This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

Feature Selection for Segmentation of Factories Using Load Profile Data

Imran Khan[1], Joshua Zhexue Huang[2], M.A. Masud[2], and Qingshan Jiang[1]
[1] Shenzhen Key Laboratory of High Performance Data Mining.
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences,
Shenzhen 518055, China.
[2] College of Computer Science and Software Engineering, Shenzhen University,
Shenzhen 518060, China.
Corresponding author: Imran Khan (imran.khan@siat.ac.cn)

In recent years, the new achievements in the field of technology and data science allowed to gather detailed and well-structured information about electricity consumption behaviors of industrial enterprises. Such type of information can find numerous applications in the power distribution industry. The utilities often use the data from contracts to assign each industrial customer a class label according to his type defined in predetermined industry segmentation. Such kind of fixed-chart segmentation is not able to satisfy the needs of modern enterprises for flexible and dynamic determination of production modes. In the present work, we address this problem by proposing a new method for segmentation of various kinds of factories based on their electricity consumption patterns represented in load profile data. It exploits the evolution-based characteristics of smart meter data of multiple types of factories to remove irrelevant features. We use data visualization to estimate the number of clusters and apply the well-known k-means algorithm on filtered data to generate segmentation. Experimental results on real load profile data collected with smart meters from manufacturing industries in Guangdong province of China have shown that the new clustering approach produced the meaningful segmentation of factories that reflect production operations.

INDEX TERMS Segmentation, Power Consumption, Smart Grid, Load Profiles, Feature Selection

I. INTRODUCTION

The extensive application of smart meters as a part of smart grids provides enormous opportunities, but it, however, also leads to challenges for power distribution operators. Significant investments in the Advanced Metering Infrastructure (AMI) enable the smart grids to be well monitored, controlled, managed and optimized, and customers to be well serviced. On the other hand, power providers face more challenges in handling big data due to the need to satisfy a list of business imperatives. Such a list includes reliability and efficiency, safety and security, profitability, and implementation of evolving intelligent grid that can serve a heterogeneous customer base. This list of business essentials could appear overwhelming, particularly in the context of efficient integration of big data content and solutions.

Smart metering data can often show substantial changes in trends over time. Therefore, it is useful to understand, visualize and diagnose the evolution of these patterns. Such data often poses challenges as huge size, irrelevant dimensionality, skewed distribution, sparsity and seasonal variations. The presence of irrelevant dimensions could arguably lead to degraded performance and increased computation time of the most learning algorithms. Irrelevant dimensions of AMI data are a source of inconsistencies and inefficiency that make it difficult to discover the production modes of an industry sector on the basis of power consumption behavior. Consequently, such dimensions may lead to poor decisions with an adverse impact on the reliable and economic grid operation and planning. To the best of our understanding, no research or industrial community considered the evolution-based characteristic of smart grid data to obtain strongly correlated data subset for defining business process operations.

In this paper, we suggest a solution to this problem by presenting a new method for segmentation of different types of factories based on their electricity consumption patterns represented in load profile data. It utilizes an innovative concept called density estimation to discover the irrelevant dimensions of AMI data in an efficient manner. Our method detects the local densities in different special regions (individual dimensions) of the data. When computing the local densities, we also include those of temporal regions that are the combination of subsequent dimensions. We classify the local densities into two classes, the high-density one represented by

1 and the low-density one represented by 0. We use a binary matrix to represent the density classes of factories at different time slots. From the binary matrix, we compute the similarity of density vectors between every two subsequent time slots and identify the irrelevant dimensions of density vectors to be deleted from the time series data. We finally use a visualization approach to determine the total number of clusters and use the k-means algorithm to cluster the filtered data to generate factory segmentation results.

Experimental results have been obtained using smart meter data sampled at 15-minute intervals, collected from manufacturing industries in Guangdong province of China. According to results, the new feature selection algorithm outperformed the well-known state-of-the-art algorithms. The new clustering approach produced the meaningful segmentation of factories that reflect production operations. Such segmentation can be used in utility applications such as the design of variable rates.

The rest of the present work is organized as follows. Section 2 presents smart grid data. In Section 3, a detailed description of the proposed method is presented. Section 4 shows a thorough evaluation of clustering results on a real-world dataset. Conclusions are given in Section 5.

II. RELATED WORK

Nowadays, the increasing availability of energy consumption data allows unique opportunities in designing segmentation strategies of industrial energy use to support smart grid data applications. The introduction of smart meters has driven studies on high-resolution time series modeling and customer clustering.

The large size of smart meter data suggests that new approaches are needed to maintain demand response, design programs for improving the energy efficiency and ensure efficient customer targeting [1], [2]. In spite of the high number of clustering algorithms available in the literature e.g. automated variable weighting in k-means (W-k-means) [3], clustering with fastmap projection [4], swarm intelligence based clustering [2]. The self-organizing maps [5], k-means [6], and hierarchical clustering [7] are often applied for load pattern mining. Though, existing algorithms do not focus on the identification of characteristics of clustering of customers. They extract load profiles from electricity data by considering the global properties of power consumption patterns, rather than undertaking the local ones. Moreover, they always operate over all feature spaces of an input dataset to learn as much as possible, which degrades the performance due to the lack to discover the hidden patterns in noisy and irrelevant dimensions. The scalability is another significant issue of existing algorithms for load profiling.

Feature selection plays an important role to improve the quality of clustering in machine learning and data mining. The Feature selection approaches can be classified into wrapper and filter techniques. Wrapper techniques [8], [9] wrap feature selection around the learning process and explore for features which improve the performance of the learning task. Filter methods [10]–[12], on the other hand, investigate the intrinsic characteristics of the data and select highly-ranked features according to some criterion before starting the learning task. Wrapper methods are computationally more expensive than filter methods as they depend on deploying the learning models several times until a subset of relevant features is found.

Only a few of the current filter methods are unsupervised. The Laplacian score [11] is measured to reflect its locality preserving power. This approach is based on the observation that two data points are probably related to the same subject if they are close to each other. In fact, in various learning problems such as classification, the local structure of the data space is more important than the global structure. The Sparse K-Means score [12] uses a lasso-type penalty to select the features. This framework to develop simple methods for sparse K-means for feature selection. Data variance [10] might be the simplest unsupervised evaluation of the features. The variance along a dimension reflects its representative power. Although the data variance criteria find features that are helpful for describing data, there is no reason to expect that these features must be helpful for discriminating between data in different classes.

We addressed these problems by proposing a new feature selection technique for smart meter data to enhance the performance of clustering algorithms. The obtained results are useful to efficiently adopt the strategies by utilities to increase the business gain.

III. Smart Grid Data

AMI deployment is a significant trend in the electricity distribution industry. The enormous volume of generated AMI data is associated with two fundamental challenges: to retain the data and extract business value from it. Such challenges make AMI prime candidates for the application of big data processing and analytics. Fig. 1 shows the typical AMI architecture with multiple smart meters.

