Pattern Recognition 171 (2026) 112280

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Unsupervised heterogeneous group streaming feature selection

Peng Zhou [a,b], Qianzhen Chen [a,b], Lei Sang [a,b,*], Shu Zhao [a,b], Xindong Wu [c]

[a] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, 230601, Anhui Province, PR China
[b] School of Computer Science and Technology, Anhui University, 230601, Hefei, Anhui Province, PR China
[c] Ministry of Education of China, Key Laboratory of Knowledge Engineering with Big Data, 230009, Anhui Province, PR China

ARTICLE INFO

Keywords:
Feature selection
Unsupervised feature selection
Streaming feature
Heterogeneous feature group
Feature clustering

ABSTRACT

Feature selection aims to select the optimal feature subsets from the dataset and has been widely applied in many fields and systems. However, data are not always static, and most of them are unlabeled. Besides, features may be heterogeneous and generated dynamically in practical applications. Therefore, online streaming feature selection was proposed that assumes the features are generated one by one or group by group on the fly while the number of instances remains fixed. This paper focuses on a new practical issue of online unsupervised streaming feature selection where the features are heterogeneous and dynamically generated in groups. Difficulties come from three aspects: the lack of label information, the uncertainty about the feature space, and the dynamic generation of heterogeneous streaming features. To solve this issue, we propose a new online Unsupervised Heterogeneous Group Streaming Feature Selection method named UHGSFS. To handle the problem of heterogeneous streaming features without the feature type information, UHGSFS applies MIC (Maximal Information Coefficient) to evaluate feature relationships without assuming data distribution in advance. To address the challenge of unlabeled information, UHGSFS clusters streaming features by the density based on the Gaussian kernel function and minimizes redundancy by selecting representative features. Extensive experiments were conducted on 13 benchmark datasets, with comprehensive comparisons against state-of-the-art supervised and unsupervised streaming feature selection methods. The experimental results demonstrate that our proposed method achieves comparable or even superior performance relative to supervised streaming feature selection methods.

1. Introduction

The primary objective of feature selection is to discern optimal subsets from the original feature set, thereby enhancing model performance [1]. In the contemporary era of big data, the exponential proliferation of data has precipitated the advent of expansive feature space, thereby engendering what is commonly called the “dimensionality disaster” [2]. Feature selection emerges as a pivotal instrument to address this predicament. By choosing informative features and discarding irrelevant ones, feature selection not only refines learning performance but also engenders heightened computational efficiency and diminished memory storage requirements [3,4].

With the increase in data volume and dimensionality, traditional feature selection methods can no longer meet the demand in terms of efficiency [2]. Meanwhile, traditional feature selection methods assume that all instances and features in the target dataset can be obtained before learning. However, in actual application scenarios, e.g., in image analysis [5], Mars exploration [6], and healthcare [7], we are often confronted with streaming features (features are generated and keep arriving over time) [8]. As shown in Fig. 1, suppose all the employees of a company go for a health check-up. Each employee needs to be examined by different items, and groups of streaming features are constantly generated by different examination items. Meanwhile, the number of samples (employees) remains fixed. With the arrival of new streaming features, we want to execute feature selection and machine learning models as soon as possible instead of waiting long for all features. Therefore, online streaming feature selection faces two main challenges: 1) We cannot know the information of the entire feature space before learning; 2) The method needs to decide whether to keep or discard newly arrived features since storing all these streaming features is impossible [9].

In recent years, many techniques for supervised streaming feature selection have been proposed, such as Fast-OSFS (Online feature selection with streaming features) [8], OFS-Density (Ofs-density: A novel online streaming feature selection method) [10], OFS-3WD (Online scalable streaming feature selection via dynamic decision) [9] and LOSSA (A

[*] Corresponding author.
E-mail addresses: doodzhou@ahu.edu.cn (P. Zhou), qianzhen@stu.ahu.edu.cn (Q. Chen), sanglei@ahu.edu.cn (L. Sang), zhaoshuzs@ahu.edu.cn (S. Zhao), xwu@hfut.edu.cn (X. Wu).

https://doi.org/10.1016/j.patcog.2025.112280
Received 9 December 2024; Received in revised form 20 June 2025; Accepted 9 August 2025
Available online 16 August 2025

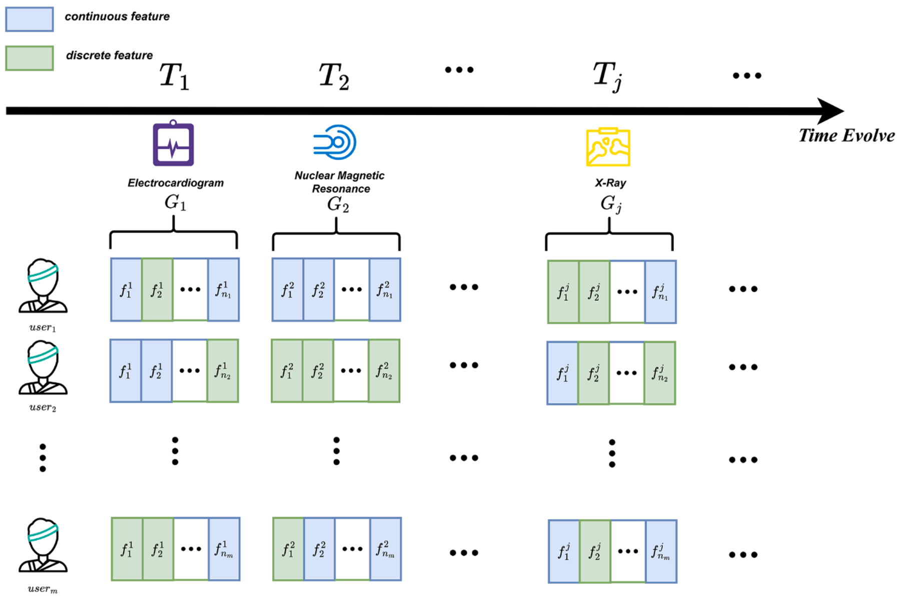

continuous feature
discrete feature
T_1
T_2
T_j
Time Evolve
Electrocardiogram
G_1
Nuclear Magnetic Resonance
G_2
X-Ray
G_j
$user_1$
$user_2$
$user_m$
f_1^1 f_2^1 $f_{n_1}^1$
f_1^2 f_2^2 $f_{n_1}^2$
f_1^j f_2^j $f_{n_1}^j$
$f_{n_2}^1$ $f_{n_2}^2$ $f_{n_2}^j$
$f_{n_m}^1$ $f_{n_m}^2$ $f_{n_m}^j$

Fig. 1. Illustration of a real-world scenario for unsupervised heterogeneous group streaming feature selection. When a group of users enters the hospital, several inspection items are required to determine the cause of the illness. The results of different inspection items (group features) arrive randomly over time, resulting in groups of heterogeneous streaming features. The issue we want to investigate is how to deal with heterogeneous streaming features without label information.

latent factor analysis-based approach to online sparse streaming feature selection) [11]. They mainly contain these four steps: 1) Receive new streaming features (individual/group); 2) Determine whether the new features are to be added to the candidate subset by analyzing the feature relevance; 3) Update the selected feature subset by redundancy analysis; 4) Repeat steps 1 to 3 until there are no more streaming features. However, labeled data is expensive for real-world applications. Although supervised streaming feature selection methods have made significant progress, their applicability remains limited in real-world scenarios dominated by unlabeled data.

So far, few studies have focused on unsupervised streaming feature selection. For example, Li et al. [12] designed a method for unsupervised streaming feature selection for social media using link information, called USFS. However, USFS can only handle isomorphic streaming features and requires stable link information with high computational complexity. In [13], UFSSF based on k-means clustering achieved unsupervised streaming feature selection for continuous individual streaming feature. Nevertheless, because the k-means algorithm is sensitive to noise, the UFSSF cannot deal with discrete streaming features. In [14], Yan et al. proposed a density formulation adapted to streaming features and implemented unsupervised streaming feature selection based on density clustering, named OUFSDFC. However, OUFSDFC cannot deal with the feature selection of mixed streaming data. It requires prior knowledge of the type of streaming feature (discrete or continuous) as well as the need to set clustering parameters artificially. Zheng et al. [15] proposed a new feature selection method for feature streaming, which introduces a dynamic similarity graph to evaluate irrelevant features adaptively and utilizes similarity graph diffusion to eliminate those irrelevant features in the feature streaming. However, this method needs adjusting multiple parameters to achieve the best effect and requires manually specifying the number of selected features. In summary, most state-of-the-art unsupervised streaming feature selection methods exhibit significant computational complexity, are constrained to isomorphic features, and cannot effectively handle heterogeneous streaming features. In practice, the order of arrival of streaming features is random. Therefore, knowing the feature type of the following arriving streaming features in advance is unrealistic.

The motivation of this paper lies in three aspects: 1) Traditional unsupervised feature selection cannot handle online scenarios where features are continuously arriving; 2) Existing supervised streaming feature selection methods require label information, and most of the data in real-world applications is unlabeled; 3) Existing unsupervised streaming feature selection methods are designed for homogeneous streaming features and cannot solve the issue of unknown feature type under heterogeneous streaming features. For example, a group of users needs various medical diagnostic procedures in sequence to determine the cause of the disease. Over time, these inspection items will continue to generate various streaming features, such as image data from X-rays, signal data from ECGs, and text data from electronic case reports. For an emergency patient, online streaming feature selection needs to be executed as fast as possible rather than waiting for the results of all examination items. Unsupervised heterogeneous group streaming feature selection has three challenges: 1) Unlabeled data: Since the features we obtain are all unlabeled, a clear standard must be used to distinguish and measure the importance of features. 2) Continuity of streaming feature: Features arrive as streams. Before learning, we could not know all the feature space information and could not cache all features. 3) Randomness of streaming feature: We cannot determine the type of features arriving in each group and their order of arrangement.

Motivated by this, we propose a new online unsupervised feature selection method where the streaming features are heterogeneous and dynamically generated in groups. Meanwhile, we assume that the feature types of streaming features are unknown in advance for practical situations. In order to solve this problem, we apply MIC (Maximum Information Coefficient) [16] to measure the relationship between heterogeneous features, which can discover nonlinear relationships and complex correlation structures. Then, we utilize feature density analysis to compute the density of each streaming feature based on the Gaussian kernel function. Based on the density information, we propose a new adaptive feature clustering method to group arriving streaming features. Finally, we select the representative features with the highest density from feature clusters. The main contributions of this paper are as follows:

- We focus on the new practical issue of online unsupervised heterogeneous streaming feature selection where streaming features are generated dynamically in groups without the information of feature types in advance. Compared with the idealized settings of existing online streaming feature selection methods, our work is closer to real application scenarios.

- We propose a new online unsupervised heterogeneous group streaming feature selection method based on density-based streaming feature clustering. Specifically, we first evaluate the density of arriving streaming features based on MIC and Gaussian kernel function. Then, we design a novel adaptive feature clustering method that does not require parameter setting and utilizes feature redundancy minimization to select the most representative subset of features. Our new method is nonparametric and does not need to consider the incoming streaming feature type, so it can better meet various practical application scenarios.
- Extensive experiments were conducted on 13 datasets, involving a meticulous examination and comparison with eight supervised streaming feature selection methods and one unsupervised streaming feature selection algorithm. The experimental results, coupled with rigorous statistical analyses, conclusively indicate that, in the absence of label information and under the circumstance of unknown streaming feature types, the performance of our new method is tantamount to that of supervised streaming feature selection algorithms or even better.

The novelty of this work lies in two aspects. First, compared with existing online streaming feature selection approaches, our new method aims to handle unsupervised heterogeneous streaming feature selection without the information of feature types before learning, which is closer to real application scenarios. Second, our proposed new method applies adaptive feature clustering based on natural search neighbors, making our new algorithm not need to specify any parameter values in advance. Meanwhile, our algorithm does not have to consider the streaming feature type, allowing us to handle arbitrary kinds of datasets without any prior knowledge.

The rest of this article is organized as follows. In Section 2, we describe related work. Section 3 proposes the formal definition of the problem, the technology involved, and a new approach for unsupervised heterogeneous streaming feature selection. In Section 4, comprehensive experimental analyses are presented. Section 5 discusses and concludes the proposed new method.

2. Related work

Feature selection has been studied for decades and has been an integral part of data mining and machine learning [2]. This section will briefly introduce popular traditional and online streaming feature selection methods.

2.1. Traditional feature selection methods

Depending on how the label information is used, traditional feature selection algorithms can be classified as supervised, unsupervised, and semi-supervised.

Supervised feature selection assumes we can acquire all the label information for the training data. More specifically, Fisher Score [17] achieved the same eigenvalues in the same class and different eigenvalues in different classes by calculating the ratio of inter-class separation and intra-class variance for each feature. Mutual information can measure the correlation between each feature and the target variable, and [18] discussed the practical implementation of mutual information (MI) in feature selection. The algorithm addresses noise and feature selection issues in some multi-label environments by establishing a robust label enhancement model. PMSNE [19] addressed feature noise and feature selection issues in some multi-label environments by establishing a robust label enhancement model.

