International Journal of Database Theory and Application
Vol.9, No.8 (2016), pp.269-280
http://dx.doi.org/10.14257/ijdta.2016.9.8.25

Load Pattern Window Aware Power Supply Device Clustering

Wanxing Sheng[1], Ke-yan Liu[1], Yixi Yu[2,3], Rungong An[2,3], Ningnan Zhou[2,3] and Xiao Zhang[2,3,a]

[1]*China Electric Power Research Institute, Beijing 100192, China*
[2]*Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, Renmin University of China, Beijing 100872, China*
[3]*School of Information, Renmin University of China 100872, China*
[1]*{wxsheng,liukeyan}@epri.sgcc.com.cn*
[2, 3]*{yuec, anrungong, zhouningnan123, zhangxiao}@ruc.edu.cn*

Abstract

Data-driven decision in big data era is becoming ubiquitous in electronic grid. In particular, daily collected power consumption records enable workload aware device clustering, which is crucial for critical domain applications such as device functionality identification. In this paper, we propose a load pattern window aware method for clustering power supply devices. Our approach overcomes the drawbacks in existing works, such as fuzzy based clustering, K-means based clustering and neutral network based clustering. After investigating the large scale records from power supply devices, our approach partitions device records into disjoint time intervals with parameterized window size, which indicate the load pattern feature for a period of time given a specific device. Devices are then decomposed into a mixture of these features, and those devices with similar dominating features are grouped together. The experimental results demonstrate the effectiveness and efficiency of our solution based on the real data collected from power grid in China.

Keywords*: load pattern aware, clustering, power grid, k-means*

1. Introduction

Data-driven decision in big data era is becoming ubiquitous in power grid. In this paper, we focus on the issue of device functionality identification. Although the address of each device is registered when the device is installed, the power usage through the device is out of control from the power grid company. For example, in a residential distinct, power supply device is aimed for daily use. However, some factories may borrow power from these residential buildings with no permission from power grid company. To force resource-consuming industries to save energy, electronic power is much more expensive for factories than houses, following the laws of state grid. To enforce such laws, the capability of device functionality identification is the promise. In our example, the power grid company should be able to aware that the power for houses is transferred to factories.

Power supply devices clustering is a straightforward and intuitive solution to this issue. After devices with similar workloads are grouped together, power supply companies can identify the representative workload and thus the functionality of each device is revealed. For example, Figure 1 shows the workloads from three power supply devices for a factory, a residential building and the case where the factory borrows power from a residential building respectively. Although the address associated with the third power supply device is a residential building, we can see its workload is more similar to that of

[a] Corresponding Author: Xiao Zhang, E-mail: zhangxiao@ruc.edu.cn

ISSN: 2005-4270 IJDTA

the factory. In this way, if we cluster the first and the third device together, it will be revealed to the power supply company that they share the similar functionality.

Existing device clustering works can be classified into three categories [1-14]. The first fuzzy-based clustering methods extract power load patterns and represent them by different mathematical models. However, their effectiveness is sensitive to the initial manually chosen models. The second are K-means based that are totally unsupervised but suffer from random fluctuation in power measurement. The third neutral network based methods require plenty of labeled data. Our load pattern window aware clustering approach inherits the unsupervised property from the K-means based clustering algorithm and overcomes all drawbacks in existing works by computing the average power in a load pattern window so that the random fluctuation can be canceled out.