Electricity meters have been provided with microprocessors and storage units that allow for intelligent functions and turn them into smart meters. They also

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

ensure bi-directional communication and remote operating capabilities. A large number of smart meters have been deployed in different residential and commercial buildings. In the industry sector, they are usually installed at factory sites to record the data about the power consumption of ongoing production activities.

A. Data collection

Typically, smart meters generate readings at small intervals of 15, 30, or 60 minutes. Smart meter data is collected and forwarded via a local area network (LAN) to the data collection center. In terms of data processing, some tasks could be carried out at the regional collection centers. Often, the data are transferred to central collection centers via a wide area network (WAN). Deploying a substantial number of smart meters and connecting them to collection centers is an expensive and time-consuming process that often takes many years.

For the goal of the present research, we obtained the electricity consumption data of a manufacturing center located in the Pearl River Delta (PRD) Region, Guangdong Province of China. This province is an important industrial center, where the volume of the smart meter data for one month collected from the factories of one city amounts to approximately 80 GB. There are different types of factories in the PRD region, and each one has many installed smart meters. Each smart meter records power consumption at 15-minute intervals and sends measured information back to the collection center. The collection center maintains a text file for each smart meter that contains the following attributes: date, timestamp, a unique identifier for the meter that produced the reading, and consumption value (kW).

The data collection task usually involves a costly and time-consuming process. We obtained data from 21330 smart meters sampled at 15-minute intervals of the year 2012 in the form of text files. We imported each file into a raw dataset with n rows and d dimensions. Each dimension of a raw dataset denotes a time slot, and each row represents a particular factory with its power consumption at multiple sequential time slots.

B. Data exploration

A load profile provides information about electricity consumption for a given factory over a given period, e.g. a day or month, at a particular frequency, typically every 15 minutes.

Our target was to extract production mode of multiple types of factories based on their daily power consumption behavior. Therefore, we need to analyze one-day data for the analysis. However, visual analysis of all individual load profile is a difficult and time-consuming process. Thus, we randomly chose and analyzed a few of them to discover the generic types of the load profiles that show abnormal behavior.

Data transmission errors can affect data streams leading to evaluation and simulation problems. The connection between smart meters and data collection centers could be both wired or wireless. Due to the nature of wireless transmission, signal attenuations that affect data transmission could occur. On the other hand, wired channels also are susceptible to equipment and power supply failures, a sudden interruption of lines, etc. Thus, missing values can take place in the data streams.

Fig.2 illustrates some data streams with missing values. For example, stream 1 and 8 in Fig. 2(a) and 2(b), respectively, show periodicity with sudden power consumption falling to zero because of missing values. One significant indication of this issue is stated in [14]. As a rule of thumb, a typical, well-run, large-scale smart meter system misses up to 4 % of the interval usage data that is supposed to record and retrieve each month. For a million meter-system, this amounts to over 28 million missing data intervals per month. The smart grid requires a high level of confidence in the data for its applications.

A recent study by an independent testing group found that 99.91 percent of smart meters are accurate within 0.5 percent [15]. Besides, smart meters are continuously controlled by the responsible authorities to ensure that they are working correctly. The industrial utilities on their side continuously monitor the data transmitted from smart meters to prove that power usage is within the expected limits. If readings show a big deviation from the normal levels, specialists examine the meter. For example, in Fig. 2(a), 2(b), the load profiles 4 and 7, counted from the upper left corner to the right down corner, show a significant difference from other data streams. Moreover, load profiles 7 and 6 exhibit power consumption below zero. Such load profiles may represent that their corresponding smart meters have a technical fault.

C. Segmentation of Load Profiles

The electricity demand of customers varies daily and seasonally. A production plant assembly line begins and ends operation during the whole day and week. During peak times, a tremendous amount of electricity is required (this is the so-called peak load), but a $base\ load$ requirement is needed year-round. Since electricity for industrial consumers cannot be stored, electricity distribution network operator must predict electrical power demands for even the most extreme conditions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

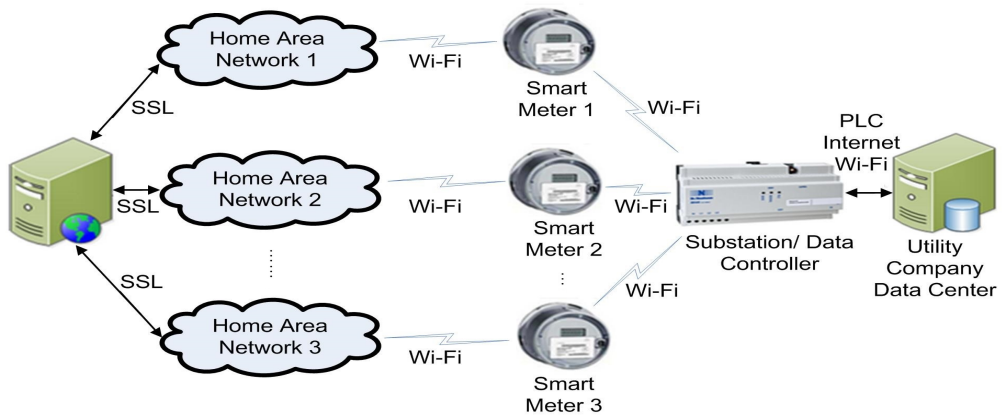
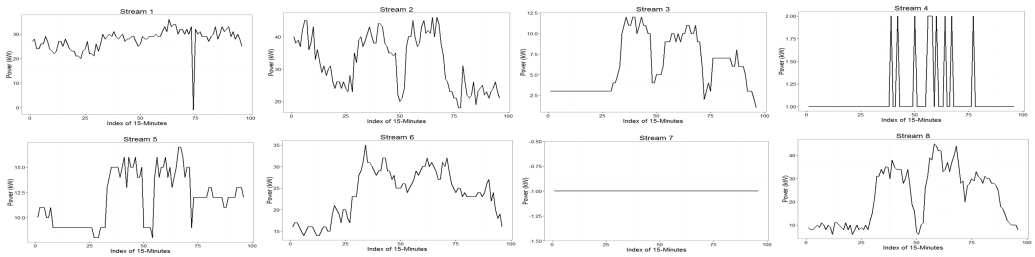
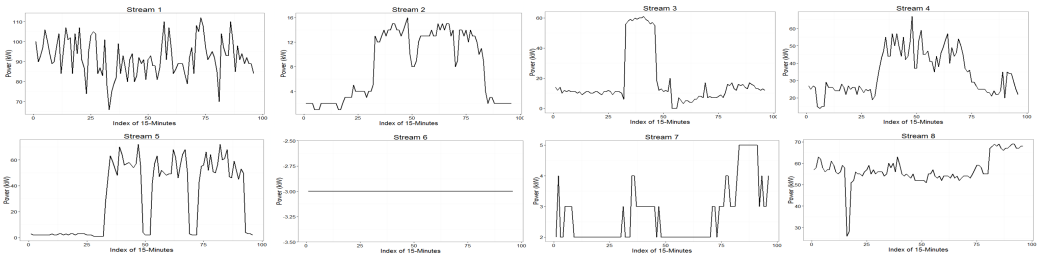


Fig. 1. Typical Smart Metering Architecture [13].