Without the label information, unsupervised feature selection methods cannot use the label to measure the importance of each feature. For example, Laplacian Score [20] evaluated the importance of features by their variance and power of locality preservation and matched similar features by constructing the nearest neighbor graph. FSFS [21] introduced the Maximum Information Compression Index (MICI) to reduce feature redundancy by clustering features with high similarity and selected the most compact feature in each cluster (determined by the distance between features). MRMGRFS [22] aimed to maximize the relevance of features while minimizing the redundancy between features by integrating spectral clustering with the global redundancy minimization model, thereby effectively performing unsupervised feature selection.

Besides, semi-supervised feature selection assumes that we have a small amount of labeled data and a large amount of unlabeled data. For instance, SRFS [23] was a semi-supervised feature selection method that treats unlabeled data in a Markov blanket as labeled data through relevance gain. A-SFS [24] was a semi-supervised feature selection based on multi-task self-supervision, which innovatively introduces a self-supervised mechanism based on deep learning into the feature selection problem.

In summary, traditional feature selection methods necessitate full knowledge of the feature space before learning, rendering them unsuitable for online scenarios. Moreover, the substantial volume of data often results in computational overhead and time complexity. Therefore, when applied to large-scale datasets, traditional feature selection methods frequently fail to meet real-time requirements.

2.2. Online streaming feature selection methods

2.2.1. Supervised streaming feature selection

With the advent of the big data era, online streaming feature selection has become a research hotspot for processing high-dimensional datasets [25]. In general, online supervised streaming feature selection can be divided into individual streaming feature selection and group streaming feature selection.

Individual streaming feature selection methods assume features arrive one by one over time. For example, Grafting [26], which employs a stagewise gradient descent approach, represented the first individual-level streaming feature selection (SFS) method. Grafting considered feature selection as an integral part of learning a predictor within a regularized framework. Zhou et al. [27] proposed the Alpha-investing algorithm, which leverages streamwise regression for online streaming feature selection. Specifically, Alpha-investing was a streaming feature selection that evaluates the representativeness of features by means of a predefined p-value, but it ignored the redundancy of features and was unable to discard features inside a subset of selected features. OSFS [8] evaluated feature relevance based on the class label information of samples; it discarded irrelevant features according to specific criteria and dynamically identified which features were redundant. However, the method to identifying redundant features was inefficient. Based on OSFS, the Markov Blanket-based Fast-OSFS [8] can efficiently remove redundant features. In [28], SAOLA efficiently identifies the most informative features from large-scale and high-dimensional datasets by pairwise correlation techniques, and the algorithm is scalable. OFS-3WD [9] was a new online scalable streaming feature selection framework from a dynamic decision-making perspective, which dynamically classifies input features as selected, discarded, or delayed to minimize decision risk. [29] proposed a multi-objective online streaming feature selection method, which utilized mutual information theory and Pareto optimal set theory to select streaming features using a multi-objective search strategy. OCFSSFs [30] was an online feature selection method based on causal discovery by mining and identifying relationships in Markov blankets about parents and children (PCs) and spouses. LOSSA [11] was an online sparse streaming feature selection algorithm based on latent factor analysis, which uses latent factor analysis to solve the problem of missing data in sparse streaming features. In [31], redundancy analysis in streaming feature selection is defined as a binary optimization problem, and the binary bat algorithm (BBA) was used to reconsider the redundant features that were previously removed. OHSFS [32] formulated the streaming feature selection problem as a minimax problem by utilizing MIC (Maximum Information Coefficient) and did not require prior knowledge of the feature type information in advance. Meanwhile, some Rough Set-based streaming feature selection methods were proposed recently, including K-OFSD [33], OFS-A3M [34], OFS-density [10], OGSFS-FNGBRS [35], and ASFS [36].

Existing supervised streaming feature selection methods have achieved good performance in different aspects. However, these methods cannot be applied to unsupervised application scenarios since they require labeled data for training.

2.2.2. Unsupervised streaming feature selection

In practice, data is mostly unlabeled. Online unsupervised streaming feature selection methods were proposed to handle online feature selection without label information.

Specifically, in [12], the authors performed unsupervised streaming feature selection against social media data by identifying link information. This method is only applicable to social media data. Moreover, it can only handle isomorphic features and requires stable link information, leading to high computational complexity. In [13], the authors proposed unsupervised streaming feature selection based on k-means for continuous individual streaming features, which cannot handle discrete features because k-means is sensitive to noise. In [14], the authors developed a feature density formulation adapted to streaming features and achieved feature correlation maximization and redundancy minimization based on density clustering. However, it could not handle feature selection for mixed streaming data and artificially set a parameter for clustering, which is inconsistent with the unsupervised and non-parametric nature of real-world applications. In [15], the authors proposed a new feature selection method for feature streaming, which introduced dynamic similarity graphs to adaptively evaluate irrelevant features and used similarity graph diffusion to eliminate those irrelevant features in the feature streaming.

In sum, most existing streaming feature selection approaches are constructed upon supervised information, which overlooks the scarcity of labels in practical applications. While some researchers have introduced unsupervised streaming feature selection methods, their high computational complexity, restriction to isomorphic features, and incapacity to handle unknown type features render these methods unsuitable for addressing unsupervised heterogeneous streaming feature selection challenges.

3. The proposed method

In this section, we first present the formal definition of online unsupervised heterogeneous group streaming feature selection. Then, we introduce the proposed new method in detail. Finally, we analyze the time complexity of our new method. Table 1 summarizes the notation used in this paper.

Table 1
Summary on mathematical notations.

Notations	Definition
t	the timestamp
G_t	a group of streaming features arrive at timestamp t
$f_i^{G_t}$	the ith streaming feature in G_t
S_t	the selected feature subset after timestamp t
D_f	the density value of streaming feature f
$\mathbb{C}_t$	the extracted streaming feature cluster set at timestamp t
C_k	the kth streaming feature cluster in $\mathbb{C}_t$
r_C	the radius of the cluster C
$d(f_a, f_b)$	the distance from the feature f_a to the feature f_b
f_C	the center feature in cluster C
$\lvert \cdot \rvert$	the size of $\cdot$

3.1. Definitions and assumptions

[Online unsupervised heterogeneous group streaming feature selection]

Let $\mathbb{F} = \{G_1, G_2, \dots, G_T\}$ represent the entire streaming features, where $G_t = \{f^1_{G_t}, f^2_{G_t}, \dots, f^m_{G_t}\}$ denotes the streaming feature group arriving at timestamp t. For each streaming group G_t, the features are heterogeneous with each other, and we cannot know the feature type information and label information for each streaming feature $f^i_{G_t}$ in G_t. S_t denotes the selected feature subset at each timestamp t. Our aim is to obtain an optimal feature subset S_t from $\mathbb{F}$ at each timestamp.

The assumptions for online unsupervised heterogeneous streaming feature selection are as follows:

- The quantities of samples are fixed;
- Streaming features are heterogeneous and arrive as groups over time;
- The label information and feature type information for each streaming feature are unknown.

Since feature space changes over time or even infinite, we cannot store all features for online streaming feature selection. Meanwhile, since there is no label information, we have no basis for judging the importance of features. Therefore, unsupervised streaming feature selection often requires discovering relationships between features to select or discard features. In addition, we need an effective metric to explore the relationships between heterogeneous streaming features.

To solve this problem, we apply density-based streaming feature clustering to select the representative features and discard the redundant features. Suppose the streaming features are clustered into multiple clusters $\mathbb{C}_t$ at timestamp t, and C is one of the clusters in $\mathbb{C}_t$.

Definition 1 (Representative feature)**.** The representative feature in streaming feature cluster C is defined as the feature with the maximal density as $f_R = \underset{f \in C}{max}\{D_f\}$, where D_f is the density value of streaming feature f.

For the new arriving streaming group G_t, we calculate the density of each streaming feature and cluster them into multiple clusters. We designate the feature based on the local maximum density value as the cluster center and the representative feature.

Definition 2 (Redundant features)**.** For the features in the same cluster C, features other than representative features are considered redundant.

Based on streaming feature clustering, highly correlated features are clustered in the same cluster. According to **Definition 1**, we selectively choose the most representative feature from each cluster and consider the other features redundant. This approach can effectively reduce feature redundancy.

Based on these two definitions, our online unsupervised streaming feature selection method can effectively implement a strategy of maximizing relevance and minimizing redundancy.

3.2. Our new method

The depicted framework of our proposed Unsupervised Heterogeneous Group Streaming Feature Selection (UHGSFS) is illustrated in Fig. 2.

Specifically, we first apply the Maximum Information Coefficient (MIC) to measure the information between heterogeneous streaming features, which can discover complex non-linear relationships for a distance evaluation. Then, we calculate the density of each streaming feature. We select the feature with the current highest density to perform adaptive feature clustering to group the features into different clusters. Finally, we obtain the optimal feature subset by selecting the most relevant features and discarding redundant features.

3.2.1. The measure for unknown type and heterogeneous streaming feature

The generated streaming features in practical applications often exhibit heterogeneity. Meanwhile, each streaming feature arrives

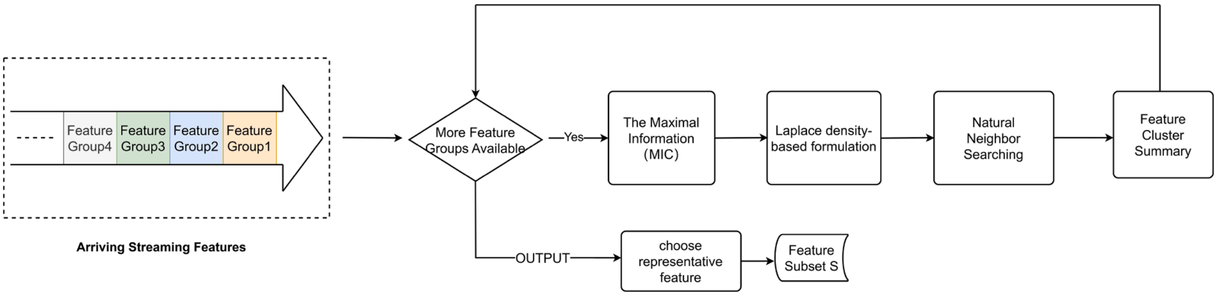
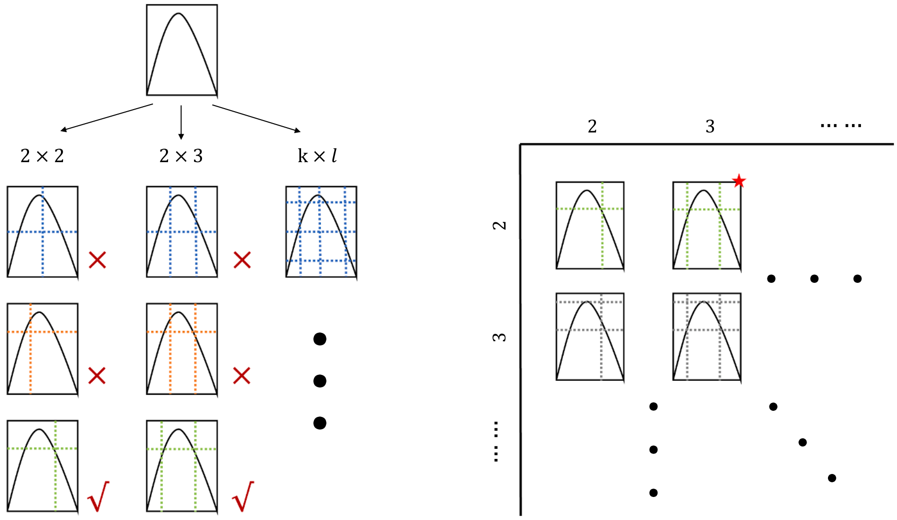

Feature Group4
Feature Group3
Feature Group2
Feature Group1
Arriving Streaming Features
More Feature Groups Available
Yes
The Maximal Information (MIC)
Laplace density-based formulation
Natural Neighbor Searching
Feature Cluster Summary
OUTPUT
choose representative feature
Feature Subset S

Fig. 2. An overview of the online unsupervised heterogeneous group streaming feature selection method.



(a) original instances distribution

(b) new spatial instances distribution after transductive warping

Fig. 3. A schematic illustration of the calculation of MIC using a parabola as an example.

randomly, and we cannot know their feature types in advance. Previous streaming feature selection methods are mainly designed to handle a single feature type (discrete or continuous). Although a few methods (e.g., [32]) have been proposed to address this challenge in the supervised learning domain, to the best of our knowledge, no one has yet investigated this problem in the unsupervised learning domain. Therefore, there is an urgent need to develop novel metrics that can effectively address the challenges posed by heterogeneous streaming features of unknown types.

MIC (Maximal Information Coefficient) is a data analysis algorithm that evaluates relationships between variables without assuming data distribution [16]. MIC partitions a scatter plot into grids and computes similarity using mutual information. The highest mutual information value across all grid partitions, normalized by the grid size, represents the Maximum Information Coefficient (MIC). MIC has proven to be effective in bridging the gap of mutual information and quantifying the correlation between nonlinear data more comprehensively.

