Intelligent Automation And Soft Computing, 2019
Copyright © 2019, TSI® Press
Vol. 25, no. 3, 585–593
https://doi.org/10.31209/2019.100000113

An Efficient Supervised Energy Disaggregation Scheme for Power Service in Smart Grid

Weilie Liu, Jialing He, Meng Li, Rui Jin, Jingjing Hu, and Zijian Zhang

School of Computer Science and Technology Beijing Institute of Technology
No.5 Yard Zhong Guan Cun South Street Haidian District Beijing China

ABSTRACT
Smart energy disaggregation is receiving increasing attention because it can be used to save energy and mine consumer's electricity privacy by decomposing aggregated meter readings. Many smart energy disaggregation schemes have been proposed; however, the accuracy and efficiency of these methods need to be improved. In this work, we consider a supervised energy disaggregation method which initially learns the power consumption of each appliance and then disaggregates meter readings using the previous learning result. In this study, we improved the fast search and find of density peaks clustering algorithm to cluster appliance power signals twice to learn appliance feature matrices. Additionally, we improved the max-min pruning matching optimization algorithm to decompose the aggregate power consumption into individual appliance. Experimental results obtained using the reference energy disaggregation dataset demonstrate that the proposed scheme achieves 81.9% accuracy and requires only 8 s to analyze 20-m readings for each sliding window. Thus, the proposed scheme exhibits better accuracy and efficiency compared with existing schemes.

KEY WORDS: supervised energy disaggregation, energy saving, privacy mining, smart meter

1 INTRODUCTION

ENERGY disaggregation or non-intrusive load monitoring algorithms, which can be considered as the decomposing of aggregate households', commercial, or industrial power consumption as the power consumption of related single devices, is currently an area of focus in energy saving research (Ehrhardt-Martinez et al (2010), Berges, M. et.al (2011), Patel, S.N. et al. (2007), Jin, Y. et al. (2011), Raju L et al. (2017), Yuhua Peng. et al. (2018)). The residential power consumption in the European Union alone accounts for 30% of global electricity consumption (Cox, R. et al. (2006), Schoofs, A. et al. (2010), Giri, S. et al. (2012), Najmeddine, H. et al. (2008), Ruzzelli, A.G et al. (2010)). Furthermore, the International Energy Agency has forecasted that the probability of increase for global energy demand would be up to 30% between now and 2040, equivalent to China's plus India's energy consumption; therefore, reduction of residential power consumption is critical. Compared with aggregated consumption data, the data for individual appliance have advantages relative to reflecting users' electrical behavior modes and moderating electricity consumption. Furthermore, individual appliance data can be utilized to rapidly detect appliance malfunctions and precisely predict the power demand (Hart, G. (1992), Z Guan et. al. (2017), Z Guan et al. (2018), Zoha, A. (2012), Hosseini, S. S. et. al. (2017), K. Gai et. al(2018)).

Energy disaggregation algorithms are commonly divided into supervised and unsupervised algorithms. The former uses individual appliance data for training, whereas the latter does not. Supervised algorithms utilize individual appliance data in the learning stage to form feature matrices used to disaggregate meter readings. Some prior knowledge can be useful in supervised schemes, such as the total number of activated electrical appliances or the collaborative working information of multiple appliances. (Kolter, J. et. al. (2010), Dong, Roy, et al. (2013), Wytock, Matt, and J. Zico Kolter. (2014), Yeqing Li et al. (2014), Altrabalsi, Hana, et al. (2014), Elhamifar, Ehsan, and Shankar Sastry. (2015), Mauch, Lukas, and Bin Yang. (2015)). Unsupervised algorithms (Parson, Oliver, et al. (2014), Bonfigli, R, et al. (2015), S. Pattem (2012),

CONTACT Jingjing Hu hujingjing@bit.edu.cn

K. S. Barsim, R. Streubel, and B. Yang (2014)) do not need to train the power consumption data of individual appliance; instead, they cluster aggregated meter readings directly. However, the performance of unsupervised algorithms is often not as good as that of supervised algorithms because unsupervised methods require manual labels after the learning stage, which may result in inaccurate and inefficient disaggregation results. Therefore, we focus on a supervised energy disaggregation scheme.

Most supervised energy disaggregation algorithms utilize sparse coding (Kolter, J. et al. (2010), Elhamifar, Ehsan, and Shankar Sastry (2015), Wang, D. et al. (2017)), in which the power consumption of each individual appliance over a long period is used to model a sparse linear mixture of the elements in a learned feature dictionary. The drawback is that they require a large training set to determine the status of each appliance for each period by analyzing all meter readings and the time complexity for classification is very high. The PED (Elhamifar, Ehsan, and Shankar Sastry. (2015)) scheme proposes the “powerlet” dictionary, which is formed by the signature consumption pattern of individual appliance and utilizes the dissimilarity-based sparse subset selection (Elhamifar, E. et. al (2016)) algorithm to decompose the aggregated power consumption. The average accuracy of this scheme is approximately 72%, and the disaggregation time is approximately 12 s for every 15 readings. The PED scheme is efficient; however, it is based on many constrains. Thus, efficiency is unstable with some other datasets. The ESCD (Wang, D. et. al (2017)) scheme proposes an efficient sparse coding-based framework that utilizes the fast search and finds density peaks (FSFDP) clustering algorithm (Rodriguez, A., and Laio, A. (2014)) to learn an appliance’s feature matrix and the max-min pruning matching (MMPM) optimization algorithm to decompose the aggregate consumption data. This method achieves 77% accuracy within a 10 s disaggregation time for 20 meter readings in each sliding window. The ESCD scheme promotes PED's performance; however, a series of parameters of the FSFDP and MMPM algorithms must be set manually, which results in poor stability and extensibility. Thus, we propose an efficient supervised energy disaggregation scheme. The primary contributions of this study are as follows.

