Real-time Energy Disaggregation at Substations with Behind-the-Meter Solar Generation

Wenting Li, *Member, IEEE,* Ming Yi, *Student Member, IEEE,* Meng Wang, *Member, IEEE*,

Yishen Wang, *Member, IEEE*, Di Shi, *Senior Member, IEEE*, Zhiwei Wang, *Senior Member, IEEE*

***Abstract*—Energy Disaggregation at substations (EDS) is challenging because measurements are mostly aggregated over multiple types of loads, and the existence of some loads such as behind-the-meter solar is unknown to the operator. This paper for the first time addresses this so-called "partial labels" issue in energy disaggregation and develops a model-free EDS method to separate individual loads, including BTM solar, from the total energy consumption in real-time. Our approach learns the patterns of all loads offline from recorded historical datasets with partial labels. Compared with conventional model-free methods that require either pure measurements of each load for training or full labels of each training sample, our method can extract load patterns from partially labeled aggregated data and thus, is more applicable to practical scenarios and alleviates the annotation burden for the operator. Specifically, we propose to solve a new dictionary learning problem, where column-sparsity and incoherence regularization terms are added to identify unlabeled loads and learn distinctive patterns of each load. In real-time disaggregation, our approach solves an improved sparse decomposition problem where one decomposes the aggregated measurements as a linear combination of some representative recorded measurements with known disaggregation learned in the offline stage. Numerical experiments are reported to validate our method.**

***Index Terms*—Energy disaggregation, behind-the-meter solar generation, dictionary learning, sparse coding**

I. Introduction

Although modern meters are installed at some residential households, commercial buildings, and the output of solar farms and wind turbines, the coverage in distribution systems is still very sparse such that the visibility is not comparable to that of transmission systems. The power consumption measured at a substation is the net consumption of all the industrial loads, residential users, and renewable (like solar or wind) generations that are connected to this substation, and the operator does not have separate measurements for each type of load[1]. The objective of energy disaggregation at the substation level (EDS) is to extract the energy consumption of each type of load from the aggregated net load measurements. EDS becomes increasingly challenging due to behind-the-meter (BTM) solar generations, which are not directly measured by the operator. Accurate real-time estimation of various loads, especially BTM solar, is necessary for distribution systems planning and operations such as hosting capacity analysis [1], distribution network reconfiguration [2], Volt Var control [3], load forecasting [4], and demand response [5].

One recent line of works estimates the BTM solar generation from the aggregate measurements. Both model-based methods and data-driven methods have been proposed. The model-based methods such as [6]–[13] usually require meteorological data and physical models of PV characteristics. The estimation accuracy largely depends on the availability and accuracy of model parameters. Data-driven methods do not require model information but require either high time-resolution measurements [9], [14]–[17] or some fully observable customers with known PV generations [18]. Moreover, all these works focus on estimating BTM solar only and does not disaggregate other loads, while profiles of different loads such as residential loads and industrial sites are needed for accurate load forecasting and demand response. A few works such as [19], [20] disaggregate multiple loads at the substation of feeder level. These methods require parametric models for all loads and the types of loads in the aggregate data need to be available.

Other related works focus on non-intrusive load monitoring, which has two directions: energy disaggregation at the household level (EDH) [21]–[26], and load classification [27]–[29]. EDH estimates the status and power consumption of different home appliances from the total household energy consumption. Existing approaches mostly require estimating the patterns of individual appliances from recorded historical data. Various methods have been explored, such as modeling using hidden Markov Chain [30], [31] or on/off steady states [32], training deep learning models [33], [34], choosing proper features based on the domain knowledge [35], and learning a discriminative dictionary of load patterns by non-negative matrix factorization [21]. Among these methods, dictionary learning-based approaches have the advantage of no modeling of the load patterns and no training of complicated learning models needed. One major challenge that limits the disaggregation accuracy is that different types of loads may share similar patterns, resulting in ambiguity in disaggregation. Various techniques have been proposed to increase disaggregation accuracy including discriminative dictionary learning

The first two authors contribute equally to the paper and they are listed in an alphabetical order.