(a) Active Power Data Streams(weekend)



(b) Active Power Data Streams(workday)

Fig. 2. The one day load profiles (power consumption behaviours) of some factories.

(such as high ambient temperature due to hot weather). Consumption depends predominantly on the time of day and the season. The well-defined production modes (such as two-shift mode, three-shift mode or one-day off) on the basis of load profile segments could facilitate the handling of demand and supply.

Data engineering is deemed to be a fundamental problem in the development of smart grid applications. To build models of data, the success of the most clustering techniques hinges on the reliable selection of a small set of highly correlated features. The presence of irrelevant, redundant, and noisy features at the stage of model development could result in a poor clustering performance.

As shown in Fig. 2, the load profiles of some factories show clear daily electricity consumption patterns. These patterns reflect daily production operations of factories and the daily patterns repeat on work days and weekend days. The variation of electricity consumption values at multiple time scales makes smart grid data streams different from other data streams like the stock time series. Furthermore, the variation is caused by many factors, such as production order, weather condition, working hours, price incentives, etc. Therefore, segmentation of load profiles is a challenging task for clustering methods to investigate production modes of factories from load

profile data.

IV. METHODOLOGY

In this section, we present a new method for segmentation of different types of factories based on their electricity consumption patterns represented in load profile data. These electricity consumption patterns represent daily production operations of factories. The proposed method consists of two steps: feature selection and clustering. According to the characteristics of AMI streaming data, we propose a new feature selection method that utilizes evolving characteristics of AMI streaming data.

A. Notations

The electricity consumption data of a factory i is represented as a time series $X_i = x_1, x_2, ..., x_d, ...$, where each x_j is a measurement of electricity consumption at a given time interval, i.e., 15 minutes. Let $\mathbf{X}$ be a set of N time series from N factories. For a given time window with d time slots (intervals), $\mathbf{X}$ is a $N \times d$ matrix $\{x_{i,j}\}$ where $x_{i,j}$ is the measurement of time series i at the jth time slot. Let Y_j be a vertical vector of N elements representing the measurements of N factories at the jth time slot. $\mathbf{X} = \{Y_1, Y_2, ..., Y_d\}$ represents a sequence of d vectors. Let W be a time window of d time slots. $\mathbf{X}$ is a matrix representing N time series $\{X_1, X_2, ..., X_N\}$ with d dimensional attributes $\{Y_1, Y_2, Y_3, ..., Y_d\}$.

Based on the above notation, we have a simple data representation model as shown in Fig.3. The left figure is a data matrix of N time series in a time window of d time intervals. Each column of the matrix represents the distribution of the total electricity consumption at a time interval over N factories as shown in the middle figure. We call this distribution as spatial distribution. The electricity consumption along the neighboring time intervals is called temporal distribution as shown in the right figure. In this work, we use one day as the time window for electricity consumption pattern analysis. The window length is 24 hours starting and ending at midnight. There are 96-time intervals in the time window, i.e., $d = 96$. Using this data model and the electricity consumption distribution concepts, we developed a feature selection method described below.

B. Features selection method

1) Local density estimation:

At a given time slot j, Y_j is a vector representing the electricity consumption distribution of N factories. We use the k-means clustering algorithm to cluster the N factories into $\sqrt{N}$ clusters according to the N measurement values of Y_j. We estimate the distribution density of the factories in cluster k as

$$f_k(x) = \frac{1}{m} \sum_{i=1}^{m} K_h(x - x_i) \tag{1}$$

where k ($1 \leq k \leq \sqrt{N}$) is a cluster number, m is the number of factories in cluster k and $K_h(.)$ is a Gaussian kernel defined as

$$K_h(x - x_i) = \frac{1}{\sqrt{(2\pi)}h} e^{\frac{-(x-x_i)^2}{2h^2}} \tag{2}$$

where h is a smoothing parameter.

Using (1) and spatial density concept, we calculate the spatial density for each factory at each time interval Y_j and produce a spatial density vector D_{sj}. The density estimate is cluster-based, so it is a local spatial density.

Using the temporal distribution concept, we calculate the density change in a small time window h_w that contains neighboring time slots j and $j + 1$ with the spatiotemporal kernel function [16] as

$$K'_{(h_s,h_w)}(Y, t) = \left(1 - \frac{t}{h_w}\right) K_{h_s}(Y) \tag{3}$$

where $K_{h_s}(Y)$ is a Gaussian kernel in (1), h_w is a temporal kernel width and h_s is a spatial kernel width, and t (i.e. t = j) is the arrival time of vertical vector Y_j and $t = j/d$ where $0 < t \leq 1$.

Using (3), we calculate the velocity-density as

$$V_{(h_s,h_w)}(Y, t_j) = \frac{K'_{(h_s,h_w)}(Y_j, t_j) - K'_{(h_s,h_w)}(Y_{j+1}, t_{j+1})}{h_w} \tag{4}$$

where t_j and t_{j+1} indicate the time slots of Y_j and Y_{j+1}, respectively.

For each time slot j, we can use Y_j and Y_{j+1} to calculate its velocity-density vector D_{vj} with (4).

Given the spatial density vector D_{sj} and the velocity density vector D_{vj}, we calculate the spatiotemporal density vector D_{stj} as

$$D_{stj} = \{d_{stj}(i)\} = \{d_{sj}(i) \times d_{vj}(i)\} \tag{5}$$

where $1 \leq i \leq N$.

The spatiotemporal density is a modification of the spatial density by the velocity density. Given a sequence of vectors $Y_1, Y_2, ..., Y_m$, we can calculate a sequence of spatiotemporal density vectors $D_{st1}, D_{st2}, ..., D_{stm}$. The last spatiotemporal density vector D_{stm} is computed by using $D_{v(m-1)}$ to modify D_{sm} because Y_{m+1} is not available. The algorithm to compute the spatiotemporal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

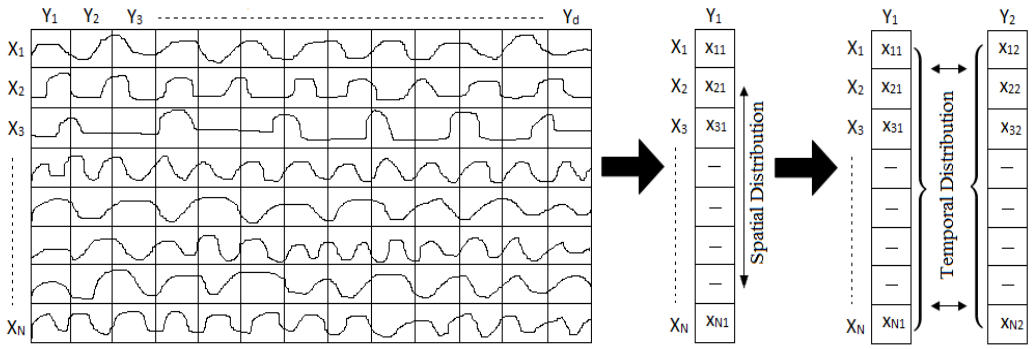


Fig. 3. The time series data of N factories in a time window of d time intervals is represented as an $N \times d$ matrix of the left figure. Each column Y_j is considered as spatial distribution of the total electricity consumption at the jth time interval among N factories as shown in the middle figure. The distribution along the neighboring time intervals is called temporal distribution as shown in the right figure.

density is given in **Algorithm 1**.

