Received July 9, 2018, accepted August 28, 2018, date of publication September 13, 2018, date of current version October 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869129

Cluster Survival Model of Concept Drift in Load Profile Data

MD ABDUL MASUD, JOSHUA ZHEXUE HUANG, MING ZHONG, AND XIANGHUA FU
College of Computer Science and Software Engineering, Big Data Institute, Shenzhen University, Shenzhen 518060, China

Corresponding author: Md Abdul Masud (masud@szu.edu.cn)

This work was supported in part by the National Natural Science Foundations of China under Grant 61473194 and Grant 61472258 and in part by the Shenzhen-Hong Kong Technology Cooperation Foundation under Grant SGLH20161209101100926.

ABSTRACT An accurate scenario of customer's power consumption patterns is a worthwhile asset for electricity provider. This paper proposes a cluster survival model of concept drift in load profile data. The cluster survival model of concept drift retrieves the dynamic behaviors of the clusters over time. We formulate a new data stream clustering algorithm, I-niceStream, which identifies the number of clusters and initial cluster centers automatically for producing the clustering results. We derive a modified Kullback–Leibler divergence for computing the concept drift scores from the clustering results. The concept drift scores are used to estimate the related clusters and the clustering patterns. The survival model categorizes the clustering patterns into sustaining, fading, and emerging types. Experiments were conducted on both synthetic datasets and real-world load profile dataset collected from different factories at Guangdong province in China. Experimental results show that the cluster survival model is able to identify the clustering patterns effectively from load profile data stream. The I-niceStream algorithm significantly outperformed three state-of-the-art algorithms in clustering accuracy on synthetic stream datasets.

INDEX TERMS Concept drift, clustering pattern, data stream clustering, load profile data, survival model.

I. INTRODUCTION

Survival model focuses the survival analysis of evolving events where events describe the length of time from a starting to end point. The survival analysis is an analysis of data including some sequential events of interest over time [1]. It is assigned as time-to-event analysis, which is based on analogy to the statistical model of lifetime data such as breast cancer data [2], and gene expression data [3]. For example, an origin time of a cancer patient can be assigned to be the time point of diagnosis and end point can be assigned to be death time. Then the length of time can be calculated and predicted for further cases. Survival analysis has many applications such as fatigue failure of aircraft structures [4], depression and anxiety [5], and risk assessment of traffic congestion [6].

A data stream can be divided into several time windows. Objects in each window are distributed into different clusters which are represented as concepts. As new data elements arrive over time, the structure of the clusters changes, which is known as concept drift. Survival of clusters is represented as the status of concepts over time, which studies the time between initial structure of concepts and a subsequent changing event of concepts. The detection of concept drift identifies the dynamic behaviors of the clustering patterns.

A load profile is a set of electrical power consumption of a consumer within a range of time. Generally, power providers use the data to make a plan how much electricity they will need to make available at any given time for the consumer.

The collection of electricity consumption from factories in different industrial areas in a specific study period is represented as the load profile data. In this paper, we use power consumption of manufacturing factories as load profile data. We collected the load profile data from smart meters sampled at 15 minutes interval at Guanagdong province in China in 2012.

A multidimensional load profile data is represented as a matrix, where N is the number of rows and D is the number of columns, in Fig. 1. Row N represents the number of factories and column D represents the number of time attributes. The factory i with j time slot generates $x_{i,j}$ power consumption as data object. D time attributes in load profile data are divided into W monthly time windows. Each load profile window data $\mathcal{X}$ is also represented as a matrix with N rows and d columns. Load profile window data $\mathcal{X}$ contains N time series $X_1, X_2, \ldots, X_N$ data with d dimensional attributes $F_1, F_2, \ldots, F_d$. Let F_j be a vertical vector of N elements representing the measurements of N factories at the jth time slot.

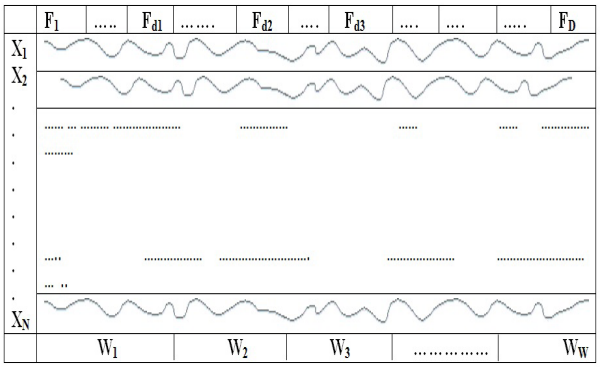

FIGURE 1. Load profile data is represented with a matrix where rows are the manufacturing factories and columns are the measurement time of power consumption.

Each column of the matrix represents the distribution of the total electricity consumption of N factories at a time measurement.

These consecutive windows of load profile data represent power consumption behaviors which may change during the study time. For example, the consumption patterns during public vacation may differ from regular time. In Fig. 2, we observe that an industry sector can consume different power consumption over time. The electricity consumption of an industry sector represents the production figure. The production figure mainly depends on production demand, weather condition and public vacation. The identification of dynamic behaviors of load profile in different factories is important for analysis the production capacity.

Each window load data is investigated to obtain the clustering solution. We use the concept drift detection to estimate the clustering patterns and survival analysis to categorize the clustering patterns among consecutive windows.

Several traditional clustering algorithms are addressed in [7]. The cluster ensembles are developed to improve the performance of the standard clustering algorithms [8]. The multitask clustering algorithm performs robustness of the clustering partition by using the relationship of multiple tasks simultaneously [9]. The unsupervised dimensionality reduction methods are used to enhance the performance of clustering results. The multilayer bootstrap network is a recently proposed method [10], which reduces the non-linear variation of data by an unsupervised deep ensemble architecture on data domain. These algorithms are effectively used to perform unsupervised learning on static data.

Data streams are considered as continuously flow of data and the underlying distribution of data stream may change to the next event of time. Data streams deserve the learning algorithms that are capable of continuous learning and forgetting the obsolete objects, can also adapt models over time. To address these criteria, several data stream clustering algorithms have been proposed [11]–[17].

The stable segmentation of load profile data is essential to support marketing strategy for distribution companies and retailers. Some of the significant contributions in clustering on load profile data are [18]–[24]. In addition, the clustering problem with concept drift is addressed to understand the dynamic behaviors of patterns in many real-world applications [13], [14], [25]–[28]. In literature, most of the data stream clustering algorithms need a given number of clusters as input parameter and these existing methods do not address the survival analysis of concept drifting patterns in data stream.

In this paper, we propose a cluster survival model of concept drift in load profile data stream. The cluster survival model detects the changing behaviors of clusters and performs the survival analysis of clustering patterns over time. For discovering the structure of clusters, we introduce a new data stream clustering algorithm named as I-niceStream, the abbreviation of **I**dentifying **n**umber of clusters and **i**nitial cluster **c**enters for clustering on data **S**tream. We use a modified Kullback-Leibler (KL) divergence to compute the concept drift scores from the clustering results. These concept drift scores are used to estimate related clusters which are used to form clustering patterns. We use survival model to categorize the clustering patterns into sustaining, fading, and emerging types. The survival model is also used to estimate the survival level and survival probability of clustering patterns.

We conducted a series of experiments on both synthetic stream datasets, and real-world load profile dataset. Experimental results show that the proposed method is able to identify the clustering patterns from load profile data and categorize the clustering patterns according to their dynamic behaviors. Furthermore, the I-niceStream algorithm significantly outperformed other data stream clustering algorithms in terms of clustering accuracy.

The remainder of this paper is organized as follows: First, we present a brief overview of related work in Section 2. We describe the problem formulation in Section 3. We introduce the cluster survival model of concept drift in load profile data in Section 4. We present the experimental results of the proposed method in Section 5. Finally, we discuss the survival model of concept drift in load profile data and the conclusion of this research in Section 6.

II. RELATED WORK

Research community in data mining has paid much attention for clustering on data stream. Several state-of-the-art algorithms [11], [12], [29]–[32] have been proposed to extract the intrinsic structure from non-stationary data.

The clustering feature vector is introduced for estimating the statistical summary from large volume of data stream in BIRCH algorithm [29]. This feature vector has several components, such as the number of objects in data, the linear sum of objects, and the sum of squared objects. These components are used to compute cluster means, radius and diameter. A clustering feature tree is built from the continuous objects of data stream. An user-given radius parameter defines whether a new object may be absorbed by a clustering

