An Online Unsupervised Streaming Features Selection Through Dynamic Feature Clustering

Xuyang Yan, *Student, IEEE*, Abdollah Homaifar, *Member, IEEE*, Mrinmoy Sarkar, *Student, IEEE*, Benjamin Lartey and Kishor Datta Gupta, *Member, IEEE*

***Abstract*—Streaming feature selection (SFS) is emerging as a key research direction which addresses the non-stationary property of feature streams when the sample size is fixed. Most existing SFS techniques are supervised methods, and ignore the label scarcity. Real-world datasets are typically unlabeled and the labeling costs are expensive. Although some unsupervised SFS approaches are proposed, these approaches are either limited to the homogeneous feature types or require substantial computational complexity. To address these problems, we propose an online unsupervised feature selection framework using dynamic feature clustering in this paper. We derived a recursive density lower bound to estimate the density distribution of feature streams and developed a density-based dynamic clustering method to perform the online feature stream clustering for exploring feature redundancy. An unsupervised online feature relevance maximization and redundancy minimization strategy is introduced to extract a subset of important features with low redundancy from the feature stream. Experimental results on thirteen well-known benchmark datasets and comparison studies with seven state-of-the-art supervised SFS methods demonstrate that the proposed unsupervised method provides statistically comparable performance with the supervised SFS techniques while the label information is unknown.**

***Impact Statement*—The exponential growth in the volume of data generated from real-world applications initiates the explosion of research progress on streaming feature selection (SFS) problems. However, lack of label information, high computational complexity, and heterogeneous feature types, strongly challenged the existing streaming feature selection methods. The framework proposed in this paper addressed these issues through an unsupervised streaming feature clustering procedure as well as a simple yet effective online feature selection strategy. With comparable performance to the existing supervised approaches, the proposed framework overcomes the dependency of supervised approaches on label information while handling the heterogeneousness among feature types for unsupervised streaming feature selection methods with a relatively low time and memory complexity. As a generalized framework, the proposed technique provides an alternative perspective on the streaming feature selection problem using any available unsupervised density-based streaming clustering approaches.**

***Index Terms*—Feature stream clustering; Relevance maximization and redundancy minimization; Unsupervised online learning; Streaming feature selection.**

I. Introduction

TRADITIONAL feature selection (FS) approaches assume the entire feature space is known in advance and the characteristics of feature space are stationary [1], [2]. However, this assumption does not always hold in some real-world applications such as medical-related data analysis and image annotations. As one of the most representative examples, in medical applications, the feature information of patients grows as more inspection results become available and the diagnostic decisions vary accordingly. In this situation, features arrive sequentially over time and the concept of feature stream is initiated. Feature stream refers to the scenario where features arrive individually or as a sequence of chunks while the sample size is fixed [3]–[6].

Over the past decades, numerous supervised streaming feature selection (SFS) techniques have been proposed [1], [2], [7]–[13]. Among these techniques, a two-step supervised SFS procedure through feature redundancy minimization and relevance maximization is employed to obtain an appropriate feature subset. The feature redundancy minimization step eliminates features that provide similar information related to the class labels. The feature relevance maximization step ensures the quality of the selected feature subset by removing noisy and non-significant features using the label information. Several recent studies considered the interaction among features to enhance the performance of SFS methods [13], [14].

Despite substantial progress on supervised SFS techniques, one obvious limitation in prior studies is the dependency on label information. The label scarcity affects not only the relevance maximization but also the minimization of redundancy among features, which greatly limits the existing supervised SFS methods. In [1], an unsupervised SFS method, namely unsupervised streaming feature selection (USFS), was recently introduced for social media data analysis using the link information, which is only applicable to social media data and requires high computational complexity. In [15]–[18], several causal discovery-based unsupervised SFS methods were proposed to handle feature streams with extensive time and memory complexity.

In recent years, several unsupervised FS techniques using feature clustering analysis have shown competitive performance with supervised FS methods in traditional FS problems with the fixed feature space [19]–[22]. These approaches utilize the dependency among feature distributions to group

This work is supported by the Air Force Research Laboratory and the OSD under agreement number FA8750-15-2-0116. Also, this work is partially funded by the NASA University Leadership Initiative (ULI) and the National Science Foundation under grants number 80NSSC20M0161, and 2000320 respectively.

Xuyang Yan, Abdollah Homaifar, Mrinmoy Sarkar, and Benjamin are with Autonomous Control and Information Institute in North Carolina A& T State University, Greensboro, NC, 27411, USA. (e-mail: xyan@aggies.ncat.edu; homaifar@ncat.edu; msarkar@aggies.ncat.edu; blartey@aggies.ncat.edu).

Kishor Datta Gupta is with the computer science department in Clark Altanta University, GA, 30314. USA. (e-mail: gkishordatta@ncat.edu).

This paragraph will include the Associate Editor who handled your paper.

features that are highly redundant with each other and obtain a feature subset with low redundancy by selecting a single feature from each cluster. It has been shown that feature clustering-based unsupervised FS methods achieved a competitive performance with supervised FS methods in static FS problems. In [23], the unsupervised feature selection for dynamic features (UFSSF) extended the k-means clustering to cluster the continuous feature stream only from the individual level. Nevertheless, the performance of k-means clustering is sensitive to noise and is not applicable to discrete features.

Motivated by the merits of feature clustering-based FS methods, an **O**nline **U**nsupervised streaming **F**eature **S**election framework through **D**ynamic density-based **F**eature **C**lustering, namely OUFSDFC, is proposed in this paper. Unlike existing SFS methods, OUFSDFC takes advantage of feature stream clustering analysis to continuously group highly redundant features from the feature streams and maintain a summary of feature clusters to perform SFS. Based on the maintained cluster summary, OUFSDFC introduces an unsupervised FS strategy to maximize the feature relevance while minimizing feature redundancy.

In summary, the contributions of this work are three-fold:

- Propose an unsupervised SFS framework using dynamic density-based feature stream clustering analysis. To the authors' best knowledge, this is the first work that utilizes the density-based stream feature clustering analysis to handle unsupervised SFS problems. The proposed framework explores the redundancy among the streaming features through a dynamic density-based clustering procedure and selects a subset of highly representative features with low redundancy using the clustering information.
- Develop a density-based feature stream clustering method to group both continuous and discrete feature streams. We derive a new recursive lower bound to estimate the density distribution of the feature stream using the Laplacian density kernel function, and a complete version of mathematical derivations is provided in the Appendix. Additionally, an unsupervised strategy is developed to maximize the feature relevance and minimize feature redundancy in SFS.
- Conduct extensive experiments and comparison studies with well-known state-of-the-art SFS methods. Experimental results and statistical analysis justified that OUFSDFC provides better or comparable performance without label information.

The remainder of this paper is organized as follows: Section II provides a review of the related works on the SFS methods. The details of the OUFSDFC framework and its computational complexity are discussed in Section III. Section IV presents the experimental results and comparison study between OUFSDFC and the state-of-the-art methods. Finally, concluding remarks and future works are outlined in Section V.

II. Related Works

In this section, we reviewed the existing studies on the supervised and unsupervised SFS approaches. Besides, traditional redundancy-based FS methods and a summary of the existing density-based data stream clustering methods are discussed briefly.

A. Supervised Streaming Feature Selection

Most existing SFS methods are supervised and these approaches usually fall into two major groups [13]: (i) individual-level SFS; and (ii) group-level SFS. For individual-level SFS methods, features are assumed to arrive as a sequence of single independent variables. Grafting [8] was the first individual-level SFS method that utilized a regularized framework to integrate the SFS problem as a sub-task of the predictive modeling from the data. Information-investing and alpha-investing were employed as two penalized likelihood ratios in the stream-wise regression for SFS [24]. A scoring measure was introduced to evaluate the importance of an incoming feature with respect to a base model in supervised SFS techniques [25]. Scalable and Accurate Online feature selection Approach (SAOLA) [26] maintained a parsimonious model over time and conducted the online pairwise comparison for SFS. An online streaming feature selection (OSFS) framework and its variation, namely Fast OSFS [3] were proposed based on the redundancy minimization and relevance maximization procedure. Several recent SFS methods, including OSFSMI [11], OFS-A3M [27], OS-NRRSARA-SA [28], K-OFSD [29], and OFS-density [2], were developed using fuzzy rough set theory.

Group-level supervised SFS methods address the situation where feature streams arrive as a sequence of groups over time. Group-SAOLA extended SAOLA to handle feature streams that arrive as groups. In [10], the authors utilized the information theory to explore the group structure for SFS. An efficient group-level SFS framework, namely Online Group Feature Selection (OGFS), was developed using a hybrid of intra-group and inter-group feature selection stages in [12]. In [13], [14], the interaction among features is considered to improve the efficacy of group-level supervised SFS methods. In summary, both the individual-level and group-level supervised SFS methods require the label information as a prior and the label scarcity is rarely considered in the literature.