Algorithm 1: Local Density Estimation

Input: $X_{N\times d}$
Output: Spatiotemporal density vectors $D_{st1}, D_{st2}, ..., D_{std}$
for *j := 1 to d* **do**
  Select attributes Y_j and Y_{j+1} from $X_{N\times d}$;
  Apply k-means on Y_j using the number of clusters $\sqrt{N}$;
  if *j != d* **then**
    Compute clusters vise spatial density vector D_{sj} of Y_j using (1);
    Compute velocity density vector D_{vj} for Y_j using Y_j and Y_{j+1} using (4);
    Compute D_{stj} for Y_j using D_{sj} and D_{vj} using (5).
  else
    Compute clusters vise spatial density vector D_{sd} of Y_d using (1);
    Compute velocity density vector D_{vd} for Y_d using Y_d and Y_{d-1} using (4);
    Compute D_{std} for Y_d using D_{sd} and D_{vd} using (5).
Spatiotemporal density vectors $D_{st1}, D_{st2}, ..., D_{std}$;

2) Density threshold estimation:
Let $D = \{D_{st1}, D_{st2}, ..., D_{std}\}$ be a sequence of d spatiotemporal density vectors. Each vector contains $\sqrt{N}$ clusters. We compute the average spatiotemporal density value d_x for each cluster and rank the clusters on the average spatiotemporal density values. We plot the average spatiotemporal density values against the order of clusters from the highest average spatiotemporal density values to the lowest ones. Fig. 4 shows examples of four-time slots. We can see that the average spatiotemporal density distributions are different in different time slots. Some time slots have more high average density clusters than others.

We rank the clusters of all time slots on average spatiotemporal density values. Fig. 5 shows the distribution of the average spatiotemporal densities on all time slots. From the aggregated distribution of densities of all clusters, we set a threshold to divide clusters into two classes, i.e., high-density clusters and low-density clusters.

To determine the threshold, we use Minimal Description Length (MDL) principle to divide all clusters into two subsets as used in [17]. Let A be the set of high-density clusters and B the set of low-density clusters. Let l be the cluster in A whose average density is smaller or equal to the average density of any cluster in A but greater than the average density of any cluster in B. Let μ_A and μ_B be the averages of the average cluster densities in A and B, respectively. Cluster l is found by minimizing the code length (CL) of the MDL principle as

$$\begin{aligned} CL_l = log(1+\mu_A) + \sum_{c_i \epsilon A} log(1+|c_i - \mu_A|) \\ -log(1+\mu_B) + \sum_{c_i \epsilon B} log(1+|c_i - \mu_B|) \end{aligned} \quad (6)$$

where c_i is the average density of cluster i in set A or B. The average density c_l of cluster l is used as the threshold to separate high-density clusters from low-density ones. We use Algorithm 2 to minimize the minimum code length CL to find cluster l and threshold c_l.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

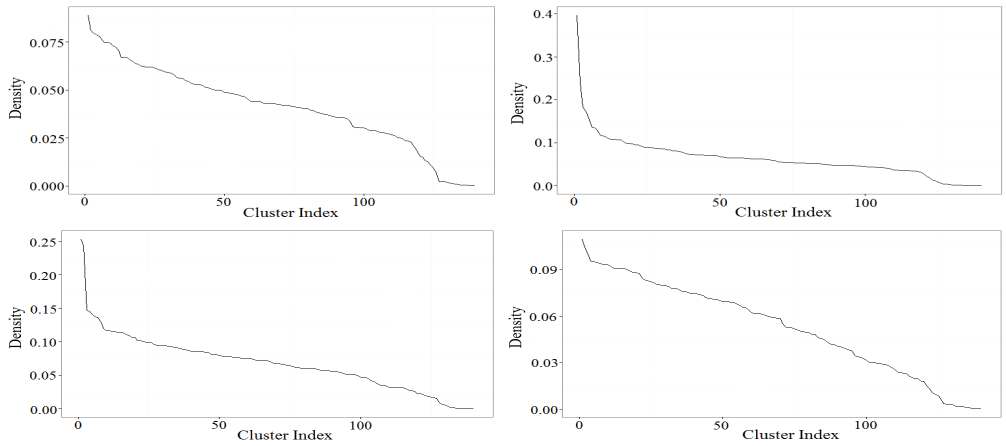


Fig. 4. The average spatiotemporal density distributions of four time slots.

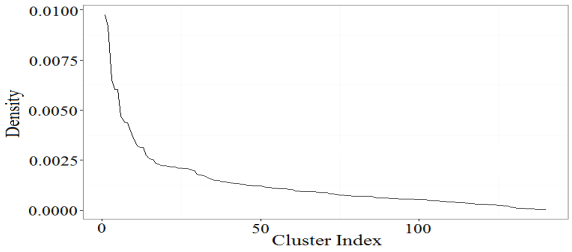


Fig. 5. Aggregated distribution of four time slots in Fig. 4.

3) Detection of irrelevant features:
Given the density matrix $D = (D_{st1}, D_{st2}, ..., D_{std})$ and the density threshold c_l found using Algorithm 2, we compute a binary matrix B as

$$b_{(r_i,d_j)} = \begin{cases} 1, & if \quad d_{(r_i,d_j)} > c_l \\ 0, & Otherwise \end{cases} \tag{7}$$

where r_i and d_j are the i_{th} rows and j_{th} dimension, respectively.

Matrix B classifies densities into two classes with 1 representing high density and 0 representing low density as shown Fig. 6.

Given $B_{N\times d}$, we use Jaccard similarity coefficient to compute the similarity between two time slots Y_i and Y_j as

$$JC(Y_i, Y_j) = \frac{n_{11}}{n_{01} + n_{11} + n_{10}} \tag{8}$$

where

- n_{11} is the total number of elements where Y_i and Y_j both have a value of 1.
- n_{01} is the total number of elements where Y_i is 0 and Y_j is 1.
- n_{10} is the total number of elements where Y_i is 1 and Y_j is 0.

Using (8), we compute the similarity matrix $S_{d\times d}$ from $B_{N\times d}$. $S_{d\times d}$ has values between 0 and 1. A large value between two time slots represents that they have high similarity. For each row of $S_{d\times d}$, we compute the average similarity value of d dimensions. If the average similarity value of the row is smaller than a given threshold τ, the dimension represented by the row is considered irrelevant and is deleted from the data matrix $\mathbf{X}$. In our work, τ is determined by the user. The filtered matrix is aggregated into one-day data (96 dimensions) by averaging power consumption measurements at the corresponding time slots in days.

