linear dataset	Design method	1.0000	99.94	12.7
	Reference [6] method	0.7992	86.01	15.3
	Reference [7] method	0.8873	89.09	18.9
	DenStream	0.8124	84.73	16.5
	CluStream	0.8315	86.95	13.8
	StreamKM++	0.7852	82.04	11.2
KDD CUP 99 dataset	Design method	0.9921	99.85	198.4
	Reference [6] method	0.8970	92.03	235.1
	Reference [7] method	0.9113	94.62	286.7
	DenStream	0.8637	90.25	252.3
	CluStream	0.8829	91.78	205.6
	StreamKM++	0.8415	88.43	185.9

According to Table 5, the method proposed in this article achieved optimal clustering accuracy (NMI and accuracy) and reasonable running time on all three datasets. The specific analysis is as follows: on the Forest Cover Type dataset, the NMI value of the proposed method reaches 0.9644, which is 14.1% higher than the traditional method CluStream and 20.8% higher than the modern method StreamKM++; In terms of running time, the proposed method (285.3s) outperforms the methods in reference [7] (412.5s) and DenStream (378.9s). Although it is slightly 7.3% slower than StreamKM++, the accuracy improvement is significant. On the GMD-4C2D800 Linear dataset, the proposed method achieved perfect clustering performance (NMI=1, accuracy 99.94%), with a running time of 12.7 seconds at a moderate level, achieving a good balance between accuracy and efficiency. On the KDD CUP 99 dataset, the proposed method also performs well, with an NMI value of 0.9921, which is 8.9% higher than the second-best method in reference [7], and a running time of 198.4 seconds, which is better than most comparison methods. Overall, the proposed method utilizes grid coupling and soft constraint mechanisms to significantly improve the clustering quality of mixed-attribute data streams while maintaining high operational efficiency, especially when dealing with high-dimensional and imbalanced data.

3.3 Comparative analysis of dimensionality reduction methods

To verify the effectiveness of Local Linear Embedding (LLE) in clustering mixed-attribute data streams, this paper compared three dimensionality reduction methods in the same experimental environment: LLE, Principal Component Analysis (PCA), and t-Distributed Random Neighborhood Embedding (t-SNE). On the Forest Cover Type dataset, three methods were used to reduce the original 54-dimensional data to 10 dimensions, and the clustering structure preservation ability of the reduced data was evaluated. The results are shown in Table 6:

Table 6: Comparison of cluster structure preservation ability of different dimensionality reduction methods

Dimensionality reduction method	Silhouette coefficient	Average distance within the cluster	Average distance between clusters	Dimensionality reduction time/s
LLE	0.724	1.23	5.89	12.4

PCA	0.536	2.15	4.32	3.2
t-SNE	0.681	1.45	5.41	28.7

According to Table 6, the LLE method outperforms PCA and t-SNE in terms of contour coefficient (0.724), average distance within clusters (1.23), and average distance between clusters (5.89), indicating that LLE can better preserve the clustering structure of the original data. Although LLE has a higher computation time than PCA, it is significantly lower than t-SNE, achieving a good balance between clustering quality and computational efficiency. This result validates the rationality of choosing LLE as the dimensionality reduction method in this paper.

3.4 Hyperparameter adjustment and experimental stability analysis

The method proposed in this article involves multiple key hyperparameters, namely grid size, sliding window length, and fuzzy attenuation factor. To ensure the reliability and accuracy of the experimental results, the optimal parameter combination was determined by combining grid search with 5-fold cross validation. On the Forest Cover Type dataset, search for grid sizes within the range of [2,44] with a stride of 2, sliding window lengths within the range of [300,600] seconds with a stride of 60 seconds, and fuzzy attenuation factors within the range of [0.001,0.01] with a stride of 0.001. After determining the optimal parameter combination, to further evaluate the stability of the experimental results, 10 different random seeds were used to rerun the experiment, and the values of each evaluation index were recorded. The fluctuation of the results was analyzed by calculating the mean and standard deviation. In addition to Forest Cover Type, there are also Dataset A and Dataset B. After running experiments using 10 different random seeds on the three datasets, the mean and standard deviation results of CMM and Purity indicators are shown in Table 7.