B. Unsupervised Streaming Feature Selection

To overcome the limitation of supervised SFS methods described above, several unsupervised SFS approaches were developed in [1], [15]–[18], [23]. In [1], the authors proposed an unsupervised streaming feature selection method to handle social media data by exploring the link information. It is customized for social media data analysis and requires high computational complexity. In [23], the k-means based feature clustering procedure is extended to perform SFS on continuous feature streams only. However, the performance of k-means clustering is sensitive to noises, and it can not handle discrete features. In [15]–[17], several causal discovery-based unsupervised SFS approaches were proposed to handle feature streams from the individual level while these approaches showed poor performance on continuous datasets. To address this issue, an unsupervised individual-level SFS method, namely partial rank correlation-based streaming feature causal discovery

(PRCDSF) [18], was developed to explore the causal relationship in the continuous feature streams. Nevertheless, the causal discovery procedure does require high time complexity.

C. Redundancy-based feature selection methods

Several well-known traditional supervised redundancy-based FS methods, such as correlation-based feature selection (CFS) [30], maximum relevance and minimum redundancy (MRMR) [31], and Relief [32], utilized the label information to filter highly redundant features in the static feature space. Feature clustering-based FS methods have been recently investigated to address the redundancy among features in an unsupervised manner [19]–[22], [33]. In [19], the authors first utilized the k-means clustering to group features that are highly redundant with each other and obtain a feature subset with low redundancy by selecting the mean of each feature cluster. Later, a feature clustering-based unsupervised FS method was introduced by adopting a rival penalization-controlled competitive learning framework into the feature clustering procedure [20]. Then, similar to [19], the mean of each feature cluster is selected to obtain the final feature subset. Instead of partitioning the feature space directly, in [33], a graph representation of the static feature space is extracted to obtain a set of non-redundant features, and then the clustering of non-redundant features is performed to select features. Another supervised filter-based feature selection method was introduced by combining the graph-based feature clustering with ant colony optimization to reduce the redundancy among features [34].

In [21], a density-based clustering method, namely efficient unsupervised feature selection through feature clustering (EUFSFC), was extended to explore the dependency among features to reduce the feature redundancy for homogeneous feature types in the static feature space. The authors presented two different feature clustering schemes to handle the continuous and discrete features separately. As an extension of EUFSFC, a supervised feature selection method through density-based feature clustering, namely (SFSDFC) [22], was introduced to handle heterogeneous feature types simultaneously in the static space. Unlike [21], a systematic feature clustering procedure was employed for both the continuous and discrete features. Besides, the label information is utilized to refine the quality of the selected feature subset. Overall, these feature clustering-based FS methods are only applicable to handle the static feature space, and none of them address non-stationary feature streams.

Several approaches were proposed to handle the feature redundancy by mining the graph structure in high-dimensional data [35]–[37]. A rank-constrained spectral clustering method with flexible embedding is proposed in [35] to filter irrelevant and noisy features by learning an intrinsic low-dimensional projected feature representation for high-dimensional data. In [36], the authors constructed a dynamic affinity graph to perform the spectral clustering on high-dimensional data and obtained a low-dimensional feature representation with low redundancy. Although these approaches demonstrated promising performance in addressing redundancy among features, they are not directly suitable to handle streaming features.

TABLE I
TABLE OF NOTATIONS .

Notations	Definition
m	the size of the feature chunk G_t
n	the number of samples
t	the time index
G_t	a chunk of features arrives at time t
F_t	the feature stream up to time t
T	the total number of feature chunks in F_t
FS_t	the selected feature subset from F_t
FC_t	the extracted feature clusters from F_t
CS_t	the extracted feature cluster summary from F_t
D_t	the density values of feature clusters from F_t
$f^i_{G_t}$	the i^{th} feature in the current feature chunk at t
$D_{f^i_{G_t}}$	the density value of i^{th} feature in G_t
FC^k_t	the k^{th} feature cluster from F_t
D^k_t	the density value of the k^{th} feature cluster from F_t
FC_0	a set of feature clusters in F_{t-1}
FC^k_0	the center of k^{th} cluster in F_{t-1}
D^k_0	the density value of the k^{th} historical feature cluster
$f^l_{k,0}$	the l^{th} sample in the cluster k at $t-1$
f^p_t	the p^{th} feature in F_t
$\|FC^k_0\|$	the number of features in FC^k_0
$\|FC_0\|$	the number of feature clusters at $t-1$
$\|FC_t\|$	the number of feature clusters at t
d_{il}	the distance from $f^i_{G_t}$ to $f^l_{k,0}$
d_{lk}	the distance from $f^l_{k,0}$ to FC^k_0
d_{ik}	the distance from $f^i_{G_t}$ to FC^k_0
d_{ij}	the distance from $f^i_{G_t}$ to another feature $f^j_{G_t}$ in G_t
β_t	the variance of F_t
β_{C_t}	the variance of G_t
$\hat{D}_{f^i_{G_t}}$	the estimated lower bound density value of $f^i_{G_t}$

D. Density-based data stream clustering

Density-based data stream clustering approaches are widely used to mine information from streaming data in the presence of clusters with an arbitrary shape, overlap clusters, and noises. In [38], a comprehensive review of different density-based data stream clustering approaches such as D-Stream [39], DenStream [40], DBStreams [41], and HDDStreams [42], etc. was provided. The density peak clustering (DPC) [43] was proposed as a novel fast density-based clustering approach by identifying cluster centers as local maximum density peaks. Several recent extensions of DPC approaches, including DPC-KNN [44], DPC-DBFN [45], and DPC-DLP [46], were proposed to handle the limitations of the original DPC method such as clusters with uneven densities, high time complexities, and parameter tuning. Motivated by the success of the DPC methods, the EDMStream [47] method was developed to handle the data stream through the exploration of density mountains in the data stream.

III. PROPOSED METHODOLOGY

In this section, we describe the basic notations, assumptions, and definitions first. The details of the proposed framework are then discussed. Besides, the time and space complexities of OUFSDFC are analyzed.

A. Notations, assumptions, and definitions

Let F_t and G_t be a feature stream and a chunk of features arrive at time t such that $F_t = \bigcup_{t=1}^{T} G_t$ where T refers to the total number of feature chunks. G_t consists of m features and each feature is denoted as: $f^i_{G_t}$, $i = 1, ..., m$The notation FS_t refers to the selected feature subset from the feature stream at time t. Also, let FC_t and CS_t be the extracted feature cluster centers and the cluster summary from the feature stream F_t until t such that $CS_t = \{FC_t, D_t\}$ where $FC_t = \bigcup_{k=1}^{|FC_t|} FC^k_t$ and $D_t = \bigcup_{k=1}^{|FC_t|} D^k_t$. The notations FC^k_t and D^k_t refer to the k^{th} feature cluster and its density value, respectively. In addition to these basic notations, a list of mathematical notations related to the derivations of the recursive lower bound of the Laplacian density function is provided in Table I. According to these notations, the objective of this paper is to obtain a good feature subset FS_t from F_t with the following assumptions:

- The numbers of samples and classes are fixed.
- Feature stream arrives as chunks and drifts of class distribution happen over time.
- The label information of the dataset is unknown.

Similar to the traditional feature clustering-based FS problems [19], [21], [48], the following definitions are considered in this paper.

Definition 1. *Relevant/Representative features: a set of features with the local maximum density values in each feature cluster.*

According to Definition 1, the centers of those feature clusters are usually considered as the most descriptive features and thus are obtained as a set of *Relevant/Representative features*.

Definition 2. *Redundant features: features are considered redundant with each other if they have high dependency/correlation.*

Using Definition 2, features with similar characteristics can be grouped together and the redundancy of features can be reduced by avoiding selecting features from the same cluster. With these two definitions, we propose a stream feature clustering-based SFS framework with an efficient online unsupervised relevance maximization and redundancy minimization strategy.

B. Online unsupervised streaming feature selection through dynamic feature clustering (OUFSDFC)

The proposed OUFSDFC method consists of two primary steps: (i) dynamic feature stream clustering; and (ii) online feature selection based on cluster summary. For dynamic feature stream clustering, we integrated the backbone of a state-of-the-art density-based data stream clustering method with an effective cluster merge or initialization procedure to perform the feature redundancy minimization. During the selection stage, the representativeness of features is used to achieve feature relevance maximization. An overview of the OUFSDFC procedure is provided in Figure 1.

1) Feature similarity evaluation: The Symmetric Uncertainty (SU) [49] and Maximal Information Compression Index (MICI) [19] are two widely used similarity measures for feature clustering analysis. SU reflects the relative information gain between two discrete feature distributions with respect to the sum of their entropy. MICI measures the linear dependency between two continuous feature distributions based on the covariance matrix. In OUFSDFC, MICI is used to measure the similarity for the continuous feature streams and SU is employed for the discrete feature streams.