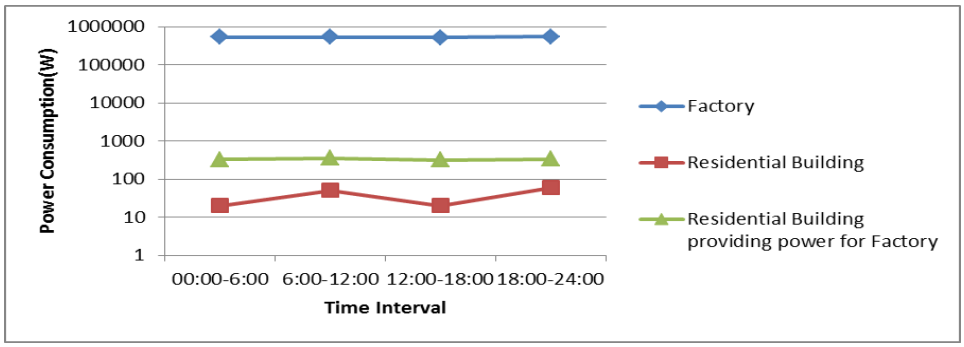


Figure 1. Power Supply Measurement for Devices with Different Functionalities

To overcome these drawbacks, in this paper, we investigate the load curves of power consumptions for each device and then summarize the concept of load patter window (LPW), a daily property for power consumption of each device. LPW enables us to propose a load pattern window aware Power Supply Device Clustering approach. In this approach, we first partition the device records into disjoint time intervals with the length of LPW and then summarize each time interval by the average power consumption for each device in each time interval. Finally, we adopt the well-known K-means clustering algorithm to mine the similarity from these summarized records of all devices.

The main contributions of this paper are as follow:

(1) It first studies the problem of power supply device clustering to facilitate critical domain applications such as device functionality identification.

(2) This paper first introduces load pattern window aware clustering approach to overcome all drawbacks, such as human intervention during model selection, clustering precision drop under random fluctuation and the requirement for plenty of labeled data. Records from every device are regarded as a mixture of different functionalities and the functionality dominating the device reveals the device identification.

(3) The extensive experiments are conducted to evaluate the effectiveness and efficiency of our solution. We evaluate our solution using a real data from the actual power grid in the very province of China.

2. Related Work

In this section, we review existing clustering approaches on State Grid dataset. Existing methods can be categorized into three classes, Fuzzy based clustering, K-means based clustering and neutral-network based clustering.

Fuzzy based clustering methods [1-6] extract power load patterns, represent them by different mathematical models and view each individual load curve as a mixture of different models. And the coefficients of a load curve on these models compose the

feature space. For example, one may assume the load is a mixture of multiple Gaussian distribution and thus the curve can be fit into a weighed linear combination of multiple Gaussian distribution and the weights on all distributions compose a vector defined as the feature for the device. However, such methods suffer from diversity of load patterns and thus are sensitive to the selection of the initial mathematical models. Compared to these methods, our approach is totally unsupervised and no human intervention is required.

K-means based approach is simple and efficient [7-10]. Initially, time series power consumption values from the same device become a long vector for the device and K-means is applied on these vectors to cluster devices. However, such approach suffers from 1) the number of records is not the same for all devices and their similarity is not well-defined; 2) the huge number of dimensionality of vectors causes the curse of dimensionality and all devices are far from each other. In this way, recent methods represent each record in the vector space by its recorded power consumption. Then, each device can be regarded as a set of points in the coordination system. After processed by the original K-means algorithm, each point is assigned with its cluster identification. In this way, for each device, the most frequent cluster id corresponding to its point set determines its cluster assignment. However, these methods suffer from fluctuation in power consumption measurement and the most frequent cluster id tends to come from outline records. Our approach improves the K-means based approach in two aspects. *Firstly*, we introduce the load pattern window. This parameter replaces the accurate power consumption by the average power consumption in a time window. Such replacement eliminates the fluctuation brought about by outline records. *Secondly*, by applying load pattern window, our approach is much faster than the compared K-means algorithm.

For huge volume datasets, there are also efforts to implement K-means algorithm on cloud platform. Such distribution and parallel frameworks are not our focus in this paper. However, since we do not modify the K-means algorithm, we believe that these alternatives can also be applied to our algorithm.

Neutral-network [11-14] based clustering is a supervised-based clustering; it thus can achieve good precision. Some works adopt KOHONEN neutral network to capture the dynamic characteristic of load patterns in load curves. However, since it requires huge labeled datasets, it is labor-consuming and difficult to apply to the China State Grid dataset with huge volume and dynamic load patterns. In addition, neutral network is usually time-consuming to tune its parameters in each neutral node and existing methods have not yet guaranteed when the algorithm can terminate with stable parameters.

3. Load Pattern Window Aware Clustering

3.1. Overview

In this section, we first introduce the idea underlying our load pattern window aware clustering approach and then present the detailed algorithms.