Table 7: Experimental stability analysis

Dataset	CMM mean	CMM standard deviation	Purity mean	Purity standard deviation
Forest Cover Type	0.925	0.012	0.918	0.009
Dataset A	0.883	0.015	0.876	0.012
Dataset B	0.901	0.008	0.895	0.006

According to Table 7, the proposed method exhibits good stability on all three datasets. In terms of CMM indicators, the standard deviation range is between 0.008-0.015, indicating that under different random seeds, the fluctuation of CMM indicators is small and the values are relatively stable. In terms of the Purity index, the standard deviation range is 0.006-0.012, which also indicates that the fluctuation degree of this index is relatively low under different random seeds. This low volatility fully demonstrates that the proposed method is insensitive to random initialization, has excellent robustness, and can stably perform on different datasets, providing reliable guarantees for practical applications.

4 Discussion

4.1 Analysis of method advantages

From the quantitative results, the accuracy (up to 99.94%) and NMI value (up to 1) of our method on three datasets are significantly better than the comparative methods, verifying its effectiveness in clustering mixed-attribute data streams. Specifically, compared to the fixed window method in reference [6] and the generative adversarial network method in reference [7], our method exhibits significant advantages in the following aspects:

Concept drift adaptability: Through the dynamic sliding window and grid coupling update mechanism, this method can track real-time changes in data distribution. When there is a sudden change in the data flow (such as the change in attack mode in the KDD CUP 99 dataset), the dynamic adjustment of the grid centroid makes the CMM index above 0.894, while the comparison method shows a significant decrease under the same conditions (the accuracy of the method in reference [6] drops to 87.63%).

Density constraint optimization: Introducing soft constraints based on fuzzy mathematics effectively limits the high-frequency movement of grid clusters. In the GMD-4C2D800 dataset, density soft constraints maintain clustering purity above 0.95, while traditional micro clustering methods (such as literature [4]) suffer from cluster splitting due to the lack of such constraints.

Mixed-attribute processing: Through local linear embedding dimensionality reduction and standardization processing, this method effectively preserves the distribution characteristics of categorical attributes while maintaining the distance relationship between numerical attributes. On the 44-dimensional nominal attributes of the Forest Cover Type dataset, the NMI value of our method reached 0.9644, significantly higher than the 0.9057 value of the method in reference [7].

4.2 Limitations and parameter sensitivity analysis

Although the method proposed in this article performs well in experiments, its effectiveness may be limited under the following conditions:

High dimensional sparse data: When the data dimension exceeds 50 dimensions and the sparsity is higher than 90%, the neighborhood reconstruction error of local linear embedding increases, which may lead to structural distortion after dimensionality reduction. At this point, it may be considered to combine feature selection or switch to other dimensionality reduction methods (such as t-SNE).

Parameter sensitivity: The sliding window size and fuzzy attenuation factor have a significant impact on clustering performance. Experiments have shown that when the window size deviates from the optimal value by ± 15% or the fuzzy attenuation factor is greater than 0.005, the clustering accuracy may decrease by 5-8%. Therefore, in practical applications, parameter tuning needs to be carried out through grid search.

Computing resource requirements: Grid coupling and centroid updates during the online phase require a significant amount of memory resources. In memory constrained environments such as embedded devices, it may be necessary to simplify the grid structure or adopt approximate calculations.

5 Conclusion

This article proposes a mixed-attribute data stream clustering algorithm that combines soft constraints. The algorithm integrates grid coupling and soft constraint principles, focusing on minimizing deviation values by standardizing data clustering boundaries and actual distributions; Using grid centroids instead of raw data to clarify the dynamic relationships of mixed-attribute data streams and perform data clustering effectively addresses concept drift tracking failures and improves clustering accuracy. Through experimental analysis, it can be concluded that the clustering algorithm proposed in this paper can effectively distinguish data with different category attributes in data streams with different attributes. It demonstrates strong adaptability to changes in data stream distributions and maintains a relatively efficient clustering effect.

Author contributions

W.B. Wu contributed to the study conception, design and writing.

Data availability statements

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests

The author has no competing interests to declare that are relevant to the content of this article.

References

[1] Y. Gao, Z. Fang, J. Xu, S. Gong, C. Shen and L. Chen, (2024) "An Efficient and Distributed Framework for Real-Time Trajectory Stream Clustering," *in IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 5, pp. 1857-1873, doi: 10.1109/TKDE.2023.3312319.

[2] K.T. Jafseer, S. Shailesh, and A. Sreekumar. (2024) "CPOCEDS-concept preserving online clustering for evolving data streams". *Cluster Computing*, vol. 27, pp. 2983–2998. https://doi.org/10.1007/s10586-023-04121-8.