2) Dynamic feature stream clustering: We extended a recent density-based data stream clustering method, namely dynamic fitness proportionate sharing clustering (DFPS-clustering) [50], to develop a feature stream clustering approach for grouping highly redundant features into clusters over time. Unlike [50], we derived a new recursive lower bound function using the Laplacian density formula and substituted it for the calculations of density distributions in the incoming feature chunks. The developed feature stream clustering method is named DFPSL-clustering. Considering the fact that the Laplacian mixture model is more robust to outliers than the Gaussian mixture model discussed in [51], the derived recursive Laplacian density lower bound effectively addresses the existence of noisy features in the stream. Let $f^i_{G_t}$ and d_{ij} be the i^{th} feature in G_t and its feature distance to the j^{th} feature in G_t, the density value of $f^i_{G_t}$ is denoted as $D_{f^i_{G_t}}$ and it is calculated using the derived recursive lower bound as follows.

$$D_{f^i_{G_t}} = \sum_{j=1}^{m} e^{(-\frac{d_{ij}}{\beta_t})^{\gamma_t}} + \sum_{k=1}^{|FC_0|} (e^{-\frac{d_{ik}}{\beta_t}})^{\gamma_t} \times D^k_0. \qquad (1)$$

Where β_t and γ_t refer to the normalization and stabilization parameters at time t, respectively. We use the same parameter estimation procedure from [50] to obtain the optimal values. d_{ik} is the distance between $f^i_{G_t}$ and the k^{th} historical feature cluster. The notations D^k_0 and $|FC_0|$ are the density value of the k^{th} historical clusters and the cardinality of the historical feature cluster set. The detailed mathematical derivations of equation 1 are provided in the Appendix. Based on the derived Laplacian recursive density lower bound, the DFPSL-clustering method effectively estimates the density values of an incoming feature chunk from both the historical and current feature chunks using only the cluster summary information. Consequently, instead of keeping all historical features from the feature stream, the derived recursive lower bound significantly reduces the memory space by maintaining the cluster summary only.

Algorithm 1 summarizes the details of the density-based dynamic feature clustering procedure. It consists of two phases: (i) offline clustering and (ii) online cluster merge. During the offline clustering procedure, as shown in Algorithm 1, the DFPSL-clustering approach starts with the density evaluation of features from the most recent feature chunk using equation 1. Then, it searches for all possible feature cluster centers with the local maximum density values from G_t. A cluster merge operation is performed on the obtained candidate feature clusters to extract a set of feature clusters FC_{G_t} from the offline

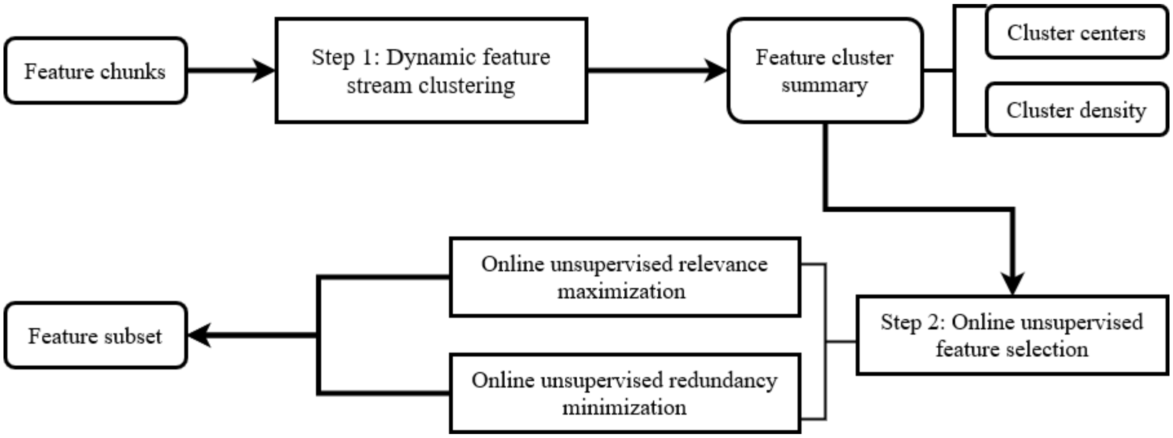

Feature chunks
Step 1: Dynamic feature stream clustering
Feature cluster summary
Cluster centers
Cluster density
Step 2: Online unsupervised feature selection
Online unsupervised relevance maximization
Online unsupervised redundancy minimization
Feature subset

Fig. 1. An overview of the OUFSDFC framework. (When a chunk of features arrive, OUFSDFC groups highly redundant features together using the dynamic feature stream clustering procedure. Then, an unsupervised online feature selection strategy is employed to reduce the feature redundancy and increase the feature relevance.)

clustering procedure. The online cluster merge step checks the possible merge between FC_{G_t} with the historical clusters FC_0 to extract the set of feature clusters in F_t. To handle discrete features, we employed the cluster merge procedure from [21] to reduce the redundancy between historical feature clusters and new feature clusters. Also, this merge procedure helps to capture the drift of class distributions through the merge of historical feature clusters and the initialization of novel feature clusters. During the cluster merge stage, the DFPSL-clustering method obtains a set of features from the boundary of historical and new feature clusters to decide whether a merge should happen or not. For example, a merge should happen when there is a significant drop in the density values among features that locate at the boundary between a new feature cluster and its nearest historical feature cluster. Otherwise, a new feature cluster is initialized as a new feature pattern that describes the drift of existing class distributions.

3) Online feature selection based on feature cluster summary: After the dynamic feature stream clustering procedure, a set of feature clusters are obtained from the feature stream and we employ an unsupervised online selection strategy to consider both relevance maximization and redundancy minimization. Due to the lack of data labels, the relevance of each feature is quantified based on its representativeness in each feature cluster. In density-based clustering methods, a cluster center usually has the local maximum density value and the highest descriptiveness of the cluster. Therefore, feature cluster centers are the most relevant features, and a set of *Relevant/Representative* features can be obtained using Definition 1. Meanwhile, according to Definition 2, features that belong to the same cluster are considered to be redundant to each other, and the redundancy minimization is guaranteed by only choosing the feature cluster center from each feature cluster. Following this strategy, a subset of important features with low redundancy can be obtained from the feature stream over time.

Algorithm 1 Dynamic feature stream clustering procedure

Input: G_t,F_t

Parameters: FC_{G_t}: the set of feature cluster centers extracted from G_t; FC_t: a set of feature cluster centers discovered from F_t until time t; D_t: the density values of all feature clusters in F_t; D_{G_t}; the density values of all features in G_t.

Output: CS_t

```
for $t = 1$ to $T$ do
    Perform the feature similarity evaluation for $G_t$ using MICI or SU
    $D_{G_t} = \varnothing$
    for $i = 1$ to $m$ do
        Calculate $D_{f^i_{G_t}}$ using equation 1
        $D_{G_t} = D_{G_t} S D_{f^i_{G_t}}$
    end for
        Rank all features of $G_t$ according to their density values and perform the search of possible feature clusters
        Merge highly overlapped feature clusters to obtain $FC_{G_t}$
        if $t == 1$then
            $CS_t = \{FC_{G_t}, D_{FC_{G_t}}\}$.
        else
            Merge $FC_{G_t}$ with historical feature clusters in $CS_t$ to obtain $FC_t$ and $D_t$ based on the existence of density valley in cluster boundaries
            $CS_t = \{FC_t, D_t\}$.
        end if
    end for
    Return $CS_t$
```

C. Complexity analysis

The time and space complexity of OUFSDFC are discussed using the following notations:

- n: total number of samples
- m: number of the features in the chunk
- $|FC_0|$: number of historical feature clusters in CS_t
- $|FC_{G_t}|$: number of feature clusters discovered in G_t

a) Time complexity.: During the dynamic feature stream clustering procedure, it takes $O(m^2)$ and $O(m^2n)$ distance calculations to obtain the feature distance matrix for continuous and discrete feature streams, respectively. For the search of possible feature clusters, the ranking of features imposes $O(nlogn)$ computations. The cluster merge stage between historical and new feature clusters requires $O(|FC_{G_t}||FC_0|)$ distance calculations. Therefore, the worst-case time complexity between continuous and discrete feature streams is $O(nm^2 + nlogn + |FC_{G_t}||FC_0|)$.

b) Space complexity.: The feature cluster summary requires $O(2|FC_0|)$ space to hold the extracted feature cluster centers and their density values. The processing of an incoming feature chunk G_t also takes $O(m)$ space complexity. Overall, the space complexity of the OUFSDFC method for both continuous and discrete feature streams becomes $O(m + 2|FC_0|)$.

IV. Experimental Studies and Discussions

In this section, experiments are conducted on well-known benchmark datasets, and comparison studies with the state-of-the-art supervised SFS methods are presented to prove the efficacy of the OUFSDFC framework. The time complexity comparison, parameter analysis, and execution time analysis are discussed as well.

A. Benchmark Dataset.

Similar to the state-of-the-art SFS techniques in [13], we selected thirteen benchmark datasets, including three discrete and ten continuous datasets, from the ASU feature selection repository [1] to validate the efficacy of the OUFSDFC framework. Table III summarizes the properties of those datasets in terms of sample size, feature size, class size, application domains, and types. As shown in Table III, all thirteen datasets have relatively large feature sizes that are suitable for the simulation of feature streams.

B. Baseline SFS methods

Due to the lack of unsupervised SFS methods that handle both the continuous and discrete feature streams, seven popular state-of-the-art supervised SFS methods, including Alpha investing, SAOLA, Fast OSFS, OFS-Density, OFS-A3M, Group SAOLA, and OGFSS-FI, are used to conduct the comparison study. The first five are individual-level supervised SFS methods and the last two are group-level supervised SFS methods. The MATLAB codes of Alpha investing, SAOLA, Fast OSFS, and Group SAOLA are available at link [2]. For the remaining methods, we obtained the MATLAB codes from link [3]. The python code of the OUFSDFC framework is provided in [4]. All experiments are conducted on an Intel Xeon (R) machine with 64GB RAM operating on Microsoft Windows 10.