Manuscript received March 11, 2020; revised July 10, 2020 and September 18, 2020; accepted October, 25 2020. Paper no.TPWRS-00384-2020. This work was supported in part by the SGCC Science and Technology Program. (Corresponding author: Meng Wang.)

W. Li, M. Yi, and M. Wang are with the Dept. of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. W. Li, M. Yi, Y. Wang, D. Shi, Zh. Wang are with GEIRI North America, 250 W Tasman Dr, San Jose, CA, 95134. Email:{liw14, yim3, wangm7}@rpi.edu, {yishen.wang, di.shi, zhiwei.wang}@geirina.net.

[1]Both loads and renewable generations are referred to as "load" to simplify the presentation in this paper. Renewable generation is considered as a negative load.

[21], [36], [37] and adding regularization terms [35]. All these methods require recorded data of individual loads separately to learn the respective patterns and do not generalize to the case of aggregate data of multiple types of loads.

At the substation level, most meters only measure at the net point so the obtained measurements are aggregated loads [12]. With the recent installation of smart meters at households, in principle one can disaggregate the solar generation of each house and sum them up to obtain the feeder level solar generation [11]. In scenarios when smart meters are not available or when the large number of households make solving individual EDH problems time consuming, an alternative approach is to solve the EDS problem directly using the net load measurements. Note that most electric appliances at the household level are single-state or multi-state and demonstrate repeatable characteristics [29], [38], while very few of them (like power drills and dimmer lights) have continuous power draw and do not have the repeatable patterns [38]. The energy consumption measured at a substation is the sum of all the loads connected and often not repeatable, making the disaggregation problem challenging [38]. In addition, it may be difficult to identify a set of pure measurements of a certain type of load (such as BTM solar) at the substation level to estimate its patterns. Furthermore, the approaches for EDH are often based on high resolution data (one sample every few seconds), which is very rare in the substation level [18]. Thus, the existing methods for EDH do not directly generalize to EDS. Another challenge for EDS is that although the distribution operator may know all the loads connected to a substation, it may not know whether a particular type of load is consuming energy or not within a given time interval. For instance, the operator does not know whether the BTM solar panels are generating power or not at a given time considering cloud coverage and PV malfunction, while residential loads are expected to exist between 5 pm to 10 pm. Thus, each aggregate measurement is the total consumption of multiple types of loads, and the types that exist in this measurement are not fully known. We call it "partially labeled aggregate data." One fundamental distinction of our problem from dictionary learning with incomplete label information for applications like image classification [39]–[42] is that each measurement here is the sum of energy consumption of different types of loads, and the patterns of different loads are obfuscated in the aggregate measurements.

The contributions of this paper are as follows. (1) We formulate the substation load disaggregation problem from partially labeled aggregate data, and this particular aspect of partial labels has not been studied before for load disaggregation. (2) We develop a data-driven disaggregation method to disaggregate all loads including hidden loads like BTM solar generations. Our model-free approach includes the offline learning of load patterns from partially labeled aggregate data and online disaggregation of different loads from the obtained real-time measurements. (3) In the offline training, we exploit the sparsity of unlabeled loads in the measurements to address the issue of partial labels. We also add an incoherence term to discriminate features of different loads. Compared with conventional dictionary learning methods that require either pure measurements for each type of load or fully labeled aggregate measurements, our method can extract information from aggregate measurements with partial labels. (4) In online disaggregation, we propose to decompose the total load into a sparse combination of some recorded historical data rather than the sparse combination of dictionary atoms in the conventional approach. That exploits the repetitive patterns of loads and enhances the disaggregation accuracy. Our method outperforms existing methods in numerical experiments.

The rest of the paper is organized as follows. Section II introduces our problem formulation. We present our approach in Section III, and the evaluations of our method are in Section IV. Section V concludes the paper. Table I lists the major notations in this paper.