In Fig. 3, MIC uses a dynamic axis division method, which allows the calculation of mutual information between discrete and continuous data, rendering it highly versatile across diverse applications. For variables x,y belonging to the two-dimensional data set $D = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, $MIC(x, y)$ is computed as follows:

$$MIC(x; y) = \max_{a*b<B} \frac{I(x, y)}{\log_2 min(a, b)} \tag{1}$$

$I(x, y)$ represents the mutual information value of variables x and y. The integers a and b, representing the number of partitions along the x and y directions, and determined via exhaustive that maximizes $I(x, y)$. The parameter B is a function on the number of samples n on the dataset D, with B set to $n^{0.6}$ in [16]. When calculating $I(x, y)$, it is necessary to divide the scatter plot into grids and traverse all possible segmentation schemes to find the optimal value. This is not realistic for large sample data. Therefore, David et al. [16] proposed a fast approximation algorithm based on a dynamic programming approach, which makes it more efficient to find the grid division that maximizes $I(x, y)$ within a limited search range, and finally obtaining an accurate MIC value.

For online heterogeneous streaming feature selection, the inherent uncertainty lies in the inability to ascertain the feature type of the next streaming feature preemptively. The Maximum Information Coefficient (MIC) uniquely quantifies correlations across various types of variables. In light of our objective to compute feature distances, we leverage the concept that higher correlations result in shorter distances. Therefore, we incorporate the complementary metric of $1 - MIC$ to measure and represent the distances between these heterogeneous streaming features effectively within our novel information metric framework.

3.2.2. The density of streaming feature

The Gaussian kernel function has many applications in many density-based offline clustering methods, which can be a good solution to the problem of outliers or noise. For example, [37], considering the dynamics of data streaming, a recursive Gaussian kernel lower-bound function formula is derived, which can be used in the absence of a large amount of a priori knowledge by considering historical samples to obtain the density of each sample for the calculation. The formula derived by the Laplace density in [14] is a new lower bound function for feature

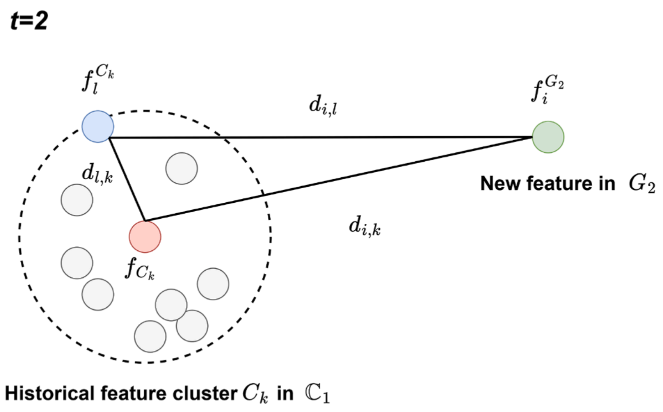


Fig. 4. Triangle inequality relationship between distances: $d_{i,l} \le d_{i,k} + d_{l,k}$. At timestamp $t = 2$, the new streaming feature $f_i^{G_2}$ (belonging to the feature group G_2) can form a triangle with the historical feature cluster center f_{C_k} and the historical feature $f_l^{C_k}$ (belonging to historical feature cluster C_k).

streaming. Since the derived Laplace mixture model is more robust in handling anomalous features, the derived Laplace density lower bound formula can perform stream feature density calculations that contain noise.

Specifically, assuming G_1 is the first streaming feature group and the distance from the current feature $f_i^{G_1}$ to the rest of the features in G_1 obeys the Laplace distribution, the density of feature $f_i^{G_1}$ can be expressed as:

$$D_{f_i^{G_1}} = \sum_{j=1}^{|G_1|}\left(e^{-\frac{d(f_i^{G_1},f_j^{G_1})}{\beta_1}}\right)^{\gamma_1}, \tag{2}$$

where β_1 is the normalization parameter at timestamp $t = 1$, and we set β_1 to the maximum value of the distance between two features in the streaming feature group G_1. γ_1 is the stabilization parameter at timestamp $t = 1$, and we estimate this parameter via [37]. In the experiments, we set γ_t to 5. $d(f_i^{G_1}, f_j^{G_1})$ refers to the dissimilarity between an input feature $f_i^{G_1}$ and another feature $f_j^{G_1}$ in G_1. Here we calculate it by $d(f_i^{G_1}, f_j^{G_1}) = 1 - MIC(f_i^{G_1}, f_j^{G_1})$.

Then, we receive the streaming feature group G_2, and calculate the new density of each streaming feature as:

$$D_{f_i^{G_2}} = \sum_{k=1}^{|G_1|}(e^{-\frac{d_{i,k}}{\beta_2}})^{\gamma_2} + \sum_{j=1}^{|G_2|}(e^{-\frac{d_{i,j}}{\beta_2}})^{\gamma_2}, \tag{3}$$

where $d_{i,k}$ denotes the distance between $f_i^{G_2}$ and $f_k^{G_1}$, $d_{i,j}$ denotes the distance between $f_i^{G_2}$ and $f_j^{G_2}$, β_2 and γ_2 denotes the value at timestamp $t = 2$.

For online streaming group selection, we cannot cache all the streaming features at each timestamp. Therefore, we cluster the streaming features in G_1 and use the cluster summary information to calculate the density of features in G_2. Specifically, suppose the feature cluster set for features in G_1 is $\mathbb{C}_1$, and C_k denotes the kth cluster in $\mathbb{C}_1$. Then, Eq. (3) equals to:

$$D_{f_i^{G_2}} = \sum_{k=1}^{|\mathbb{C}_1|}\sum_{l=1}^{|C_k|}(e^{-\frac{d_{i,l}}{\beta_2}})^{\gamma_2} + \sum_{j=1}^{|G_2|}(e^{-\frac{d_{i,j}}{\beta_2}})^{\gamma_2}, \tag{4}$$

where $d_{i,l}$ denotes the distance between $f_i^{G_2}$ and feature $f_l^{C_k}$ in cluster C_k.

According to Fig. 4, we can know that feature $f_i^{G_2}$, feature $f_l^{C_k}$ and the cluster center f_{C_k} can form a triangle. Based on the trigonometric inequality $\Delta\left(f_i^{G_2}, f_l^{C_k}, f_{C_k}\right)$,we introduce the relation that

$$d_{i,l} \le d_{i,k} + d_{l,k}, \tag{5}$$

where $d_{i,l}$ denotes the distance between $f_i^{G_2}$ and feature $f_l^{C_k}$ in cluster C_k, $d_{i,k}$ denotes the distance between $f_i^{G_2}$ and the cluster center f_{C_k}, $d_{l,k}$ denotes the distance between feature $f_l^{C_k}$ and f_{C_k}.

According to Eq. (5) and the monotone decreasing property of e^{-d}, we can conclude:

$$\sum_{l=1}^{|C_k|}\left(e^{-\frac{d_{i,l}}{\beta_2}}\right)^{\gamma_2} \ge \sum_{l=1}^{|C_k|}\left[\left(e^{-\frac{d_{l,k}}{\beta_2}}\right)^{\gamma_2} \times \left(e^{-\frac{d_{i,k}}{\beta_2}}\right)^{\gamma_2}\right] \tag{6}$$

Also since $d_{i,k}$ is seen as constant for all features in the cluster C_k to $f_i^{G_2}$, Eq. (6) can be modified as:

$$\sum_{l=1}^{|C_k|}\left(e^{-\frac{d_{i,l}}{\beta_2}}\right)^{\gamma_2} \ge \left(e^{-\frac{d_{i,k}}{\beta_2}}\right)^{\gamma_2} \times \sum_{l=1}^{|C_k|}\left(e^{-\frac{d_{l,k}}{\beta_2}}\right)^{\gamma_2} \tag{7}$$

When a new cluster emerges, the variance of the streaming feature will change while the following relationships remain constant:

$$\frac{\beta_2}{\gamma_2} = \frac{\beta_1}{\gamma_1}. \tag{8}$$

So we can launch:

$$\sum_{l=1}^{|C_k|}\left(e^{-\frac{d_{l,k}}{\beta_2}}\right)^{\gamma_2} = \sum_{l=1}^{|C_k|}\left(e^{-\frac{d_{l,k}}{\beta_1}}\right)^{\gamma_1}. \tag{9}$$

Let $D_{f_{C_k}} = \sum_{l=1}^{C_k}\left(e^{-\frac{d_{l,k}}{\beta_1}}\right)^{\gamma_1}$, where $D_{f_{C_k}}$ represents the density of the f_{C_k}. Then Eq. (3) can be written as

$$D_{f_i^{G_2}} \ge \sum_{k=1}^{|\mathbb{C}_1|}\left[\left(e^{-\frac{d_{i,k}}{\beta_2}}\right)^{\gamma_2} \times D_{f_{C_k}}\right] + \sum_{j=1}^{|G_2|}\left(e^{-\frac{d_{i,j}}{\beta_2}}\right)^{\gamma_2} \tag{10}$$

Based on Eq. (10), we can define the estimated lower bound on the density value of the new streaming feature $f_i^{G_2}$ as:

$$\hat{D}_{f_i^{G_2}} = \sum_{k=1}^{|\mathbb{C}_1|}\left[\left(e^{-\frac{d_{i,k}}{\beta_2}}\right)^{\gamma_2} \times D_{f_{C_k}}\right] + \sum_{j=1}^{|G_2|}\left(e^{-\frac{d_{i,j}}{\beta_2}}\right)^{\gamma_2}, \tag{11}$$

and $D_{f_i^{G_2}} \ge \hat{D}_{f_i^{G_2}}$.

Similarly, we can calculate the density of each new streaming feature in G_t at timestamp t by using the equation:

$$D_{f_i^{G_t}} = \sum_{k=1}^{|\mathbb{C}_{t-1}|}\left[\left(e^{-\frac{d_{i,k}}{\beta_t}}\right)^{\gamma_t} \times D_{f_{C_k}}\right] + \sum_{j=1}^{|G_t|}\left(e^{-\frac{d_{i,j}}{\beta_t}}\right)^{\gamma_t}, \tag{12}$$

where C_k is the kth cluster center in $\mathbb{C}_{t-1}$, $d_{i,k}$ is the distance between $f_i^{G_t}$ and f_{C_k}, and $d_{i,j}$ is the distance between $f_i^{G_t}$ and $f_j^{G_t}$, and $D_{f_{C_k}}$ is the density of the cluster center feature f_{C_k}.

Based on the Eq. (12), we can use the historical cluster density information to calculate the newly arrived streaming feature density without saving the entire history of streaming feature.

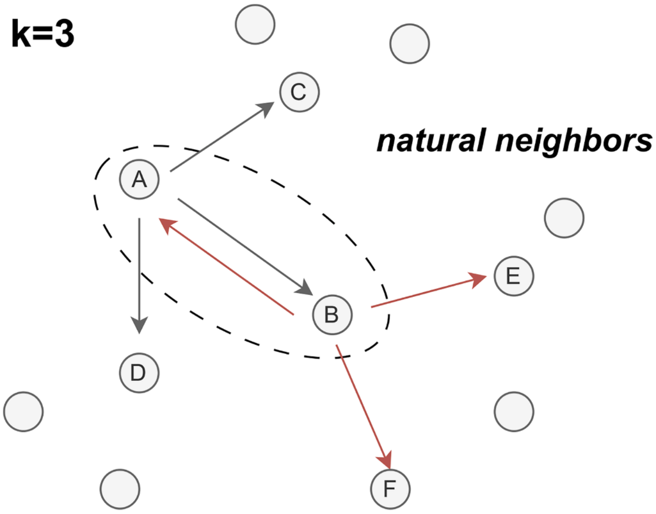


Fig. 5. Natural neighborhood search. Suppose we are looking for k nearest neighbors (k = 3). It can be observed that node A is closest to nodes B, C, and D. Meanwhile, node B is closest to nodes A, E, and F. Based on the natural neighbors search, it can be inferred that nodes A and B are neighbors.

3.2.3. *Adaptive feature cluster method*

“Natural Neighbors” introduces a novel paradigm of proximity akin to human social interactions: for instance, a trio of individuals denoted as A, B, and C. If A lacks amicability towards B, while B reciprocates with friendliness towards A, one may infer that A’s affinity towards B is unidirectional rather than a reciprocal friendship. Conversely, if A and C both display mutual animosity, it establishes that they do not share a close bond. In contrast, the amicability between B and C suggests a genuine friendship. Unlike traditional k-nearest neighbors search, Fig. 5 illustrates the natural neighbor search process.

Diverging from the conventional k-nearest neighbor (KNN) approach, the natural nearest neighbor method leverages the intrinsic structure of the dataset to discern the closest neighbors for each data instance. Noteworthy is its unique capacity to dynamically ascertain the optimal number of nearest neighbors for each data point, obviating the need for a priori parameter specification. Motivated by the notion of natural neighborhoods, we extend this concept to devise an adaptive feature clustering method tailored for feature streaming data.

Given a data set $\mathbb{X}$, we define *d(p,q)* as the distance between data points p and q in $\mathbb{X}$. Let data point q be the kth nearest neighbor of data point p. The interpretations of k-nearest neighbors, reversed k-nearest neighbor, natural stable state, and natural search neighbor are given as follows.

Definition 3 (K-Nearest Neighbors)**.** KNN: For data point p, its k-Nearest Neighbors set can be expressed as:

$$KNN_k(p) = Find_{knn}(p, k) \tag{13}$$

where $Find_{knn}(p, k)$ represents the searching function of KNN which searches for the kth nearest neighbors of p. Here, a KD tree can be utilized to accelerate the KNN searching process.