C. Benchmark classifiers and evaluation metrics

The classification performance is used to show the efficacy of the OUFSDFC method versus seven supervised SFS methods. We used the decision tree and k-nearest-neighbors (KNN) classifiers to evaluate the quality of the final selected feature subset for different SFS methods. Two well-known evaluation metrics, including accuracy (Acc) and f-score (F_{mac}) [53], are employed as performance evaluation metrics. To account for the imbalance of class distributions, we use the macro-average of f-score and it is expressed as follows.

$$F_{mac} = \frac{1}{n_c} \sum_{i=1}^{n_c} F_i, \tag{2}$$

where F_i and n_c denote the *F-measure* for the i^{th} class and the number of classes, respectively.

D. Parameter settings

To simulate the feature stream, the feature chunk size is set to 250 for all datasets except for Lung. For the Lung dataset, the chunk size is set to 50. According to [13], the value of α is set to 0.01 for Fast OSFS, SAOLA, and Group-SAOLA. For Alpha-Investing, OFS-density, OFS-A3M, and OGFSS-FI methods, the default parameter settings from [2], [8], [27] are used to obtain the final selected feature subset. The number of the selected features from each SFS method is summarized in Table I. For the KNN classifier, the value of K is set to 5. We repeated each experiment ten times and performed ten-fold cross-validation on each dataset. The average values of the Acc and $f1_{mac}$ are reported in Tables II and IV, respectively. For each dataset, the best results among all compared methods are highlighted in bold-face. Two statistical analysis, including the Friedman rank test and Nemenyi post-hoc test [54], are conducted on the experimental results with a significance level of 0.05 The Nemenyi post-hoc test evaluates the pairwise statistical difference between two compared methods and it constructs a critical distance (CD) diagram [55], [56]. From the CD diagram, two compared methods are statistically comparable if they are connected by a dark solid line.

E. Results and discussions

a) OUFSDFC vs. Compared SFS methods on KNN: Table II summarizes the experimental results of OUFSDFC and the other seven supervised SFS methods using the KNN classifier in terms of Acc and F_{mac}. From Table II, we can observe that OUFSDFC achieves the highest average ranks of 2.69 and 2.31 on Acc and F_{mac}, respectively. Among all thirteen datasets, OUFSDFC shows better performance than all compared supervised SFS methods on Orlraws10P, Pixraws10P, WarpPIE10P, and SMK datasets on both metrics. For the remaining datasets, OUFSDFC presents a comparable performance with both group-level and individual-level supervised SFS methods. In Figures 2 and 3, the CD-diagrams are obtained to show the statistical comparison between OUFSDFC and the supervised SFS methods using the Nemenyi post-hoc test. From Figures 2 and 3, the Nemenyi post-hoc test

[1] https://jundongl.github.io/scikit-feature/datasets.html
[2] https://github.com/kuiy/LOFS
[3] https://github.com/doodzhou/OSFS
[4] https://github.com/XuyangAbert/OUFSDFC

Datasets	Metrics	Alpha-Investing [24]	SAOLA [26]	Fast-OSFS [3]	OFS-Density [2]	OFS-A3M [27]	Group SAOLA [26]	OGFSS-FI [13]	OUFSDFC
ALLAML	Acc	0.7534 (8)	0.9233 (5)	0.9252 (4)	0.9393 (2)	0.8612 (7)	0.9292 (3)	0.8708 (6)	**0.9446 (1)**
	F_{mac}	0.7481 (8)	0.9070 (5)	0.9116 (3)	**0.9319 (1)**	0.8301 (7)	0.9119 (4)	0.8397 (6)	0.9293 (2)
Lung	Acc	0.4174 (8)	0.5646 (4)	0.4699 (7)	0.5525 (5)	0.5255 (6)	0.5720 (3)	0.5902 (2)	**0.6388 (1)**
	F_{mac}	0.3678 (8)	0.4296 (5)	0.3783 (7)	0.4252 (6)	0.4480 (4)	0.4696 (2)	0.4628 (3)	**0.5322 (1)**
Arcene	Acc	0.7452 (5)	0.6426 (7)	0.7023 (6)	**0.8391 (1)**	0.8283 (2)	0.5929 (8)	0.7647 (4)	0.7892 (3)
	F_{mac}	0.7416 (4)	0.6308 (7)	0.6976 (5)	**0.8308 (1)**	0.8248 (2)	0.5807 (8)	0.6891 (6)	0.7915 (3)
Lymphoma	Acc	0.4992 (8)	0.7571 (3)	0.6523 (7)	0.6726 (6)	**0.7695 (1)**	0.6832 (5)	0.7150 (4)	0.7425 (2)
	F_{mac}	0.4545 (8)	0.6851 (2)	0.5881 (5)	0.5756 (6)	0.6038 (4)	0.6505 (3)	0.5078 (7)	**0.8063 (1)**
Orlraws10P	Acc	0.7801 (6)	0.8201 (3)	0.7404 (7)	0.8621 (2)	0.8090 (4)	0.8012 (5)	0.6801 (8)	**0.8900 (1)**
	F_{mac}	0.5789 (8)	0.6917 (5)	0.6620 (6)	0.7900 (2)	0.7584 (4)	0.7610 (3)	0.6278 (7)	**0.8583 (1)**
Pixraw10P	Acc	0.7912 (5)	0.7453 (7)	0.7685 (6)	0.9112 (2)	0.8630 (4)	0.7142 (8)	0.8650 (3)	**0.9273 (1)**
	F_{mac}	0.7905 (5)	0.7501 (7)	0.7011 (8)	0.8742 (2)	0.8252(3)	0.7609 (6)	0.8230 (4)	**0.9100 (1)**
WarpPIE10P	Acc	0.8152 (4)	0.6437 (7)	0.8050 (5)	0.9053 (2)	0.8972 (3)	0.6210 (6)	0.6052 (8)	**0.9233 (1)**
	F_{mac}	0.8175 (4)	0.6114 (7)	0.7913 (5)	0.8963 (2)	0.8896 (3)	0.6222 (6)	0.5772 (8)	**0.9125 (1)**
COIL20	Acc	**0.9813 (1)**	0.6635 (7)	0.7986 (6)	0.9687 (2)	0.9657(3)	0.6233 (8)	0.9493 (4)	0.9059 (5)
	F_{mac}	**0.9809 (1)**	0.6537 (7)	0.7835 (6)	0.9559 (3)	0.9649 (2)	0.5932 (8)	0.9280 (4)	0.8965 (5)
Colon	Acc	0.7186 (8)	0.7219 (7)	0.7600 (3)	0.7590 (4)	0.7571(5)	**0.7898 (1)**	0.7605 (2)	0.7548 (6)
	F_{mac}	0.6550 (8)	0.6997 (6)	0.7215 (2)	0.7200 (3)	0.7039 (4)	**0.7541 (1)**	0.6945 (7)	0.7006 (5)
GIL-85	Acc	0.7087 (8)	0.8261 (3)	0.8269 (2)	**0.8352 (1)**	0.7920 (7)	0.8004 (5)	0.7958 (6)	0.8032 (4)
	F_{mac}	0.6284 (8)	0.8232 (2)	0.8154 (3)	**0.8287 (1)**	0.7282 (7)	0.8096 (4)	0.7845 (5)	0.7676 (6)
GILMO	Acc	0.4121 (8)	0.6818 (4)	0.6296 (7)	0.6678 (5)	0.6639 (6)	0.6995 (2)	0.6939 (3)	**0.7012 (1)**
	F_{mac}	0.4018 (8)	0.5806 (6)	0.5482 (7)	**0.6374 (1)**	0.5957 (5)	0.5967 (4)	0.6131 (3)	0.6172 (2)
SMK	Acc	0.6061 (7)	0.6654 (2)	0.6511 (5)	0.5964 (8)	0.6363 (6)	0.6648 (3)	0.6643 (4)	**0.6793 (1)**
	F_{mac}	0.5975 (7)	0.6575 (2)	0.6451 (5)	0.5889 (8)	0.6270 (6)	0.6568 (3)	0.6557 (4)	**0.6691 (1)**
Carcinom	Acc	0.7858 (5)	0.7963 (3)	0.7226 (8)	0.8163 (2)	**0.8235** (1)	0.7505 (6)	0.7881 (4)	0.7484 (7)
	F_{mac}	0.6232 (6)	0.6761 (4)	0.4761 (8)	0.6428 (5)	0.7467 (2)	0.6053 (7)	0.6804 (3)	**0.8323 (1)**
Avg. ranks	Acc	6.23	4.69	5.62	3.23	4.23	4.85	4.46	**2.69**
	F_{mac}	6.38	5.00	5.46	3.15	4.07	4.46	5.15	**2.31**

TABLE II
PERFORMANCE COMPARISON WITH SEVEN SUPERVISED BASELINE SFS METHODS USING KNN CLASSIFIER (K=5). (THE RELATIVE RANK OF EACH ALGORITHM IS SHOWN WITHIN THE PARENTHESES.)