(1) We utilize the FSFDP clustering algorithm to cluster individual appliance consumption data twice to learn an appliance’s feature matrix. In the first clustering procedure, we preprocess an individual appliance’s consumption data to automatically determine a number of cluster centers that is consistent with the number of different appliance state modes, which are manually set as a fixed value for all appliances in the ESCD scheme.

(2) We improve the MMPM algorithm, which greatly improves the efficiency of the decomposing process, in the decomposing process for aggregated power consumption.

(3) Experiments performed using the public reference energy disaggregation dataset (REDD) (Kolter, J.Z. et al. (2011)) demonstrate that the proposed scheme can greatly reduce disaggregation time and improve disaggregation accuracy.

2 PRELIMINARIES

THE core idea of the FSFDP clustering algorithm is to select cluster centers that with higher densities than their neighbors and the distance from other data points with higher density is relatively large.

Here, assume that dataset $S = \{x_i\}_{i=1}^{N}$ represents the data points to be clustered, $I_S = \{1,2,\ldots,N\}$ is the set of relative indices, and d_{ij} is a defined distance between data points x_i and x_j. For each data point x_i, the FSFDP clustering algorithm computes its local density ρ_i and distance δ_i from points with higher densities.

Note that ρ_i can be calculated using either a cut-off or Gaussian kernel as shown in Equation (1) and (2), respectively.

$$\rho_i = \sum_{j \in I_S \{i\}} \chi(d_{ij} - d_c) \quad \chi(x) = \begin{cases} 1, x < 0 \\ 0, x \geq 0 \end{cases} \tag{1}$$

$$\rho_i = \sum_{j \in I_S \{i\}} e^{-(\frac{d_{ij}}{d_c})^2} \tag{2}$$

Here, $d_c > 0$ is a predefined cut-off distance. For the cut-off kernel computation, ρ_i is equal to the number of points closer than d_c is to point x_i. For the Gaussian kernel computation, the value of ρ_i increases with an increasing number of data points that are less than d_c from x_i. The ρ_i using a cut-off kernel takes a discrete value, and ρ_i using a Gaussian kernel takes a continuous number. Therefore, ρ_i of a cut-off kernel has a lower probability of conflict (i.e., different data points having the same value of ρ).

Let $\{q_i\}_{i=1}^{N}$ descend sort, and is the subscript indexes of $\{\rho_i\}_{i=1}^{N}$, i.e., $\rho_{q_1} \geq \rho_{q_2} \geq \cdots \geq \rho_{q_N}$. δ_i is calculated as follows.

$$\delta_i = \begin{cases} \min\limits_{j \in I_S^i}, I_S^i \neq \emptyset \\ \max\limits_{j \in I_S}, I_S^i \neq \emptyset \end{cases} \tag{3}$$

Here, $I_s^i = \{k \in I_s : \rho_k > \rho_i\}$. Equation (3) shows that δ_i is obtained through computing the minimum distance between point x_i and any of other points whose density is higher. When x_i has the highest local density (ρ_i), δ_i represents the distance between x_i and the data points from S that are the most distant from

x_i; otherwise, δ_i is the shortest distance between x_i and any other data points with a higher density.

With the FSFDP algorithm, we select cluster centers with the greater ρ_i and δ_i values.

Next, we introduce the process of the clustering algorithm.

We assume dataset $S = \{x_i\}_{i=1}^{N}$ represents the data points to be clustered. Here there are $n_c(\geq 1)$ clusters in total. $\{m_j\}_{j=1}^{n_c}$ demonstrates the number of data points corresponding to the cluster centers, i.e., X_{m_j} is the $j-$th cluster center. $\{c_i\}_{i=1}^{N}$ denotes the clusters, where c_i is the $i-$th data point in S belonging to the $c_i - th$ cluster. $d_{max} = {}^{max}_{i<j}\{d_{ij}\}$ represents the distance between the two most distant data points from S.

$\{n_i\}_{i=1}^{N}$: n_i is the number of data points in S whose ρ is greater than that of x_i and is closest to x_i. The definition is expressed as follows.

$$n_{q_i} = \begin{cases} arg \min_{q_j j<i}\{d_{q_i q_j}\}, i \geq 2; \\ 0, \qquad\qquad i = 1. \end{cases} \quad (4)$$

Here, $\{q_i\}_{i=1}^{N}$ is as defined previously.

$\{h_i\}_{i=1}^{N}$ represents the cluster halo or cluster core. All data points in a cluster belong to the cluster core or cluster halo. Note that the cluster halo represents data points with greater ρ_i. If $h_i = 1$, then X_i belongs to the cluster halo; otherwise, X_i belongs to the cluster core.

The FSFDP algorithm proceeds as follows.

Step 1. Initialization and preprocessing.