Table I: Major notations

N	the number of training data
T	the length of time window
n	the total number of load types
X	the $T \times N$ data matrix for training
X_i	the load i component in X, $X = \sum_{i=1}^{n} X_i$
$\bar{x}^j$	the jth column of X
z^j	the indices of loads in $\bar{x}^j$
y^j	partial labels in z^j
D_i	the dictionary of load type i
A_i	the coefficients that correspond to load type i, $X_i = D_i A_i$
K_i	the number of columns in dictionary D_i
d^m	the mth column of $D = [D_1, ..., D_n]$
x	the testing aggregate data

II. Problem Formulation and Discussion

A. Problem Formulation

We assume that n types of loads in total are connected to a substation. The consumption[2] of a load is positive if it consumes power and is negative if generates power. Here, the consumption is the total value of all the users of the same type. For example, the consumption of the residential load is the total consumption of some residential users. At a given time interval, some types of loads may not appear, e.g., solar panels are not producing power at night. Let $\bar{X} \in \mathbb{R}^{T \times N}$ contain the recorded measurements of real power[3] consumption in a time window of length T at N different time intervals. The jth column of $\bar{X}$, denoted by $\bar{x}^j \in \mathbb{R}^T$, represents a time series of the aggregate consumption of all loads during T consecutive time steps. Let set $z^j \subseteq \{1, ..., n\}$ denote the indices of loads that appear in $\bar{x}^j$. The operator may only know partial labels in z^j. Let $y^j \subseteq z^j$ denote the types of loads that are known to exist in $\bar{x}^j$ to the operator. If the operator does not know which load or loads out of n total loads are nonzero in $\bar{x}^j$, then y^j is an empty set.

Our proposed model characterizes the two properties of energy consumption at the substation. First, each time series

[2]We remarked that the power loss in the distribution systems is not directly modeled in this paper. The energy consumption discussed here is the net consumption observed at a substation and can include both the actual load consumption and some power loss in the system.

[3]We focus on real power because oftentimes only real power is measured. Our approach can also disaggregate loads based on reactive power if measures are available. If both real and reactive power consumptions are measured, our method can be easily modified by using a complex value to represent the power consumption.

of energy consumption in a time interval is usually aggregated over multiple types of loads rather than measuring individual loads only. Second, at the substation level, it is sometimes difficult to know if a certain type of load, such as BTM solar, generates/consumes a significant amount of energy or not during a given time interval. The distribution operator may have partial information of the loads that exist.

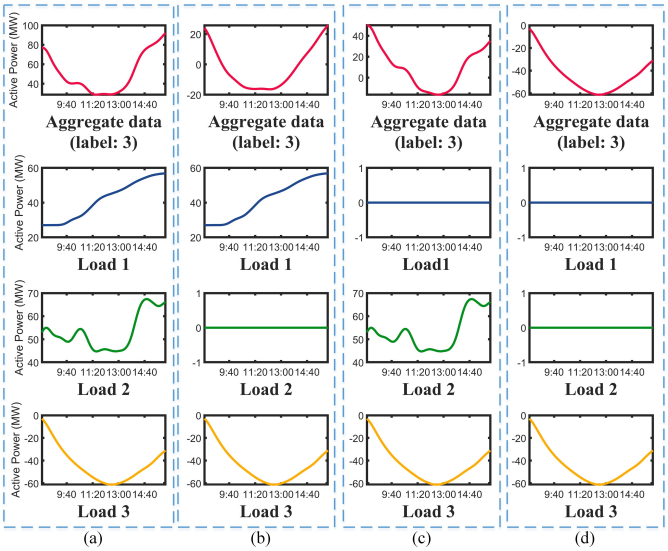


Fig. 1: An example of partially labeled aggregate data. The aggregate data may contain load 1, load 2 and 3. All four aggregate data are labeled as load 3. (a) all loads exist; (b) load 1 and 3 exist; (c) load 2 and 3 exist; (d) only load 3 exists.