Datasets	No. of S.	No. of F.	No. of C.	Domain	Type
ALLAML	72	7129	2	Medical	Continuous
Lung	73	325	7	Medical	Discrete
Arcene	200	10000	2	Medical	Continuous
Lymphoma	96	4026	9	Medical	Discrete
Orlraws10P	100	10304	10	Image	Continuous
Pixraws10P	100	10000	10	Image	Continuous
WarpPIE10P	210	2420	10	Image	Continuous
COIL20	1440	1024	20	Image	Continuous
Colon	62	2000	2	Biological	Discrete
GLI-85	85	22283	2	Biological	Continuous
GLIMO	50	4434	4	Biological	Continuous
SMK	187	19993	2	Biological	Continuous
Carcinom	174	9182	11	Biological	Continuous

TABLE III
DATASET PROPERTIES [52].(NO. OF S.: NUMBER OF SAMPLES; NO. OF F.: NUMBER OF FEATURES; NO. OF C.: NUMBER OF CLASSES.)

reveals that OUFSDFC achieves statistically better or comparable performance with the individual-level and group-level supervised SFS methods on the KNN classifier while it does not require any label information. Besides, the performance of the OUFSDFC method is statistically comparable to OGFSS-FI method without explicitly exploring the interactions among features when the KNN classifier is used.

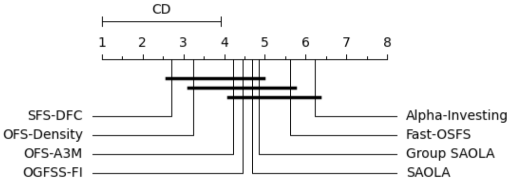


Fig. 2. Comparison of OUFSDFC against baseline methods with the Nemenyi test using KNN in terms of Acc.

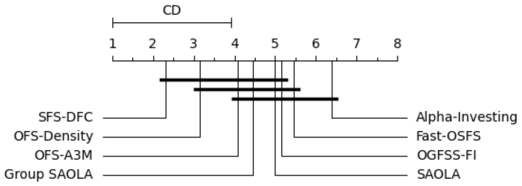


Fig. 3. Comparison of OUFSDFC against baseline methods with the Nemenyi test using KNN in terms of F_{mac}.

b) OUFSDFC vs. Compared SFS methods on decision tree classifier: For the decision tree classifier, in Table IV, the comparison between OUFSDFC and the baseline SFS methods demonstrates that OUFSDFC has the highest average rank on the F_{mac} and the third-highest average rank on Acc. Based on Acc, Figure 4 indicates that OUFSDFC achieves statistically comparable performance with all seven supervised methods. Similarly, in Figure 5, OUFSDFC provides statistically comparable performance with all seven supervised SFS methods in terms of F_{mac} using the decision tree classifier. In particular, OUFSDFC shows very similar performance with the OGFSS-FI method on the decision tree classifier. This observation can be attributed to the fact that the decision tree classifier can explore the interaction among features to improve the classification performance such that the advantage of the OGFSS-FI method becomes less significant. Overall, the OUFSDFC method yields statistically comparable performance with seven supervised baseline techniques on the decision tree classifier without using any label information, which proves the efficacy of the proposed framework.

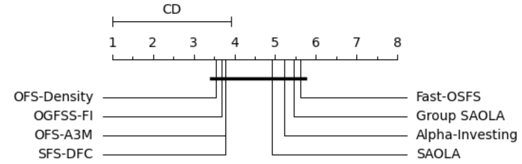


Fig. 4. Comparison of OUFSDFC against baseline methods with the Nemenyi test using decision tree in terms of Acc.

Datasets	Metrics	Alpha-Investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group SAOLA	OGFSS-FI	OUFSDFC
ALLAML	Acc	0.7169 (8)	0.8761 (4)	0.8861 (2)	0.8649 (5)	0.8116 (7)	**0.8935 (1)**	0.8552 (6)	0.8789 (3)
	F_{mac}	0.7024 (8)	0.8584 (3)	0.8600 (2)	0.8360 (5)	0.7787 (7)	**0.8772 (1)**	0.8302 (6)	0.8425 (4)
Lung	Acc	0.3384 (8)	0.4813 (5)	0.4751 (7)	0.4865 (4)	**0.5455 (1)**	0.4813 (6)	0.5364 (2)	0.5351 (3)
	F_{mac}	0.2911 (8)	0.3503 (5)	0.3326 (7)	0.3589 (4)	**0.4385 (1)**	0.3503 (6)	0.4106 (3)	0.4252 (2)
Arcene	Acc	0.6704 (6)	0.6211 (7)	0.6739 (5)	**0.7441 (1)**	0.7318 (2)	0.6074 (8)	0.7125 (3)	0.7009 (4)
	F_{mac}	0.6671 (5)	0.6082 (7)	0.6627 (6)	**0.7365 (1)**	0.7238 (2)	0.5944 (8)	0.6970 (3)	0.6941 (4)
Lymphoma	Acc	0.4813 (8)	0.5710 (6)	0.5678 (7)	0.6586 (2)	0.6137 (4)	0.5939 (5)	0.6512 (3)	**0.7069 (1)**
	F_{mac}	0.4195 (8)	0.5260 (6)	0.5264 (5)	0.5625 (3)	0.5011 (7)	0.5581 (4)	0.5652 (2)	**0.5984 (1)**
Orlraws10P	Acc	0.7250 (3)	0.5080 (7)	0.5620 (6)	0.5890 (5)	0.7240(4)	0.4620 (8)	**0.7333 (1)**	0.7950 (2)
	F_{mac}	0.5690 (4)	0.5139 (7)	0.5268 (6)	0.5339 (5)	0.6604 (3)	0.5060 (8)	**0.7001 (1)**	0.7423 (2)
Pixraw10P	Acc	0.9380 (2)	0.8420 (6)	0.7790 (8)	**0.9500 (1)**	0.9070 (3)	0.8370 (7)	0.8820 (5)	0.8970 (4)
	F_{mac}	0.9180 (2)	0.7968 (6)	0.7232 (8)	**0.9337 (1)**	0.8787 (4)	0.7867 (7)	0.8457 (5)	0.8650 (3)
WarpPIE100	Acc	0.7508 (2)	0.5431(8)	0.6910 (5)	**0.7776 (1)**	0.7374(3)	0.5663 (7)	0.6081 (6)	0.7102 (4)
	F_{mac}	0.7343 (2)	0.5154 (8)	0.6673 (5)	0.**7632 (1)**	0.7182(3)	0.5394 (7)	0.5763 (6)	0.6884 (4)
COIL20	Acc	0.8144 (3)	0.7582 (7)	0.7611 (6)	0.8133 (4)	**0.8457 (1)**	0.7554 (8)	0.7889 (5)	0.8225 (2)
	F_{mac}	0.5951 (6)	0.5232 (8)	0.5902 (7)	0.6944 (4)	**0.8443 (1)**	0.6104 (5)	0.8016 (3)	0.8113 (2)
Colon	Acc	0.7264 (6)	**0.7821 (1)**	0.7760 (2)	0.7276 (5)	0.7210 (7)	0.7600 (3)	0.7313 (4)	0.7095 (8)
	F_{mac}	0.6725 (6)	**0.7426 (1)**	0.7317 (2)	0.6744 (5)	0.6619 (7)	0.7109 (3)	0.6837 (4)	0.6538 (8)
GIL-85	Acc	0.7905 (6)	**0.8210 (1)**	0.7877 (4)	0.8144 (2)	0.7397 (8)	0.7983 (3)	0.7825 (5)	0.7736 (7)
	F_{mac}	0.5839 (8)	0.6189 (4)	0.5872 (7)	0.6148 (5)	0.6768 (2)	0.6013 (6)	0.6705 (3)	**0.6968 (1)**
GILMO	Acc	0.4178 (8)	0.6337 (4)	0.6172 (5)	0.5954 (6)	0.5920 (7)	0.6327 (3)	**0.6289 (1)**	0.6035 (2)
	F_{mac}	0.3941 (8)	0.6183 (2)	0.5543 (3)	0.5484 (5)	0.5075 (7)	**0.6185 (1)**	0.5275 (6)	0.5393 (4)
SMK	Acc	0.5942 (7)	0.6521 (2)	0.6252 (6)	0.5770 (8)	0.6257 (5)	**0.6563 (1)**	0.6355 (3)	0.6348 (4)
	F_{mac}	0.5885 (7)	0.6465 (2)	0.6203 (4)	0.5670 (8)	0.6089 (6)	**0.6502 (1)**	0.6285 (3)	0.6163 (5)
Carcinom	Acc	0.6032 (3)	0.5855 (6)	0.5380 (8)	0.5933 (5)	**0.6348 (1)**	0.5451 (7)	0.5971 (4)	0.6295 (2)
	F_{mac}	0.4940 (3)	0.4726 (4)	0.4145 (8)	0.4429 (6)	0.5130 (2)	0.4395 (7)	0.4656 (5)	**0.5308 (1)**
Avg. ranks	Acc	5.23	4.92	5.62	**3.54**	3.77	5.46	3.69	3.76
	F_{mac}	5.77	4.85	5.38	4.00	3.92	4.92	3.85	**3.31**

TABLE IV

PERFORMANCE COMPARISON WITH SEVEN SUPERVISED BASELINE SFS METHODS USING DECISION TREE. (THE RELATIVE RANK OF EACH ALGORITHM IS SHOWN WITHIN THE PARENTHESES.)