Definition 4 (Stable Searching State)**.** Suppose the searching round r increases from 1 to n. The natural neighbor searching becomes stable when the following condition is satisfied:

$$(\forall p \in \mathbb{X}) \wedge (\exists q \in \mathbb{X}) \wedge (p \neq q) \longrightarrow p \in KNN_r(q) \wedge q \in KNN_r(p) \tag{14}$$

When the stable searching state is reached, the maximal search round r is called the natural neighbor eigenvalue.

Definition 5 (Natural Search Neighbors)**.** When the natural neighbor searching process keeps a stable searching state, the data point p and data point q are natural neighbors if $p \in KNN_r(q)$ and $q \in KNN_r(p)$.

Motivated by the concept of natural neighbor search, we have formulated an algorithm for adaptive feature clustering, drawing upon the definition and analysis outlined above. A comprehensive exposition of the clustering process employing natural feature neighbor search is presented in Algorithm 1.

Specifically, for the target feature f, we aim to find its maximal natural neighbors as the cluster of f. Step 1 initializes $r = 1$. Then, we continuously search for rth neighbor of f as q in Step 3. Steps 4-7 check whether f belongs to the r nearest neighbors of q. If it does, then r is incremented by 1. Otherwise, it means that feature f reaches the **STABLE SEARCHING STATE** and we end the searching.

Algorithm 1 Natural Neighbor Clustering.

```
Input:
    $F$: Feature set;
    $f$: Feature $f$;
Output:
    $C_f$: The cluster of feature $f$;
1: Initialization: $r = 1$;
2: Repeat
3:     Find the $r$th nearest neighbor of $f$ as $q$
4:     If $f \in KNN_r(q)$
5:         $C_f = C_f \cup \{q\}$;
6:         r = r + 1;
7:     End If
8: Until $f$ reaches stable searching state
9: Return $C_f$
```

This paper assumes that the shape of the feature clusters formed by the Algorithm 1 is a sphere. In future work, we will consider the irregular shape of clusters. We determine whether two clusters can be merged by judging the distance between the two cluster centers, as shown in Fig. 6. Assuming that f_B represents the boundary feature of cluster $C1$, f_{C1} and f_{C2} represent the cluster centers of two adjacent clusters. If

$$d(f_B, f_{C1}) > d(f_{C1}, f_{C2}), \tag{15}$$

it means that there is an overlapping area between these two clusters. Then, these two clusters can be merged.

3.2.4. *The proposed method*

Since the online streaming feature selection operates unsupervised, it necessitates consideration of feature correlation and redundancy.

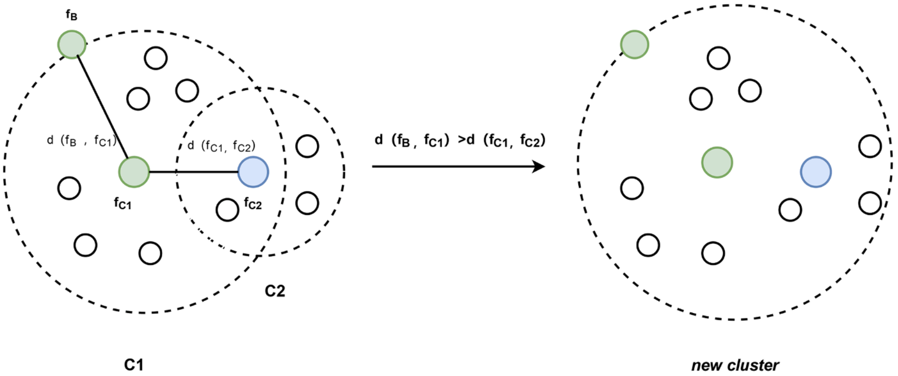


Fig. 6. Clusters merging. Cluster $C1$ (with center f_{C1}) and cluster $C2$ (with center f_{C2}) can be merged into a new cluster if the distance between f_{C1} and f_B is bigger than the distance between f_{C1} and f_{C2}.

Without data labels for measuring feature importance and relevance, we rely on the representation within each feature cluster to assess feature significance. In the context of density-based clustering methods, it is customary for the cluster center to exhibit the highest local density value, thus establishing the feature cluster center as the most pertinent feature within the cluster.

Moreover, given that features with more remarkable similarity are typically clustered within the same feature clusters, it is reasonable to infer that features in the same cluster exhibit redundancy. By exclusively selecting feature centers, we can effectively minimize redundancy. Therefore, we choose one representative feature from each feature cluster that is closest to the cluster center. Employing these two strategies enables the extraction of crucial feature subsets from a feature streaming while maintaining low redundancy and high correlation among them. Drawing from the preceding solution, we introduce UHGSFS, an unsupervised method designed to select heterogeneous streaming features, as shown in Algorithm 2.

Algorithm 2 Unsupervised Heterogeneous Group Streaming Feature Selection(UHGSFS).

```
Input:
    $G_t$: a group of streaming features arrive at timestamp t;
Output:
    $FS_t$: the selected feature subset;
 1: Repeat
 2:     $D_{G_t} = \{\}$: the density values of each feature in $G_t$;
 3:     For each feature $f_i^{G_t} \in G_t$
 4:         Calculate $D_{f_i^{G_t}}$ using Eq. 12 with MIC;
 5:         $D_{G_t} = D_{G_t} \bigcup \{D_{f_i^{G_t}}\}$;
 6:     End For
 7:     Rank all features in $G_t$ based on their density values;
 8:     $FC_t = \{\}$: the extracted streaming feature cluster set;
 9:     Repeat
10:         Select the feature $f$ in $G_t$ with the maximal density value;
11:         Generate the cluster $C_f$ by Algorithm 1;
12:         $FC_t = FC_t \cup C_f$;
13:         $G_t = G_t - C_f$;
14:     Until no more features in $G_t$
15:     Update $FC_t$ by merging clusters based on Eq. 15;
16:     If $FS_{t-1}$ in not empty
17:         Merge $FC_t$ with historical feature clusters in $FC_{t-1}$;
18:     End If
19:     $FS_t$ = select one representative feature from each cluster in $FC_t$;

20: Until no more streaming feature groups
21: Return $FS_t$
```

Specifically, suppose UHGSFS gets a new streaming feature group G_t at timestamp t. Steps 3-6 calculate the density values of each feature in G_t in terms of Eq. (12) and the maximum information coefficient (MIC). In step 7, we sort all the features by their density values. Steps 9-14, we select feature f with the highest density in G_t at each time and apply Algorithm 1 to generate the cluster of f until all the features are assigned to the clusters. Step 15 determines whether merging is required by the presence of density valleys between the two clusters. In steps 16-18, we check the possible merging of current clusters FC_t and historical clusters in FC_{t-1}. Finally, we select the representative feature from each cluster as the selected feature subset.

3.3. Time complexity

In our algorithm, the computation of the feature distance matrix for the unknown heterogeneous feature streaming necessitates $O(m^2)$ distance calculations. The search for feature clusters through our natural neighbor search method incurs $O(nlogn)$ computational complexity. Likewise, the merging process involving historical feature clustering and new feature clustering entails $O(|FC_t||FC_{t-1}|)$ calculations. As a result, the worst-case time complexity of our algorithm is expressed as $O(m^2 + nlogn + |FC_t||FC_{t-1}|)$.

4. Experiments

In this section, we conduct experiments on several benchmark datasets and comparative studies with state-of-the-art supervised and unsupervised streaming feature selection methods to demonstrate the effectiveness of our new framework.

4.1. Experiment setup

4.1.1. Datasets

To verify the effectiveness of our proposed new method, we conducted extensive experiments on 13 real-world datasets, including three discrete datasets and ten continuous datasets. We can find these datasets from the ASU feature selection repository[1]. These 13 datasets are widely used across various real-world applications, particularly in healthcare, image recognition, and speech processing. For instance, datasets like Arcene, ALLAML, Lymphoma, Colon, GLIOMA, Nci-9, and SMK_CAN_187 are crucial in cancer research, helping to classify and diagnose different types of cancer based on gene expression data, which aids in early detection, personalized treatments, and prognosis prediction. Datasets such as COIL20, Orlraws10P, Pixraw10P, and WarpPIE10P are used in object and facial recognition tasks, which

[1] Publicly available at https://jundongl.github.io/scikit-feature/datasets.html

Table 2
Experimental data dets.

Data Set	Instances	Features	Classes	Domain	Type
Arcene	200	10000	2	Medical	continuous
ALLAML	72	7129	2	Medical	continuous
COIL20	1440	1024	40	Image	continuous
Colon	62	2000	2	Biological	discrete
GLIOMA	50	4434	4	Biological	continuous
Isolet	1560	617	26	recognition	continuous
Lymphoma	96	4026	9	Medical	discrete
Nci-9	60	9712	9	Biological	discrete
Orlraws10P	100	10304	10	Image	continuous
Pixraw10P	100	10000	10	Image	continuous
SMK_CAN_187	187	19993	2	Biological	continuous
USPS	9298	256	10	Image	continuous
WarpPIE10P	210	1024	15	Image	continuous

have applications in security systems, identity verification, and human-computer interaction. The USPS dataset, focused on handwritten digit recognition, is applied in automatic document processing and postal sorting. Finally, the Isolet dataset is widely used in speech recognition systems, enhancing technologies like virtual assistants and voice-controlled devices. Table 2 summarises all the attributes of these data.

4.1.2. Comparing algorithms

We compare our new method with ten state-of-the-art streaming feature selection methods[2], including Alpha-investing [27], SAOLA [28], Fast-OSFS [8], OFS-Density [10], OFS-A3M [34], OFS-3WD [9], OHSFS [32], Group-SAOLA [28], OGSFS-FI [38] and OUFSDFC [14]. The first nine algorithms are supervised, and OUFSDFC is an unsupervised method. Meanwhile, the first seven algorithms are individual streaming feature selection methods, while the last three are group-based approaches. According to [38], the α value of Fast-OSFS, SAOLA, and Group-SAOLA is set to 0.01. We use the default parameters from [10,34] to go for feature selection.

4.1.3. Experimental settings

In experiments, we simulate the streaming features for each dataset in Table 2. Specifically, one feature is randomly selected from the dataset at each timestamp and sent to the competing algorithms. In other words, each competing streaming feature selection method can only require one random feature at each timestamp without the entire feature space information. After the feature streams end, we compare the quality of the selected features for each competing algorithm.

4.1.4. Evaluation metrics

Classification performance is used to show the effectiveness of these competing algorithms. Fundamental classifiers were employed, including KNN (k = 5), SVM (with the linear kernel), and DT (decision tree), to evaluate the feature subset selected in the experiment. For each dataset, we use a 5-fold cross-validation method, dividing the dataset into five equal parts, with four parts for training and one part for testing each time. We utilize the same training and testing set division for each competing algorithm. We repeat these experiments ten times for each dataset and report the average value and standard deviation as the final experiment results in each table.

We use the accuracy (Acc) and f-score to evaluate the performance of the selected feature subset. Considering the possible imbalance in the distribution of the data categories, we use F_{mac}, which is the macro-mean of the f-score, for the performance evaluation, and the expression of F_{mac} is as follows:

$$F_{mac} = \frac{1}{n_c}\sum_{i=1}^{n_c} F_i \qquad (16)$$

where F_i and n_c denote the F-measure for the ith class and the number of classes, respectively.

Besides, we conducted the statistical test using Friedman's test at a significance level of 95 %, assuming the null hypothesis is true. If the null hypothesis is rejected, we conduct the Nemenyi test as a post-hoc test and construct critical distance (CD) graphs [39].

4.1.5. Computational device

All experiments were conducted on a computer running Windows 10, AMD Ryzen 7 3700X 8-core processor 3.6 GHz, 16 GB RAM.

4.2. Results and analysis

Tables 3–8 present a comprehensive summary of the accuracy (ACC) and F_{mac} achieved by these competing algorithms, utilizing KNN, SVM, and DT classifiers. Additionally, Table 9 provides insights into each algorithm's average number of selected features. The best results are highlighted in boldfaces on the tables.

UHGSFS and OUFSDFC are implemented in Python, while the other nine algorithms are implemented in Matlab. Thus, we can only compare the running time between UHGSFS and OUFSDFC. However, we give the time complexity comparison between UHGSFS and 10 competing algorithms in Table 10. And Fig. 7 compares the running times between UHGSFS and OUFSDFC algorithms.

For the accuracy metric (ACC), the p-values obtained from the Friedman test for KNN, SVM, and DT(decision tree) classifiers are 2.4708e-04, 2.4708e-04, and 0.0019, respectively. Similarly, for F_{mac}, the p-values for KNN, SVM, and DT classifiers are 3.7915e-04, 0.0011, and 1.9891e-04, respectively. These p-values indicate significant differences exist in both accuracy and F_{mac} among these competing algorithms. The CD value for all classifiers is 4.1853. Fig. 8 illustrates the statistical analysis conducted to evaluate the prediction accuracy and F_{mac} of these competing algorithms in the context of KNN, SVM, and DT(decision tree) classifiers.

From Tables 3–9 and Fig. 8, we can observe:

- UHGSFS *vs.* Alpha-investing: UHGSFS outperforms Alpha-investing in prediction accuracy and F_{mac} with KNN, SVM, and DT classifiers. Alpha-investing generally selects fewer features than UHGSFS, especially for large datasets like COIL20, USPS, and Isolet. It selects only a few features for some datasets, resulting in poor performance. This suggests Alpha-investing is less suitable for diverse datasets. In contrast, UHGSFS adapts feature selection based on dataset characteristics, ensuring better performance and stability.
- UHGSFS *vs.* SAOLA: The Nemenyi test shows UHGSFS significantly outperforms SAOLA in ACC-KNN and ACC-SVM. However, SAOLA selects fewer features overall. SAOLA is limited to handling only feature pair relationships and struggles with mixed types of streaming feature.