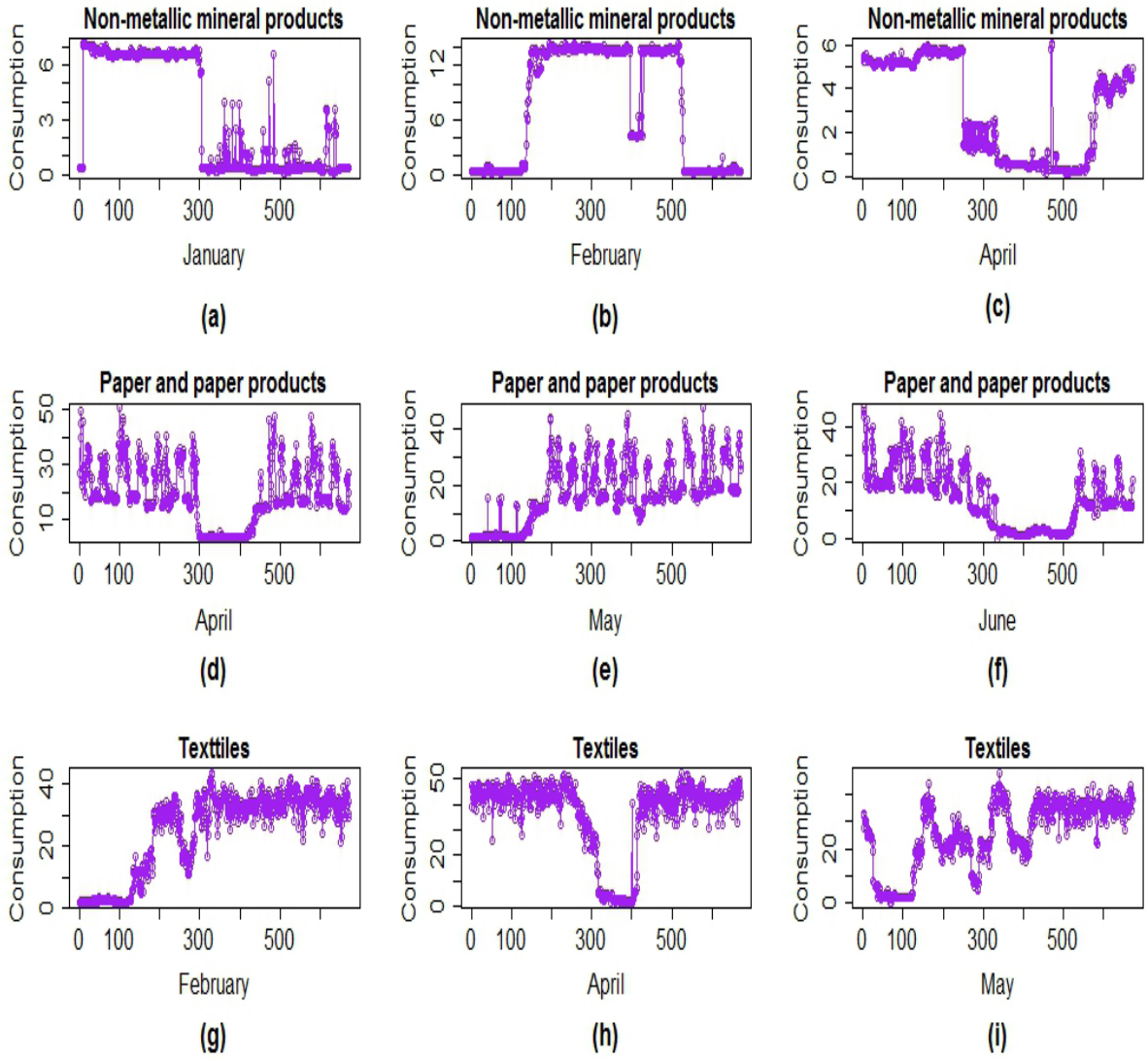

FIGURE 2. Power consumption behaviors of different industries in load profile data. (a)-(c) Non-metallic mineral products industry. (d)-(f) Paper and paper products industry. (g)-(i) Textile industry.

feature vector. Then, the objects are clustered with k-means into a user-given number of clusters.

The concept of clustering feature vector is extended with a concept of micro-cluster in CluStream algorithm [11]. This algorithm performs the clustering on continuous data into two steps, including online, and offline. In online step, the statistical summary is periodically stored into the given number of micro-clusters. The statistical summary of micro-clusters is used to generate the high level clusters with a given number of macro-clusters and time horizon at off-line step.

DenStream is a density-based clustering algorithm on data stream, which also uses clustering feature vector [12]. This algorithm has two structures for estimating statistical summary from data, such as potential-micro-clusters, and outlier-micro-clusters. A new object is inserted into the nearest potential-micro-cluster by updating the statistical summary of the cluster. The set of parameters are also adapted in manually in the algorithm.

In [31], the ClusTree algorithm is developed based on the concept of micro-clusters as compact representation of data distribution. This algorithm maintains the clustering features by extending index structure from R-tree family and automatically adapts the size of clustering model.

In online-offline steps based algorithms, the off-line step suffers high computational cost for clustering data. The derivation of the statistical summary from data stream at online step and the utilization of them for generating cluster at offline step are time consuming process. In addition, when data records are coming with real-time streaming fashion, due to the unavailability of whole training data, the prediction of the structure of clusters is far from the original structure of clusters. The major drawback of these types of algorithms is that it is needed to assign the number of clusters in advance.

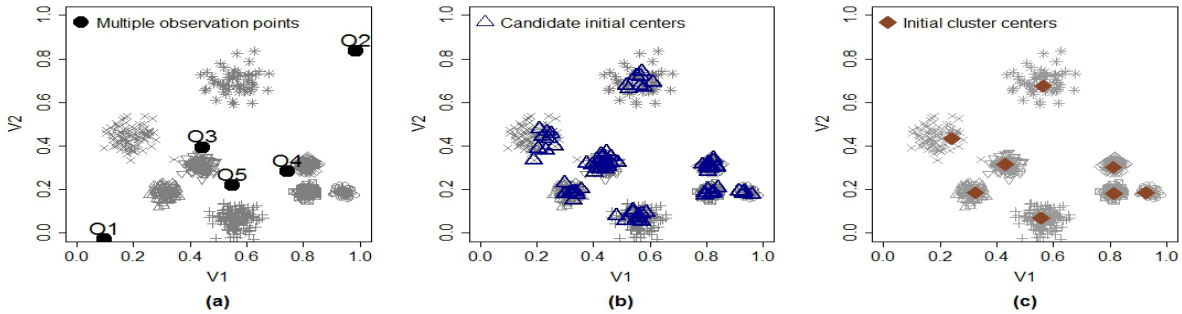

FIGURE 3. Selection process of initial cluster centers on a synthetic dataset (a) Multiple observation points observe the clusters on data (b) Estimation of the candidate initial centers from data (c) Estimation of initial cluster centers from the candidate initial centers.

Silva and Hruschka have formulated a method based on ordered multiple runs of k-means (OMRk) and bisecting k-means (BkM) for estimating the number of clusters on evolving data stream [33]. The OMRk method uses k-means repeatedly for an increasing number of clusters, then, uses the simplified Silhouette to assess the best number of clusters. Validation-index-based solutions are not likely to provide consistent results across different clustering algorithms and data structures [34], [35]. The OMRk method assigns $\sqrt{N}$ (N is the number of objects) as the maximum number of increasing clusters, which is really an inefficient initialization to deal with large volume of records on data stream.

The exploration of patterns over time windows can provide a great understanding of the evolving behaviors of the estimated clusters. Some works [28], [36]–[38] have been developed to focus on the changing behaviors of data stream. Khan *et al.* [38] have discussed the changes of patterns from load profiles along with time windows. This method uses a hierarchical binary k-means algorithm for generating base clusterings and an ensemble method to obtain final clustering solution. In pattern tracking, this work addresses the distributions of fading and emerging patterns only from load profiles.

A very simple and effective data stream clustering algorithm, I-niceStream, is developed for resolving the challenges of existing algorithms. Then, an innovative cluster survival model of concept drift is formulated for addressing the details study of all clustering patterns among time windows.

III. PROBLEM FORMULATION

This paper is formulated into two consecutive steps: clustering solution and survival analysis of clustering results based on concept drifting behaviors.

Several works focus the clustering problem on load profile data as data stream [18]–[24], [36].

In [18], household electricity consumption data is collected from the real-time meters at the clients during the period in study. A set of representative consumption patterns is discovered with several clustering algorithms where k-means outperformed others. In order to form similar customer clusters and exhibit similar patterns, a comparative clustering analysis has been performed with unsupervised clustering algorithms and self-organization maps (SOM) on electricity consumption data [20]. Similarly, the best performing technique is evaluated from k-means, k-medoid and SOM in order to segment individual households into clusters based on the patterns of electricity used across the day [39].

Another household energy consumption data is investigated with an adaptive k-means algorithm to find the K representative load shapes as clusters. The hierarchical clustering technique is used to summarize the clusters, which have close centers, as final clusters [21]. An incremental density based ensemble clustering method aggregates the obtained clusterings of subsequent time windows incrementally on load profile data [36]. A clustering framework is discussed for the automatic classification of electricity customers loads in [24]. Here the number of clusters is again an issue, which is needed to assign in advance, to run most of the data stream clustering methods.

In the first part of the proposed work, we introduce a new data stream clustering algorithm, I-niceStream, to cluster the load profile data stream. The algorithm can also identify the number of clusters and initial cluster centers automatically. Fig. 3 shows the selection process of initial cluster centers on a synthetic two-dimensional dataset, which contains eight clusters. First, we allocate several observation points in the data domain in Fig. 3(a). We transform one-dimensional data by computing the distances between each observation point and all objects in the high-dimensional data. Different locations of observation points result in different distance distributions, which represent the dense and sparse regions of objects in the original dataset.

The distance data is used to build multiple models with Gamma mixture model (GMM) and the multiple models are fitted with expectation-maximization(EM) algorithm. We obtain a set of best-fitted models for a set of observation points. Objects in each component of best-fitted models are analyzed to estimate candidate initial centers in Fig. 3(b). These candidate initial centers are abstract representation of original data.

A distance matrix is computed among the candidate initial centers. We build a tree structure from the distance matrix. An estimated height value is used to separate the tree into

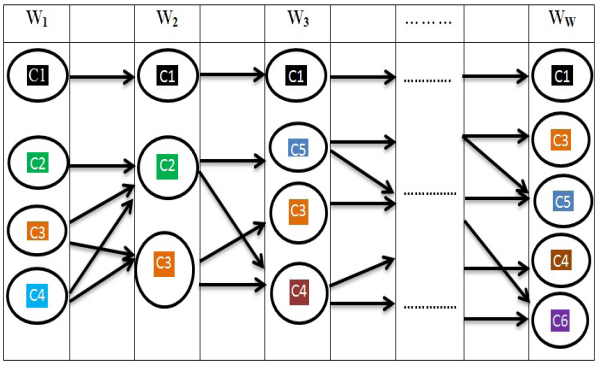

FIGURE 4. Cluster structures of different load profile windows where some clusters are continued from initial to last window, some old clusters are disappeared at intermediate window, and some new clusters are appeared at the different time windows.

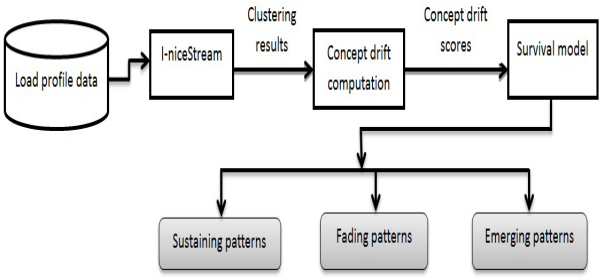

FIGURE 5. Cluster survival model of concept drift in load profile data. This model estimates the clustering patterns from load profile data.

different branches where candidate initial centers in a branch are close to each other. The number of branches is considered as the number of clusters. Then, we select one candidate initial center, which has the smallest distance to others in a branch, as an initial cluster center in Fig. 3(c). We use the initial cluster centers to cluster the load profile data.

After the segmentation of load profile data, the distribution of objects in clusters may change over time. Several methods have been developed to address the concept drift problem on data stream [13], [14], [25]–[28].

In [13], a PCA based change detection framework detects the abrupt changes in multidimensional data stream. The framework is based on projecting data with principle components. The change score is computed with a divergence metric on the densities in reference and test windows for each projection. Similarly, the margin density drift detection algorithm tracks the number of objects in the uncertainty region of a classifier as a metric to detect drift in [26]. A three-layer concept drift detection approach detects the text data by dividing into three categories: label space, feature space, and mapping relationship of labels and features in [27]. In the method [28], the data stream is divided into a set of time windows and a concept change method is applied to analyze the trend on categorical data stream. A data labeling algorithm is used to determine the clusters of current windows from the clusters of previous window. The cluster emerging and fading scenarios are presented on only two kinds of clusters.