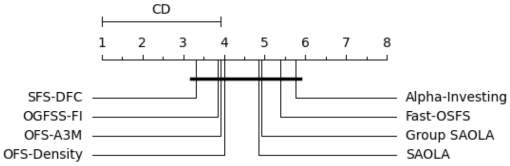


Fig. 5. Comparison of OUFSDFC against baseline methods with the Nemenyi test using decision tree in terms of F_{mac}.

SFS Methods	Time Complexity
Alpha-Investing	$O(t * m * \|F S_t\|^2)$
SAOLA	$O(t * m * \|F S_t\|)$
Fast OSFS	$O(\|F S_t\| * q^{\|F S_t\|})$
OFS-Density	$O(m^2 n^2 log(n))$
OFS-A3M	$O(m^2 n^2 log(n))$
Group SAOLA	$O(t * m * \|F S_t\|)$
OGFSS-MI	$O(\|F S_t\|^3 + m)$
OUFSDFC	$O(nm^2 + nlog(n) + \|F C_{G_t}\|\|F C_0\|)$

TABLE V

TIME COMPLEXITY COMPARISON. (HERE, q IS A CONSTANT VALUE.)

F. Number of final selected features

Following the parameter settings in Section IV-D, the number of final selected features for all compared SFS methods and the OUFSDFC framework is summarized in Table VI. Since the ten-fold cross-validation is used, we reported the average number of selected features from each method in Table VI.

As shown in Tables II and VI, OUFSDFC shows better or comparable performance with more selected features in most benchmark datasets. This can be explained by the fact that supervised SFS methods utilized the label information to filter out highly redundant and less relevant features. However, OUFSDFC is an unsupervised method and it only explores the group structure among features to select relevant features with high descriptive power and remove redundant features primarily based on the correlation or dependency among features. Consequently, the OUFSDFC method selects more features to achieve better or comparable performance than supervised SFS methods. In general, the performance of the classification model improves with the increase in the number of selected features if those features are not highly correlated. In case some selected features are highly correlated with each other, the multicollinearity issue can significantly degrade the model performance, especially for classifiers such as KNN, linear regression, and logistic regression [57]. Since OUFSDFC is designed to reduce the redundancy among features by obtaining a subset of less correlated features, it demonstrates better or comparable performance on the KNN with more selected features than other supervised SFS methods. Due to the low correlations among features in the COIL20 dataset, the redundancy minimization does not show obvious performance improvement, and thus OUFSDFC presents worse performance than the Alpha-Investing method.

G. Parameter analysis and execution time analysis

a) Parameter analysis.: To study the effects of different chunk sizes on the proposed framework (OUFSDFC), three representative datasets, including COIL20, ALLAML, and Carcinom, were selected to perform the parameter sensitivity analysis. These three datasets come from the medical, image, and biology domains. We set the chunk size from 50 to 400 with an increment of 50 and recorded the performance of OUFSDFC on both the KNN and decision tree classifier. Figure 6 displays the curves of $accuracy$ and F_{mac} on both the KNN and decision tree classifiers, respectively. As shown in

Datasets	Alpha-Investing	SAOLA	Fast-OSFS	OFS-Density	OFS-A3M	Group SAOLA	OGFS-FI	OUFSDFC
ALLMALL	14.62	20.83	5.17	4.89	15.82	14.45	13.63	57.12
Lung	3.58	9.75	2.86	5.40	21.55	8.71	12.91	64.00
Arcene	16.00	33.82	10.29	73.10	43.26	30.63	105.12	156.50
Orlraws10P	12.00	5.39	4.7	5.48	11.99	3.71	15.24	122.00
Pixraw10P	12.00	8.09	5.11	132.25	7.34	7.39	21.38	79.79
WarpPIE100	47.00	3.09	5.94	29.10	14.48	3.71	34.74	105.00
Lymphoma	4.09	15.12	7.42	8.60	27.90	18.57	46.05	266.30
COIL20	250.20	5.18	7.97	161.50	19.05	4.87	54.32	98.00
Colon	2.84	4.6	3.48	6.05	26.48	2.95	14.55	92.00
GIL-85	20.00	32.61	7.7	14.60	21.06	16.42	44.58	81.12
GILMO	4.12	15.99	5.43	8.80	23.92	12.71	57.47	112.20
SMK	4.22	4.37	5.11	12.29	18.89	5.09	23.12	264.30
Carcinom	27.00	41.46	13.65	39.46	38.54	29.78	109.41	106.60

TABLE VI
AVERAGE NUMBER OF SELECTED FEATURES FROM SEVEN SUPERVISED SFS METHODS AND THE OUFSDFC FRAMEWORK.

Figure 6, the change in chunk size does not cause a significant performance variation on the OUFSDFC method in terms of the Acc and F_{mac} when the KNN classifier is used as the benchmark classifier. For the decision tree classifier, similar observations can be obtained from Figure 6. According to these observations, it demonstrates that the performance of the OUFSDFC approach is not sensitive to the change in chunk size.

b) Execution time analysis.: As described in Section IV-B, our experiments are conducted on an Intel Xeon (R) machine with 64GB RAM operating on Microsoft Windows 10. We obtained the execution time of the OUFSDFC method for each feature chunk in all benchmark datasets, and the results are presented in Figure 7. From Figure 7, OUFSDFC takes a longer time to handle features in the discrete feature streams such as Lymphoma and Colon datasets. According to Section III-C, the sample size imposes additional time complexity in the feature similarity evaluation procedure for the discrete feature streams. Consequently, OUFSDFC requires more time to perform the feature similarity evaluation operation in the discrete feature streams. For the continuous feature streams, the maximum execution time of the proposed OUFSDFC technique is less than one second for a single feature chunk.

H. Time complexity comparison

In Table V, the time complexity comparison between the OUFSDFC method and seven supervised SFS methods is provided. As shown in Table V, it is obvious that OUFSDFC has less time complexity than OFS-Density and OFS-A3M methods. For the remaining five baseline methods, OUFSDFC requires more time complexity. Overall, OUFSDFC achieves comparable performance with OFS-Density and OFS-A3M methods with less time complexity. At the same time, although OUFSDFC takes more time complexity than the remaining five supervised SFS methods, it presents better performance and does not require any label information.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we developed an unsupervised online group-level SFS framework using a dynamic density-based feature stream clustering procedure (OUFSDFC) to handle the label scarcity for both continuous and discrete feature streams. To decrease the redundancy among features, OUFSDFC performs feature stream clustering analysis on both continuous and discrete feature streams using the developed DFPSL-clustering method. An unsupervised online feature selection strategy is developed to ensure the feature relevance maximization and redundancy minimization using the feature cluster summary. OUFSDFC is a generalized SFS framework that is independent of any density-based feature stream clustering methods. Experimental results and comparison studies proved that the OUFSDFC method achieves statistically better or comparable performance with the state-of-the-art supervised SFS approaches without using label information.

Current SFS approaches, as well as the proposed framework, assume the number of classes is fixed over time, and the occurrence of novel classes known as concept evolution is ignored. In the future, we will extend the OUFSDFC framework to address the concept evolution in feature stream analysis.

REFERENCES

[1] J. Li, X. Hu, J. Tang, and 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.

[2] P. Zhou, X. Hu, P. Li, and X. Wu, "Ofs-density: A novel online streaming feature selection method," *Pattern Recognition*, vol. 86, pp. 48–61, 2019.

[3] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 5, pp. 1178–1192, 2012.

[4] K. Yu, W. Ding, and X. Wu, "Lofs: A library of online streaming feature selection," *Knowledge-Based Systems*, vol. 113, pp. 1–3, 2016.

[5] X. Hu, P. Zhou, P. Li, J. Wang, and X. Wu, "A survey on online feature selection with streaming features," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 479–493, 2018.

[6] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Online learning from capricious data streams: a generative approach," in *International Joint Conference on Artificial Intelligence Main track*, 2019.