The concept of load pattern window is motivated by investigation over the huge number of power supply records collected from devices. Each record is in the form of (DeviceId, Power, Timestamp)[b] and Figure 1 plots records from three devices. For example, the device near a factory keeps high power consumption in all time intervals, which means the factory is producing all the time. In contrast, the power consumption curve for residential building exhibits rises and falls, indicating whether TVs or air-conditioners are in use or not. Similarly, when a residential building lends power to factory, its power usage pattern is similar to that of a factory, which keeps high all the time.

In this observation, the choice of time interval in the plotting is critical. Too small time interval will yield curves with all falls and rises due to fluctuation and too large time

[b] We omit other attributes here for brevity.

interval will result in all horizontal curves. In fact, the time interval chosen in Figure 1 matches the load pattern for workloads in factories and residential buildings, which take 6 hours to transfer from one power usage pattern to another.

Because different regions own different work behavior, suitable time interval length varies from region to region. In order to capture such dynamic property of suitable time interval, we introduce a variable called load pattern window parameter LPW to represent the length of these time intervals. The definition 1 formalizes this concept.

Definition 1. Given a power supply device, the period time where its recording power consumption is dominated by a mixture of sources with stable coefficient is called **load pattern window**. We use LPW to denote the length of this period of time.

Specifically, in the example above, in the load pattern window [6:00, 12:00], the power consumption recorded in the power supply device is dominated by a mixture of sources with stable coefficient where the source representing the chemical factory takes the most weight in the mixture.

Given the parameter LPW, our Load Pattern Window aware Clustering approach consists of three phases:

At first, we partition the device records into disjoint time intervals with length of LPW and then summarize each interval by the average power consumption for each device in that interval.

Next, we adopt the well-known K-means clustering algorithm to mine the similarity from these summarized records of all devices.

Finally, the K-means clustering provides the probability for each device to be assigned with each cluster identification. Thus, the most probable cluster identification is chosen to be the actual cluster ID for the device.

3.2. Algorithm

This section presents three algorithms to describe the three phases of load pattern window aware clustering (LPWAC) respectively.

To eliminate fluctuation caused by outline records, we calculate the average power consumption in each time interval. Before the calculation, we first produce these time intervals by the load pattern window parameter. Algorithm 1 illustrates the partitioning of records into disjoint time intervals and the average power consumption computation. We first initialize an empty set *L* to store the representative power consumption of load curves (line 1). To get the representative power consumption, there are several tricks (line 2-15). LPWAC normalizes each load curve *Z* in dataset *D* to discard the impact of heterogeneous data attributes (line 3). Each load curve *Z* is partitioned into segments with fixed length LPW (line 4-7). LPWAC gets the records in each time interval and computes the average value of records as the representative power consumption *E* (line8-12). Then entry *E* is added into *L* (line 14). The set *L* of all the representative power consumption entries is returned (line 16).

For example, Table 1 illustrates records of three devices sampled every 2 hours and the outline records are in bold and italic. After we set LPW to 6 hours, since there are 24 hours per day, these records are partitioned into 24/6/2=2 disjoint time intervals. Values in plain in Table 2 shows the partition result and we can see that the outline records are canceled in each time interval and the bold values show the result after the normalization (Line 3).

Table 1. An Example for Power Consumption for Three Devices Sampled Every 2 Hours

Power consumption (W)	00:00	2:00	4:00	6:00	8:00	12:00	14:00	16:00	18:00	20:00	22:00	24:00
Device1	***6687***	5297	***3124***	5126	5364	5146	5467	4986	5678	***3147***	***6124***	5234
Device2	20	20	20	50	40	20	20	20	20	50	80	20
Device3	354	312	368	347	396	345	328	371	363	316	348	375

Table 2. An Example for the Average Value in Each Time Interval

Power consumption(W)	00:00-6:00	6:00-12:00	12:00-18:00	18:00-24:00
Device1	5036/**0.93**	5212/**0.97**	5377/**1**	4835/**0.90**
Device2	20/**0.4**	36.7/**0.734**	20/**0.4**	50/**1**
Device3	344.7/**0.95**	362.7/**1**	354/**0.98**	346.3/**0.955**