In the second part of the proposed work, we develop a cluster survival model which analyzes the clustering patterns from different concept drifting scenarios on load profile data stream. We can explain the cluster survival analysis of patterns and define them as follows. In the Fig. 4, each window data is partitioned into several clusters. A cluster contains a set of factories which have similar power consumptions. The continuation of clusters, which have maximum number of common factories, in consecutive windows is referred as a clustering pattern. Unique number and similar color of two clusters in consecutive windows indicate the continuation of clusters. We define three types of clustering patterns in the cluster structure of load profile data:

Sustaining pattern: The sustaining pattern is a complete one, which starts from a cluster at initial window and reaches to a cluster at the end window. The sustaining pattern $P1^1$ starts from cluster $C_{1^{w_1}}$ and reaches to the cluster $C_{1^{w_W}}$ at last window.

Fading pattern: The fading pattern is an incomplete one, which starts from a cluster at any window but cannot continue to final window. The fading pattern $P2^1$ starts form cluster $C_{2^{w_1}}$ and disappears at w_3 window.

Emerging pattern: The emerging pattern is also an incomplete one, which appears from a cluster at the intermediate window and continues to next window. The emerging pattern $P4^3$ appears from a new cluster $C_{4^{w_3}}$ and continues to last window.

IV. CLUSTER SURVIVAL MODEL OF CONCEPT DRIFT

In this section, we propose a cluster survival model of concept drift. This model estimates the interesting changing behaviors of the clustering results on load profile data. Each window load data is investigated to produce the clustering results. The concept drift scores are computed from clustering results among windows. The clustering patterns are estimated from the concept drift scores. The clustering patterns are analyzed for categorizing into three types, including sustaining, fading, and emerging. Fig. 5 shows a block diagram of the cluster survival model of concept drift in load profile data.

A. CLUSTERING WITH A NEW I-niceStream ALGORITHM

I-niceStream is a data stream clustering algorithm, which estimates the number of clusters and initial cluster centers for clustering process to generate the clustering results. The I-niceStream algorithm performs a data abstraction by estimating a set of candidate initial centers from input data. A tree is built from the distance matrix of candidate initial centers. We cut the tree with a threshold height, then tree is divided into several compact branches. The number of branches of tree is considered as the number of clusters. Objects in each branch are analyzed to identify an initial cluster center. The number of clusters and the initial cluster centers are applied

as input parameters to the k-means for partitioning the data stream.

The I-niceStream algorithm uses the following steps:

1) SELECT THE BEST-FITTED GMM MODEL

Let R^d be a load profile window data domain of d dimensions and $\mathcal{X} \subset R^d$ a set of N objects in R^d. Assume $p \in R^d$ be a randomly generated point with a uniform distribution, we allocate p as an observation point to $\mathcal{X}$. We compute all distances between observation point p and N objects of $\mathcal{X}$ and obtain a set of distances $X_p = \{x_1, x_2, \ldots, x_N\}$. Given a different observation point, we can compute a different distance distribution from $\mathcal{X}$.

GMM is used instead of Gaussian mixture models because computed distance values are non-negative [40]–[42].

Let $X_p = (x_1, x_2, \ldots, x_N)$ be a set of normalized distance values calculated from $\mathcal{X}$ with respect to observation point p. The GMM of X_p is defined as:

$$p(x|\theta) = \sum_{j=1}^{M} \pi_j g(x|\alpha_j, \beta_j), \quad x \geq 0 \tag{1}$$

where θ is the vector of parameters of the GMM. M is the number of the Gamma components and π_j, α_j and β_j are the mixing proportion, shape and scale parameters of component j, respectively.

The parameters of the GMM are solved by maximizing the log-likelihood function. The log-likelihood function is defined as

$$\mathcal{L}(\theta|X_p) = \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} \pi_j g(x_i|\alpha_j, \beta_j) \right) \tag{2}$$

The EM algorithm is used to solve Eq. (2) [43]. Given X_p, latent discrete random variables $Z = \{z_i\}$ are introduced to identify the elements of X_p in the components of the GMM. $z_i = j$ indicates that element x_i in X_p is assigned to component j of the GMM. However, the values of $Z = \{z_i\}$ are unknown in advance [41].

After doing some mathematical manipulations of Eq. (2), the mixing proportion, shape and scale parameters are estimated as follows:

$$\hat{\pi}_j = \frac{1}{N} \sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n) \tag{3}$$

$$\log(\hat{\alpha}_j) - \psi(\hat{\alpha}_j) = \log \left(\frac{\sum_{i=1}^{N} x_j p(Z_i = j|x_i, \theta^n)}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)} \right) - \left(\frac{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n) \log x_i}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)} \right) \tag{4}$$

where $\psi(x) = \frac{\partial \log(\Gamma(x))}{\partial x} = \frac{\Gamma'(x)}{\Gamma(x)}$ is called the Digamma function. Since Eq. (4) has no closed solution, we can use a Newton-type algorithm [44] to calculate $\hat{\alpha}_j$. This algorithm is fast in getting the results in few iterations to converge.

$$\hat{\beta}_j = \frac{1}{\alpha_j} \frac{\sum_{i=1}^{N} x_i p(Z_i = j|x_i, \theta^n)}{\sum_{i=1}^{N} p(Z_i = j|x_i, \theta^n)} \tag{5}$$

With Eq. (5) and the estimated value of $\hat{\alpha}_j$ from Eq. (4), we can obtain the estimates of $\hat{\beta}_j$.

Multiple fitted models are obtained with respect to each observation point. The minimum value of the second order Akaike information criterion (AICc) [45], [46] is used to select the best-fitted model. The second order Akaike information criterion is calculated as below.

$$AICc = -2 \log (\mathcal{L}(\theta^*)) + 2q \left(\frac{N}{N - q - 1} \right) \tag{6}$$

where $\mathcal{L}(\theta^*)$ is the maximum log-likelihood, N is the number of objects, and q is the number of parameters. With AICc, we select the best-fitted GMM model for each observation point.

2) ESTIMATE THE CANDIDATE INITIAL CENTERS

The best-fitted GMM model for each observation point belongs with certain number of components. Each component extracts a group of objects as clustering information. When the number of objects in different clusters are imbalance, many objects are extracted from a cluster with majority objects whereas few objects are extracted from a cluster with minority objects. To consider all representative clustering information, we choose an object, which has the smallest distance to other objects in a component, as the first candidate initial center, and select another object, which has the largest distance to other objects in the component, as second candidate initial center. In this way, we estimate candidate initial centers from all components of the best-fitted models related to all observation points. These candidate initial centers are representative objects, therefore, they can carry the clustering information from the original data.

3) DETERMINE THE NUMBER OF CLUSTERS AND INITIAL CLUSTER CENTERS

We compute Euclidean distance matrix [47] among candidate initial centers and build a tree structure from distances. We estimate a threshold height value th for dividing the tree into different branches. The number of branches is considered as the number of clusters. The candidate initial centers in each branch are close to each other. The candidate initial center which has the smallest distance from others in a branch is selected as an initial cluster center. A set of initial cluster centers is selected from all of the estimated branches of the tree structure.

4) CLUSTERING

The number of clusters and the initial cluster centers are used to cluster the window load data. We use k-means algorithm for its efficiency and simplicity. The estimation of initial cluster centers and the clustering process are presented in **Algorithm** 1.

Algorithm 1 I-niceStream: A New Data Stream Clustering Algorithm

Input: Window load profile data $\mathcal{X}$
Output: Clustering result
Initialization: P observation points and $Mmax$ as the maximum number of GMMs
Select best-fitted models:
for $p := 1$ *to* P **do**
    Compute distance vector X_p between $\mathcal{X}$ and p ;
    for $M := 2$ *to* $Mmax$ **do**
        Select best-fitted model GMM(p) using minimum AICc ;
Keep all selected models GMMs(p) with different number of components c for all p ;
Estimate candidate initial centers:
for $p := 1$ *to* P **do**
    for $c := 1$ *to* c-max **do**
        Select first candidate initial center with the smallest distance to others in c ;
        Select second candidate initial center with the largest distance to others in c ;
    Keep candidate initial centers for model GMM(p) ;
Determine initial cluster centers:
Compute distance matrix $d(k)$ among candidate initial centers k ;
Make T tree from $d(k)$ and cut T with threshold th for generating K branches ;
K branches is considered as K number of clusters ;
for $b := 1$ *to* K **do**
    Select dense candidate initial center as initial cluster centers from branch b ;
Clustering:
Assign K initial cluster centers to k-means to cluster window load profile data $\mathcal{X}$;

B. COMPUTE CONCEPT DRIFT SCORE

The concept drift score represents the distributional difference between two groups of objects. The estimation of related clusters is the first step for generating cluster chain in consecutive time windows. We identify the related clusters based on the concept drift scores. Divergence metrics are used for measuring the difference between two groups of objects. These objects are belonged in two different clusters of two consecutive time windows. We assume that objects of all clusters are drawn from Gamma distribution. The most popular distribution divergence metric is the Kullback-Leibler (KL) divergence [48]. The KL divergence is an asymmetric non-negative metric that is affected by the type of change in data variance (from large to small or vice versa).

If the distribution P has larger variance value than the distribution Q, then $D_{KL}(P||Q)$ is much larger than $D_{KL}(Q||P)$. As a result, general KL divergence fails to detect some changes with decreasing variance, or it can detect changes with a large delay. A modified divergence measure is essential to overcome this problem of KL divergence. The change score is crucial step in the change detection process of clustering results among time windows.