1.1 Determine parameter $t \in (0,1)$ corresponding to cut-off distance d_c.

1.2 Compute d_{ij}, where $d_{ij} = d_{ji}, i < j, i, j \in I_S$.

1.3 Determine cut-off distance d_c as follows. Arrange all d_{ij} values calculated in the previous step in ascending order to obtain sequence $d_1 \leq d_2 \leq \cdots \leq d_M$. Let $d_c = d_{f(Mt)}$, where f(Mt) is an integer that is half adjust by Mt.

1.4 Compute $\{\rho_i\}_{i=1}^{N}$ and $\{q_i\}_{i=1}^{N}$ in descending order, which is the subscript indexes.

1.5 Compute $\{\delta_i\}_{i=1}^{N}$ and $\{n_i\}_{i=1}^{N}$ as follows:

$n_i = 0, i \in I_S$;
for $i = 2, ..., N$ **do**
  $\delta_{q_i} = d_{max}$
    for $j = 1, 2, ..., i-1$ **do**
      if $dist(X_{q_i}, X_{q_j}) < \delta_{q_i}$
        $\delta_{q_i} = dist(X_{q_i}, X_{q_j})$
        $n_{q_i} = q_j$;
      end if
    end for
end for
$\delta_{q_1} = \max_{j\geq 2} \delta_j$

In this process, function $\mathrm{dist}(X_{q_i}, X_{q_j})$ yields the distance between X_{q_i} and X_{q_j}.

Step 2. Determine the set of cluster centers $\{m_j\}_{j=1}^{n_c}$ and initialize $\{c_i\}_{i=1}^{N}$ as follows.

$$c_i = \begin{cases} k & X_i\, is\, the\, clusterig\, center\, of\, kth\, cluster \\ -1 & otherwise \end{cases} \quad (5)$$

Step 3. Cluster all data points which are not belonging to cluster centers.

for $i = 2, ..., N$ **do**
  if $(c_{q_i} = -1 c_{q_i} = c_{n_{q_i}})$
  end if
end for

Step 4. If $n_c > 1$, classify all the data points in each cluster to "cluster core" or "cluster halo".

4.1 Initially set $h_i = 0, where\ i \in I_S$.

4.2 Compute $\{\rho_i^b\}_{i=1}^{n_c}$ for each cluster as follows:

$\rho_i^b = 0, i = 1, 2, ..., n_c$;
for $i = 2, ..., N-1$ **do**
  for $j = i+1, i+2, ..., N$ **do**
    if $(c_i \neq c_j\ and\ dist(X_i, X_j) < d_c)$
      $\bar{\rho} = \frac{1}{2}(\rho_i + \rho_j)$;
      if$(\bar{\rho} > \rho_{ci}^b)\rho_{ci}^b = \bar{\rho}$
      end if
      if$(\bar{\rho} > \rho_{cj}^b)\rho_{cj}^b = \bar{\rho}$
      end if
    end if
  end for
end for

4.3 Compute the cluster halo as follows:

for $i = 1, ..., N$ **do**
  if $(\rho_i < \rho_{c_i}^b)$
    $h_i = 1$
  end if
end for

3 SCHEME

3.1 The framework of the scheme

THE framework of our proposed energy disaggregation scheme will be demonstrated in detail in this part. In the proposed scheme, a dataset containing both the total power signal and the individual appliance's signal is utilized. Here, N denotes the number of appliances, $x_i(t)$ is the power signal of the $i-$th appliance at time t (using $x_i(t)$, we know the usage condition of the $i-$th appliance), and $y(t)$ is the aggregated power signal at time t. We obtain the following equation.

$$y(t) = \sum_{i=1}^{N} x_i(t)$$

The proposed scheme attempts to recover the electricity consumption signal of each appliance, namely, infer $x_i(t)\ i \in \{1,2,\dots,N\}$ through the aggregated power consumption $y(t)$.

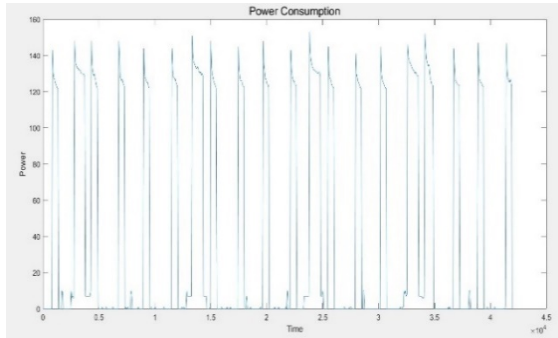
Here, T is assumed to be the length of the training data's time, w represents the length of a sliding window, where $w \ll T$, and we represent the aggregated signal and each appliance signal with w-dimensional vectors $T_w(\mathrm{y(t)}) = (y(t), y(t+1), \dots, y(t+w-1))$ and $T_w(x_i(t)) = (x_i(t), x_i(t+1), \dots, x_i(t+w-1))$ in the interval $[\mathrm{t}, \mathrm{t}+\mathrm{w}-1]$. $B_i \in R^{M_i \times w}$ represents the feature matrix of the $i-$th appliance, where M_i is the number of features of the $i-$th appliance, i.e., the number of rows of B_i. If we can obtain an appropriate feature matrix, we can approximate the ith appliance as $T_w(x_i(t)) \approx B_i c_i(t)$, where $c_i(t)$ is activation of feature matrix B_i and $c_i(t)$ is a sparse vector with mostly 0 elements and only one 1 elements.