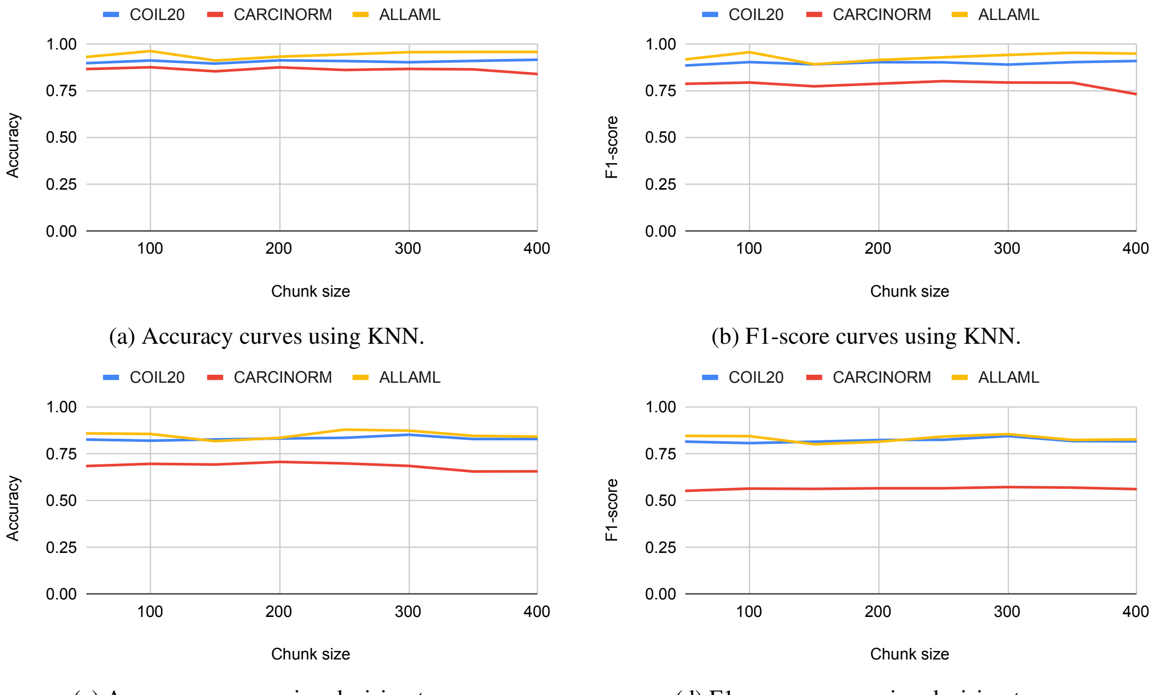
[7] K. Yu, X. Wu, W. Ding, and J. Pei, "Towards scalable and accurate online feature selection for big data," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 660–669.

[8] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, "Streamwise feature selection," 2006.

[9] X. Wu, K. Yu, H. Wang, and W. Ding, "Online streaming feature selection," in *ICML*, 2010.

[10] H. Li, X. Wu, Z. Li, and W. Ding, "Online group feature selection from feature streams," in *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

[11] M. Rahmaninia and P. Moradi, "Osfsmi: online stream feature selection method based on mutual information," *Applied Soft Computing*, vol. 68, pp. 733–746, 2018.



(a) Accuracy curves using KNN.

(b) F1-score curves using KNN.

(c) Accuracy curves using decision tree.

(d) F1-score curves using decision tree.

Fig. 6. Accuracy and F1-score curves of OUFSDFC approach in the COIL-20, Carcinorm, and ALLAML datasets.

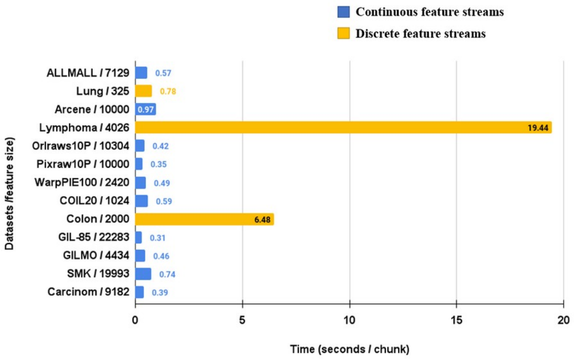


Fig. 7. The execution time of the OUFSDFC method for a single feature chunk in all thirteen benchmark datasets.

[12] J. Wang, M. Wang, P. Li, L. Liu, Z. Zhao, X. Hu, and X. Wu, "Online feature selection with group structure analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3029–3041, 2015.

[13] P. Zhou, N. Wang, and S. Zhao, "Online group streaming feature selection considering feature interaction," *Knowledge-Based Systems*, vol. 226, p. 107157, 2021.

[14] P. Zhou, P. Li, S. Zhao, and X. Wu, "Feature interaction for streaming feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[15] K. Yu, X. Wu, W. Ding, and H. Wang, "Exploring causal relationships with streaming features," *The Computer Journal*, vol. 55, no. 9, pp. 1103–1117, 2012.

[16] X. Guo and J. Yang, "Causal structure learning algorithm based on streaming features," in *2017 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 2017, pp. 192–197.

[17] J. Yang, X. Guo, N. An, A. Wang, and K. Yu, "Streaming feature-based causal structure learning algorithm with symmetrical uncertainty," *Information Sciences*, vol. 467, pp. 708–724, 2018.

[18] J. Yang, L. Jiang, A. Shen, and A. Wang, "Online streaming features causal discovery algorithm based on partial rank correlation," *IEEE Transactions on Artificial Intelligence*, 2022.

[19] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[20] Y.-m. Cheung and H. Jia, "Unsupervised feature selection with feature clustering," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1. IEEE, 2012, pp. 9–15.

[21] X. Yan, S. Nazmi, B. A. Erol, A. Homaifar, B. Gebru, and E. Tunstel, "An efficient unsupervised feature selection procedure through feature clustering," *Pattern Recognition Letters*, vol. 131, pp. 277–284, 2020.

[22] X. Yan, M. Sarkar, B. Gebru, S. Nazmi, and A. Homaifar, "A supervised feature selection method for mixed-type data using density-based feature clustering," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 1900–1905.

[23] N. Almusallam, Z. Tari, J. Chan, A. Fahad, A. Alabdulatif, and M. Al-Naeem, "Towards an unsupervised feature selection method for effective dynamic features," *IEEE Access*, vol. 9, pp. 77 149–77 163, 2021.

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

[25] D. Sengupta, S. Bandyopadhyay, and D. Sinha, "A scoring scheme for online feature selection: Simulating model performance without retraining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 2, pp. 405–414, 2016.

[26] K. Yu, X. Wu, W. Ding, and J. Pei, "Scalable and accurate online feature selection for big data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 2, pp. 1–39, 2016.

[27] P. Zhou, X. Hu, P. Li, and X. Wu, "Online streaming feature selection using adapted neighborhood rough set," *Information Sciences*, vol. 481, pp. 258–279, 2019.

[28] M. M. Javidi and S. Eskandari, "Online streaming feature selection: a minimum redundancy, maximum significance approach," *Pattern Analysis and Applications*, vol. 22, no. 3, pp. 949–963, 2019.

[29] P. Zhou, X. Hu, P. Li, and X. Wu, "Online feature selection for high-dimensional class-imbalanced data," *Knowledge-Based Systems*, vol. 136, pp. 187–199, 2017.

[30] M. A. Hall *et al.*, "Correlation-based feature selection for machine learning," 1999.

[31] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[32] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine learning proceedings 1992*. Elsevier, 1992, pp. 249–256.

[33] S. Bandyopadhyay, T. Bhadra, P. Mitra, and U. Maulik, "Integration of dense subgraph finding with feature clustering for unsupervised feature selection," *Pattern Recognition Letters*, vol. 40, pp. 104–112, 2014.

[34] P. Moradi and M. Rostami, "Integration of graph clustering with ant colony optimization for feature selection," *Knowledge-Based Systems*, vol. 84, pp. 144–161, 2015.

[35] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 6073–6082, 2018.

[36] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, "Dynamic affinity graph construction for spectral clustering using multiple features," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 6323–6332, 2018.

[37] R. Zhou, X. Chang, L. Shi, Y.-D. Shen, Y. Yang, and F. Nie, "Person reidentification via multi-feature fusion with adaptive graph learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 5, pp. 1592–1601, 2019.

[38] A. Zubaroğlu and V. Atalay, "Data stream clustering: a review," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1201–1236, 2021.

[39] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 133–142.

[40] F. Cao, M. Estert, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proceedings of the 2006 SIAM international conference on data mining*. SIAM, 2006, pp. 328–339.

[41] M. Hahsler and M. Bolaños, "Clustering data streams based on shared density between micro-clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1449–1461, 2016.

[42] I. Ntoutsi, A. Zimek, T. Palpanas, P. Kröger, and H.-P. Kriegel, "Density-based projected clustering over high dimensional data streams," in *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 2012, pp. 987–998.

[43] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[44] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135–145, 2016.

[45] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognition*, vol. 107, p. 107449, 2020.

[46] S. A. Seyedi, A. Lotfi, P. Moradi, and N. N. Qader, "Dynamic graph-based label propagation for density peaks clustering," *Expert Systems with Applications*, vol. 115, pp. 314–328, 2019.

[47] S. Gong, Y. Zhang, and G. Yu, "Clustering stream data by exploring the evolution of density mountain," *Proceedings of the VLDB Endowment*, vol. 11, no. 4, pp. 393–405, 2017.

[48] N. Almusallam, Z. Tari, J. Chan, and A. AlHarthi, "Ufssf-an efficient unsupervised feature selection for streaming features," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 495–507.

[49] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in c," *Cambridge University Press*, vol. 1, p. 3, 1988.

[50] X. Yan, M. Razeghi-Jahromi, A. Homaifar, B. A. Erol, A. Girma, and E. Tunstel, "A novel streaming data clustering algorithm based on fitness proportionate sharing," *IEEE Access*, vol. 7, pp. 184 985–185 000, 2019.

[51] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993.

[52] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 94, 2018.