Let two probability densities p and q which are drawn from C_p and C_q clusters, respectively. We compute the change score between p and q using a divergence metric. For this purpose, we adopt the KL divergence with fully defined model parameters of cluster data. The KL divergence of two probability density functions (PDFs) $p(x;\theta_p)$ and $q(x;\theta_q)$ is defined as:

$$\mathcal{D}_{KL}(p(x;\theta_p)||q(x;\theta_q)) = \int_x p(x;\theta_p)\log\frac{p(x;\theta_p)}{q(x;\theta_q)}dx \quad (7)$$

The PDF $p(x,\theta_p)$ can be defined with p Gamma component as

$$p(x;\theta_p) = p(x;\alpha_p,\beta_p) = \frac{x^{\alpha_p-1}e^{-(x/\beta_p)}}{\beta_p^{\alpha_p}\Gamma(\alpha_p)} \quad (8)$$

By inserting Eq. (8) in Eq. (7), we derive $\mathcal{D}_{KL}$ between two Gamma PDFs:

$$\begin{aligned}\mathcal{D}_{KL}(p(.;\alpha_p,\beta_p)||q(.;\alpha_q,\beta_q)) &= \ln\frac{\beta_p^{\alpha_p}\Gamma(\alpha_q)}{\beta_q^{\alpha_q}\Gamma(\alpha_p)} + [\psi(\alpha_p) \\ &\quad + \ln\frac{1}{\beta_p}](\alpha_p-\alpha_q) + \alpha_p(\frac{\beta_p-\beta_q}{\beta_p}) \end{aligned} \quad (9)$$

The KL divergence is a nonnegative and non-symmetric measure. It is 0 when the two distributions are completely identical, and becomes larger as the two distributions deviate from each other. The non-symmetric property makes unfair grade for detecting the degree of change of two distributions (clusters). To overcome this problem of KL divergence, Liu *et al.* [49] used modified symmetric KL divergence for evaluating the correlation between two matching scores in optimal feature selection. In this paper, we use the averaged symmetric KL divergence which is computed as follows:

$$\mathcal{D} = [\mathcal{D}_{KL}(p||q) + \mathcal{D}_{KL}(q||p)]/2 \quad (10)$$

Using Eq. (10), we compute concept drift scores $\mathcal{D}$ from clustering results among consecutive windows. Based on this concept drift score, we identify the related clusters in consecutive window. Suppose, windows w and $w+1$ have clusters $\{C_{1^w}, C_{2^w}, C_{3^w}\}$ and clusters $\{C_{1^{w+1}}, C_{2^{w+1}}, C_{3^{w+1}}\}$, respectively. We want to identify the related cluster at window $w+1$ from cluster C_{1^w} at window w. We compare the concept drift scores between cluster C_{1^w} at window w and

clusters $\{C_{1^{w+1}}, C_{2^{w+1}}, C_{3^{w+1}}\}$ at window $w+1$. Let the concept drift score $\mathcal{D}_{C_{1^w}, C_{3^{w+1}}}$ between clusters C_{1^w} and $C_{3^{w+1}}$ is smaller than other combinations, and both concept drift scores $\mathcal{D}_{C_{2^w}, C_{3^{w+1}}}$ and $\mathcal{D}_{C_{3^w}, C_{3^{w+1}}}$ are larger than concept drift score $\mathcal{D}_{C_{1^w}, C_{3^{w+1}}}$. Then we obtain that the pair of clusters C_{1^w} and $C_{3^{w+1}}$ is related between windows w and $w+1$. We estimate the chain from pairs of related clusters as clustering pattern over time windows.

C. CLUSTER SURVIVAL MODEL

Survival model is related with the studying time between entry to a study and a subsequent event. We perform survival analysis on clusters at different windows and clustering patterns among windows. We categorize the clustering patterns based on similar changing behaviors and estimate the statistical proportion of each kind of patterns.

1) CATEGORIZE THE CLUSTERING PATTERNS

From the previous section, we have computed the concept drift scores among windows load profile data. The concept drift scores are used to categorize three kinds of clustering patterns, including sustaining, fading and emerging.

Let two consecutive time windows w and $w+1$ contain the number of clusters k^w and k^{w+1}, respectively. We compare the concept drift score $\mathcal{D}$ between one cluster of w and all clusters $C_{k^{w+1}}$ of $w+1$ for assigning the status of clusters. We detect three kinds of clustering patterns based on the concept drift score $\mathcal{D}$ of clusters among windows.

a: SUSTAINING PATTERN

In this case the concept drift score $\mathcal{D}_{s^w, s^{w+1}}$ between a specific cluster C_{s^w} at window w and cluster $C_{s^{w+1}}$ at window $w+1$ is smaller than concept drift scores from other combinations, and the concept drift score $\mathcal{D}_{s^w, s^{w+1}}$ between a specific cluster $C_{s^{w+1}}$ at window $w+1$ and cluster C_{s^w} at window w is smaller than concept drift scores from other combinations. For observing the value of concept drift scores, one cluster is fixed from the current window and another cluster is chosen sequentially from all of the clusters at the next window and vice versa. We represent it as:

If $\exists(\min \mathcal{D}(C_{s^w}, C_{j^{w+1}}) \Lambda \min \mathcal{D}(C_{i^w}, C_{s^{w+1}}))$ then cluster C_{s^w} is sustaining to next window, where $1 \leq i \leq k^w$, $1 \leq j \leq k^{w+1}$, $s^w \subset k^w$ and $s^{w+1} \subset k^{w+1}$. The collection of sustaining clusters among windows make a sustaining pattern.

b: FADING PATTERN

The concept drift score $\mathcal{D}_{f^w, g^{w+1}}$ between a specific cluster C_{f^w} and cluster $C_{g^{w+1}}$ is smaller than concept drift scores from other combinations but the concept drift score $\mathcal{D}_{f^w, g^{w+1}}$ between a specific cluster $C_{g^{w+1}}$ and cluster C_{f^w} is not smaller than concept drift scores from other combinations. We represent it as: If $\exists(\min \mathcal{D}(C_{f^w}, C_{j^{w+1}}) \Lambda \neg \min \mathcal{D}(C_{i^w}, C_{g^{w+1}}))$ then cluster C_{f^w} is faded at window w, where $1 \leq i \leq k^w$, $1 \leq j \leq k^{w+1}$, $f^w \subset k^w$ and $g^{w+1} \subset k^{w+1}$. The cluster chain of cluster C_{f^w} until current window w is considered as fading pattern and the cluster C_{f^w} will be disappeared at the next window.

c: EMERGING PATTERN

There is no smallest concept drift score between any clusters at window w and a specific cluster C_e at window $w+1$. We represent it as: If $\neg \min(\mathcal{D}_{i^w}, \mathcal{D}_{e^{w+1}})$ then cluster $C_{e^{w+1}}$ is emerged at window $w+1$, where $1 \leq i \leq k^w$, and $e^{w+1} \subset k^{w+1}$. The cluster chain is started from cluster $C_{e^{w+1}}$, which is represented as emerging pattern.

2) SURVIVAL ANALYSIS OF CLUSTERING PATTERNS AMONG WINDOWS

To analyze the survival of the clusters at window and the clustering pattern among windows, we need to consider several timing factors such as starting time, lifetime, end time of pattern, and time-event based function. A pattern is a collection of related clusters among time periods. We compute the survival factor from each pair of related clusters. Survival factor is a ratio of the number of common objects and average objects between two clusters. Survival factor is calculated as follows.

$$S.F = \frac{C_{i^w} \cap C_{j^{w+1}}}{(C_{i^w} \cup C_{j^{w+1}})/2} \tag{11}$$

where C_{i^w} is the cluster at window w and $C_{j^{w+1}}$ is the cluster at window $w+1$.

Using Eq. (11), we compute the survival factors of all pairs of related clusters from clustering patterns. Then we estimate the survival level of all clustering patterns among time windows. Survival and hazard functions are used for estimating the survival probability of clusters among different windows. The survival and hazard functions are described below.

a: SURVIVAL FUNCTION

The survival function is the probability of survival as a function of time. It gives the probability that the survival time of an individual exceeds a certain value.

Let $T \geq 0$ have a pdf $f(t)$ and cdf $F(t)$. The survival function takes on the form, $S(t) = P\{T > t\}$. The survival function for a continuous distribution, $S(t)$, is the complement of the cumulative distribution function:

$$S(t) = P\{T \geq t\} = 1 - F(t) = \int_t^\infty f(x)dx \tag{12}$$

$S(t)$ is positive and rages from 0 to 1. It is defined as $S(0) = 1$ and as t approaches ∞, $S(t)$ approaches 0. Survival curve describes the relationship between the probability of survival and time.

b: HAZARD FUNCTION

The hazard function gives the instantaneous failure rate of an individual conditioned on the fact that the individual survived until a given time. A characterization of the distribution of

T is given by the hazard function, this rate of occurrence of the event, defined as:

$$h(t) = \lim_{dt \to 0} \frac{P\{t \leq T < t + dt | T \geq t\}}{dt} \tag{13}$$

where a cluster has survived for a time t and we desire the probability that it will not survive for an additional time dt. The hazard function is also defined as follows:

$$h(t) = \frac{f(t)}{S(t)} = \frac{f(t)}{1 - F(t)} \tag{14}$$

The hazard function estimates the survival probability of clusters among time windows.

The cluster survival model of concept drift in load profile data is presented in **Algorithm** 2.

Algorithm 2 CSCD: Cluster Survival Model of Concept Drift in Load Profile Data

Input: W load profile data windows, $\{\mathcal{X}\}_{w=1}^{W}$
Output: Sustaining, fading and emerging patterns and survival level and probability
Initialization: W is the number of windows, $\mathcal{D}$ concept drift score, k^w is number of clusters at w.
Generate clustering results:
Call **Algorithm** 1 to generate the clustering results on $\{\mathcal{X}\}_{w=1}^{W}$;
Compute concept drift score:
for w := *1 to* $(W - 1)$ **do**
  for i := *1 to* k^w **do**
    temp1 = $\mathcal{X}$[[w]][[i]] ;
    Compute probability density $pd1$ on temp1 with Eq. (8) ;
    for j := *1 to* k^{w+1} **do**
      temp2 = $\mathcal{X}$[[w+1]][[j]] ;
      Compute probability density $pd2$ on temp2 with Eq. (8) ;
      Compute $KL[i, j]$ between $pd1$ and $pd2$ using Eq. (10) ;
  $\mathcal{D}$[[w]] = KL ;
Estimate related clusters:
If $\mathcal{D}_{C_{a^w}, C_{b^{w+1}}}$ is smaller than other combinations between w and $w + 1$, where $a^w \subset k^w$ and $b^{w+1} \subset k^{w+1}$. C_{a^w} and $C_{b^{w+1}}$ are related clusters between windows w and $w + 1$
Categorize the clustering patterns:
for w := *1 to* $(W - 1)$ **do**
  $\mathcal{D}[[w]] = \mathcal{D}[[w]][1 : k^w, 1 : k^{w+1}]$;
  if$(\exists(\min \mathcal{D}[[w]][s^w, j = 1 : k^{w+1}]) \Lambda \min \mathcal{D}[[w]][i = 1 : k^w, s^{w+1}]))$;
  C_{s^w} is a sustaining cluster and estimate sustaining pattern with related clusters of C_{s^w} ;
  else if$(\exists(\min \mathcal{D}[[w]][f^w, j = 1 : k^{w+1}]) \Lambda \neg \min \mathcal{D}[[w]][i : 1k^w, g^{w+1}]))$;
  C_{f^w} is a fading cluster and estimate fading pattern with related clusters of cluster of C_{f^w} ;
  else$(\neg \min(\mathcal{D}[[w]][i = 1 : k^w, e^{w+1}]))$;
  $C_{e^{w+1}}$ is an emerging cluster and estimate emerging pattern with related clusters of cluster $C_{e^{w+1}}$;
Survival analysis of clustering patterns:
Compute S.F of all pairs of related clusters from Eq. (11) and estimate survival level of patterns ;
Estimate the survival probability with (14) ;

V. EXPERIMENTS

In this section, we present the experimental results of the cluster survival model of concept drift in load profile data. Experimental results demonstrate that the proposed model retrieves different kinds of clustering patterns and their dynamic behaviors among windows.