C. Clustering

With the feature selection method discussed above, we delete some Y vectors from the streaming sequence of $Y_1, Y_2, ..., Y_d$ in time window W. The remaining Ys form a reduced data matrix $\mathbf{X}_r$.

We use a visual method to determine the number of clusters in $\mathbf{X}_r$. As stated in [18], visualizations are powerful tools to help the users to explore and make

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

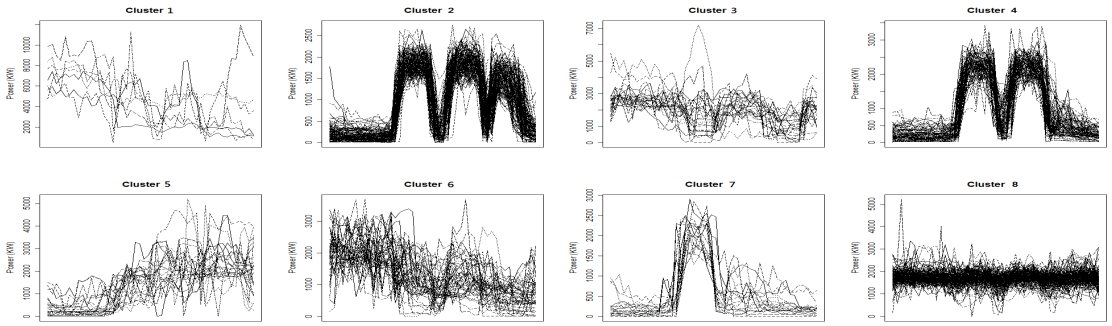


Fig. 7. Exploration of cluster patterns to determine the number of clusters.

Algorithm 2: MDL-Based Threshold Selection

Input: The sorted sequence of average density of all clusters S
Output: l_{min} and c_{min}
for l := *1 to* $s_{total-1}$ **do**
  Assign the first l cluster average densities to A;
  Assign the next $s_{total} - l$ cluster average densities to B;
  c_l=the average density of cluster l in A;
  Compute μ_A, the average density of clusters in A;
  Compute μ_B, the average density of clusters in B;
  Calculate the code length ;
  CL_l = $log(1+\mu_A)$+$\sum_{c_l \epsilon A} log(1+|c_l-\mu_A|)$ - $log(1+\mu_B)$+$\sum_{c_l \epsilon B} log(1+|c_l-\mu_B|)$;
  if (c_{min}¿c_l);
  c_{min}=c_l; $l_{min} = l$;
Output l_{min} and c_{min};

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
r_1	1	0	1	0	1	0	1	1	0	0
r_2		0		0		0			0	0
r_3			1			1	1		1	
	1		0		1	1	0	1	1	
	0		0	0	0		0	0	0	
				0					0	
		0		1		1			1	0
	0	0		1	0	1				0
		1	0			0	0			1
	0	1	0		0	0	0	0	1	1
	1		1		1		1	1	0	
		1		1		0				1
r_n	1	1	1	1	1	0	1	1	0	1

Fig. 6. The binary density matrix $B_{N \times d}$.

sense of data, intuitively revealing trends, outliers, and clusters from large and complex datasets. We use the k-means algorithm to cluster $\mathbf{X}_r$ into a large number of clusters and visually investigate the potential clusters and outliers. Fig.7 shows some examples of clusters produced by k-means.

From the array of clusters in the figure, we can see the top row contains 2 clusters of clear patterns, i.e., cluster number 2 and 4. The bottom row also contains two cluster patterns, i.e., cluster number 7 and 8. The patterns of two clusters in the leftmost column are not clear. Cluster number 3 in the top row and cluster number 6 in the bottom row show some patterns but are not explainable on work patterns. To determine the number of clusters, we do not count the clusters in the first column and only consider the remaining 6 clusters. Therefore, the true number of clusters is between 4 and 6 in this case. We say [4, 6] is a possible range.

To find the optimal number from the obtained range of clusters, we run the k-means algorithm multiple times on $\mathbf{X}_r$ using randomly chosen k from the estimated clusters range. For each clustering, we again visualize the clusters in the dimensions of two principle components to explore the highest variances of the data. The plots are also used to compute the separation and compactness of the clustering results. These two methods are collectively used to find the optimal number of clusters that provides plots with compact and well-separated clusters, where each cluster shows clear electricity consumption patterns. The procedure for segmentation of factories based on their power consumption behaviours is summarized in Algorithm 3.

V. Experimental Results

In this section, we present experiment results of the new method on a real-world AMI dataset. We compare the performance of clustering results of the new method with three feature selection methods of state-of-the-art algorithms for time series data. The comparisons have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

Algorithm 3: Segmentation of Factories using Load Profile Data

Input: One-month Data D
Output: Daily-Basis Segmentation λ of Factories
Remove anomalous data records from D ;
Apply feature selection technique on D;
Aggregate filtered data into one-day data (96 dimensions) by averaging power consumption measurements at the corresponding time slots in days;
Apply data visualization on aggregated data to estimate the number of clusters;
Apply the k-means algorithm on aggregated data;
Output: λ ;

shown that the new feature selection method can produce better clustering results than other three methods. We also discuss applications of factory segmentation on electricity consumption behaviors in tariff setting, demand response management and the quality of service.

A. Data

The real world dataset used in experiments was collected from Guangdong province of China. The AMI streaming data were obtained with smart meters installed at 21330 manufacturing factories. One month data in November 2012 was selected. Each time series contains 2880 measurements collected at 15 minutes time interval. 21330 manufacturing factories were from 33 industrial categories.

We use one day as the pattern analysis time window and divide the days in the month into workdays from Monday to Friday and weekends from Saturday to Sunday because work patterns at a workday and a weekend day are usually different. Each day has 96 electricity consumption measurements. There were 22 workdays and eight weekend days in November 2012. We represent workday and weekend day data in two matrices. Fig.2 plots some workday and weekend day electricity consumption time series. We can see that there are anomaly time series that need to be deleted from the dataset. We can observe two types of anomaly time series in Fig.2, one with constant electricity consumption measurement values that often result from fault readings of smart meters and one with very low average electricity consumption which indicates irregular production operations such as lack of production orders.