Algorithm 1 GetRepresentativePowerConsumption

```
Input:  D:load curves of devices, LPW: load pattern window size
Output: L: set of representative power consumption
1:  L←ϕ;
2:  for each load curve Z ∈ D do
3:     Z ← Normalized(Z);
4:     Leg←getTotalLength(Z);
5:     Intervals←⌈Leg/LPW⌉;
6:     for i=0:(Intervals-1) do
7:          Records ← GetRecordsInInterval(Z,i*LPW,(i+1) *LPW);
8:          E←new RepresentativePowerConsumption();
9:          for each record R ∈ Records do
10:              for each attribute a ∈ R do
11:                  E.a = E.a+R.a/m;
12:         end for
13:       end for
14:       L.add(E);
15:    end for
16: return L;
```

After the records of each device are partitioned into disjoint time intervals and the fluctuation in power measurement is eliminated by the average value, similar values are grouped together by the classic K-means algorithm. In this way, each time interval of the device is assigned with a workload pattern, which can be interpreted as factory pattern or residential building pattern. Algorithm 2 illustrates the clustering process for all the representative power consumption entries in *L*. First, K-means randomly selects k initial cluster centers (line 1). For each element *E* in *L*, K-means measures the distances from *E* to each of the cluster centers and tags *E* as belonging to the cluster which is the nearest to *E* (line 4-13). K-means computes the average value in each cluster as new cluster center (line 16). Loop the above two steps until the difference between new cluster center and the last cluster center is less than threshold (line 17-22).

As the example in Table 2, the three rows for the three devices form three vectors and we can easily see that the first vector is more similar to the third vector after normalization. In fact, when we indicate that there are two clusters to K-means, the first record is assigned with the same cluster identifier as the third record.

Algorithm 2 KmeansClustering

```
Input:  L: set of representative power consumption, k: cluster number
Output: Clusters: k clusters set
1:  CC ←SelectClusterCenter(L,k);
2:  Clusters← ϕ;
3:  while true
4:     for each element E∈ L do
5:        min = distance(E,CC[1]);  C=1;
6:      for i=2:k do
```

```
7:          dist ← distance(E,CC[i]);
8:          if (dist < min)  then
9:          min =dist;  C=i;
10:         end if
11:       end for
12:     Cluster[C].add(E);
13:     end for
14:     flag←true;
15:   for i=1:k do
16:     NCC[i] ←ComputeClusterCenter(Cluster[i]);
17:     if (NCC[i] – CC[i] > θ )  then
18:               flag ←false;
19:     end if
20:     if  flag  then
21:           break;
22:       end if
23:       CC[i] = NCC[i];
24:     end for
25:   end while
26:   return Clusters;
```

With the help of clustering results in Algorithm 2, each device consists of different workload patterns. Algorithm 3 aims to find the dominating pattern for each device as its cluster identification. In particular, Algorithm 3 counts the entries belonging to each device in each cluster (line 2-7). The device is assigned to the cluster in which the occurrence probability of this device is the largest (line 9-16).The final result is returned (line 17).

Algorithm 3 ClusteringAssign

```
Input:   Clusters: k clusters set, N: total devices number
Output:  Result: n clusters set
1:   i ←0; Counts← new int[N][k];
2:   for each cluster CL ∈ Clusters do
3:     for each element E ∈ CL do
4:       j ←E.id;
5:       Counts[j][i] = Counts[j][i]+1;
6:     end for
7:   end for
8:   for i=1:N do
9:     max←0, C←0;
10:    for j=1:k do
11:      if (Counts[i[j] > max) then
12:      max= Counts[j][i]; C=j;
13:      end if
14:    end for
15:    Results[C].add(i);
16:  end for
17:  return Clusters;
```

LPWAC improves the clustering quality by adjusting the LPW. The paper evaluates the precision and recall while varying LPW in section Experimental evaluation. The results demonstrate that our solution is effective.

4. Experimental Evaluation

In this section, we first describe our experiment settings and then demonstrate that our approach can achieve accurate clustering with reasonable execution time.