A. EXPERIMENT SETUP

1) DATASETS

In the experiments, we used two kinds of datasets: synthetic stream datasets and load profile dataset.

a: SYNTHETIC STREAM DATASETS

We generated eight synthetic stream datasets to evaluate the clustering performance of the I-niceStream algorithm and existing state-of-the-art data stream clustering algorithms. The synthetic stream dataset was generated with given number of objects, number of classes, dimensions and percentage of noise objects. Each class was formed by multivariate normal distribution with randomly selected mean and covariance matrix. A new object was added to a class which was chosen with the probability weight. Then, the object was drawn from the multivariate normal distribution with the mean and covariance matrix of the class. Noise objects were generated in a bounding box from uniform distribution. The details of these synthetic stream datasets are summarized in Table 1.

b: LOAD PROFILE DATASET

We collected the load profile data of different manufacturing factories at Guangdong province from January to August 2012. The load profile data contains the power consumption of 21330 manufacturing factories. The power consumption was measured every 15 minute with a smart meter. There are 96 measurements in the load profile data in a day. The volume of load profile data from 21330 manufacturing factories is about 80 GB.

2) PREPROCESSING OF LOAD PROFILE DATA

A large volume of continuous power consumption data is generated with smart meters. We aggregated data from all individual smart meters into a raw load profile data. The raw load profile data contains missing values, noise and anomalies due to data transmission error and incorrect readings of smart meters. Anomaly can be identified with the visualization of raw load profile data. We removed missing

TABLE 1. Characteristics of the synthetic stream datasets.

Number	Datasets	Objects	Features	Noise	Classes
1	SynStream1	500	2	5%	3
2	SynStream2	1000	3	7%	5
3	SynStream3	2000	5	7%	8
4	SynStream4	5000	8	8%	10
5	SynStream5	7000	10	8%	15
6	SynStream6	10,000	12	8%	16
7	SynStream7	12,000	14	10%	18
8	SynStream8	15,000	30	5%	20

TABLE 2. Comparison of clustering results with I-niceStream and other data stream clustering algorithms on synthetic stream datasets.

Datasets	Purity				Rand index			
	Clu-Stream	Den-Stream	Clus-Tree	I-niceStream	Clu-Stream	Den-Stream	Clus-Tree	I-niceStream
SynStream1	0.909	0.844	0.919	**0.926**	0.722	0.937	0.713	**0.970**
SynStream2	0.875	0.818	0.871	**0.880**	0.863	0.801	0.595	**0.952**
SynStream3	0.862	**0.939**	0.936	0.886	0.889	0.740	0.870	**0.977**
SynStream4	0.920	0.854	**0.974**	0.911	0.886	0.938	0.878	**0.997**
SynStream5	0.956	0.954	**0.974**	0.914	0.865	0.926	0.833	**0.999**
SynStream6	0.965	**0.982**	0.975	0.812	0.648	0.920	0.831	**0.984**
SynStream7	**0.966**	0.8933	0.964	0.900	0.548	0.949	0.840	**0.999**
SynStream8	0.952	0.854	0.954	**0.955**	0.062	0.343	0.120	**1.000**

values, noise and anomalies from raw load profile data and obtained load profile data. In the experiments, load profile data from January to August were used as consecutive eight windows. We chose power consumption for fist seven days from each month as a window load profile data. One window load profile data contains 672 measurements.

3) EXPERIMENT SETTINGS

The objective of load profile data partition is the observation of evolving behaviors among consecutive months.

In I-niceStream algorithm, the number of observation points, P, is set as 6. The threshold th is computed as follows. We compute the distance matrix among candidate initial centers and use the distance matrix to build a tree. The threshold th is estimated from height of tree. The height, which makes the tree into two branches, is assigned as the top height of the tree. We choose a set of height values with repeatedly decreasing the top height for assigning a set of numbers of branches from tree. The number of branches for each height is assigned as the number of clusters. Using the number of clusters for each height, we cluster the candidate initial centers and estimate the total within-sum-of-squares from clustering result. One height value is chosen as threshold th so that the next height value does not improve the total within-sum-of-squares. The number of branches, which is obtained with the threshold height value th, is considered as the number of clusters. The most densest candidate initial center in each branch is selected as the initial cluster center. The estimated initial cluster centers are used to cluster the window load profile data. The clustering results are used for further analysis.

To evaluate the clustering accuracy, we compared the clustering results of the I-niceStream algorithm with the clustering results of three state-of-the-art data stream clustering algorithms, including CluStream [11], DenStream [12], and ClusTree [31], on synthetic stream datasets. The clustering results of these algorithms were measured with purity [50] and rand index [51].

B. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present the experimental results of the proposed method on synthetic stream datasets and real-world load profile dataset. First, we present the clustering accuracy of I-niceStream and other data stream clustering algorithms. Then, we present the experimental results on load profile data into two phases. The first phase is the identification of clustering patterns among windows. The demonstration of changing behaviors in power consumption patterns is another phase.

1) PERFORMANCE IN TERMS OF IMPROVEMENT OF CLUSTERING WITH THE I-niceStream ALGORITHM

Table 2 presents the comparison of the proposed I-niceStream algorithm with three state-of-the-art data stream clustering algorithms in terms of clustering accuracy. Purity and rand index are used to measure the clustering accuracy. The results show that the I-niceStream algorithm generated clustering accuracy in purity is better than other algorithms generated the clustering accuracy in purity. We see that the I-niceStream algorithm significantly outperformed other existing algorithms in terms of rand index in all cases. From this results,

TABLE 3. Clustering results on eight window load profile datasets. The value of each window refers the number of factories in the corresponding cluster.

Cluster	January	February	March	April	May	June	July	August
1	4902	6476	5387	6229	4827	4552	4507	4451
2	206	391	28	173	72	110	98	2
3	28	843	30	28	191	30	303	394
4	32	38	704	278	13	2	471	30
5	8	78	38	1646	392	11	27	29
6	706	23	20	459	13	24	35	96
7	462	34	2	1127	21	239	3437	148
8	386	351	149	22	3628	14	26	1010
9	785	4031	19	36	427	23	16	660
10	258	18	364	279	32	13	428	3320
11	20	3	311	49	763	668	191	33
12	3612	1274	192	17	157	922	2	498
13	13	2299	200	15	10	734	22	478
14	1517	1151	3758	619	13	3460	13	14
15	14	17	10	1818	1075	12	1130	18
16	1298		4	4232	818	7	401	1222
17	2649		1465		1385	2009	2110	1303
18	31		879		2029	1492	1507	2001
19	96		1999		223	587	1353	8
20	4		515		25	1527	785	1312
21			953		5	337	165	
22					137	254		
23					17			
24					541			
25					213			

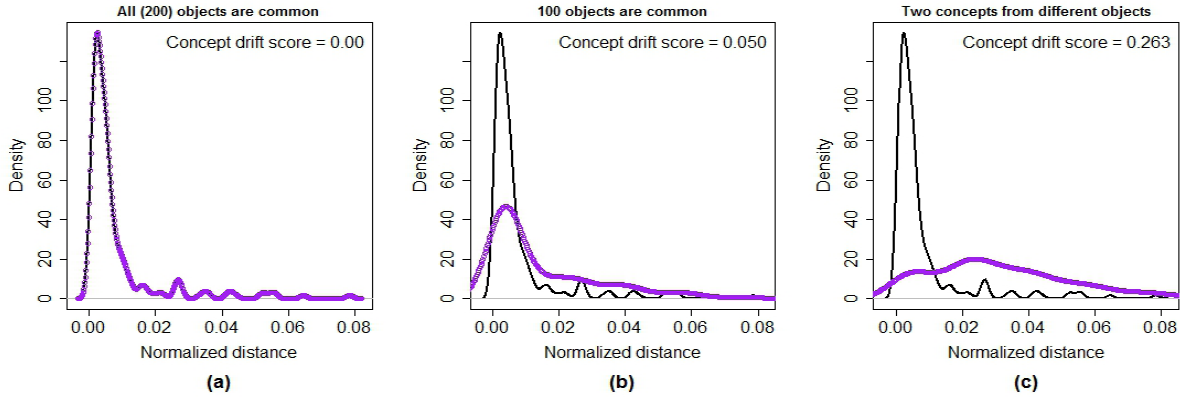

FIGURE 6. Relationship between the concept drift scores and two density curves which generated from two concepts with the different number of common objects from January load profile data. (a) Concept drift score is zero when all objects between two concepts are common. (b) Concept drift score increases when the number of common objects between two concepts decreases. (c) Concept drift score is maximum when two concepts from different sets of objects.

it is observed that the I-niceStream algorithm found clusters are more compact than existing algorithms generated clusters.

2) IDENTIFICATION OF CLUSTERING PATTERNS AMONG WINDOWS

Load profile data streams are divided into eight windows where each window data is segmented into several clusters. A collection of related clusters is represented as a clustering pattern. Patterns are identified and categorized into sustaining, fading and emerging types.

Table 3 shows the clustering results on eight load profile data windows from January to August. A load profile data window is segmented into certain number of clusters where each cluster contains a set of factories. The value 4902 in January window refers that the number of factories in the first cluster on load profile data in January is 4902. It is observed that the clustering results of different monthly load profile data windows are different. From this clustering results, we compute the concept drift score between consecutive windows.

Fig. 6 shows the relationship between the concept drift scores and different number of common objects from two concepts in January load profile data. We compute the concept drift scores (CDS) from three combinations of common objects between two concepts. Fig. 6(a) shows the unique

TABLE 4. Identification of the related clusters based on the smallest concept drift score among windows.

JanFeb	CDS	FebMar	CDS	MarApr	CDS				JulAug	CDS
C_{9^1},C_{2^2}	0.0224	C_{3^2},C_{4^3}	**0.0006**	C_{4^3},C_{6^4}	0.0389	...	...	...	C_{4^7},C_{12^8}	**0.0014**
C_{9^1},C_{3^2}	**0.0135**	C_{3^2},C_{10^3}	0.0058	C_{4^3},C_{7^4}	**0.0061**	...	...	...	C_{4^7},C_{16^8}	0.0071
C_{9^1},C_{4^2}	0.0727	C_{3^2},C_{12^3}	0.0092	C_{4^3},C_{9^4}	0.0122	...	...	...	C_{4^7},C_{20^8}	0.0077

TABLE 5. Different types of clustering patterns among monthly time windows.