We employ a two-step process to complete the energy disaggregation task. In the first step, we learn a feature matrix B_i for each appliance. Here, we use the improved semi-automatic FSFDP clustering algorithm. In the second step, we use the improved MMPM algorithm and utilize the appliances' feature matrices to decompose the aggregated signal data to obtain $c_i(t)$.

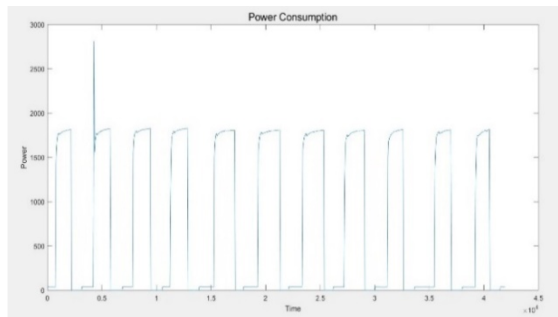
3.2 *Learning appliances' feature*

In the proposed scheme, we assume the length of the training time for the appliance signal is T and w is the length of a sliding window; therefore, the size of a sliding window can be computed as $(\mathrm{T}-\mathrm{w}+1)$. In this study, all signal data were processed in units of sliding windows, i.e., we consider the sliding window of the data as data points Q_i, where $\mathrm{i} \in \{1,2,\dots,\mathrm{T}-\mathrm{w}+1\}$. We go through all data points Q_i, remove duplicate points, form a dataset Q_{uni} with L unique data points, and record their repeated times using L-dimensional vector $\mathrm{S} = \{s_1, s_2, \dots, s_L\}$. We define the distance matrix for these L unique data points as $\mathrm{D} = \{d_{ij}, \mathrm{i,j} \in \{1,2,\dots,\mathrm{L}\}\}$, where $d_{ij} = \|Q_i - Q_j\|_2$. We then obtain the feature matrix of each appliance using the FSFDP clustering algorithm. Here, every feature matrix is obtained by executing the FSFDP algorithm twice. We discuss these two processes in the following.

In the first clustering process, we compute ρ_i, δ_i for each data point Q_i $\mathrm{i} \in \{1,2,\dots,\mathrm{L}\}$ of the target appliance (ρ_i is computed as a Gaussian kernel in the proposed scheme). The FSFDP clustering algorithm selects data points with greater ρ_i and δ_i values as cluster centers. However, the number of cluster centers cannot be determined automatically. In the ESCD (Wang, D. et al. (2017)) scheme, a fixed m value is determined for all appliances, which is unreasonable and inflexible. To address this issue, we observe the appliance signal data and find that, for each appliance, the same operation state produces similar fluctuations in the power consumption process (**Figure 1**). Thus, the value of m is more reasonable as the number of cluster centers than that in the ESCD scheme.



(a)



(b)

Figure 1. Power consumption signals of (a) a refrigerator and (b) a heat pump

Algorithm 1 Data Preprocessing

Input: Appliance training data set D_{In},
Output: The number of data states m.
1. Compute the number of rows P_{num} of D_{In}.
2. Set $m = 1$, $stm = [\ \ ]$, $stm[1] = D_{In}^1$.
3. **for** $i = 2, \dots, P_{num}$ **do**
4. **if** $(D_{In}^i < D_{In}^{(i-1)} \cdot 0.5 \parallel D_{In}^i > D_{In}^{(i-1)} \cdot 1.5)$
5. **for** $j = 1, 2, \dots, m$ **do**
6. **if** $(D_{In}^i > stm[j] \cdot 0.5 \parallel D_{In}^i < stm[j] \cdot 1.5)$
7. D_{In}^i belongs to the $j-th$ class, and calculate their average value
8. **else**
9. continue the loop until $j = m$, $m = m + 1$ D_{In}^i is a new class stm[m]=D_{In}^i
10. **end if**
11. **end for**
12. **end if**
13. **end for**
14. **return** m;

We employ a data preprocessing algorithm (**Algorithm 1**) to deal with each target appliance to obtain the number of appliance states m. In Algorithm 1, array stm[ ] stores the data corresponding to different states. The algorithm then traverses all input data. If the value of the data is 0.5 times smaller or 1.5 times greater than the value of the previous data, then the data may represent a new state. Then, the algorithm traverses all data in stm. If the input data are 0.5 times greater or 1.5 times less than the value of a single data element in stm (denoted stm[j]), then the input data belong to the $\mathrm{j}-\mathrm{th}$ class and the algorithm calculates their average value. Otherwise, the input data represent a new state, and we set $\mathrm{m}=\mathrm{m}+1$ and add this input data to stm.

After obtaining m, we begin clustering from the point with the greater ρ_i and δ_i values. Then, cyclic clustering is performed. If the number of cluster centers is less than m, we continue the clustering process, where the ρ and δ values are reduced by 1% each iteration. The algorithm will de ended until the number of cluster centers is greater than or equal to m. After the first clustering process, we classify all data points into m classes. Consequently, the cluster centers obtained in the first process are the data points with greater distances, which may result in some appliance features denoted by some data points with a smaller distance to be masked. Therefore, we perform the second clustering process to obtain a better result.