4.1. Setting

The experiments are conducted on a PC with Intel 2.33 GHz quad-core CPU and 4GB of RAM, and the disk size is 500G. The K-means algorithm is implemented by Java and the JDK 1.6 is adopted.

We use a real-world dataset. The dataset is collected from the real-time power consumption of 24 power supply devices belonging to China State Gird Company. We sample each device per 15 minutes and thus obtain a one-year dataset with 840960 records in the form of (DeviceId, Power Consumption, and Timestamp).

The ground truth of the clustering result is provided by the China State Grid Company. To measure the accuracy of our clustering algorithm, we adopt the precision (denoted by *P*) and recall (denoted by *R*) measurement for clustering in [15]. They are shown in formula 1 and 2. The symbols and notations are listed in Table 3.

$$P = TP/(\mathrm{TP}+\mathrm{FP}) \tag{1}$$

$$R = TP/(\mathrm{TP}+\mathrm{F}N) \tag{2}$$

Table 3. The Symbols and Notations Measuring Precision (P) and Recall (R)

symbol	Description
N	record size
N(N-1)/2	record pairs involved in the clustering
TP	the number of record pairs that are similar and assigned with the same cluster identification (True Positive)
TN	the number of record pairs that are dissimilar and assigned with the same cluster identification(True negative)
FN	the number of record pairs that are dissimilar and assigned with the different cluster identification(False Negative)
FP	the number of record pairs that are dissimilar and assigned with the same cluster identification(False Positive)

4.2. Experimental Analysis

In this section, we first show that our approach is efficient and effective in real world dataset. Then, we change the size of the data set to show that the effect of our approach is robust in terms of the data volume.

First, we compare the precision and recall with two promising and representative approaches, neutral network based clustering (NNC) [11] and fuzzy based Gaussian mixture model based clustering (GDC) [1]. Figure 2 shows that under the whole dataset, when we set the LPW of our LPW approach to 4 hours, our approach outperforms other competitors significantly. Figure 3 further shows that our approach consumes the least execution time.

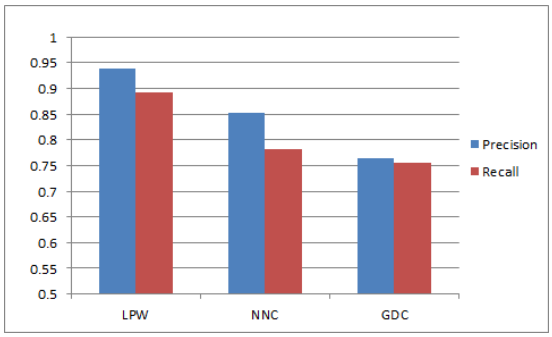


Figure 2. Clustering Effect Comparison

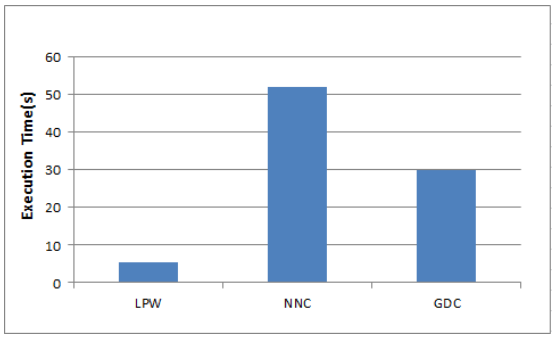


Figure 3. Clustering Performance Comparison

In the next experiment, we show the effect of the Load Pattern Window LPW on precision and recall. As Figure 4 and Figure 5 illustrate, when the window is set to about 4 hours, the precision and recall is the highest. This is because when the LPW is too small, random fluctuation in the power supply may influence the clustering effect. On the other hand, when the LPW is too large, interesting patterns hidden in the power supply are missing because the average value is computed over a long period. When the LPW is set to 4 hours, it matches the normal working hours for daily life. That is, people usually work from 8:00 to 12:00 in the morning and 14:00 to 18:00 in the afternoon and so do machines.