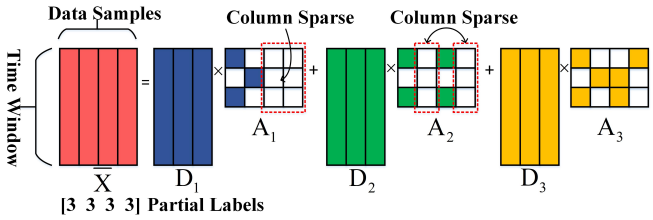


Fig. 2: Decomposition of data in Fig. 1. A_1 and A_2 are column-sparse. Load 3 is labeled in all four data samples, while load 1 and load 2 are not labeled in any data.

The example in Fig. 1 illustrates our model. The time window is 8:00 am-4:00 pm with a 5-minute resolution. There are three types of loads, two industrial loads and one solar generation. Each column shows the aggregate data and the corresponding individual loads at one time window. These four different aggregate data are all labeled as load 3 (Solar), although the other two loads may also exist. The existence of loads 1 and 2 in these data are not known from the partial labels. Note that in practice, the power consumption of a load type might not be exactly zero in a time interval in Fig. 1. In this paper, we treat loads consuming a very small amount of power during a time interval as zero when disaggregating the loads.

We assume the time series of load i share common patterns. Let $D_i^* \in \mathbb{R}^{T\times K_i}, i = 1,\cdots,n$ contain K_i representative patterns of load i. Then the energy consumption of load i in a window of length T can be written as a linear combination of columns in D_i^*. Specifically, we have[4]

$$\bar{X} = \Sigma_{i=1}^{n}\bar{X}_i = \Sigma_{i=1}^{n} D_i^* A_i^*, \tag{1}$$

where $\bar{X}_i \in \mathbb{R}^{T\times N}$ denotes consumption of load i, and the jth column of $A_i^* \in R^{K_i\times N}$ denotes the pattern coefficients of load i in $\bar{x}^j$. For example, the dictionary representation of the four data samples in Fig. 1 is illustrated in Fig. 2. D_i represents the dictionary of load i, and we show three dictionary atoms for simplification. Because load 1 does not exist in Fig.1 (c) and (d), the third and fourth columns of A_1 in Fig.2 are all zeros, i.e., column-sparse. Similarly, the second and fourth columns of A_2 are all zeros. Because load 3 always exists, A_3 is not column sparse. Note that although some loads such as solar generation may be variable and intermittent, a large number of time series indeed share a relatively small number of dominant patterns. For instance, dictionary learning algorithm has been exploited to learn the patterns of solar generation for load forecasting [43].

Given $\bar{X}$ with partial labels, *the objective of this paper* is to develop a disaggregation method such that for any aggregate measurements in real-time, denoted by $x \in \mathbb{R}^T$, one can estimate [5] $x_i \in \mathbb{R}^T$ $(i = 1, ..., n)$ where x_i is the consumption of load i, and $x = \sum_{i=1}^{n} x_i$.

Note that if load l does not appear in any of the partial labels of the training data, it is very challenging to identify and disaggregate load l, because the patterns of load l are all masked by other loads, and its existence is unknown to the operator. For example, in Figs. 1 and 2, load 1 and load 2 are not labeled in any training data, while load 3 is labeled. It is harder to estimate them accurately than load 3.

B. Examples to Illustrate the Limits of Disaggregation

Before introducing our disaggregation method, we first use simple examples to illustrate that the disaggregation error, resulting from the similarities of load patterns of different types of loads, usually exists despite the disaggregation method.

Consider the case of two types of loads with $D_i^* \in \mathbb{R}^{T\times K_i}$ $(i = 1, 2)$. The aggregate data x can be written as

$$x = x_1 + x_2 = D_1^* a_1^* + D_2^* a_2^*, \tag{2}$$

where a_1^* and a_2^* are the corresponding coefficients for load 1 and 2, respectively. Even if the ground-truth dictionaries D_1^* and D_2^* are known, from basic linear algebra, x_1 and x_2 can be accurately estimated if and only if D_1^* and D_2^* are orthogonal to each other. To see this, consider two simple cases.