[3] Riquelme A I, Ortiz J M (2024) "A Riemannian Tool for Clustering of Geo-Spatial Multivariate Data". *Mathematical geosciences*, vol. 56, no. 1, pp. 121-141. DOI:10.1007/s11004-023-10085-7

[4] S. Agarwal, C. R. K. Reddy (2024) "A smart intelligent approach based on hybrid group search and pelican optimization algorithm for data stream clustering". *Knowledge and information systems*, vol. 66, no. 4, pp. 2467-2500. https://doi.org/10.1007/s10115-023-02002-5

[5] S. Gorrab, F. BEN REJAB and K. Nouira. (2024) "Split incremental clustering algorithm of mixed data stream". *Progress in Artificial Intelligence*, vol. 13, no. 1, pp. 51-64. https://doi.org/10.1007/s13748-024-00316-1

[6] X.B. Jiang, Y.C. Jiang, L.P. Liu, M. Xia, Y.L. Jiang. (2024) "Time dimension feature extraction and classification of high-dimensional large data streams based on unsupervised learning". *Journal of computational methods in sciences and engineering*, vol. 24, no. 2, pp. 835-848. https://doi.org/10.3233/JCM-237085

[7] S. Matheswaran, N. Nachimuthu, and G. Prakash. (2024) "Efficient Online Big Data Stream Clustering Using Dual Interactive Wasserstein Generative Adversarial Network". *International Journal on Artificial Intelligence Tools*, vol. 33, no. 5, pp. 153-178. https://doi.org/10.1142/S021821302450009X

[8] R. A. Patil, P. D. Patil. (2024) "Efficient approximation and privacy preservation algorithms for real time online evolving data streams". *World wide web*, vol. 27, no. 1, pp. 1.1-1.20. https://doi.org/10.1007/s11280-024-01244-9

[9] Wen H, Liang M, Zhao S, et al (2025) "Unsupervised attribute reduction algorithm framework based on spectral clustering and attribute significance function". *Applied Intelligence*, vol. 55, no. 1, pp. 1-26. DOI:10.1007/s10489-024-05878-0

[10] M. Soleymanian, H. MashayekhI, M, Rahimi. (2024) "An incremental clustering algorithm based on semantic concepts". *Knowledge and information systems*, vol. 66, no. 6, pp. 3303-3335. https://doi.org/10.1007/s10115-024-02063-0

[11] A. K. Kar, A. C. Mishra, S. K. Mohanty. (2025) "EDMIX: an entropy-based dissimilarity measure to cluster mixed data comprising of numerical–nominal–ordinal attributes". *Knowledge and Information Systems*, vol. 67, no. 3, pp. 3023-3045. https://doi.org/10.1007/s10115-024-02319-9

[12] K.X. Chu, M. Zhang, Y.L. Xun, et al. (2024) "A hybrid similarity measure-based clustering approach for mixed attribute data". *International journal of machine learning and cybernetics*, vol. 15, no. 4, pp. 1295-1311. https://doi.org/10.1007/s13042-023-01968-6

[13] B.W. Lin, C.H. Liu, D.Q. Miao. (2024) "An improved decision tree algorithm based on boundary mixed attribute dependency". *Applied Intelligence*, vol. 54, no. 2, pp. 2136-2153. https://doi.org/10.1007/s10489-023-05238-4

[14] J. MO, J. KE, H. ZHOU, et al. (2025) "Hybrid network intrusion detection system based on sliding window and information entropy in imbalanced dataset". *Applied Intelligence*, vol. 55, no. 6, pp. 433. https://doi.org/10.1007/s10489-025-06307-6

[15] E. A. Asiamah, N. K. Akrasi-mensah, P. Odame, et al. (2025) "A storage-efficient learned indexing for blockchain systems using a sliding window search enhanced online gradient descent". *The Journal of Supercomputing*, vol. 81, no. 1, pp. 321. https://doi.org/10.1007/s11227-024-06805-3

[16] M. Ranalli, and R. Roberto. (2024) "Composite likelihood methods for parsimonious model-based clustering of mixed-type data". *Advances in data analysis and classification*, vol. 18, no. 2, pp. 381-407. https://doi.org/10.1007/s11634-023-00539-5

[17] J.X. Chen, Y.J. Gong, W.N. Chen and J. Zhang. (2024) " EvoS&R: Evolving Multiple Seeds and Radii for Varying Density Data Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 5, pp. 1964-1978. https://doi.org/10.1109/TKDE.2023.3312760

[18] J.Q. Weng, L. Lv, T.H. Fan, and P. Kang. (2024) "Evolving Data Stream Clustering Algorithm Based on Density Peaks". *Computer Simulation*, vol. 41, no. 6, pp. 448-454.