Pattern	Type	JanFeb	FebMar	MarApr	AprMay	MayJun	JunJul	JulAug
$P1^1$	Sustaining	C_{1^1},C_{1^2}	C_{1^2},C_{1^3}	C_{1^3},C_{1^4}	C_{1^4},C_{1^5}	C_{1^5},C_{1^6}	C_{1^6},C_{1^7}	C_{1^7},C_{1^8}
$P2^1$	Sustaining	C_{2^1},C_{5^2}	C_{5^2},C_{13^3}	C_{13^3},C_{2^4}	C_{2^4},C_{3^5}	C_{3^5},C_{2^6}	C_{2^6},C_{2^7}	C_{2^7},C_{6^8}
$P4^1$	Sustaining	C_{4^1},C_{4^2}	C_{4^2},C_{5^3}	C_{5^3},C_{11^4}	C_{11^4},C_{20^5}	C_{20^5},C_{3^6}	C_{3^6},C_{6^7}	C_{6^7},C_{11^8}
$P5^1$	Sustaining	C_{5^1},C_{6^2}	C_{6^2},C_{6^3}	C_{6^3},C_{3^4}	C_{3^4},C_{4^5}	C_{4^5},C_{8^6}	C_{8^6},C_{9^7}	C_{9^7},C_{5^8}
$P7^1$	Sustaining	C_{7^1},C_{2^2}	C_{2^2},C_{10^3}	C_{10^3},C_{6^4}	C_{6^4},C_{5^5}	C_{5^5},C_{21^6}	C_{21^6},C_{3^7}	C_{3^7},C_{3^8}
$P9^1$	Sustaining	C_{9^1},C_{3^2}	C_{3^2},C_{4^3}	C_{4^3},C_{7^4}	C_{7^4},C_{11^5}	C_{11^5},C_{11^6}	C_{11^6},C_{4^7}	C_{4^7},C_{12^8}
$P12^1$	Sustaining	C_{12^1},C_{9^2}	C_{9^2},C_{14^3}	C_{14^3},C_{16^4}	C_{16^4},C_{8^5}	C_{8^5},C_{14^6}	C_{14^6},C_{7^7}	C_{7^7},C_{10^8}
$P13^1$	Sustaining	C_{13^1},C_{10^2}	C_{10^2},C_{15^3}	C_{15^3},C_{12^4}	C_{12^4},C_{6^5}	C_{6^5},C_{10^6}	C_{10^6},C_{13^7}	C_{13^7},C_{15^8}
$P14^1$	Sustaining	C_{14^1},C_{14^2}	C_{14^2},C_{18^3}	C_{18^3},C_{14^4}	C_{14^4},C_{9^5}	C_{9^5},C_{13^6}	C_{13^6},C_{20^7}	C_{20^7},C_{9^8}
$P17^1$	Sustaining	C_{17^1},C_{13^2}	C_{13^2},C_{19^3}	C_{19^3},C_{15^4}	C_{15^4},C_{18^5}	C_{18^5},C_{17^6}	C_{17^6},C_{17^7}	C_{17^7},C_{18^8}
$P10^1$	Fading	C_{10^1},C_{8^2}	C_{8^2},C_{11^3}	-	-	-	-	-
$P15^1$	Fading	C_{15^1},C_{15^2}	C_{15^2},C_{9^3}	C_{9^3},C_{13^4}	C_{13^4},C_{14^5}	C_{14^5},C_{5^6}	-	-
$P16^1$	Fading	C_{16^1},C_{12^2}	C_{12^2},C_{21^3}	-	-	-	-	-
$P18^1$	Fading	C_{18^1},C_{7^2}	C_{7^2},C_{3^3}	C_{3^3},C_{9^4}	C_{9^4},C_{23^5}	-	-	-
$P20^1$	Fading	C_{20^1},C_{11^2}	C_{11^2},C_{16^3}	-	-	-	-	-
$P12^3$	Fading	-	-	C_{12^3},C_{10^4}	C_{10^4},C_{19^5}	-	-	-
$P7^5$	Fading	-	-	-	-	C_{7^5},C_{9^6}	C_{9^6},C_{5^7}	-
$P12^5$	Fading	-	-	-	-	C_{12^5},C_{22^6}	C_{22^6},C_{21^7}	-
$P12^3$	Emerging	-	-	C_{12^3},C_{10^4}	C_{10^4},C_{19^5}	-	-	-
$P17^3$	Emerging	-	-	C_{17^3},C_{5^4}	C_{5^4},C_{15^5}	C_{15^5},C_{12^6}	C_{12^6},C_{15^7}	C_{15^7},C_{8^8}
$P7^5$	Emerging	-	-	-	-	C_{7^5},C_{9^6}	C_{9^6},C_{5^7}	-
$P10^5$	Emerging	-	-	-	-	C_{10^5},C_{6^6}	C_{6^6},C_{8^7}	C_{8^7},C_{4^8}
$P12^5$	Emerging	-	-	-	-	C_{12^5},C_{22^6}	C_{22^6},C_{21^7}	-
$P16^5$	Emerging	-	-	-	-	C_{16^5},C_{18^6}	C_{18^6},C_{18^7}	C_{18^7},C_{20^8}
$P17^5$	Emerging	-	-	-	-	C_{17^5},C_{20^6}	C_{20^6},C_{19^7}	C_{19^7},C_{17^8}
$P21^5$	Emerging	-	-	-	-	C_{21^5},C_{16^6}	C_{16^6},C_{14^7}	C_{14^7},C_{19^8}
$P22^5$	Emerging	-	-	-	-	C_{22^5},C_{7^6}	C_{7^6},C_{11^7}	C_{11^7},C_{7^8}

and overlapping density of two distributions where all objects (200) are common between two concepts and the concept drift score is 0.00. Fig. 6(b) shows an amount of density deviation between two distributions where 100 objects are common between two concepts and the concept drift score is 0.050. The Fig. 6(c) shows a large amount of density deviation between two distributions where all objects are drawn from completely two different concepts and the concept drift score is 0.263. Therefore, we are able to estimate the similarity of objects between two clusters by observing the concept drift score of them.

Different number of common objects are contained between a specific cluster at one window and some clusters of next window. Therefore, there is a one-to-many relation based on the concept drift scores of corresponding clusters in consecutive windows. We use smaller concept drift score from clusters combination in consecutive windows to identify the related clusters. Table 4 shows the identification of related clusters among windows. The concept drift score between cluster C_{9^1} in January and cluster C_{3^2} in February is smaller than other combinations. So, the cluster C_{9^1} in January and cluster C_{3^2} in February are related. Then, the cluster C_{3^2} in February is related to cluster C_{4^3} in March. Similarly, we can identify the related clusters until the last window. A continuation of related clusters makes a cluster chain which is assigned as a clustering pattern. As a result, we obtain a complete cluster chain as clustering pattern, from cluster C_{9^1} in January to clusters C_{12^8} in August, $P9^1 = (C_{9^1}, C_{3^2}), (C_{3^2}, C_{4^3}), \ldots, (C_{4^7}, C_{12^8})$.

Table 5 presents different clustering patterns among windows. Each pattern is constructed with the related clusters among windows. The first part of the table presents the sustaining patterns where all patterns are started from January and ended to August. The second part presents the fading patterns where patterns are disappeared in any intermediate window. The third part presents the emerging patterns where patterns are appeared in any intermediate window. These patterns are used to demonstrate the changing behavior of load profile over time.

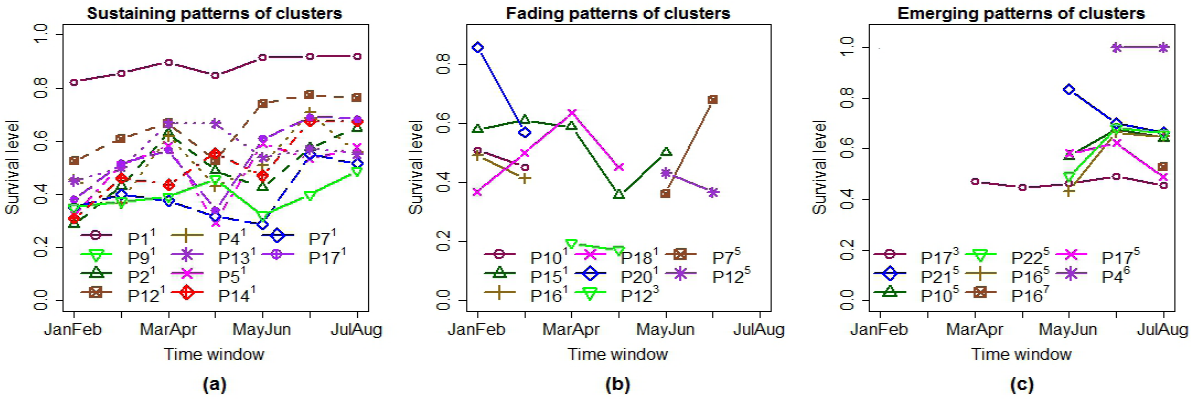

FIGURE 7. Clustering patterns and their survival levels among windows. (a) Sustaining patterns of clusters from January to August. (b) Fading patterns which are ended before the ultimate window. (c) Emerging patterns which are appeared at any intermediate window.

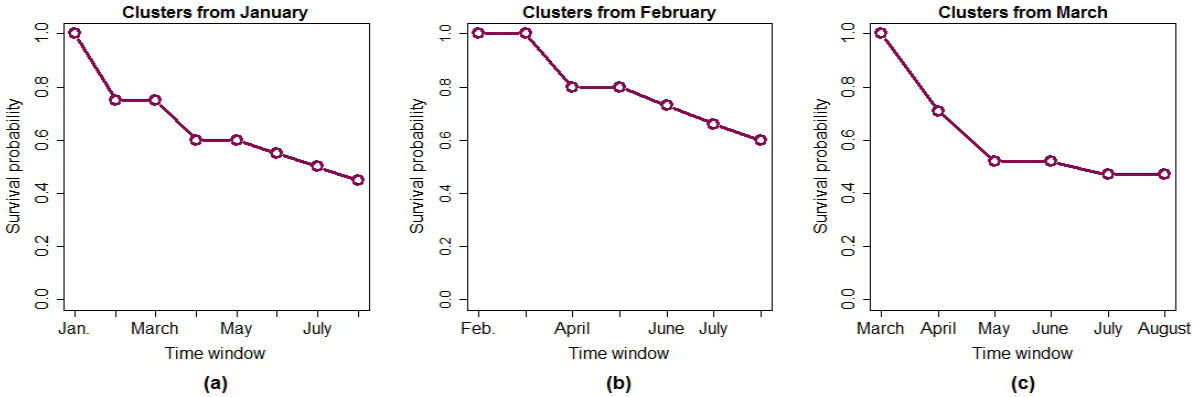

FIGURE 8. Survival probability of clusters among windows. (a) Survival probability of clusters from January to August. (b) Survival probability of clusters from February to August. (c) Survival probability of clusters from March to August.