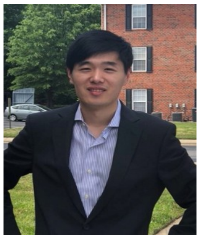
[53] J. D. Kelleher, B. Mac Namee, and A. D'arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

[54] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

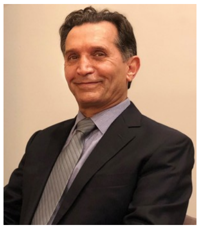
[55] D. G. Pereira, A. Afonso, and F. M. Medeiros, "Overview of friedman's test and post-hoc analysis," *Communications in Statistics-Simulation and Computation*, vol. 44, no. 10, pp. 2636–2653, 2015.

[56] T. Pohlert, "The pairwise multiple comparison of mean ranks package (pmcmr)," *R package*, vol. 27, no. 2019, p. 9, 2014.

[57] M. Nilashi, O. Ibrahim, H. Ahmadi, and L. Shahmoradi, "A knowledge-based system for breast cancer classification using fuzzy logic method," *Telematics and Informatics*, vol. 34, no. 4, pp. 133–144, 2017.



Xuyang Yan (Student Member, IEEE) received his B.S. degree in Electrical Engineering from North Carolina Agricultural and Technical State University (NC A &T) and Henan Polytechnic University in 2016. In 2018, he earned his M.S. degree in electrical engineering at NC A &T. He is currently pursuing his Ph.D. degree in electrical engineering at NC A &T. His research interests include clustering, classification, feature selection, data stream analysis, active learning, multi-label classifications, and the application of machine learning techniques in autonomous vehicles.



Abdollah Homaifar (Member, IEEE) received his B.S. and M.S. degrees from the State University of New York at Stony Brook in 1979 and 1980, respectively, and his Ph.D. degree from the University of Alabama in 1987, all in Electrical Engineering. He is the NASA Langley Distinguished Professor and the Duke Energy Eminent professor in the Department of Electrical and Computer Engineering at North Carolina A &T State University (NCA &TSU). He is the director of the Autonomous Control and Information Technology Institute and the Testing, Evaluation, and Control of Heterogeneous Large-scale Systems of Autonomous Vehicles (TECHLAV) Center at NCA&TSU. His research interests include machine learning, unmanned aerial vehicles (UAVs), testing and evaluation of autonomous vehicles, optimization, and signal processing. He also serves as an associate editor of the Journal of Intelligent Automation and Soft Computing and is a reviewer for IEEE Transactions on Fuzzy Systems, Man Machines and Cybernetics, and Neural Networks.

Mrinmoy Sarkar (Student Member, IEEE) received his B.S. degree in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology in 2016, and his Ph.D. degree in Electrical and Computer Engineering at North Carolina A&T State University in 2022. He worked as a software engineer at Samsung R&D Institute Bangladesh Ltd. from July 2016 - July 2017. His research interests include eVTOL performance evaluation for UAM missions, testing and evaluation of autonomous behavior of UAV agents using machine learning technique, developing different complex scenarios for testing UAVs and a heterogeneous system consisting of UAVs as well as UGVs, analyzing the behavior of large-scale autonomous systems and the application of machine learning techniques in robotics.

Benjamin Lartley received the BSc. Degree in Electrical and Electronics Engineering from Kwame Nkrumah University of Science and Technology, Kumasi, Ghana in 2018. He is currently pursuing the PhD. Degree in Electrical Engineering at North Carolina A&T State University, NC, USA. His research interests includes machine learning, mathematical optimization, and mobility on demand systems.

Kishor Datta Gupta (Senior Member, IEEE) is an assistant professor in the Cyber-Physical department at Clark Atlanta University. He obtained his Ph.D. from the University of Memphis. His research interests include bio-inspired algorithms, computer security, computer vision, etc. He is a senior Member of IEEE and serves as the program committee member in the flagship artificial intelligence conference AAAI-23. He has one patent and several peer publications related to adversarial machine learning.

APPENDIX

In this section, we provide detailed mathematical derivations for Equation 1 to approximate the density distribution of the feature stream. The derivations are outlined as follows:

Proof: Assume that the distance from the current feature $f^i_{G_t}$ in G_t to the rest of the features in the feature stream follows a Laplacian distribution, then the density value of the feature distribution $f^i_{G_t}$ can be expressed as:

$$D_{f^i_{G_t}} = \sum_{p=1}^{|F_t|} \left(e^{-\frac{\|f^i_t - f^p_t\|}{\beta_t}}\right)^{\gamma_t}, \tag{3}$$

where Table I provides the definitions of all the necessary parameters. The norm here refers to the feature dissimilarity between an incoming feature $f^i_{G_t}$ and another feature f^p_t in F_t.

The parameter $|F_t|$ is the total number of the previous and current features that can be decomposed by the sum of the number of the previous features $|F_{t-1}|$ and the number of new arriving features m, hence: $|F_{t-1}| + m = |F_t|$. Also, the number of the old features can be further decomposed by the sum of features in the previous feature clusters: $|F_{t-1}| = \sum_{k=1}^{|FC_0|} |FC^k_0|$. The density value of the new sample $f^i_{G_t}$ can be rewritten as the sum of the densities from the previous features and new features:

$$D_{f^i_{G_t}} = \left[\sum_{k=1}^{|FC_0|}\sum_{l=1}^{|FC^k_0|} \left(e^{-\frac{d_{il}}{\beta_t}}\right)^{\gamma_t} + \sum_{j=1}^{m} \left(e^{-\frac{d_{f_{ij}}}{\beta_t}}\right)^{\gamma_t}\right], \tag{4}$$

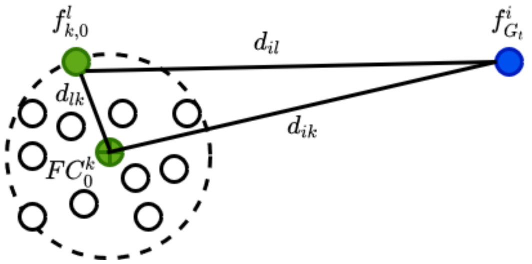


Fig. 8. The historical feature instance $f^l_{k,0}$ in the previous cluster k and a new feature instance $f^i_{G_t}$.

According to triangular inequality from $\triangle(f^i_{G_t}, f^l_{k,0}, FC^k_0)$, the following relationship holds

$$d_{il} \le d_{ik} + d_{lk}, \tag{5}$$

such that

$$\sum_{l=1}^{|FC^k_0|} \left(e^{-\frac{d_{il}}{\beta_t}}\right)^{\gamma_t} \ge \sum_{l=1}^{|FC^k_0|} \left(e^{-\frac{d_{lk}}{\beta_t}}\right)^{\gamma_t} \times \left(e^{-\frac{d_{ik}}{\beta_t}}\right)^{\gamma_t}. \tag{6}$$

Since d_{ik} is constant for all features in the cluster FC^k_0, Equation (6) can be rewritten as

$$\sum_{l=1}^{|FC^k_0|} e^{-\frac{d_{il}}{\beta_t}} \ge e^{-\frac{d_{ik}}{\beta_t}} \times \sum_{l=1}^{|FC^k_0|} e^{-\frac{d_{lk}}{\beta_t}}. \tag{7}$$

When a new cluster appears, the variance of the data stream will change while the following relationship holds:

$$\frac{\beta_t}{\gamma_t} = \frac{\beta_{t-1}}{\gamma_{t-1}}, \tag{8}$$

such that

$$\sum_{l=1}^{|FC^k_0|} \left(e^{-\frac{d_{lk}}{\beta_t}}\right)^{\gamma_t} = \sum_{l=1}^{|FC^k_0|} \left(e^{-\frac{d_{lk}}{\beta_{t-1}}}\right)^{\gamma_{t-1}}. \tag{9}$$

Let $D^k_0 = \sum_{l=1}^{|FC^k_0|} \left(e^{-\frac{d_{lk}}{\beta_{t-1}}}\right)^{\gamma_{t-1}}$, Equation (4) can be rewritten as

$$D_{f^i_{G_t}} \ge \sum_{k=1}^{|FC_0|} \left(e^{-\frac{d_{ik}}{\beta_t}}\right)^{\gamma_t} \times D^k_0 + \sum_{j=1}^{m} \left(e^{-\frac{d_{f_{ij}}}{\beta_t}}\right)^{\gamma_t}. \tag{10}$$

Based on above Equation (10), an estimated lower boundary of the density value of feature $f^i_{G_t}$ is defined as

$$\hat{D}_{f^i_{G_t}} = \sum_{k=1}^{|FC_0|} \left(e^{-\frac{d_{ik}}{\beta_t}}\right)^{\gamma_t} \times D^k_0 + \sum_{j=1}^{m} \left(e^{-\frac{d_{f_{ij}}}{\beta_t}}\right)^{\gamma_t}, \tag{11}$$

where $D_{f^i_{G_t}} \ge \hat{D}_{f^i_{G_t}}$.

Therefore, Equation (1) is derived as a recursive lower bound of the Laplacian density function. Based on this recursive lower bound, the developed DFPSL-clustering can continuously estimate the density values of an incoming feature chunk using the historical cluster information. Instead of storing all historical features from the feature stream, it reduces the memory space by keeping only the cluster summary from the feature stream. ■