Case 1: orthogonal dictionaries. Suppose $K_1 = K_2 = 1$, and $T = 3$. Let $D_1^* = [1, 0, 0]^T$, $D_2^* = [0, 1, 0]^T$, $x = [2, 3, 0]^T$. Since D_1^* and D_2^* are orthogonal, there is one unique way to disaggregate x, i.e., $x_1 = [2, 0, 0]$, $x_2 = [0, 3, 0]$.

[4]Eq. (1) is based on the assumption that the power consumption is the linear function of representative features. This is a general assumption as the power consumption is indeed additive. The dictionary decomposition might not provide the most compact representation of load profiles. For example, if a feature is time-shifted significantly, the resulting time series will likely be treated as a different feature. That may increase the number of dictionary atoms for a given time period.

[5]To disaggregate load i from x, load i should appear in the training data.

Case 2: nonorthogonal dictionaries. Let $\beta_1 = [1,0,0]^T$, $\beta_2 = [0,1,0]^T$, and $\alpha = [0,0,1]^T$. $D_1^* = [\alpha, \beta_1]$, and $D_2^* = [\alpha, \beta_2]$. D_1^* and D_2^* are nonorthogonal due to the common dictionary atom α. Say $x = [2,3,4]^T$, then there exists an infinite number of ways to decompose x with $x_1 = [2,0,c_1]$, $x_2 = [0,3,c_2]$, as long as $c_1 + c_2 = 4$. Without further information, one cannot accurately disaggregate x, and the aggregation error is inevitable.

One can see from the above discussion that dictionaries of different types of loads should be orthogonal to each other to avoid disaggregation error. In practice, however, this condition is rarely met due to the similarities in different load patterns. Thus, the disaggregation error is inevitable despite the disaggregation method employed. The objective of this paper is to **reduce but not eliminate** the disaggregation error.

C. Related Works

Our disaggregation approach follows the line of research [21], [37] that first learns the dictionaries from the training data and then exploits the dictionaries to disaggregate the real-time aggregate measurements. We defer the discussion of our proposed approach to Section III and first introduce the conventional dictionary learning and sparse decomposition approaches that have been studied in face recognition [44], [45], image denoising [46], [47] and energy disaggregation [21], [37].

Given a data matrix $X \in \mathbb{R}^{T\times N}$, the objective of dictionary learning is to learn a dictionary $D = [d^1, \cdots, d^K] \in \mathbb{R}^{T\times K}$ and a sparse coefficient matrix $A \in \mathbb{R}^{K\times N}$ such that $X = DA$. This can be achieved by solving

$$\min_{D,A} \|X - DA\|_F^2 + \lambda \|A\|_1 \tag{3}$$

$$\text{s.t } \|d^m\|_2 = 1, m = 1, 2, \cdots, K, \tag{4}$$

where $\|A\|_1 = \Sigma_{j=1}^N \|a^j\|_1$ is a regulation term that promotes a sparse solution. a^j denotes the jth column of A, and $\|a^j\|_1$ is the ℓ_1-norm of vector a^j. d^m denotes the mth column of D, and each dictionary atom is normalized to one.

A dictionary is overcomplete if the number of dictionary atoms K is much larger than the dimension T. A dimension overcomplete dictionary is desirable to reduce the reconstruction error $\|X - DA\|$. However, for tasks like classification and disaggregation, an overcomplete dictionary may have similar dictionary atoms for different classes, and that reduces the classification and disaggregation accuracy [48]. Discriminative dictionary learning aims to learn a dictionary such that the dictionary atoms corresponding to different classes are as separate as possible [21], [45], [49]. For instance, one approach is to add the constraint that coefficients corresponding to the dictionary atoms of class j ($j \neq i$) should be close to zero for training data labeled with class i. It can enhance the classification accuracy but require training data from each individual class.