In the second clustering process, each class obtained in the first clustering process is further subdivided by computing the frequency of occurrence of duplicate data points for the given class. For each class $C_i\ i \in \{1,2,\ldots,m\}$, we assume p_i is the probability of the occurrence of duplicate points. Here, $p_i=\frac{num(C_i)}{\sum_{i=1}^{L} r_i}$, where $\mathrm{num}(C_i)$ is the number of duplicate points. Finally, the number of cluster centers for each class is given as $M_i=\mathrm{m}\times\alpha\times\sum_{i=1}^{m} p_i$, where α is determined experimentally.

After performing these two clustering processes, we obtain feature matrix $B_i\ i \in \{1,2,\ldots,N\}$ for each appliance. B_i is a matrix comprising M_i row vectors. Each row vector is w-dimensional, where w is the size of the sliding window. The algorithm used to learn the appliance feature matrix is given in Algorithm 2.

Algorithm 2 Learning Feature Matrix

Input: Appliance training data set D_{In},
Sliding window size w,
Output: Appliance feature matrix B.
Call **Algorithm 1** and get the output m.
Compute Q_{uni} and S for D_{In}
Compute distance matrix D:
set $i=1, j=1$.
while$(i, j \leq size(S))$ **do**
$d_{ij}=\|Q_i-Q_j\|_2$
end while
First Clustering: FSFDP(D,m)
Second Clustering:
for $i=1,\ldots,m$ **do**
  $p_i=\frac{num(C_i)}{\sum_{i=1}^{L} r_i}$
  $M_i=m\times\alpha\times\sum_{i=1}^{m} p_i$
  FSFDP(D_i,M_i)
  Clustering result is collected to B
end for
retrun B

3.3 Total power data disaggregation

In Section 3.1, the $\mathrm{i}-\mathrm{th}$ appliance is expressed as $T_w(x_i(t)) \approx B_i c_i(t)$. If we can obtain feature matrix B_i and activation vector $c_i(\mathrm{t})$, we can obtain $x_i(t)$, which is the objective of the energy disaggregation problem. Note that ≈ indicates that we cannot find an exact solution where $T_w(x_i(t))=B_i c_i(t)$. Thus, the disaggregation task is to find an optimal solution. Therefore, given feature matrix B_i and the constraint that $\mathrm{y(t)}=\sum_{i=1}^{N} x_i(t)$, the goal is to obtain $c_i(\mathrm{t})$.

Specifically, the goal is to obtain a global optimal solution of $x_i(t)$. If the disaggregation problem involves N appliances, each appliance will have a feature matrix B_i with M_i rows. However, the time complexity to solve this problem is $\mathrm{O}(\prod_{i=1}^{N} M_i)$, which is unacceptable. Therefore, we improve the MMPM algorithm to address this disaggregation problem. The improved algorithm (Algorithm 3) involves three main steps.

Algorithm 3 Aggregate Power Data Decomposing

Input: Appliance feature matrix $B_1B_2\ldots B_N$
Test aggregate data Y.
Output: result matrix $\tilde{B}_1\tilde{B}_2\ldots\tilde{B}_N$
get the size of sliding window w,
the number of rows of each feature matrix M_i
for$i=1,\ldots,N$ **do**
  for$j=1,\ldots,M_i$ **do**
    if$(T_w(B_i^j(t))>T_w(y(t)))$
      Eliminate B_i^j and form the new B_i
    end if
  end for
end for
find the maximum element of $T_w(y(t))$,
the column number is j
for$i=1,\ldots,N$
  sort B_i in descending order according to
  the $j-th$ element
  record max_i
end for
for$i=1,\ldots,N$ **do**
  for$k=1,\ldots,size(B_i)$ **do**
    $V_{max}=T_w(y(t+j-1))-\sum_{i=1}^{N} max_i$
    if$(V_{max}>0)$
      break;
    if$(min(T_w(r(t)))<0)$
      k=k+1;
      record the feature vector to $\tilde{B}_1\tilde{B}_2\ldots\tilde{B}_N$
  end for
end for

First, the algorithm traverses all feature matrices $B_i, i \in [1, N]$, and, for each B_i, the algorithm eliminates rows where the meter reading is greater than $T_w(y(t))$ in the same column. This is performed because the reading of any appliance can never be greater than the aggregate reading at any time. This elimination operation reduces the size of the matrices significantly; thus, the time required to solve the problem is reduced.

Second, maximum pruning is performed. The maximum element of $T_w(y(t))$, whose column order is j , is obtained. Then, all rows of each matrix $B_i, i \in [1, N]$ are sorted in descending order according to the value of the $j-$th element. The maximum $j-$th element in each matrix $B_i, i \in [1, N]$ is then determined. The maximum pruning parameter is computed as follows.

$$V_{max} = T_w\big(y(t+j-1)\big) - \sum_{i=1}^{N} max_i$$

In the matching operation, if $V_{max} > 0$, the V_{max} value of the remaining loops must be also be greater, which cannot contain the optimal solutions. Therefore, we cut off the remaining loops, which will reduce matching time.

Third, minimum pruning is performed. We define a vector $T_w(r(t, t+1, \ldots, t+w-1))$ to represent the remainder power, which is computed by the aggregate power minus the corresponding value of the upper loop. If $\min(T_w(\mathrm{r(t)})) < 0$, we can cut off the invalid loop, which reduces matching time.