[2] Publicly available at https://github.com/kuiy/LOFS, https://github.com/doodzhou/OSFS, and https://github.com/XuyangAbert/OUFSDFC

Table 3
Predictive accuracy using KNN as the classifier.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	0.7400±0.0341	0.5900±0.0277	0.6900±0.0306	**0.8150±0.0284**	0.7550±0.0185	0.6200±0.0354	0.5550±0.0309	0.6750±0.0240	0.8050±0.0569	0.7200±0.0234	0.8095±0.03154
ALLAML	0.8429±0.0305	0.9286±0.0178	0.9143±0.0207	0.9286±0.0205	0.8714±0.0233	0.8571±0.0395	0.9286±0.0359	0.9000±0.0193	**0.9429±0.0458**	0.9028±0.0165	0.9162±0.0377
COIL20	0.9611±0.0043	0.6056±0.0569	0.7729±0.0011	**0.9854±0.0002**	0.9597±0.0037	0.9799±0.0034	0.4583±0.0040	0.5667±0.0302	0.9576±0.0838	0.8625±0.0242	0.8472±0.0181
Colon	0.4833±0.0680	0.7667±0.0270	0.7833±0.0589	0.8000±0.0353	0.7500±0.0448	0.7333±0.0458	**0.8667±0.0299**	0.8167±0.0312	0.8000±0.0596	0.7756±0.0117	0.7910±0.0273
GLIOMA	0.6200±0.0319	0.6000±0.0517	0.6200±0.0470	**0.7800±0.0340**	0.7400±0.0721	0.4200±0.0838	0.4000±0.0499	0.7400±0.0503	0.6400±0.0344	0.6799±0.0240	0.7200±0.0596
Isolet	0.7763±0.0075	0.2737±0.0252	0.4032±0.0143	0.3615±0.0827	0.7660±0.0121	**0.7949±0.0059**	0.1096±0.0145	0.6532±0.0081	0.5474±0.0604	0.6045±0.1040	0.6551±0.0475
Lymphoma	0.5263±0.0350	0.6737±0.0309	0.5684±0.0348	0.7263±0.0388	0.7789±0.0381	0.8947±0.0344	0.5474±0.0508	0.7158±0.0294	**0.9158±0.0272**	0.8957±0.0291	0.8453±0.0402
Nci-9	0.1333±0.0372	0.1667±0.0604	0.1667±0.0299	0.2667±0.0407	0.3667±0.0506	0.4000±0.0725	0.2000±0.0545	0.1167±0.0649	**0.5333±0.0407**	0.2666±0.0834	0.4167±0.0512
Orlraws10P	0.5900±0.0343	0.5600±0.0720	0.5200±0.0391	0.6000±0.0355	0.9000±0.0403	**0.9300±0.0195**	0.6700±0.0477	0.6500±0.1163	0.7800±0.0353	0.9000±0.0126	0.9300±0.0250
Pixraw10P	0.8600±0.0414	0.8100±0.0435	0.6600±0.0468	**0.9700±0.0246**	0.9200±0.0388	0.9500±0.0272	0.6300±0.0377	0.8200±0.0713	0.9400±0.0414	0.9400±0.0054	0.9300±0.0210
SMK_CAN_187	0.6108±0.0204	0.6541±0.0175	0.6054±0.0314	0.5676±0.0315	0.6595±0.0282	0.5027±0.0163	0.6162±0.0276	0.6541±0.0196	0.6162±0.0713	0.6362±0.0206	**0.6738±0.0188**
USPS	**0.9583±0.0010**	0.5123±0.0114	0.3156±0.0050	0.7495±0.0117	0.9545±0.0010	0.8824±0.0186	0.9568±0.0077	0.9491±0.0004	0.8817±0.0206	0.5988±0.1442	0.8955±0.0589
WarpPIE10P	0.8810±0.0206	0.5905±0.0355	0.7524±0.0423	0.9143±0.0169	0.8857±0.0297	**0.9714±0.0151**	0.8286±0.0171	0.6286±0.0708	0.9429±0.0827	0.8571±0.0181	0.8095±0.0529
AVG	0.691	0.5948	0.5979	0.7281	**0.7929**	0.7643	0.5975	0.6835	0.7925	0.7415	0.7877
AVG.RANK	6.9615	8.3846	8.4615	4.5769	4.5385	5.3462	7.5000	6.6154	**3.9615**	5.5385	4.1154

Table 4
Predictive Accuracy Using SVM as the classifier.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	0.7000±0.0273	0.5900±0.0403	0.6800±0.0276	0.7600±0.0447	0.7900±0.0258	0.5750±0.0250	0.5750±0.0316	0.7050±0.0260	0.7350±0.0241	0.6500±0.0166	**0.8857±0.0242**
ALLAML	0.8857±0.0376	0.9286±0.0225	0.9286±0.0284	0.9286±0.0213	0.8571±0.0376	0.8286±0.0459	0.8714±0.0331	0.9286±0.0184	0.9571±0.0363	0.9438±0.0138	**0.9448±0.0371**
COIL20	0.9896±0.0019	0.4701±0.0393	0.6722±0.0016	0.9931±0.0028	0.9688±0.0053	0.9958±0.0013	0.3632±0.0252	0.4951±0.0292	0.7688±0.0050	0.9340±0.0149	0.9125±0.0289
Colon	0.7333±0.0453	0.7167±0.0474	**0.7833±0.0609**	0.7333±0.0395	0.7333±0.0306	0.7333±0.0326	0.7167±0.0609	0.7667±0.0360	0.7500±0.0225	0.7448±0.0222	0.7756±0.0321
GLIOMA	0.5200±0.0520	0.5400±0.0610	0.6000±0.0520	**0.7600±0.0796**	0.6200±0.0558	0.4200±0.0760	0.3600±0.0350	0.6600±0.0626	0.6200±0.0403	0.6400±0.0140	0.6000±0.0384
Isolet	0.8750±0.0070	0.3494±0.0174	0.4878±0.0113	0.4045±0.0933	0.8263±0.0087	**0.8885±0.0045**	0.1179±0.0131	0.8109±0.0090	0.6327±0.0174	0.6660±0.1134	0.7417±0.0410
Lymphoma	0.4737±0.0433	0.6211±0.0335	0.6105±0.0426	0.7053±0.0419	0.8000±0.0325	**0.8632±0.0337**	0.5474±0.0385	0.7053±0.0482	0.6158±0.0610	0.7926±0.0081	0.7621±0.0581
Nci-9	0.1000±0.0446	0.1833±0.0828	0.1167±0.0458	0.2667±0.0657	**0.3833±0.0534**	0.3333±0.0682	0.2333±0.0561	0.0667±0.0454	0.3667±0.0335	0.2500±0.0194	0.2667±0.0271
Orlraws10P	0.7300±0.0269	0.6800±0.0772	0.4700±0.0488	0.5200±0.0490	0.8800±0.0288	0.9500±0.0306	0.6800±0.0790	0.7700±0.1243	0.8100±0.0433	**0.9700±0.0102**	0.9600±0.0250
Pixraw10P	0.9200±0.0337	0.9200±0.0273	0.7300±0.0478	0.9500±0.0316	0.9600±0.0267	0.8900±0.0434	0.7600±0.0698	0.8400±0.0868	0.9500±0.0273	0.9199±0.0092	**0.9700±0.0247**
SMK_CAN_187	0.6486±0.0178	0.6486±0.0249	0.5784±0.0262	0.5892±0.0407	0.6378±0.0439	0.6108±0.0210	**0.6595±0.0340**	0.6432±0.0253	0.6486±0.0419	0.6469±0.0112	0.6575±0.0101
USPS	**0.9548±0.0010**	0.5210±0.0076	0.2340±0.0022	0.7094±0.0205	0.9505±0.0009	0.8699±0.0157	0.9472±0.0004	0.9491±0.0006	0.8004±0.0267	0.4892±0.1699	0.9165±0.0596
WarpPIE10P	0.9571±0.0147	0.5524±0.0478	0.7000±0.0315	0.9571±0.0181	0.9333±0.0098	**0.9762±0.0082**	0.9048±0.0137	0.6714±0.0536	0.6524±0.0316	0.9047±0.0338	0.8857±0.0514
AVG	0.6910	0.5948	0.5979	0.7281	**0.7929**	0.7643	0.5975	0.6835	0.7925	0.7415	0.7877
AVG.RANK	6.9615	8.3846	8.4615	4.5769	4.5385	5.3462	7.5000	6.6154	**3.9615**	5.5385	4.1154

Table 5
Predictive Accuracy Using DT as the Classifier.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	0.6900±0.0315	0.5150±0.0292	0.6600±0.0354	**0.7350±0.0208**	0.7150±0.0348	0.6300±0.0189	0.5250±0.0474	0.6400±0.0322	**0.735±0.0935**	0.6800±0.0394	0.7095±0.0535
ALLAML	0.7571±0.0530	0.8857±0.0386	0.9286±0.0250	0.9286±0.0302	0.8714±0.0374	0.8571±0.0330	0.9000±0.0225	0.8429±0.0162	**0.9571±0.0306**	0.8733±0.0458	0.8610±0.0334
COIL20	0.8931±0.0077	0.5993±0.0496	0.7340±0.0022	**0.8993±0.0076**	0.8736±0.0106	0.8799±0.0118	0.4299±0.0284	0.5375±0.0359	0.7688±0.0450	0.7493±0.0268	0.7576±0.0390
Colon	0.5833±0.0331	0.7500±0.0425	0.7167±0.0643	0.7500±0.0407	0.5667±0.0625	0.6667±0.0432	**0.7833±0.0358**	0.7667±0.0312	0.7500±0.0312	0.7282±0.0329	0.6090±0.0483
GLIOMA	0.5400±0.0760	0.4000±0.0535	0.5600±0.0649	0.5800±0.0680	0.5200±0.0640	0.4600±0.0914	0.4200±0.0366	0.5600±0.0893	**0.6200±0.0829**	0.4599±0.0616	0.5800±0.0515
Isolet	**0.6763±0.0092**	0.2699±0.0184	0.3981±0.0101	0.3558±0.0731	0.6628±0.0080	0.6724±0.0188	0.1115±0.0106	0.6263±0.0131	0.6327±0.0320	0.4981±0.0935	0.5494±0.0657
Lymphoma	0.3684±0.0254	0.5158±0.0312	0.5684±0.0435	0.5684±0.0442	0.5053±0.0609	0.5684±0.0634	0.4632±0.0513	0.5895±0.0314	0.6158±0.0289	0.5647±0.0306	**0.6579±0.0458**
Nci-9	0.0500±0.0353	0.1667±0.0450	0.1167±0.0443	0.2833±0.0840	0.2833±0.0562	0.2667±0.0312	0.1833±0.0470	0.1000±0.0558	**0.3667±0.0326**	0.3166±0.0494	0.3500±0.0359
Orlraws10P	0.7300±0.0327	0.6700±0.0594	0.5000±0.0615	0.5400±0.0413	0.7400±0.0401	0.5700±0.0506	0.6600±0.0699	0.7600±0.1061	**0.8100±0.0413**	0.7300±0.0301	**0.8100±0.0291**
Pixraw10P	0.9500±0.0326	0.9100±0.0288	0.6800±0.0698	**0.9600±0.0297**	0.9300±0.0223	0.5800±0.0829	0.7800±0.0668	0.8500±0.0724	0.9500±0.0332	0.9199±0.0241	0.9500±0.0320
SMK_CAN_187	0.5514±0.0339	0.5946±0.0289	0.5730±0.0357	0.5568±0.0332	0.5892±0.0295	0.5784±0.0313	0.6324±0.0304	0.5784±0.0206	0.6486±0.0034	0.5513±0.0315	**0.6575±0.0308**
USPS	**0.8785±0.0041**	0.5371±0.0034	0.3147±0.0043	0.7483±0.0076	0.8705±0.0030	0.8266±0.0146	0.8685±0.0019	0.8635±0.0030	0.8004±0.1232	0.7003±0.1232	0.8345±0.0707
WarpPIE10P	0.7429±0.0266	0.5762±0.0270	0.6714±0.0261	0.7619±0.0300	**0.7857±0.0314**	0.6952±0.0492	0.6571±0.0396	0.5190±0.0639	0.6524±0.0270	0.6238±0.0281	0.7048±0.0383
AVG	0.6390	0.5678	0.5625	0.6588	0.6773	0.6297	0.5631	0.6429	0.6923	0.6476	**0.6939**
AVG.RANK	6.1154	8.0000	7.4615	4.6154	5.0385	6.7308	7.2308	6.5385	3.1538	7.0385	4.0769