B. Three feature selection methods for comparison

We chose three feature selection for comparisons with the proposed method. They are Variance score [10], Laplacian score [11], and Sparse K-Means score (SK-Means) [12]. These methods represent state-of-the-art individual variable weighting methods. The SK-Means method uses the well-known lasso-type penalty to select the features. The Variance score method uses the variance of instances for each of the attributes as a measure to estimate the separability. For a given feature f and instance values $v(x,f), x = 1,...,n, f = 1,...,d$, the variance score is defined as follows:

$$VS(f) = \frac{1}{n}\sum_{x=1}^{n}\big(v(x,f) - \mu_f\big)^2, \qquad \mu_f = \frac{1}{n}\sum_{x=1}^{n} v(x,f) \tag{9}$$

The Laplacian score is based on locality preserving projection and Laplacian eigenmaps. It favors on features with high locality preserving power. The Laplacian score is computed as:

$$LS(f) = \frac{\sum_{x,y}\big(v(x,f) - v(y,f)\big)^2 S_{xy}}{\sum_{x}\big(v(x,f) - \mu_f\big)^2 D_{xx}}$$
$$S_{xy} = \begin{cases} e^{\frac{||d_x - d_y||^2}{t}}, & if d_x, d_y are neighbors \\ 0, & Otherwise \end{cases} \tag{10}$$

where $D_{xx} = \sum_y S_{xy}$, μ_f is the mean of values of feature f, t is the constant parameter, and d_x and d_y are the neighbors that either d_x belongs to the k-nearest neighbors of d_y, or vice versa.

In the comparison experiments, we first used a feature selection method to produce a reduced time series dataset. Then, we used the k-means algorithm to generate the clustering results. Finally, we used clustering evaluation measures to evaluate the clustering results produced by different feature selection methods. To make the result stable, for each feature selection method, we conducted clustering five times and used the average of evaluation measures to compare the clustering results of different feature selection methods.

C. Evaluation measures

Three evaluation measures were used to evaluate the clustering results in the experiments. The first measure is Mean Index Adequacy (MIA) [19], defined as the average of the distances between the objects and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

centers of the clusters to which the objects are assigned. MIA is calculated as follows:

$$MIA = \sqrt{\frac{1}{k}\sum_{i=1}^{k} d\left(r^{(i)}, L^{(i)}\right)} \tag{11}$$

where k is the total number of clusters, $L^{(i)}$ is the set of objects in cluster i, $r^{(i)}$ is the center of cluster i and d is the sum of distances between objects in the cluster and the cluster center. MIA measures the separations of clusters. The smaller the MIA, the more separate the clusters.

The second measure is Davies-Boulden Index (DBI) [20], which measures the ratio of the within-cluster scatter and the between-cluster separation. DBI is calculated as

$$DBI = \frac{1}{k}\sum_{x=1}^{k} max\left(\frac{d'(L^{(i)}) + d'(L^{(j)})}{d(r^{(i)}, r^{(j)})}\right) \quad i \neq j \tag{12}$$

where $L^{(i)}$ is the set of objects in cluster i, $d'(L^{(i)})$ is the geometric mean of the inter-distances between objects in $L^{(i)}$, and $d(r^{(i)}, r^{(j)})$ is the distance between the centers of clusters i and j. The smaller the DBI, the better the clustering result.

The third measure is CD index [21] defined as the total distance between centers of all clusters. CD is calculated as follows:

$$CD = \frac{D_{max}}{D_{min}}\sum_{i=1}^{k}\left(\sum_{j=1}^{k} d(r^{(i)}, r^{(j)})\right)^{-1} \tag{13}$$

where D_{max} and D_{min} represent the maximum and minimum distances between the cluster centers, respectively. The larger the CD, the better the clustering result.

D. Performance comparison on feature selection methods

Using the one month AMI dataset, we conducted experiments to compare the clustering performance of the new feature selection method and other three methods. In preprocessing, we divided the dataset into workday dataset and weekend day dataset and removed 1004 anomaly time series from the workday dataset and 1363 anomaly time series from the weekend day dataset. Then, we ran the four feature selection methods on the two datasets to remove some insignificant features from the two datasets. After that, we aggregated the multiple days time series in each dataset into one-day time series by taking the averages of multiple electricity consumption values at each time slot in the one day window. Finally, we used the k-means clustering algorithm to cluster the aggregated one-day workday and weekend datasets. The clustering results were evaluated with the three evaluation measures discussed above. The number of clusters k was visually determined. For the workday dataset, k was chosen as 25, and for the weekend data, k was set as 30.

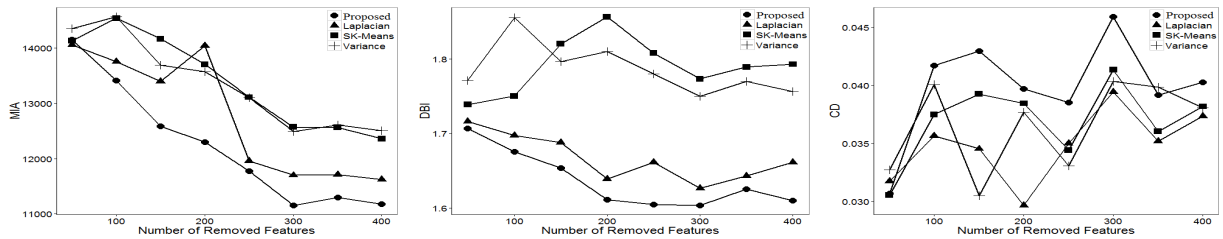
Fig.8 shows the clustering performance comparisons of the four feature selection methods on the workday and weekend datasets evaluated with three measures. The vertical axis indicates the evaluation measure and the horizontal axis is the number of features being removed with the four feature selection methods. We can observe that both MIA and DBI measures decrease as the number of removed features increase, and CD measure decreases as the number of removed features increases. These results indicate that feature selection is necessary for improving the clustering performance of both workday and weekend datasets.

The comparison of the four feature selection methods shows that the proposed method performs the best in all three measures because its performance measure line in the MIA and CBI figures is located below the *Laplacian*, *variance*, and *SK-Means* performance lines, whereas it is located above the other three performance lines in the CD figure. By analyzing the comparison results in the figures, we absorb that MIA and DBI measurements are the lowest and CDI measurement is the highest at the number of removed features 250 and 300 for the weekend and workday datasets, respectively. On the basis of this intuition, we remove 300 features from the workday dataset and 250 features from the weekend day dataset.

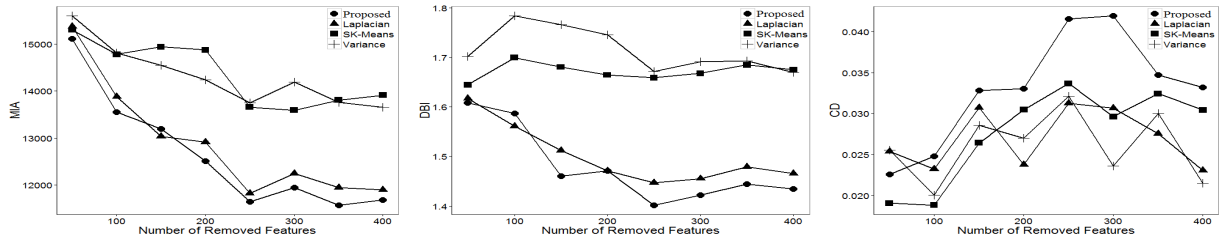
E. Cluster patterns analysis

After comparison of feature selection methods, we used the new feature selection method to remove 300 features from the workday dataset and 250 features from the weekend day dataset. The average numbers of removed features per day from the weekend and workday datasets are 13.6 and 31.3, respectively. Then, we used the k-means clustering algorithm to cluster the daily basis aggregated datasets, that have been generated from reduced one-month datasets for workday and weekend. The numbers of generated clusters for workday and weekend datasets were 30 and 25, respectively, which were determined visually.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access



(a) Evaluation on Workday Data



(b) Evaluation on Weekend Day Data

Fig. 8. Performance comparison on one-day aggregated datasets. The vertical axis shows the clustering performance measure and the horizontal axis is the number of features removed in the reduced data set.