4 EXPERIMENTS

IN Section 3, we described the flow of the proposed scheme and demonstrated that it is theoretically feasible. Here, we evaluate the proposed scheme's feasibility in a real-world scenario.

We used the REDD (Kolter, J.Z. et al. (2011)), which is the first public dataset that contains sufficient training data to obtain appliance features. The REDD contains aggregate data for six houses and 20 appliances. The data were collected over two weeks at a frequency of 1/3 Hz. In our evaluation, data for house five was excluded because these data contained few fluctuations; thus, appliance feature matrices could not be obtained.

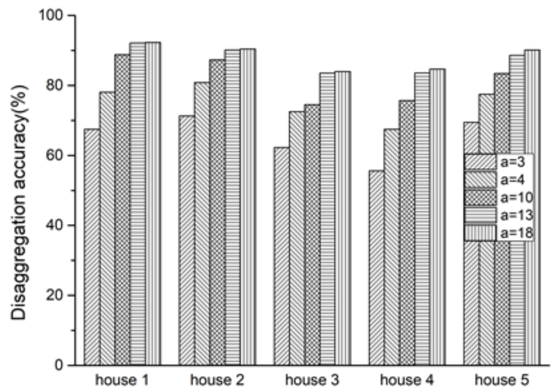
We selected five different appliances for testing. The first week of data were used to learn the feature matrix, and the remaining data were used for decomposing testing. The size of the sliding window was w = 20. Note that a larger sliding window will result in more appliance features, which may produce a more accurate disaggregation result. However, an increased number of appliance features may increase computation time. Therefore, based on a previous study (Wang, D. et al. (2017)), we set the sliding window size to 20, which is considered a compromise.

The number of features, i.e., the number of rows in each appliance feature matrix, is expressed as $M_i = m_i \times \alpha \times p_i$. Note that different α values may lead to different results.

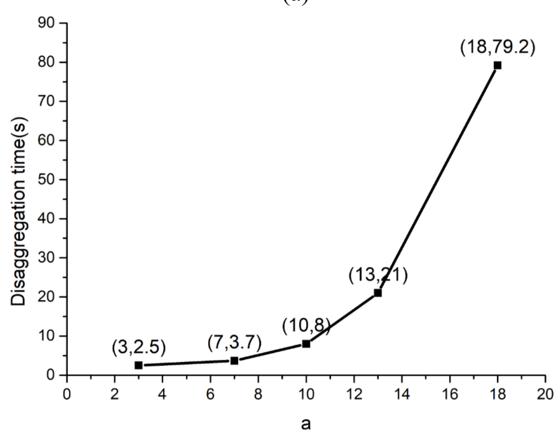
As α increases, disaggregation accuracy also increases because greater α values yield a greater Mi value, i.e., more features are acquired. More feature matrix may produce a more accurate disaggregation result; however, computation time will increase. Figure 2 shows the relationship between disaggregation accuracy and different α values. As can be seen, $\alpha = 10$ yields relatively high decomposition accuracy in a relatively short time. Thus, in our experiment, $\alpha = 10$. Note that the decomposing task for the given sliding window required approximately 8 s.

In this experiment, accuracy was calculated as follows (Elhamifar, Ehsan, and Shankar Sastry (2015)).

$$\text{energy disaggregation acc} = 1 - \frac{\sum_{t\in\varphi}\sum_{i=1}^{N}\left\|T_w(y(t)) - \tilde{T}_w(x_i(t))\right\|_1}{2\sum_{t\in\varphi}\|T_w(y(t))\|_1}$$



(a)



(b)

Figure 2. (a) Disaggregation accuracy for different α values by house and (b) disaggregation time (s) for different α values

Here, $\varphi = \{1, T_w+1, 2T_w+1, \ldots\}$, and $\tilde{T}_w(x_i(t))$ represents the power signal data of the optimal solution for the $\mathrm{i}-$th appliance. We compared the proposed scheme to the PED (Elhamifar, Ehsan, and Shankar Sastry (2015)), ESCD (Wang, D. et al. (2017)), and the naive simple mean methods. The results are shown in Table 1. As can be seen, the proposed scheme outperforms the other schemes. Compared with ESCD (Wang, D. et al. (2017)), the accuracy can be raised about 4.5% on average. Figure 3 shows the difference between an actual aggregated power signal and aggregated signal estimated by the proposed scheme. Here, the red line represents the actual waveform of the aggregated power signal and the blue line represents the estimated waveform.

Table 1. **The accuracy of energy disaggregation(%).**

	House 1	House 2	House 3	House 4	House 6	Average
Simple	41.4	39.0	46.7	52.7	33.7	42.7
PED	81.6	79.0	61.8	58.5	79.1	72.0
ESCD	84.3	82.7	70.2	71.0	78.9	77.4
ours	88.8	87.3	74.5	75.7	83.4	81.9

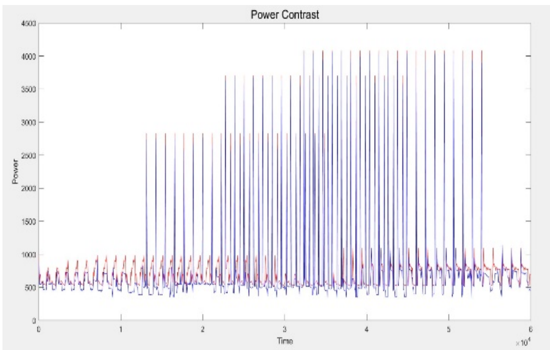


Figure 3. **Comparison of actual and estimated aggregated power consumption signals**