Table 6
F_{mac} Using KNN as the classifier.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	0.7386±0.0384	0.5800±0.0273	0.6867±0.0298	**0.8141±0.0302**	0.7543±0.0180	0.6078±0.0379	0.5331±0.0355	0.6680±0.0239	0.7995±0.0397	0.7154±0.0257	0.8027±0.0376
ALLAML	0.7600±0.0377	0.8986±0.0244	0.8951±0.0299	0.9132±0.0247	0.8303±0.0320	0.8367±0.0447	0.9228±0.0489	0.8794±0.0331	**0.9379±0.0702**	0.8852±0.0217	0.9078±0.0449
COIL20	0.9593±0.0052	0.5869±0.0598	0.7628±0.0007	**0.9849±0.0035**	0.9591±0.0037	0.9815±0.0036	0.4290±0.0301	0.5508±0.0496	0.9567±0.0497	0.8522±0.0255	0.8387±0.0207
Colon	0.4134±0.0556	0.7252±0.0387	0.7543±0.0646	0.7656±0.041	0.7315±0.0386	0.7177±0.0461	**0.8512±0.0361**	0.7728±0.0397	0.7769±0.0246	0.7081±0.0128	0.7334±0.0441
GLIOMA	0.5228±0.0559	0.4985±0.0614	0.5614±0.0550	**0.7222±0.0459**	0.6993±0.0665	0.3829±0.0759	0.3326±0.0431	0.7074±0.0482	0.5871±0.0557	0.5817±0.0336	0.6798±0.0826
Isolet	0.7775±0.0065	0.2642±0.0266	0.3976±0.0140	0.3551±0.0808	0.7585±0.0125	**0.7894±0.0056**	0.1061±0.0141	0.6477±0.0099	0.5442±0.0420	0.5973±0.1059	0.6509±0.0581
Lymphoma	0.3306±0.0456	0.4522±0.0554	0.3255±0.0350	0.5567±0.0586	0.6418±0.0702	**0.8380±0.0573**	0.2928±0.0788	0.5774±0.0497	0.8362±0.0184	0.7707±0.0487	0.6733±0.0476
Nci-9	0.1110±0.0246	0.1426±0.0534	0.1787±0.0311	0.2930±0.0386	0.4022±0.0510	0.4090±0.0629	0.2063±0.0523	0.1049±0.0557	**0.4562±0.0188**	0.1992±0.0767	0.3575±0.0461
Orlraws10P	0.5884±0.0311	0.5205±0.0827	0.4622±0.0272	0.5738±0.0488	0.8778±0.0378	**0.9353±0.0255**	0.6083±0.0415	0.6435±0.1183	0.7523±0.0011	0.8853±0.0177	0.9153±0.0365
Pixraw10P	0.8237±0.0501	0.8166±0.0420	0.6141±0.0542	0.9335±0.0301	0.9220±0.0471	**0.9597±0.0264**	0.5952±0.0421	0.7811±0.0724	0.9198±0.0317	0.9306±0.0072	0.9207±0.0244
SMK_CAN_187	0.6053±0.0211	0.6459±0.0184	0.5969±0.0308	0.5629±0.0317	0.6535±0.0296	0.4435±0.0188	0.6101±0.0292	0.6464±0.0202	0.6100±0.0586	0.6191±0.0214	**0.6582±0.0181**
USPS	**0.9542±0.0011**	0.4506±0.0112	0.2029±0.0063	0.7152±0.0145	0.7152±0.0011	0.8705±0.0200	0.9526±0.0042	0.9437±0.0005	0.8687±0.0011	0.5535±0.1569	0.8180±0.0612
WarpPIE10P	0.8495±0.0234	0.5686±0.0412	0.6921±0.0448	0.9089±0.0180	0.8681±0.0293	**0.9724±0.0180**	0.8350±0.0212	0.6310±0.0724	0.9459±0.0200	0.8497±0.0186	0.8027±0.0541
AVG	0.6488	0.5500	0.5485	0.6999	0.7549	0.7496	0.5596	0.6580	**0.7686**	0.7037	0.7507
AVG.RANK	6.8462	8.5385	8.3846	4.8077	4.6538	4.6923	7.4615	6.4615	3.9231	5.8462	4.3846

Table 7
F_{mac} Using SVM as the classifier.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	0.6978±0.0296	0.5799±0.0411	0.6762±0.0261	0.7582±0.0470	0.7873±0.0253	0.5665±0.0320	0.4909±0.0416	0.6992±0.0298	0.6946±0.0291	0.6426±0.0149	**0.8829±0.0345**
ALLAML	0.8025±0.0465	0.8962±0.0270	0.9101±0.0359	0.9132±0.0227	0.8265±0.0425	0.8056±0.0530	0.8363±0.0440	0.9151±0.0307	0.8608±0.0345	0.9342±0.0180	**0.9373±0.0454**
COIL20	0.9888±0.0018	0.4291±0.0474	0.6610±0.0015	0.9936±0.0030	0.9681±0.0052	**0.9950±0.0016**	0.3241±0.0261	0.4372±0.0360	0.8813±0.0951	0.9282±0.0171	0.9096±0.0297
Colon	0.4970±0.0207	0.6796±0.0560	**0.7543±0.0767**	0.6942±0.0582	0.7034±0.0479	0.7152±0.0339	0.6661±0.0800	0.7170±0.0443	0.7179±0.0420	0.6317±0.0394	0.6829±0.0951
GLIOMA	0.4633±0.0743	0.4819±0.0616	0.5047±0.0557	**0.6606±0.0852**	0.5672±0.0641	0.3763±0.0739	0.2912±0.0308	0.6123±0.0684	0.5046±0.0171	0.4963±0.0158	0.4489±0.0451
Isolet	0.8727±0.0074	0.3417±0.0191	0.4855±0.0113	0.3915±0.0961	0.8224±0.0073	**0.8842±0.0038**	0.0983±0.0134	0.8077±0.0094	0.5119±0.0389	0.6556±0.1157	0.7357±0.0420
Lymphoma	0.3725±0.0577	0.4356±0.0559	0.4306±0.0538	0.5639±0.0630	0.6939±0.0698	**0.7782±0.0474**	0.3930±0.0586	0.5706±0.0829	0.4372±0.0244	0.5373±0.0125	0.5280±0.0632
Nci-9	0.0948±0.0370	0.1691±0.1096	0.1244±0.0402	0.3003±0.0610	**0.4113±0.0587**	0.4089±0.0556	0.2620±0.0606	0.0468±0.0411	0.3256±0.0402	0.1514±0.0109	0.1703±0.0215
Orlraws10P	0.7324±0.0277	0.6392±0.0769	0.4245±0.0536	0.4893±0.0460	0.8639±0.0430	0.9589±0.0286	0.6705±0.0802	0.7745±0.1256	0.6565±0.0743	**0.9686±0.0113**	0.9580±0.0276
Pixraw10P	0.9277±0.0419	0.9291±0.0327	0.6963±0.0654	0.9672±0.0369	0.9571±0.0260	0.8962±0.0389	0.7283±0.0660	0.8017±0.0922	0.9307±0.0556	0.9039±0.0123	**0.9680±0.0307**
SMK_CAN_187	0.6442±0.0175	0.6448±0.0244	0.5738±0.0249	0.5823±0.0527	0.6327±0.0432	0.6079±0.0214	**0.6559±0.0343**	0.6340±0.0257	0.6369±0.0370	0.6029±0.0107	0.6500±0.0105
USPS	**0.9505±0.0012**	0.4517±0.0054	0.1166±0.0030	0.6804±0.0231	0.6804±0.0012	0.8561±0.0163	0.9416±0.0006	0.9444±0.0008	0.4942±0.0425	0.4245±0.1944	0.9083±0.0841
WarpPIE10P	0.9495±0.0138	0.5331±0.0567	0.6664±0.0373	0.9533±0.0172	0.9159±0.0142	**0.9743±0.0106**	0.9044±0.0136	0.6674±0.0508	0.7757±0.0180	0.8997±0.0364	0.8829±0.0564
AVG	0.6918	0.5547	0.5403	0.6883	**0.7562**	0.7556	0.5587	0.6637	0.6483	0.6751	0.7433
AVG.RANK	5.7692	7	7.3077	4.5	**3.8846**	4.3846	7.1538	4.9231	6.0714	5.8462	4.2308

Table 8
F_{mac} Using DT as the classifier.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	0.6846±0.0396	0.501±0.0465	0.6586±0.0352	**0.7318±0.0580**	0.7091±0.0384	0.6216±0.0330	0.5134±0.0305	0.6343±0.0305	0.7290±0.0747	0.6772±0.0334	0.7184±0.0581
ALLAML	0.7302±0.0567	0.8684±0.0486	0.9195±0.0468	**0.9207±0.0187**	0.8479±0.0502	0.8367±0.0570	0.8816±0.0384	0.8152±0.0419	0.8506±0.0332	0.8608±0.0553	0.8189±0.0336
COIL20	0.8903±0.0308	0.5904±0.0798	0.7260±0.0105	**0.8965±0.0025**	0.8722±0.0063	0.8779±0.0026	0.4139±0.0368	0.5289±0.0209	0.8682±0.0204	0.7439±0.0284	0.7122±0.0400
Colon	0.4601±0.0346	0.6936±0.0657	0.6609±0.0697	0.7213±0.0602	0.5147±0.0531	0.6347±0.0436	**0.7452±0.0740**	0.7131±0.0632	0.7220±0.0395	0.6316±0.0393	0.6821±0.0363
GLIOMA	0.4579±0.0487	0.3227±0.0627	0.4211±0.0623	0.5133±0.0812	0.4552±0.0589	0.4090±0.0830	0.3386±0.0403	**0.5658±0.0753**	0.5220±0.0614	0.4276±0.0673	0.4901±0.0461
Isolet	**0.6790±0.0154**	0.5467±0.0312	0.6990±0.0124	0.7868±0.0893	0.7001±0.0056	0.6896±0.0050	0.6873±0.0241	0.5423±0.0074	0.6292±0.0340	0.5536±0.0949	0.6518±0.0747
Lymphoma	0.5063±0.0770	0.3152±0.0303	0.2662±0.0571	0.3750±0.0580	0.3125±0.0753	0.3579±0.0539	0.3146±0.0601	0.3946±0.0902	**0.5516±0.0643**	0.3894±0.0497	0.4626±0.0604
Nci-9	0.1315±0.0457	0.2994±0.0529	0.2872±0.0391	0.4065±0.0530	0.3910±0.0643	0.455±0.0639	0.2486±0.0730	0.3967±0.0503	**0.3656±0.0019**	0.3871±0.0455	0.3877±0.0340
Orlraws10P	0.0452±0.0320	0.2106±0.0835	0.1317±0.0432	0.3071±0.0410	0.2676±0.0330	0.2399±0.0395	0.1747±0.0589	0.0758±0.2039	0.7801±0.0320	0.2621±0.0400	0.3379±0.0291
Pixraw10P	0.7363±0.0156	0.6319±0.0167	0.4555±0.0551	0.5244±0.0258	0.7525±0.0380	0.5729±0.0204	0.6560±0.0759	0.7425±0.0306	0.9496±0.0563	0.702±0.0284	**0.784±0.0347**
SMK_CAN_187	**0.9553±0.0090**	0.9171±0.0030	0.6180±0.0348	0.9479±0.0437	0.9449±0.0563	0.5492±0.0384	0.7491±0.0483	0.8197±0.0984	**0.6412±0.0395**	0.9146±0.0359	0.9053±0.0332
USPS	0.5480±0.0138	0.5932±0.0061	0.5696±0.0028	0.5498±0.0345	0.5819±0.0019	0.5755±0.0258	0.6229±0.0010	0.5742±0.0017	**0.8896±0.0201**	0.5382±0.1367	0.5924±0.0761
WarpPIE10P	0.7292±0.0419	0.5603±0.0614	0.6471±0.0164	0.7151±0.0180	**0.785±0.0190**	0.6824±0.0201	0.6507±0.0303	0.4899±0.0640	0.7467±0.0902	0.6183±0.0292	0.7336±0.0390
AVG	0.6084	0.5343	0.5152	0.6385	0.6540	0.6105	0.5349	0.6027	**0.7112**	0.6103	0.6585
AVG.RANK	6.0000	7.8462	8.0000	4.4231	4.9615	6.5385	7.0769	6.6154	**2.4615**	7.2308	4.8462

Table 9
The mean number of selected features.