Fig. 9 and Fig. 10 visualize the 30 and 25 clusters from the workday and weekend day datasets, respectively. These clusters show the daily electricity consumption patterns of different groups of factories in the month of November 2012. We can see the difference of consumption patterns during workdays and weekends. These patterns reflect production operation patterns of factories in different industrial sectors.

Two obvious patterns are the patterns of the two shift mode and the three shifts mode of production patterns. These are the common production modes in the discrete manufacturing process in the PRD region of Guangdong Province in China. Some clusters reflect the production patterns of the continuous manufacturing process which show constant electricity consumption pattern in 24 hours of a day.

From the array of Fig. 9, the 9 clusters in the first row and the first four clusters in the second row from the left column are the three shift patterns. The next 8 clusters are the two shift patterns. The next two clusters are constant patterns that represent continuous manufacturing process. The following 13 clusters present different patterns of clusters, some showing discrete manufacturing patterns and some showing continuous manufacturing patterns. The cluster patterns imply irregular manufacturing processes that may be caused by production disturbances such as insufficient production orders, frequent change of production processes or partial operation of production lines due to maintenance. For example, the first cluster of the bottom line shows a two-shift production pattern but the electricity consumptions on the morning and afternoon shifts were small. These patterns reflect either factories of small capacity or factories that production capacity is not entirely used due to insufficient production orders.

The magnitude of electricity consumption differs from one cluster to another. The difference resulted from the difference in electricity consumption in different industry sectors and difference of production capacity of factories in the same industry sector. For instance, clusters from 1 to 9 represent three-shift mode but have different peak electricity consumptions on workday from 500 kW to 10000 kW. The highest peak consumption of cluster 23 ranges from 6000 to 12000 kW. These clusters are small with a few factories.

From the array of Fig. 10, we can see that cluster patterns are more diverse than workday cluster patterns. There are less three shift patterns because weekends are not regular work days in many factories. Some factories work only on Saturdays with only morning shift and afternoon shift. Few factories work with three shifts on weekends. Many factories work irregularly on weekends as they cannot complete their production orders on workdays.

From the cluster patterns, we can further analyze the characteristics of factories represented in each cluster pattern. Table I lists examples of cluster patterns. Each cluster contains time series of factories in different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

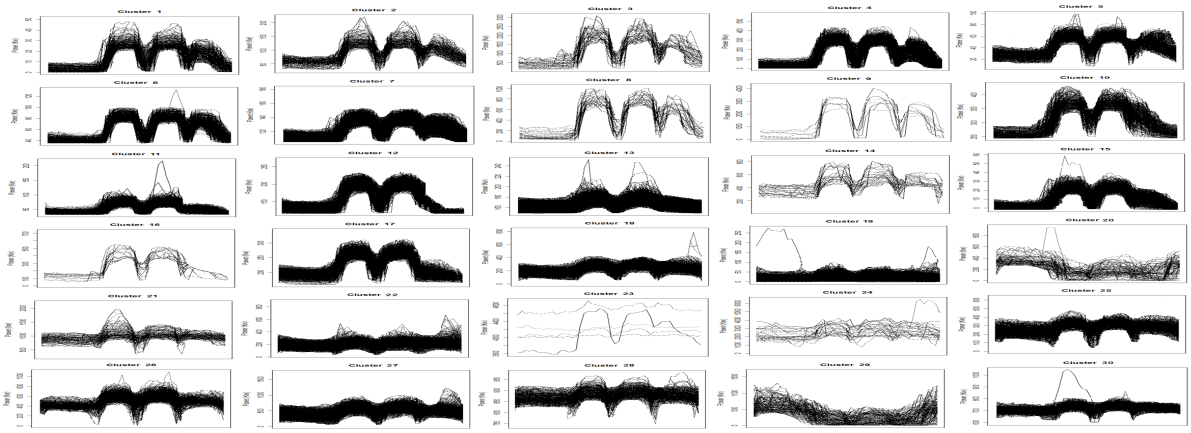


Fig. 9. Visualization of electricity consumption behaviors based segmentation on workday: y-axis: power consumption (kW); x-axis: half hour index (time).

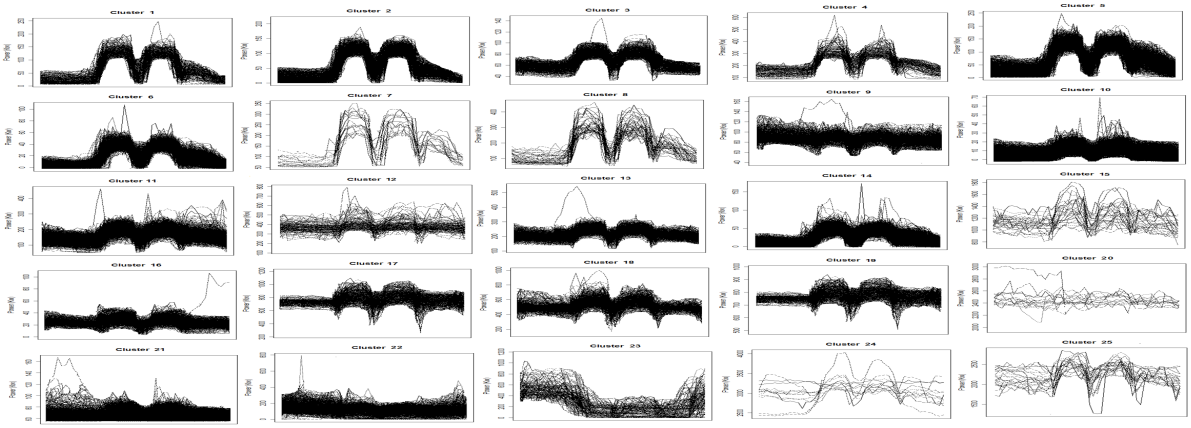


Fig. 10. Visualization of electricity consumption behaviors based segmentation on weekend: y-axis: power consumption (kW); x-axis: half hour index (time).

industry sectors. In each cluster, we list the three industry sectors of the top three frequent factories in the cluster and the percentages of the factories in each industry sector. We can see that the factories in different industry sectors use the same production mode. For example, the three shift cluster pattern contains factories most from Metal Products, Plastic Products, and Communication Equipment industry sectors. Since Metal Products and Communication Equipment industry sectors are the major industries in the PRD region of Guangdong province in China and product categories in these industries are diverse, the factories in these industry sectors have different production modes.