5 CONCLUSIONS AND FUTURE WORK

WE have proposed an efficient supervised scheme for energy disaggregation. In the proposed scheme, the semi-automatic FSFDP clustering algorithm first learns appliance feature matrices to determine the number of cluster centers, i.e., the number of the rows in the feature matrix, which improves the accuracy of the final energy disaggregation. The proposed scheme also employs an improved MMPM algorithm to perform the energy disaggregation task, which greatly reduces disaggregation time and improves efficiency. In addition, experiments using the public REDD have demonstrated the feasibility and effectiveness of the proposed scheme. The experimental results demonstrate that the proposed scheme reduces disaggregation time from 10.7 s to 8 s and increases decomposition accuracy from 77.4% to 81.9% compared with ESCD (Wang, D. et al. (2017)).

In future, to improve energy disaggregation accuracy, we plan to further improve the FSFDP clustering algorithm to fully automate determining the number of cluster centers. We also plan to find a more stable matching algorithm to replace the MMPM algorithm because its performance is overly dependent on the given dataset.

6 ACKNOWLEDGMENT

THIS work is partially supported by China National Key Research and Development Program No. 2016YFB0800301, and the National Natural Science Foundation of China No. 61772070.

7 REFERENCES

H. Altrabalsi, et al. "A low-complexity energy disaggregation method: Performance and robustness." Computational Intelligence Applications in Smart Grid (CIASG), 2014 IEEE Symposium on. IEEE, 2014.

K. S. Barsim, R. Streubel, and B. Yang, "An approach for unsupervised non-intrusive load monitoring of residential appliances," in Proceedings of the 2nd International Workshop on Non-Intrusive Load Monitoring, 2014.

M. Berges, Goldman, E.; Matthews, H.S.; Soibelman, L.; Anderson, K. User-centered non-intrusive electricity load monitoring for residential buildings. J. Comput. Civil Eng. 2011, 25, 471–480.

R. Bonfigli, et al. "Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview." IEEE, International Conference on Environment and Electrical Engineering IEEE, 2015:1175-1180.

R. Cox, Leeb, S.; Shaw, S.; Norford, L. Transient Event Detection for Nonintrusive Load Monitoring and Demand Side Management Using Voltage Distortion. In Proceedings of the 21st Annual IEEE Conference on Applied Power Electronics Conference and Exposition, New York, NY, USA, 30 August–3 September 2006; p. 7.

R. Dong, et al. "A dynamical systems approach to energy disaggregation." Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. IEEE, 2013.

E. Elhamifar and Shankar Sastry. "Energy Disaggregation via Learning Powerlets and Sparse Coding." AAAI. 2015.

E. Elhamifar, Sapiro, G., Sastry, S.S.: Dissimilarity-based sparse subset selection. IEEE Trans. Pattern Anal. Mach. Intell. 38(11), 2182–2197 (2016)

K. Ehrhardt-Martinez, Donnelly, K.A., Laitner, J.A. Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities; Technical Report E105 for American Council for an Energy-Efficient Economy (ACEE): Washington, DC, USA, 2010.

K. Gai, M. Qiu, M. Liu, and Z. Xiong, In-memory big data analytics under space constraints using dynamic programming, Future Generation Computer Systems, pp (99), 2018.

S. Giri, Berges, M. A Study on the Feasibility of Automated Data Labeling and Training Using an EMF Sensor in NILM Platforms. In Proceedings of the 2012 International EG-ICE Workshop on Intelligent Computing, Munich, Germany, 4–6 June 2012.

Z Guan, J Li, L Wu, Y Zhang, J Wu, and X Du, "Achieving Efficient and Secure Data Acquisition for Cloud-supported Internet of Things in Smart Grid," IEEE Internet of Things Journal, Vol. 4, Issue 6, pp. 1934-1944, Dec. 2017

Z Guan, G Si, X Zhang, L Wu, N Guizani, X Du, and Y Ma, "Privacy-preserving and Efficient Aggregation based on Blockchain for Power Grid Communications in Smart Communities," IEEE Communications Magazine, Vol. 56, Issue 7, pp. 1-7, Jul. 2018.

G. Hart. 1992. Nonintrusive appliance load monitoring. Proceedings of the IEEE 80(12).

S. S. Hosseini, Agbossou, K., Kelouwani, S., & Cardenas, A. (2017). Non-intrusive load monitoring through home energy management systems: A comprehensive review. Renewable and Sustainable Energy Reviews, 79, 1266-1274.

Y. Jin, Tebekaemi, E.; Berges, M.; Soibelman, L. Robust Adaptive Event Detection in Non-Intrusive Load Monitoring for Energy Aware Smart Facilities. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, 22–27 May 2011; pp. 4340–4343.

H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements." in SDM. SIAM / Omnipress, 2011, pp. 747–758.

J. Z. Kolter and Tommi Jaakkola. "Approximate inference in additive factorial HMMs with application to energy disaggregation." Artificial Intelligence and Statistics. 2012.