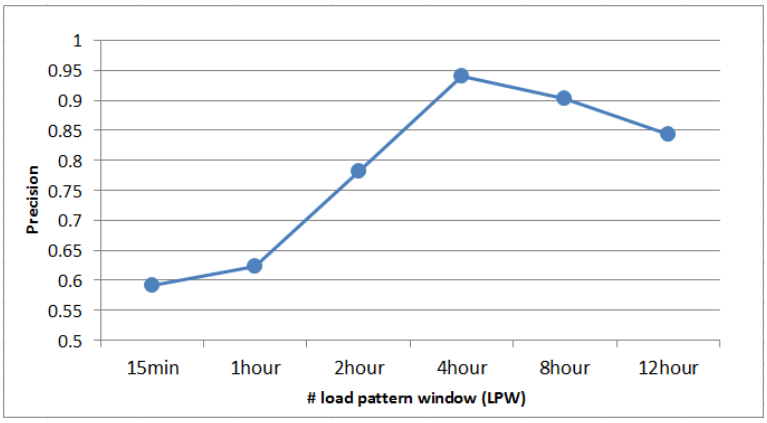


Figure 4. Effect of LPW on Precision

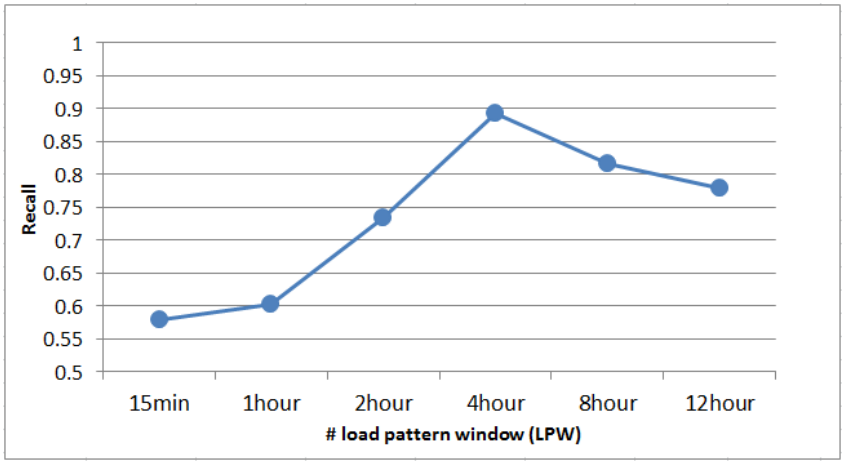


Figure 5. Effect of LPW on Recall

In the following, we show that our approach is much faster than the trivial K-means method when windows LPW is set to 15 minutes. As illustrated in Figure 6, when LPW is set to 1 hour, the execution time is reduced to 10% of the trivial K-means. This result comes from two parts: (1) the K-means algorithm essentially takes the time complexity O(KN) in each iteration and after we partition the records by window parameter, the time complexity in each iteration is reduced to O(KN/LPW); and (2) when the number of records is huge, the K-means algorithm takes much more iterations to converge and when the window LPW increases, the number of records actually reduces significantly and the K-means algorithm thus converges efficiently.

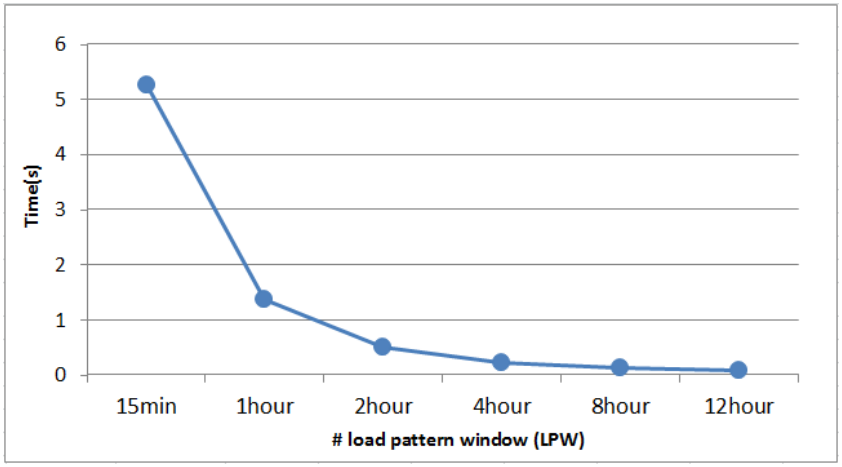


Figure 6. Effect of LPW on Execution Time

Both experiments above illustrate the effectiveness and efficiency of our approach. The third experiment below shows that the effectiveness of our approach is robust regardless of the data size. As shown in Figure 7 and 8, though the data volume decreases, the precision and recall of our approach do not deteriorate. The reason is that the core concept introduced in our approach, the load pattern window, is a daily property of power consumption and lasts for only a few hours. In this way, reducing the data size to one or two seasons will not influence the precision and recall of our approach.