3) CHANGING BEHAVIORS OF POWER CONSUMPTION PATTERNS

We present the dynamic behaviors of clusters and clustering patterns and their statistical distribution over time windows.

Fig. 7 shows the survival levels of different clustering patterns among windows. The sustaining patterns present the consistent proportion of survival levels in the whole time period. An interesting trend of patterns in Fig. 7(a) is observed that most of the patterns are going to increase their survival levels from January to March and decrease them from March to April. The load profile data was collected from Gaungdong province in China in 2012. The 22th January was the Chinese new year and from the 23th January to the 06th February were the vacation for spring festival in 2012. A large number of factories decreased their production rate in the vacation and these factories are merged into existing clusters in this time. In March, several new patterns are emerged and the existing factories are distributed into different kinds of new clusters according to their production. So, survival level going down from March to April. The survival level rises from April to May then it continues a steady level. Fig. 7(b) shows the fading clustering patterns which are short length patterns and some of them are appeared in a window and disappeared to next window. The survival levels of these patterns are comparatively small. We can assume that the load profiles belong to fading patterns consume low level of energy. Fig. 7(c) shows the emerging patterns which are appeared in any intermediate window and reached to the last window. Most of the patterns in this category are maintaining the raising survival levels.

Fig. 8 shows the survival probability of clusters from February to March windows. The survival curve demonstrates the trend of cluster dropping over time. Fig. 8(a) shows the survival probability of clusters which are generated in January. We see that the survival probability is approximately 75% (drop out 25%) in February and continues the same value in March, 60% in April and May, and 50% with slightly goes down in August. The survival probability of clusters which are generated in February in Fig. 8(b) maintains the similar rate of Fig. 8(a). The survival probability of clusters which are generated in March is decreasing sharply until May and maintains the survival proportion 50% for next windows in Fig. 8(c). We can observe that the cluster dropping

TABLE 6. Representative factories for different lengths of patterns among windows.

Pattern length	Pattern type	Starting window	Ending window	Representative factories
8 months	Sustaining	January	August	Electrical equipments, Communication and electronics equipments, Plastic products, Fabricated metal products, Textiles.
6 months	Fading	January	June	Dedicated equipments, Communication and electronics equipments, Plastic products, Textiles, Rubber products.
5 months	Fading	January	May	Textiles, Crafts products, Plastic product, General-purpose equipment.
4 months	Emerging	May	August	Textiles, Communication and electronics equipments, Electrical equipments, Rubber products, Culture sports goods.
3 months	Emerging	May	July	Paper products, Fabricated metal products, Rubber products, Plastic product.

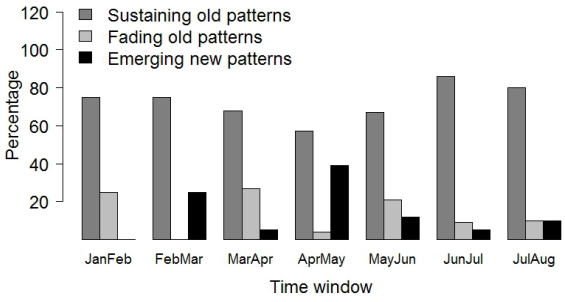

FIGURE 9. A comparison for measuring the percentage among three clustering patterns (sustaining, fading and emerging) in consecutive windows from January to August.

rate from January to February, March to April, and April to May are higher than any other consecutive months.

Table 6 describes the representative factories of different patterns over time. These patterns are distinguished with several lengths such as 8 months, 6 months, and 3 months among windows. We explore the representative load profiles of each kind of patterns. The representative factories of long length, 8 months, patterns are electrical equipments, communication and electronics equipments, plastic product, fabricated metal products and textiles whereas the representatives factories of short length patterns are paper product, rubber products, fabricated metal products, and culture and sports goods. Clusters appeared only at one window are not treated as a complete pattern but they have important contribution for measuring rate of the new concepts.

Fig. 9 shows a comparison of three kinds of clustering patterns over time windows. In the period from January to February (JanFeb), the percentage of fading patterns is smaller than the old sustaining patterns. Several new patterns are emerged and the percentage of sustaining patterns is remained fixed in the period between February and March. The reason of this changing behavior of patterns is the long vacation of Chinese new year and spring festival from January to February. A lot of production factories is closed during the vacation and a set of new production activities is started after vacation. In the period between March and April, the percentage of sustaining patterns is decreased due to the faded of the existing patterns and the percentage of emerging patterns is also decreased. In period April to May, the percentage of sustaining patterns is bit decreased and the percentage of fading patterns is sharply declined whereas the percentage of emerging patterns is rapidly developed because of high temperature in summer and more orders from local and overseas. In the period between May and June, the percentage of new emerging patterns is decreased and old sustaining and fading patterns are improved. The percentage of sustaining patterns is reached maximum and the percentage of emerging patterns is declined between June and July due to the summer vacation. The last time period maintains approximately same rate of previous time period.

Fig. 10 shows the percentage of power consumption among different representative factories from three clustering patterns over time. We explore five representative manufacturing factories, including electrical equipments, communication and electronics equipments, plastic product, fabricated metal products, and textiles, in different proportions from sustaining clustering patterns in Fig. 10(a). Fig. 10(b) shows the energy consumption ratio of individual factories from fading clustering patterns. Some of the significant factories for fading patterns are fabricated metal products, rubber products, paper products and food manufacturing products. We see that plastic products, textiles, and culture and sports goods are representative factories for emerging patterns in Fig. 10(c).

Finally, we present the power consumption behaviors of the three representative factories from the three clustering patterns in Fig. 11. We see that the communication and electronics equipments based factory is the most stable one from sustaining patterns in Fig. 10(a). Similarly, we observe that the fabricated metal products and plastic products are two major contributing factories from fading pattern in Fig. 10(b) and emerging pattern in Fig. 10(c), respectively. From Fig. 11, we observe the three scenarios of the mentioned representative factories from February to April: (i) Communication and

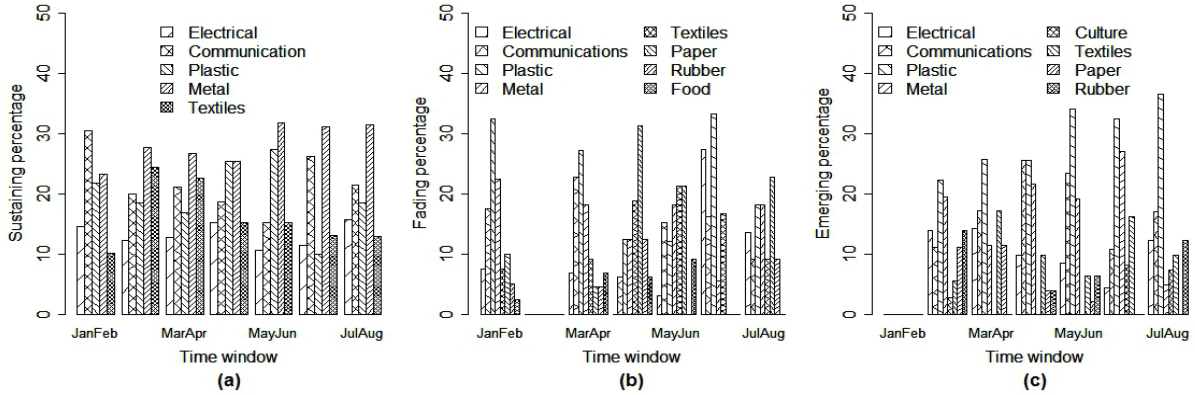

FIGURE 10. Power consumption ratio among representative factories of three clustering patterns from January to August. (a) Sustaining patterns. (b) Fading patterns. (c) Emerging patterns.

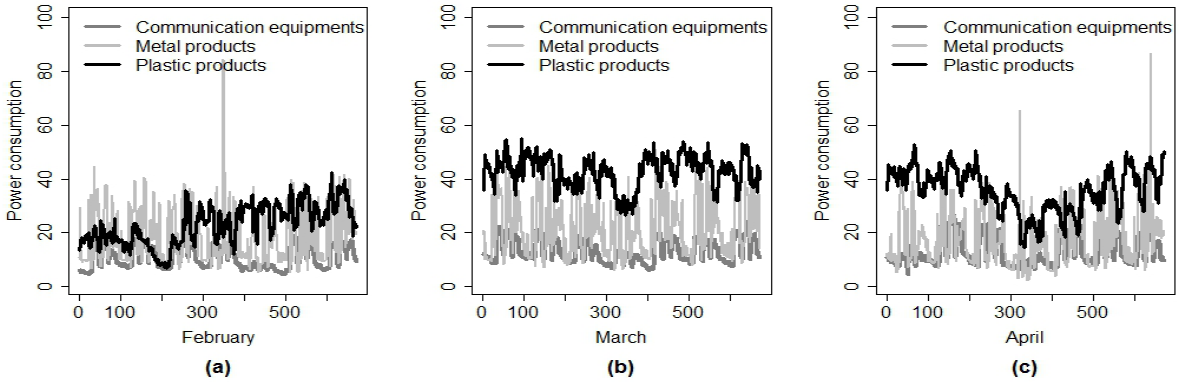

FIGURE 11. A comparison of power consumption behaviors among the three representative factories from the three clustering patterns.

electronics equipments based factory consumes a stable level of energy as a representative load profile in sustaining pattern. (ii) Fabricated metal products based factory consumes an inconsistent energy level as a representative load profile in fading pattern. (iii) Plastic product based factory consumes a significantly increasing level of energy as a representative load profile in emerging pattern. Nowadays, plastic is increasingly used as a substitute material of steel, aluminum, wood, and other materials in building and home appliances. The dynamic power consumption behavior from individual load profile is also important for understanding the power consumption demand in different industrial sectors.

Plastic products, fabricated metal products, and textiles based industries are contributed significantly for changing patterns due to weather, temperature, labor-intensive and public vacation. In addition, small factories belonged some clustering patterns are faded due to insufficient production orders and other production disturbances.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the cluster survival model of concept drift for identifying the clustering patterns and retrieving the dynamic behaviors of clustering patterns from load profile data. The window load profile data is investigated with the new data stream clustering algorithm I-niceStream for generating clustering results. We use a modified KL divergence to perform concept drift detection by estimating clustering patterns from clustering results. These clustering patterns are categorized into three kinds, including sustaining, fading and emerging. The survival probability and survival level of clusters and clustering patterns, respectively, are analyzed among time windows. The statistical distribution of representative load profiles from clustering patterns is also estimated.