J. Z. Kolter, Siddharth Batra, and Andrew Y. Ng. "Energy disaggregation via discriminative sparse coding." Advances in Neural Information Processing Systems. 2010.

J. Z. Kolter, Johnson, M.J.: REDD: a public data set for energy disaggregation research. In: Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA, vol. 25, pp. 59–62 (2011)

L. Mauch and Bin Yang. "A new approach for supervised power disaggregation by using a deep recurrent LSTM network." Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on. IEEE, 2015.

H. Najmeddine, El Khamlichi Drissi, K.; Pasquier, C.; Faure, C.; Kerroum, K.; Diop, A.; Jouannet, T.; Michou, M. State of Art on Load Monitoring Methods. In Proceedings of the 2nd IEEE International Conference on Power and Energy Conference, Johor Bahru, Malaysia, 1–3 December 2008; pp. 1256–1258.

O. Parson, S. Ghosh, M. Weal, and A. Rogers, "An unsupervised training method for non-intrusive appliance load monitoring," Artificial Intelligence, no. 217, pp. 1–19, August 2014.

S. N. Patel, Robertson, T.; Kientz, J.A.; Reynolds, M.S.; Abowd, G.D. At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line In Proceedings of the 9th International Conference on Ubiquitous Computing, Innsbruck, Austria, 16–19 September 2007; pp. 271–288.

S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in Machine Learning and Applications (ICMLA), 2012 11th International Conference on, vol. 2. IEEE, 2012, pp. 515–520.

Y. Peng, Dingyue Chen, Lihao Chen, Jiayu Yu, and Mengjie Bao. The Machine Learning based Finite Element Analysis on Road Engineering of Built-in Carbon Fiber Heating Wire. Intelligent Automation & Soft Computing, 2018: 531-539.

L. Raju, Milton R S, Mahadevan S. Application of Multi Agent Systems in Automation of Distributed Energy Management in Micro-grid using MACSimJX[J]. Intelligent Automation & Soft Computing, 2017: 1-9.

A. Rodriguez & Laio, A. (2014). Clustering by fast search and find of density peaks. Science, 344(6191), 1492-1496.

A. G. Ruzzelli, Nicolas, C.; Schoofs, A.; O'Hare, G.M.P. Real-Time Recognition and Profiling of Appliances through a Single Electricity Sensor. In Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Boston, MA, USA, 21–25 June 2010; pp. 1–9.

A. Schoofs, Guerrieri, A.; Delaney, D.; O'Hare, G.; Ruzzelli, A. ANNOT: Automated Electricity Data Annotation Using Wireless Sensor Networks. In Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks, Boston, MA, USA, 21–25 June 2010; pp. 1–9.

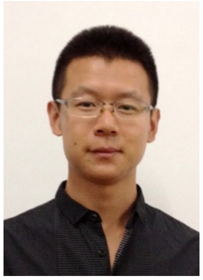
D. Wang, He, J., Rahim, M. A., Zhang, Z., & Zhu, L. (2017, December). An Efficient Sparse Coding-Based Data-Mining Scheme in Smart Grid. In International Conference on Mobile Ad-Hoc and Sensor Networks (pp. 133-145). Springer, Singapore.

M. Wytock, and J. Zico Kolter. "Contextually Supervised Source Separation with Application to Energy Disaggregation." AAAI. 2014.

L. Yeqing, Zhongxing Peng, Junzhou Huang, Zhilin Zhang, and Jae Hyun Son. Energy Disaggregation via Hierarchical Factorial HMM. In Proceeding of the 2014 NILM Workshop, pages 1-4, 2014.

A. Zoha, Gluhak, A., Imran, M. A., & Rajasegarar, S. (2012). Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. Sensors, 12(12), 16838-16866.

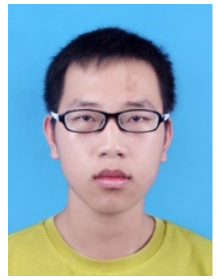
8 NOTES ON CONTRIBUTORS



Weilie Liu is a master student in the School of Computer Science and Technology, Beijing Institute of Technology, China. He received his BEng degree from PLA Information Engineering University in 2011. His research interests include NILM and Network Security.

Jialing He is currently working towards the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology, China. She received her BEng degree and master degree from Beijing Institute of Technology in 2016 and 2018. Her research interests include cloud security, smart grid, services computing and machine learning.

Meng Li is currently working towards the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology and he is a visiting Ph.D. student in Wilfrid Laurier University. He has received the National Graduate Student Scholarship in 2011 and he has been sponsored by China Scholarship Council in 2017. His research interests are applied cryptography, security and privacy, crowdsensing, fog computing and blockchain.



Rui Jin is a master student in the School of Computer Science and Technology, Beijing Institute of Technology, China. His research interests include NILM and Network Security.

Jingjing Hu received the Ph.D. degree from the School of Computer Science and Technology, Beijing Institute of Technology, China. She is currently an Associate Professor in the School of Computer Science and Technology, Beijing Institute of Technology, China. Her research interests are in the areas of web intelligence, parallel computing and cloud security.

Zijian Zhang is an Associate Professor in the School of Computer Science and Technology at Beijing Institute of Technology. He was a visiting scholar in the Computer Science and Engineering Department of the State University of New York at Buffalo in 2015. His research interests include cryptography, differential privacy, smart grid, data privacy, mobile security and data mining.