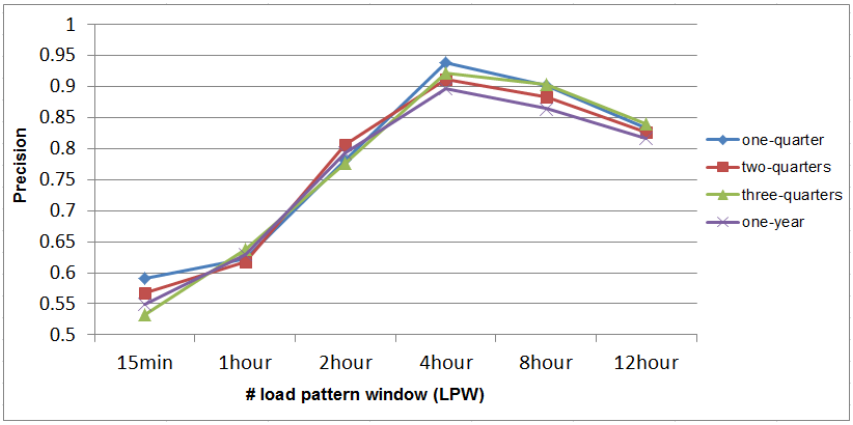


Figure 7. The Precision with LPW Influence Under Different Data Volume

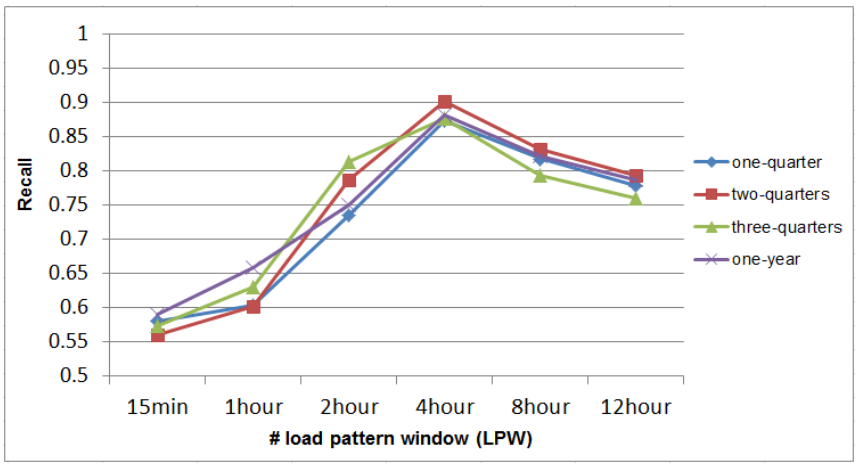


Figure 8. The Recall with LPW Influence Under Different Data Volume

5. Conclusions

To utilize the automatically collected power consumption records in big data era, this paper proposes a new load pattern window aware clustering to mine devices with similar load pattern in power system. The model partitions device records into disjoint time interval with tunable window size. Then K-means is applied to cluster power supply devices. The clustering quality is improved by setting the optimal window size compared with the baseline. The experimental results, which are based on real data set of power grid in China, demonstrate the high precision and recall of our approach. The robustness of our approach is also validated.

Acknowledgements

This research was supported by the Project of State Grid Corporation of China Research Program (EPRIPDKJ [2014] NO.3763).

References

[1] C. S. Wang, J. Cao and G. Y. Chen, "Power System Transient Stability Contingency Screening based on Clustering Analysis", Power System Technology, **(2005)**.
[2] G. Chicco, R. Noberto and F. Pigione, "Comparisons among clustering techniques for electricity customer classification", IEEE Transaction on Power Systems, **(2006)**.