Experimental results on load profile data have shown that the new method is able to estimate different types of clustering patterns and explore some representative manufacturing factories with interesting characteristics. Moreover, the I-niceStream algorithm outperformed other data stream clustering algorithms in terms of clustering accuracy.

The current method will be extended with semi-supervised clustering ensemble framework for estimating the dynamic patterns of load profile data in details.

ACKNOWLEDGMENTS

The authors are grateful to Dr. Jianfei Yin for his suggestions at the regular group discussion. Dr. Yin is an Associate Professor, College of Computer Science and Software Engineering, Shenzhen University, China.

REFERENCES

[1] C. Kartsonaki, "Survival analysis," *Diagnostic Histopathol.*, vol. 22, no. 7, pp. 263–270, 2016.

[2] V. T. Farewell, "Mixture models in survival analysis: Are they worth the risk?" *Can. J. Statist.-Revue Canadienne De Statistique*, vol. 14, no. 3, pp. 257–262, 1986.

[3] E. Glaab, A. Baudot, N. Krasnogor, R. Schneider, and A. Valencia, "Enrichnet: Network-based gene set enrichment analysis," *Bioinformatics*, vol. 28, no. 18, pp. i451–i457, 2012.

[4] S. Nadarajah, "Reliability for some bivariate gamma distributions," *Math. Problems Eng.*, vol. 2005, no. 2, pp. 151–163, 2005.

[5] H. Masanja *et al.*, "Child survival gains in Tanzania: Analysis of data from demographic and health surveys," *LANCET*, vol. 371, no. 9620, pp. 1276–1283, 2008.

[6] L. Zheng and Y.-T. Chang, "Risk assessment model of bottlenecks for urban expressways using survival analysis approach," *Transp. Res. Procedia*, vol. 25, pp. 1544–1555, Jul. 2017.

[7] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[8] T. Boongoen and N. Iam-On, "Cluster ensembles: A survey of approaches with recent extensions and applications," *Comput. Sci. Rev.*, vol. 28, pp. 1–25, May 2018.

[9] X.-L. Zhang, "Convex discriminative multitask clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 28–40, Jan. 2015.

[10] X.-L. Zhang, "Multilayer bootstrap networks," *Neural Netw.*, vol. 103, pp. 29–43, Jul. 2018.

[11] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proc. 29th Int. Conf. Very Large Data Bases (VLDB)*. San Mateo, CA, USA: Morgan Kaufmann, 2003, pp. 81–92.

[12] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proc. SIAM Int. Conf. Data Mining*, 2006, pp. 328–339.

[13] A. Qahtan, B. Alharbi, S. Wang, and X. Zhang, "A PCA-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams," in *Proc. 21th ACM SIGKDD ICKDDM*, 2015, pp. 935–944.

[14] X. Wu, P. Li, and X. Hu, "Learning from concept drifting data streams with unlabeled data," *Neurocomputing*, vol. 92, pp. 145–155, Sep. 2012.

[15] J. de A. Silva, E. R. Hruschka, and J. Gama, "An evolutionary algorithm for clustering data streams with a variable number of clusters," *Expert Syst. Appl.*, vol. 67, pp. 228–238, Jan. 2017.

[16] J. Xu, G. Wang, T. Li, W. Deng, and G. Gou, "Fat node leading tree for data stream clustering with density peaks," *Knowl.-Based Syst.*, vol. 120, pp. 99–117, Mar. 2017.

[17] S. Ren, B. Liao, W. Zhu, Z. Li, W. Liu, and K. Li, "The gradual resampling ensemble for mining imbalanced data streams with concept drift," *Neurocomputing*, vol. 286, pp. 150–166, Apr. 2018.

[18] F. Rodrigues, J. Duarte, V. Figueiredo, Z. Vale, and M. Cordeiro, "A comparative analysis of clustering algorithms applied to load profiling," in *Machine Learning and Data Mining in Pattern Recognition*. Berlin, Germany: Springer, 2003, pp. 73–85.

[19] V. Figueiredo, F. Rodrigues, Z. Vale, and J. B. Gouveia, "An electric energy consumer characterization framework based on data mining techniques," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 596–602, May 2005.

[20] G. Chicco, R. Napoli, and F. Piglione, "Comparisons among clustering techniques for electricity customer classification," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 933–940, May 2006.

[21] J. Kwac, J. Flora, and R. Rajagopal, "Household energy consumption segmentation using hourly data," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 420–430, Jan. 2014.

[22] G. Chicco, "Overview and performance assessment of the clustering methods for electrical load pattern grouping," *Energy*, vol. 42, no. 1, pp. 68–80, Jun. 2012.

[23] I. Khan, J. Z. Huang, M. A. Masud, and Q. Jiang, "Segmentation of factories on electricity consumption behaviors using load profile data," *IEEE Access*, vol. 4, pp. 8394–8406, 2016.

[24] F. Biscarri, I. Monedero, A. García, J. I. Guerrero, and C. León, "Electricity clustering framework for automatic classification of customer loads," *Expert Syst. Appl.*, vol. 86, pp. 54–63, Nov. 2017.

[25] F. G. da Costa, R. A. Rios, and R. F. de Mello, "Using dynamical systems tools to detect concept drift in data streams," *Expert Syst. Appl.*, vol. 60, pp. 39–50, Oct. 2016.

[26] T. S. Sethi and M. Kantardzic, "On the reliable detection of concept drift from streaming unlabeled data," *Expert Syst. Appl.*, vol. 82, pp. 77–99, Oct. 2017.

[27] Y. Zhang, G. Chu, P. Li, X. Hu, and X. Wu, "Three-layer concept drifting detection in text data streams," *Neurocomputing*, vol. 260, pp. 393–403, Oct. 2017.

[28] F. Cao, J. Z. Huang, and J. Liang, "Trend analysis of categorical data streams with a concept change method," *Inf. Sci.*, vol. 276, pp. 160–173, Aug. 2014.

[29] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, pp. 103–114, Jun. 1996.

[30] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2007, pp. 133–142.

[31] P. Kranen, I. Assent, C. Baldauf, and T. Seidl, "Self-adaptive anytime stream clustering," in *Proc. 9th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2009, pp. 249–258.

[32] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. De Carvalho, and J. Gama, "Data stream clustering: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, p. 13, 2013.

[33] J. de Andrade Silva and E. R. Hruschka, "Extending K-means-based algorithms for evolving data streams with variable number of clusters," in *Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops*, Dec. 2011, pp. 14–19.

[34] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.

[35] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Statist. Methodol.*, vol. 63, no. 2, pp. 411–423, 2004.

[36] I. Khan, J. Z. Huang, and K. Ivanov, "Incremental density-based ensemble clustering over evolving data streams," *Neurocomputing*, vol. 191, pp. 34–43, May 2016.

[37] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 443–448.

[38] I. Khan, J. Z. Huang, Z. Luo, and M. A. Masud, "CPLP: An algorithm for tracking the changes of power consumption patterns in load profile data over time," *Inf. Sci.*, vol. 429, pp. 332–348, Mar. 2018.

[39] F. McLoughlin, A. Duffy, and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data," *Appl. Energy*, vol. 141, pp. 190–199, Mar. 2015.

[40] G. Vegas-Sánchez-Ferrero *et al.*, "Gamma mixture classifier for plaque detection in intravascular ultrasonic images," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 61, no. 1, pp. 44–61, Jan. 2014.

[41] G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and J. M. Sanches, "A gamma mixture model for IVUS imaging," in *Multi-Modality Atherosclerosis Imaging and Diagnosis*. New York, NY, USA: Springer, 2014, pp. 155–171.

[42] A. R. Webb, "Gamma mixture models for target recognition," *Pattern Recognit.*, vol. 33, no. 12, pp. 2045–2054, 2000.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.

[44] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1983.

[45] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proc. 2nd Int. Symp. Inf. Theory*, B. N. Petrov and F. Csaki, Eds. Budapest, Hungary: Akadémiai Kiadó, 1973, pp. 267–281.

[46] N. Sugiura, "Further analysts of the data by Akaike' s information criterion and the finite corrections," *Commun. Statist.-Theory Methods*, vol. 7, no. 1, pp. 13–26, 1978.

[47] H. Kurata and P. Tarazaga, "The cell matrix closest to a given Euclidean distance matrix," *Linear Algerbra Appl.*, vol. 485, pp. 194–207, Nov. 2015.

[48] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[49] D. Liu, D.-M. Sun, and Z.-D. Qiu, "Feature selection for fusion of speaker verification via maximum Kullback–Leibler distance," in *Proc. 10th Int. Conf. Signal Process. (ICSP)*, Oct. 2010, pp. 565–568.

[50] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.

[51] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.

MD ABDUL MASUD received the M.Sc. degree in information and communication engineering from Islamic University, Bangladesh, in 2006. He is currently pursuing the Ph.D. degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is also an Associate Professor with the Department of Computer Science and Information Technology, Patuakhali Science and Technology University, Patuakhali, Bangladesh. His research interests include clustering, data stream clustering, and semi-supervised clustering algorithms for unlabeled data.

JOSHUA ZHEXUE HUANG was born in 1959. He received the Ph.D. degree from the Royal Institute of Technology, Sweden, in 1993. He is currently a Distinguished Professor with the College of Computer Science & Software Engineering, Shenzhen University. He is also the Director of Big Data Institute and the Deputy Director of the National Engineering Laboratory for Big Data System Computing Technology. He has published over 200 research papers in conferences and journals. His main research interests include big data technology and applications. In 2006, he received the first PAKDD Most Influential Paper Award.

He is known for his contributions to the development of a series of k-means type clustering algorithms in data mining, such as k-modes, fuzzy k-modes, k-prototypes, and w–k-means, which are widely cited and used, and some of them have been included in commercial software. He has extensive industry expertise in business intelligence and data mining and has been involved in numerous consulting projects in Australia, Hong Kong, Taiwan, and mainland China.

MING ZHONG is currently the Pengcheng Scholar and a Distinguished Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. He has published over 100 high-quality papers in leading conference and journals. He engages in research on the Internet of Things, cloud computing, and data mining.

XIANGHUA FU received the M.Sc. degree from Northwest A&F University, Yangling, China, in 2002, and the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2005. He led a project of the National Natural Science Foundation and several projects of the Science and Technology Foundation of Shenzhen City. He is currently a Professor and the Post-Graduate Director with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include machine learning, data mining, information retrieval, and natural language processing.