Data Set	Alpha-investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group-SAOLA	OFS-3WD	OGSFS-FI	OHSFS	OUFSDFC	UHGSFS
Arcene	9.2	26.4	5.2	37.2	31.4	8	3.4	6.2	16	61	43.6
ALLAML	11.6	30.8	4.4	3.6	12.6	1	11.8	20	134.6	120	105
COIL20	193.4	3.4	7	119.6	17.2	74.4	13.2	3.2	93.4	172.2	85.8
Colon	1.8	5.4	2.2	5.6	18.4	5.4	21.8	15	25.4	236.8	34
GLIOMA	6.2	15	3.4	8.6	21	2.4	2	16.2	134.6	144.2	51.4
Isolet	90.2	6.8	10.8	5.2	56.6	61.2	3.2	73.2	76	25.4	54.6
Lymphoma	9.8	21.4	4.6	7.4	15	715.2	6.4	32.2	412.8	241.6	38.2
Nci-9	5.4	17.6	3.6	7.2	24	215	26.8	10.2	176.4	274.4	76.4
Orlraws10P	9.8	5.6	3.2	2.6	13.2	279.6	25.2	4.8	17.2	776	115
Pixraw10P	10.4	7.6	4.2	166.2	7.4	287	20.6	2.6	269	96.4	141
SMK_CAN_187	14.8	11.2	5	15	31.6	1	26.4	28.2	68.6	96.6	128.8
USPS	166.2	4	2	9.2	106.8	12	64.2	65.2	43	19	46.2
WarpPIE10P	45	3.6	2	28.8	29.4	27	24.2	6.2	48.6	139.4	43.6
AVG	44.1	12.2	4.6	32	29.6	129.9	19.2	21.8	116.58	184.8	74.1
AVG. RANKS	5.8	4.3	2.1	4.8	6.6	5.9	4.7	5.2	8.8	9.4	8.5

- UHGSFS *vs.* Fast-OSFS: UHGSFS significantly outperforms Fast-OSFS in ACC-KNN and ACC-SVM. Fast-OSFS selects fewer features, which may lead to loss of important information and reduced prediction accuracy.
- UHGSFS *vs.* OFS-Density: UHGSFS and OFS-Density show comparable performance in prediction accuracy, with UHGSFS generally outperforming in most cases. OFS-Density selects more features for COIL20 and Pixraw10P datasets but fewer for others. It is computationally expensive for large datasets and may struggle with unevenly distributed data.
- UHGSFS *vs.* OFS-A3M: There is no significant difference between UHGSFS and OFS-A3M in prediction accuracy. However, UHGSFS performs better in most cases with KNN, SVM, and DT classifiers. OFS-A3M, like OFS-Density, is a neighborhood-based method with high time complexity and selects fewer features, making its performance sensitive to sample distribution.
- UHGSFS *vs.* OFS-3WD: UHGSFS significantly outperforms OFS-3WD in accuracy for SVM classifier. OFS-3WD selects fewer features and uses global dynamics and three-way decision-making for feature selection. However, it cannot handle heterogeneous streaming features.
- UHGSFS *vs.* Group-SAOLA: No significant difference is observed in prediction accuracy and F_{mac} between UHGSFS and Group-SAOLA. However, UHGSFS performs better in most cases. Group-SAOLA selects more features in some datasets but fewer in others. It is efficient on high-dimensional datasets but may lose important information by selecting too few features.
- UHGSFS *vs.* OGSFS-FI: Regarding accuracy (ACC) and F_{mac}, UHGSFS exhibits higher average prediction accuracy and lower average ranks than OGSFS-FI for KNN, SVM, and DT classifiers. Additionally, OGSFS-FI selects fewer features on average than UHGSFS. OGSFS-FI is an online group streaming feature selection method that considers the feature interactions between feature groups, but it uses different metrics for different feature types, so it cannot handle heterogeneous streaming features.
- UHGSFS *vs.* OHSFS: Both UHGSFS and OHSFS solve the problem of streaming feature selection with unknown feature types. Fig. 8 indicates no statistically significant difference between UHGSFS and OHSFS regarding ACC and F_{mac}. Regarding the average accuracy and average ranks, OHSFS performs a little better than UHGSFS in most cases. However, OHSFS is a supervised feature selection algorithm, while our new method is unsupervised. Regarding the average number of selected features, UHGSFS selects fewer features than OHSFS. OHSFS uses a greedy algorithm to screen features, which can achieve better results but sometimes makes the number of features too high. In terms of time complexity, UHGSFS is more complicated than OHSFS.

Table 10
Time complexity comparison. Here, m represents the number of features, n refers to the number of samples, $|FS_t|$ denotes the size of the selected feature subset at timestamp t, $|FC_t|$ denotes the size of clusters at timestamp t, and q is a constant value.

SFS Methods	Time Complexity
Alpha-Investing	$O(t * m * \|FS_t\|^2)$
SAOLA	$O(t * m * \|FS_t\|)$
Fast-OSFS	$O(\|FS_t\| * q^{\|FS_t\|})$
OFS-Density	$O(m^2 * n^2 * logn)$
OFS-A3M	$O(m^2 * n^2 * logn)$
Group-SAOLA	$O(t * m * \|FS_t\|)$
OFS-3WD	$O(\|FS_t\|^2)$
OGSFS-FI	$O(\|FS_t\|^3 + m)$
OHSFS	$O(m^2\|FS_t\|)$
OUFSDFC	$O(nm^2 + nlogn + \|FC_t\|\|FC_{t-1}\|)$
UHGSFS	$O(m^2 + nlogn + \|FC_t\|\|FC_{t-1}\|)$

- UHGSFS *vs.* OUFSDFC: Both UHGSFS and OUFSDFC are unsupervised online streaming feature selection algorithms. UHGSFS outperforms OUFSDFC in terms of average prediction accuracy (ACC) and F_{mac}, while selecting fewer features on average. OUFSDFC requires more time to identify feature relationships in continuous streaming data. Notably, OUFSDFC uses separate metrics for continuous and discrete datasets, limiting its ability to handle unknown feature types.

Table 10 presents a comparison of the time complexity between UHGSFS and ten other streaming feature selection (SFS) methods. From this comparison, it is evident that the time complexity of UHGSFS is significantly lower than that of OFS-Density and OFS-A3M, and is comparable to that of OUFSDFC. However, for the other seven methods, UHGSFS exhibits higher time complexity. As shown in Fig. 7, UHGSFS demonstrates shorter running times for discrete datasets (e.g., Nci-9) compared to OUFSDFC. In contrast, for continuous datasets (e.g., COIL20), OUFSDFC performs faster than UHGSFS. This discrepancy arises because UHGSFS requires more time to identify the relationship between two features when processing continuous streaming features.

In sum, UHGSFS performs excellently on average accuracy (ACC) and F_{mac} in cases of KNN, SVM, and DT classifiers. Even without label information, UHGSFS exhibits statistically better or equivalent performance compared to the supervised streaming feature selection methods. Furthermore, when compared to competing algorithms, UHGSFS indicates enhanced scalability regarding running time and the number of selected features. Importantly, UHGSFS can effectively handle heterogeneous streaming features without knowing the information about feature types in advance. According to Fig. 8, UHGSFS isn't definitively

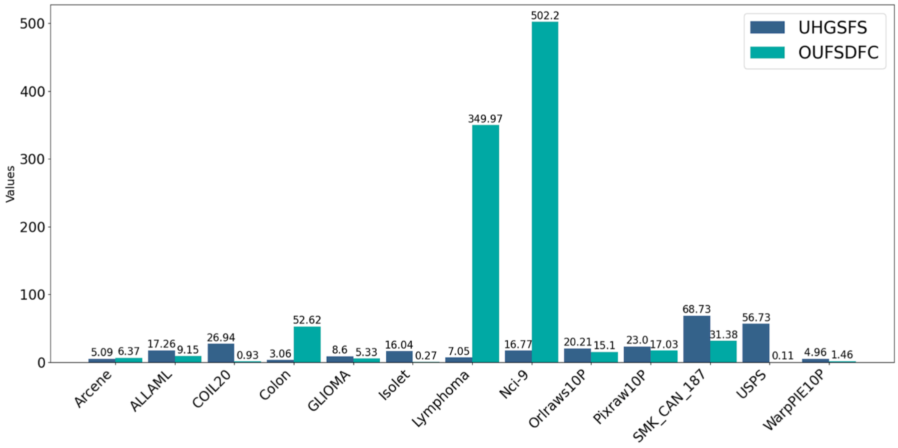
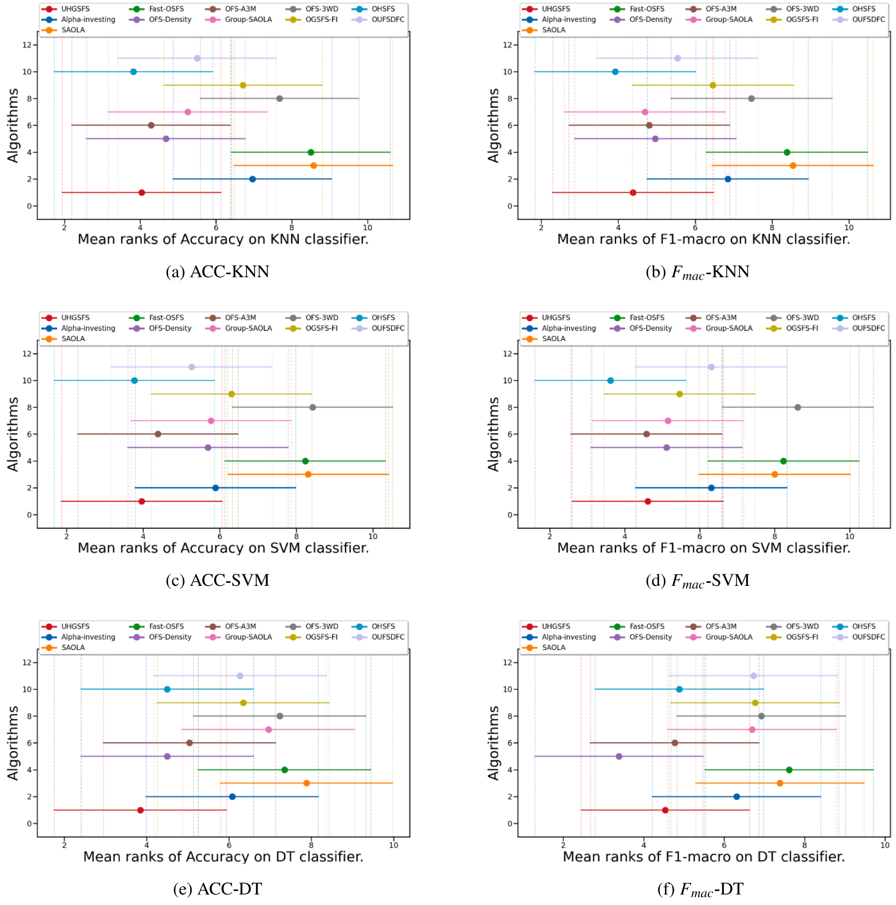


Fig. 7. Running time (seconds) of UHGSFS vs. OUFSDFC.



(a) ACC-KNN

(b) F_{mac}-KNN

(c) ACC-SVM

(d) F_{mac}-SVM

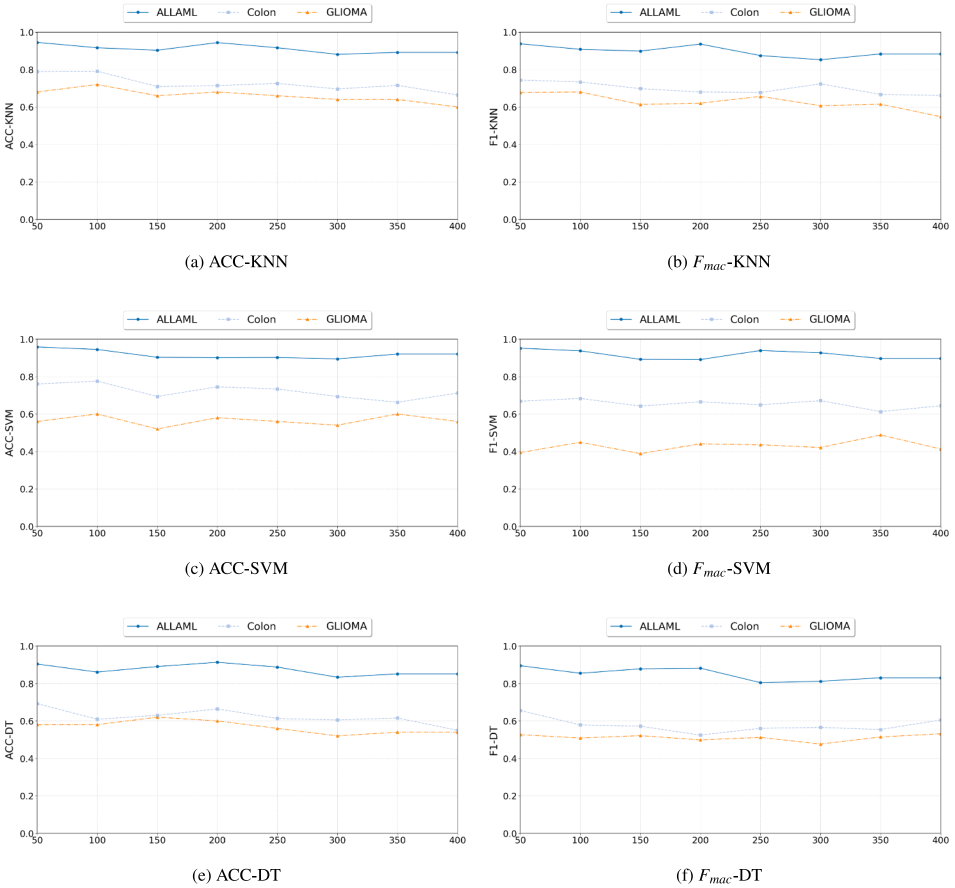
(e) ACC-DT

(f) F_{mac}-DT

Fig. 8. The statistical test graph.

Table 11
ACC of heterogeneous streaming feature.