Cluster patterns of weekend data are not clear because the production processes of different factories in different industry sectors are different on weekends. Some factories do not work on weekends. Some work only on Saturdays and some work on both Saturdays and Sundays. Table II shows the percentages of factories in different industry sectors that do not work (No-Day-Off) or work on Saturdays (1-Day-Off) or work on both Saturdays and Sundays (2-Day-Off). We can see that most factories work only one weekday on Saturday. Very few factories work two days on weekends. These different work policies on weekends make the cluster patterns on weekends different from the workday cluster patterns.

F. Applications of cluster patterns

One potential application of segmentation of factories on electricity consumption patterns is to design variable rates of the electricity price to reduce peak loads of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2619898, IEEE Access

TABLE I
DOMINANT FACTORY TYPES IN POWER CONSUMPTION BASED WORK PATTERNS.

Work Patterns	**Workday**	
	Factory Types	Top 3 Percent Shares
	Metal Products Industry	28%
	Plastic Products Industry	22%
	Communication Equipment Industry	15%
	Metal Products Industry	14%
	Textile and Garment Industry	10%
	Communication Equipment Industry	10%
	Plastic Products Industry	27%
	Metal Products Industry	15%
	Communication Equipment Industry	13%
	Plastic Products Industry	25%
	Communication Equipment Industry	19%
	Electric Equipment Industry	11%
	Plastic Products Industry	20%
	Metal Products Industry	17%
	Non-Metallic Product Industry	09%
	Transport Equipment Industry	15%
	Communication Equipment Industry	14%
	Metal Products Industry	13%

TABLE II
PERCENTAGE OF CONSUMPTION PATTERNS WITH RESPECT TO THE FACTORY TYPES.

	Weekend		
Industry	No-Day-Off	1-Day-Off	2-Day-Off
Communications Equipment	25.6	68.8	5.6
Electrical Equipment	24.9	67.6	7.5
Transportation Equipment	23.3	68.2	8.5
Metal Products	27.3	65.8	6.9
Plastic Products	25.3	70.1	4.6
Others	25.2	68.2	6.6

smart grid. The economic benefits of such time-variable electricity rates are justifiable [22]. However, the design of time-variable rates requires segmenting the electricity users according to their load profiles [23]. Segment-specific rate design determines a time-variable rate for each factory segment. As stated in [24], the segment-specific rate design is a complex process, requiring to determine the number of time zones, the start times of all time zones, the total number of price zones and the profitability of suppliers. In this process, segmentation of users on load profiles is the first necessary step.

VI. Conclusions

The extensive roll-out of smart meters on smart grids generates enormous opportunities and also creates challenges to electricity utilities. Significant investments in the AMI allow for a high level of monitoring, control, and optimization of smart grids, which, subsequently, leads to improved customer services. Utilization of the AMI data from smart meters enables utilities to achieve significant business gains. However, effective and efficient processing and analysis of big AMI data are still a big challenge to smart grid companies.

In this paper, we presented an implementation and evaluation of a cluster analysis approach for application to smart meter data. We proposed a new feature selection method to reduce the dimensions of a selected time window by removing insignificant features, thus improving the clustering performance. We demonstrated that the discovered cluster patterns allow for a bet-

ter segmentation of factories using specific patterns in behaviors of electricity consumption to judge for the different production modes. We discussed the application of segmentation in the segment-specific time variable rate design.

In our future work, we will study the change of cluster patterns over time and develop a predictive technology for prediction of the pattern change.

REFERENCES

[1] Moss, "Market segmentation and energy efficiency program design," *Berkeley, California Institute for Energy and Environment*, 2008.

[2] C.Gu, P.Shi, S.Shi, H.Huang, and X.Jia, "A tree regression-based approach for vm power metering," *IEEE Access*, vol. 3, pp. 610–621, 2015.

[3] J.Z.Huang, M.K.Ng, H.Rong, and Z.Li, "Automated variable weighting in k-means type clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 657–668, 2005.

[4] I.Khan, J.Z.Huang, N.T.Tung, and G.Williams, "Ensemble clustering of high dimensional data with fastmap projection," *Trends and Applications in Knowledge Discovery and Data Mining*, pp. 483–493, 2014.

[5] Verd, M.O.Garcia, C.Senabre, and A.G.Marin, "Classification, filtering, and identification of electrical customer load patterns through the use of self-organizing maps," *Power Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 1672–1682, 2006.

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

[7] A.Albert, "Smart meter driven segmentation: What your consumption says about you," *Power Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 4019–4030, 2013.

[8] A. Sankar and C. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE transactions on speech and Audio Processing*, vol. 4, no. 3, pp. 190–202, 1996.

[9] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," *AAAI/IAAI*, vol. 1097, 2000.

[10] C.M.Bishop, "Neural networks for pattern recognition," *Oxford university press*, 1995.

[11] X.He, D.Cai, and P.Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2005, pp. 507–514.

[12] Daniela, "A framework for feature selection in clustering," *Journal of the American Statistical Association*, vol. 105, no. 490, 2010.

[13] R. Berthier, W. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 350–355, 2010.

[14] Sioshansi, "Smart grid: integrating renewable, distributed & efficient energy," *Academic Press*, 2011.

[15] S.Meters, "Smart meter systems: a metering industry perspective," *An Edison Electric Institute-Association of Edison Illuminating Companies-Utilities Telecom Council White Paper, A Joint Project of the EEI and AEIC Meter Committees, Edison Electric Institute*, 2011.

[16] C.C.Aggarwal, "Data streams: models and algorithms," *Springer Science & Business Media*, vol. 31, 2007.

[17] R.Agrawal, J.Gehrke, D.Gunopulos, and P.Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 5–33, 2005.

[18] B.Shneiderman, "Inventing discovery tools: combining information visualization with data mining1," *Information visualization*, vol. 1, no. 1, pp. 5–12, 2002.

[19] G.Chicco, R.Napoli, F.Piglione, and C.Toader, "A review of concepts and techniques for emergent customer categorisation," in *TELMARK Discussion Forum European Electricity Markets, London*. Citeseer, 2002.

[20] D.L.Davies, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 224–227, 1979.

[21] M.Halkidi, Y.Batistakis, and M.Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2, pp. 107–145, 2001.

[22] H.Parmesano, "Rate design is the no. 1 energy efficiency tool," *The Electricity Journal*, vol. 20, no. 6, pp. 18–25, 2007.

[23] G.Chicco, R.Napoli, P.Postolache, and C.Toader, "Customer characterization options for improving the tariff offer," *Power Systems, IEEE Transactions on*, vol. 18, no. 1, pp. 381–387, 2003.

[24] C.Flath, D.Nicolay, and F.Lilia, "Cluster analysis of smart metering data," *Business and Information Systems Engineering*, vol. 4, no. 1, pp. 31–39, 2012.