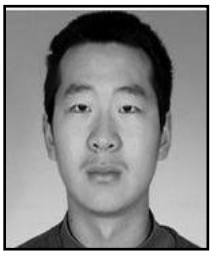
[3] S. E. Papadakis, J. B. Theocharis and A. G. Bakirtzis, "A load curve based fuzzy modeling technique for short-term load forecasting", Fuzzy Setsand Systems, **(2003)**.
[4] H. Z. Liu, K. Zhou and X. J. Hu, "Clustering Analysis of Power System Load Series based on Ant Colony Optimization Algorithm", Electric power, **(2013)**.
[5] P. Q. Li, X. R. Li, H. H. Chen and W. W. Tang, "The Characteristics Classification and Synthesis of Power Load Based on Fuzzy Clustering", Proceedings of the CSEE, **(2005)**.
[6] G. P. Shi, J. Liang and X. S. Liu, "Load clustering and synthetic modeling based on an improved fuzzy c means clustering algorithm", The 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, **(2011)**.
[7] T. Rasanen, D. Voukantsis and H. Niska, "Data-based method for creating electricity use load profiles using large amount of customer-specific hourly measured electricity use data", Applied Energy, **(2010)**.
[8] X. F. Bai and G. D. Jiang, "Load modeling based on improved K-means clustering algorithm and its application", Electric Power Automation Equipment, **(2010)**.
[9] F. Chen, H. T. Liu, Z. Huang and X. J. Zhang, "Probabilistic load model based on improved k-means clustering algorithm", Power System Protection and Control, **(2013)**.
[10] J. H. Liu, J. Wang, Y. Meng and W. S. Wang, "Clustering Analysis of Power Load by Rough Set and K-means Based on Simulated Annealing Method", Modern Electric Power, **(2012)**.
[11] R. Ma and R. M. He, "Characteristics Clustering Approach of Power System Load Based on Parallel Neural Network with Particle Swarm Optimization", Modern Electric Power, **(2006)**.
[12] Z. H. Shi, S. Z. Zhou and J. H. Zheng, "Application of improved back propagation neural network for identification of the percentage of dynamic component in composite load", Proceedings of the CSEE, **(2004)**.
[13] M. Gavrilas, O. Ivanov and G. Gavrilas, "Load profiling with fuzzy self-organizing algorithms [C].The 9th Symposium on Neural Network Application in Electrical Engineering (NEUREL), Belgrade, Serbia, **(2008)**.
[14] S. Osowski and K. Siwek, "The self-organizing neural network approach to load forecasting in the power system. International Joint Conference on Neural Networks, Washington, DC USA, **(1999)**.
[15] Information on http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

Authors

Wanxing Sheng, received the Bachelor's, Master's and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China. Since 1997, Dr Sheng has been a Full Professor in China Electric Power Research Institute (CEPRI), Beijing, China, where he is currently the Head of the Department of Power Distribution. His research interests include renewable energy generation and Grid-connected technologies, *etc*. He has published more than 150 refereed journal and conference papers, and 15 books.

Ke-yan Liu, received Ph.D. degree in electrical engineering from Beihang University in 2007.Currently, Dr. Liu is a senior researcher in China Electric Power Research Institute. His current research interests are distributed generation (DG), planning and operation analysis of distribution systems, power system computation method and simulation. He has published more than 60 refereed journal and conference papers on the transactions and journals sponsored by IEEE, ACM and Elsevier.

Yixi Yu, is a student of the Master of Computer Science at Renmin University of China. Her research focuses on database optimization and big data analysis.



Rungong An, is a student of the Master of Computer Science at Renmin University of China. His research focuses on high-performance database.

Ningnan Zhou, Ph.D. candidate of Renmin University of China. His research focuses on information retrieval and database architecture.

Xiao Zhang, received a Master degree in Computer Science and Technology in 1998 from Renmin University and his Ph.D. degree in Computer Science and Technology in 2001 from Institute of Computing Technology, Chinese Academy of Science. His current research focuses on the database architecture and big data analysis. He specializes in design and implementation of database management system.