Data Set	KNN-ACC	KNN-ACC-MIX	SVM-ACC	SVM-ACC-MIX	DT-ACC	DT-ACC-MIX
ALLAML	**0.9162**	0.8752	**0.9448**	0.9314	**0.8610**	0.7762
Colon	0.7910	**0.8051**	0.7756	**0.8077**	0.6090	**0.6115**
GILOMA	0.7200	**0.8200**	0.6000	**0.7200**	0.5800	**0.6400**
Orlraws10P	**0.9300**	0.9200	**0.9600**	0.9200	**0.8100**	0.7500
Pixraw10P	**0.9300**	0.8900	**0.9700**	0.9400	**0.9500**	0.8800
SMK_CAN_187	**0.6738**	0.6097	**0.6575**	0.5771	**0.6575**	0.5242
WarpPIE10P	**0.8095**	0.7238	**0.8857**	0.7333	**0.7048**	0.6571
USPS	**0.8955**	0.8714	**0.9165**	0.9021	**0.8345**	0.8328
AVG	**0.8244**	0.8063	**0.8277**	0.8042	**0.7389**	0.6913



(a) ACC-KNN

(b) F_{mac}-KNN

(c) ACC-SVM

(d) F_{mac}-SVM

(e) ACC-DT

(f) F_{mac}-DT

Fig. 9. Accuracy and F_{mac} curves of UHGSFS varying with different sizes of group.

Table 12
F_{mac} of Heterogeneous streaming feature.

Data Set	KNN-F_{mac}	KNN-F_{mac}-MIX	SVM-F_{mac}	SVM-F_{mac}-MIX	DT-F_{mac}	DT-F_{mac}-MIX
ALLAML	**0.9078**	0.8555	**0.9373**	0.9173	**0.8546**	0.7646
Colon	0.7334	**0.7534**	0.6829	**0.7413**	0.5784	**0.5812**
GILOMA	0.6798	**0.7421**	0.4489	**0.6146**	0.5088	**0.6255**
Orlraws10P	**0.9153**	0.9093	**0.9580**	0.9141	**0.7848**	0.7215
Pixraw10P	**0.9207**	0.8748	**0.9680**	0.9261	**0.9467**	0.8640
SMK_CAN_187	**0.6582**	0.5757	**0.6500**	0.5050	**0.5593**	0.5059
WarpPIE10P	**0.8027**	0.7150	**0.8829**	0.7274	**0.7103**	0.6497
USPS	0.8180	**0.8584**	**0.9083**	0.8922	**0.8180**	0.8153
AVG	**0.8025**	0.7751	**0.7897**	0.7637	**0.7061**	0.6732

better than all competing algorithms regarding ACC and F-mac on all three classifiers. However, practical applications are often unsupervised. Since our new method is unsupervised and can achieve competing performance with the other nine supervised feature selection algorithms, this indicates its significant practical value.

4.3. The effectiveness of UHGSFS in handling heterogeneous streaming features

Since the datasets in Table 2 are continuous or discrete feature datasets, there are no mixed datasets. To verify the effectiveness of our UHGSFS algorithm in handling heterogeneous streaming features, we take the eight continuous datasets in Table 2, randomly select 50 % features, and discretize these features into ten equal parts. Then, we conduct experiments on these two types of datasets (original and mixed). The experimental results on prediction accuracy and F_{mac} are shown in Tables 11 and 12, where KNN-ACC-MIX, SVM-ACC-MIX, and DT-ACC-MIX denote the prediction accuracy of UHGSFS on mixed streaming features.

From Tables 11 and 12, we can observe that for some datasets, the selected features on mixed version and the final classification ACC and F_{mac} do not perform as well as when using the original dataset. Meanwhile, on some other datasets, features selected on the mixed dataset outperform the original dataset. This fluctuation is slight, and the root cause is the resulting bias of the feature selection algorithm when running multiple times. Overall, our new method performs essentially equally well on both two kinds of datasets. These results demonstrate our method's capability when dealing with heterogeneous streaming features.

4.4. Analysis of streaming group size

In order to investigate the effect of different group sizes on the UHGSFS, we chose three representative datasets, ALLAML, Colon, and

GLIOMA, for parameter analysis. GLIOMA and ALLAML are continuous datasets, while Colon is a discrete dataset. We set the group size to increase from 50 to 400, and the interval between each increase was 50. Fig. 9 shows the accuracy and F_{mac} curves for KNN, SVM, and DT classifiers varying with different group sizes, respectively.

By observing the experimental results, we can find that ACC and F_{mac} performance on these three datasets presents little change regardless of different group sizes regarding KNN, SVM, and DT classifiers. This indicates that group size does not significantly affect the results of UHGSFS.

5. Discussion and conclusions

This paper addresses a novel and practical problem in online unsupervised streaming feature selection, where the generated streaming features are heterogeneous and have unknown feature types prior to learning. Although previous studies have proposed several unsupervised streaming feature selection methods, they are limited to homogeneous features and cannot effectively handle heterogeneous streaming features with unknown type information. Based on the techniques of MIC and streaming feature clustering, our new method can select good representative features from streaming groups. Compared with state-of-the-art supervised and unsupervised streaming feature selection methods, the performance of our new method is comparable to or even better than those competing methods. Besides, to verify the effectiveness of our new method in handling heterogeneous streaming features, we conduct experiments on randomized mixed data sets, and these results demonstrate the effectiveness of our new method.

There are four main advantages to our new method. 1) Compared with existing supervised online streaming feature selection approaches, our unsupervised method offers broader applicability in real-world scenarios where feature labels are unavailable. Since class label information is relatively sparse and difficult to obtain in real-world applications. 2) Compared with the existing unsupervised online streaming feature selection method, our new method does not need the feature type information before learning. In practice, it is unreasonable to assume that we can know the feature type for the next arriving streaming feature. 3) Our new method selects just one representative feature from each feature cluster, which makes the selected candidate features informative and of low redundancy. 4) Based on the strategy of natural search neighbors, our proposed new method does not need to specify any parameters for different datasets. In total, our new method is more generalizable and performs well.

Of course, our new approach has some shortcomings and limitations. 1) Although MIC can effectively measure the information between heterogeneous streaming features without prior knowledge of feature type, the time complexity of calculating the MIC value is higher than supervised information metrics. Since the current method calculates the MIC based on dynamic programming, we will consider calculating MIC based on the k-means method in the future, significantly reducing time complexity and the running time of UHGSFS. 2) Our new method considers the redundancy between features within the same feature cluster and selects just one representative feature from each cluster. However, it ignores the relationship of features between different feature clusters, such as the feature interaction between candidate streaming features. Thus, we will focus on this issue in our future work. 3) Our new method assumes that each feature cluster is a sphere. However, the shape of feature clusters may be irregular in practical applications. Therefore, our future work will consider the irregular feature cluster shape via different kernel functions.

Overall, the experimental results and comparative studies demonstrate that, without label information, the UHGSFS method statistically outperforms or performs equally well as state-of-the-art supervised streaming feature selection methods, confirming the effectiveness of our new approach in handling heterogeneous streaming features.

CRediT authorship contribution statement

Peng Zhou: Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization; **Qianzhen Chen:** Writing – original draft, Software, Methodology, Investigation, Formal analysis; **Lei Sang:** Funding acquisition, Formal analysis; **Shu Zhao:** Supervision, Formal analysis; **Xindong Wu:** Project administration, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under grants (62376001, 62206002, 62120106008), the Natural Science Foundation of Anhui Province of China under grants 2308085MF215.

References

[1] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman and Hall/CRC Press, 2007.

[2] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (6) (2018) 1–45.

[3] S. Solorio-Fernández, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, A review of unsupervised feature selection methods, Artif. Intell. Rev. 53 (2) (2020) 907–948.

[4] E. Hancer, B. Xue, M. Zhang, A survey on feature selection approaches for clustering, Artif. Intell. Rev. 53 (2020) 4519–4545.

[5] M. Wang, H. Li, D. Tao, K. Lu, X. Wu, Multimodal graph-based reranking for web image search, IEEE Trans. Image Process. 21 (11) (2012) 4649–4661.

[6] W. Ding, T.F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, X. Wu, Subkilometer crater discovery with boosting and transfer learning, ACM Trans. Intell. Syst. Technol. (TIST) 2 (4) (2011) 1–22.

[7] D. Lei, P. Liang, J. Hu, Y. Yuan, New online streaming feature selection based on neighborhood rough set for medical data, Symmetry 12 (10) (2020) 1635.

[8] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2012) 1178–1192.

[9] P. Zhou, S. Zhao, Y. Yan, X. Wu, Online scalable streaming feature selection via dynamic decision, ACM Trans. Knowl. Discov. Data (TKDD) 16 (5) (2022) 1–20.

[10] P. Zhou, X. Hu, P. Li, X. Wu, OFS-Density: a novel online streaming feature selection method, Pattern Recognit. 86 (2019) 48–61.

[11] D. Wu, Y. He, X. Luo, M. Zhou, A latent factor analysis-based approach to online sparse streaming feature selection, IEEE Trans. Syst. Man Cybern. Syst. 52 (11) (2021) 6744–6758.

[12] J. Li, X. Hu, J. Tang, H. Liu, Unsupervised streaming feature selection in social media, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1041–1050.

[13] N. Almusallam, Z. Tari, J. Chan, A. Fahad, A. Alabdulatif, M. Al-Naeem, Towards an unsupervised feature selection method for effective dynamic features, IEEE Access 9 (2021) 77149–77163.

[14] X. Yan, A. Homaifar, M. Sarkar, B. Lartey, K.D. Gupta, An online unsupervised streaming features selection through dynamic feature clustering, IEEE Trans. Artif. Intell. 4 (5) (2022) 1281–1292.

[15] W. Zheng, S. Chen, Z. Fu, J. Li, J. Yang, Streaming feature selection via graph diffusion, Inf. Sci. 618 (2022) 150–168.

[16] D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P.J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Detecting novel associations in large data sets, Science 334 (6062) (2011) 1518–1524.

[17] Q. Gu, Z. Li, J. Han, Generalized fisher score for feature selection, arXiv preprint arXiv:1202.3725 (2012).

[18] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (2014) 175–186.

[19] Y. Wu, P. Li, Y. Zou, Partial multi-label feature selection with feature noise, Pattern Recognit. 162 (2025) 111310.

[20] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. 18 (2005) 1–8.

[21] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 301–312.

[22] X. Zuo, W. Zhang, X. Wang, L. Dang, B. Qiao, Y. Wang, Unsupervised feature selection via maximum relevance and minimum global redundancy, Pattern Recognit. 164 (2025) 111483.

[23] Y. Wang, J. Wang, H. Liao, H. Chen, An efficient semi-supervised representatives feature selection algorithm based on information theory, Pattern Recognit. 61 (2017) 511–523.

[24] Z. Qiu, W. Zeng, D. Liao, N. Gui, A-SFS: Semi-supervised feature selection based on multi-task self-supervision, Knowledge-Based Syst. 252 (2022) 109449.

[25] X. Hu, P. Zhou, P. Li, J. Wang, X. Wu, A survey on online feature selection with streaming features, Front. Comput. Sci. 12 (2018) 479–493.

[26] S. Perkins, J. Theiler, Online feature selection using grafting, in: Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, pp. 592–599.

[27] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, I. Guyon, Streamwise feature selection, J. Mach. Learn. Res. 7 (9) (2006) 1861–1885.

[28] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Trans. Knowl. Discov. Data (TKDD) 11 (2) (2016) 1–39.

[29] A. Rafie, P. Moradi, A. Ghaderzadeh, A multi-objective online streaming multi-label feature selection using mutual information, Expert Syst. Appl. 216 (2023) 119428.

[30] D. You, R. Li, S. Liang, M. Sun, X. Ou, F. Yuan, L. Shen, X. Wu, Online causal feature selection for streaming features, IEEE Trans. Neural Netw. Learn. Syst. 34 (3) (2021) 1563–1577.

[31] S. Eskandari, M. Seifaddini, Online and offline streaming feature selection methods with bat algorithm for redundancy analysis, Pattern Recognit. 133 (2023) 109007.

[32] P. Zhou, Y. Zhang, Z. Ling, Y. Yan, S. Zhao, X. Wu, Online heterogeneous streaming feature selection without feature type information, IEEE Trans. Big Data 10 (4) (2024) 470–485.

[33] P. Zhou, X. Hu, P. Li, X. Wu, Online feature selection for high-dimensional class-imbalanced data, Knowledge-Based Syst. 136 (2017) 187–199.

[34] P. Zhou, X. Hu, P. Li, X. Wu, Online streaming feature selection using adapted neighborhood rough set, Inf. Sci. 481 (2019) 258–279.

[35] Y. Sun, P. Zhu, Online group streaming feature selection based on fuzzy neighborhood granular ball rough sets, Expert Syst. Appl. 249 (2024) 123778.

[36] J. Liu, Y. Lin, J. Du, H. Zhang, Z. Chen, J. Zhang, ASFS: a novel streaming feature selection for multi-label data based on neighborhood rough set, Appl. Intell. 53 (2) (2023) 1707–1724.

[37] X. Yan, M. Razeghi-Jahromi, A. Homaifar, B.A. Erol, A. Girma, E. Tunstel, A novel streaming data clustering algorithm based on fitness proportionate sharing, IEEE Access 7 (2019) 184985–185000.

[38] P. Zhou, N. Wang, S. Zhao, Online group streaming feature selection considering feature interaction, Knowledge-Based Syste. 226 (2021) 107157.

[39] D.G. Pereira, A. Afonso, F.M. Medeiros, Overview of Friedman's test and post-hoc analysis, Commun. Statistics-Simulation Comput. 44 (10) (2015) 2636–2653.