After obtaining the estimated dictionary, denoted by $\hat{D}$, by solving (3)-(4), sparse decomposition wants to find a sparse representation of testing data $x \in \mathbb{R}^T$ with respect to the dictionary $\hat{D}$. This can be achieved by solving

$$\min_{a\in\mathbb{R}^K} \|x - \hat{D}a\|_F^2 + \mu\|a\|_1, \tag{5}$$

where a is a vector that represents the dictionary coefficients of the testing data x.

When dictionary learning is applied to energy disaggregation, the dictionary $\hat{D}$ contains the representative features of all n loads, with each column being a representative temporal feature of a certain load. $\hat{D}$ is learned from (3)-(4) using N recorded time series of T consecutive time instants. Let $\hat{a}$ be the solution to (5). $\hat{a}$ is usually sparse as not every feature will be present in x. Then the load i consumption in x can be estimated by the weighted sum of all load i features in $\hat{D}$ where the weights are the corresponding coefficients in $\hat{a}$.

Due to the issue of "partially labeled aggregate data," conventional dictionary learning cannot be directly applied to solve EDS. Because the measurements include multiple loads, and load patterns usually share some similar components, a dictionary learned from (3)-(4) often contains similar features for different loads, leading to a significant disaggregation error. This paper will develop a new disaggregation method to improve the disaggregation accuracy by exploiting the information in the partial labels.

D. Significance and Applications of the Problem Formulation

At the substation level, the power consumption is usually aggregated over multiple types of loads. Even when the measurements contain only one type of load during a certain time window, the operator may not know such information. Therefore, it is difficult to obtain measurements of individual loads as needed in the conventional EDH methods. On the other hand, one can in principle learn patterns of all loads from aggregated data simultaneously using conventional dictionary learning methods, but that requires accurate labels z^j for all training data $\bar{x}^j$. Finding full labels of all the training samples requires a lot of manpower. Moreover, missing labels and wrong labels are almost inevitable. Our problem formulation relaxes the requirement on the training data such that aggregate measurements with only partial labels can be leveraged for training. That reduces the burden for the annotator significantly.

To obtain labels, one can design a detector for each type of load separately. Given a training sample $\bar{x}^j$, all these detectors can be applied in parallel to provide labels for $\bar{x}^j$. For example, references [50], [51] design detectors for the existence of solar data from the aggregated measurements. To design a detector for load i, some aggregated data without load i are needed [50], [51]. These measurements can be obtained by communicating with this load to identify some time periods when this load is off. Note that if one wants to directly label all the loads in all the training samples, the operator needs to communicate with all n loads at the same time for each training sample. Here, the operator only needs to communicate to one load to obtain the schedule that it is off for designing the detector of this load. When the number of training samples

is large, our approach costs much less in communication and manpower.

The obtained labels are often partial labels rather than full labels because a detector often mis-detects the existence of some loads [52], especially when the consumption of this type of load is relatively low compared with other loads. Moreover, a detector of some loads might not be available if no data are available to design the detector.

III. Energy Disaggregation using Partially Labeled Aggregate Data

Our proposed energy disaggregation approach includes offline learning of load patterns of loads from partially labeled aggregate data and online disaggregation of obtained aggregate measurements, presented in Sections III-A and III-B, respectively.

A. Offline pattern estimation from recorded data

Given recorded data $\bar{X}$, we propose to estimate the patterns of all types of loads by solving the following optimization problem.

$$\min_{A,D} f(A,D) = \|\bar{X} - \Sigma_{i=1}^{n} D_i A_i\|_F^2 + \Sigma_{i=1}^{n} \lambda_i \Sigma_{j:i\notin y_j} \|A_i^j\| + \lambda_D \mathrm{Tr}(D\Theta D^\intercal) \quad (6)$$

$$\text{s.t. } \|d^m\|_2 \le 1, d^m \ge 0, m = 1, \cdots, K \quad (7)$$

$$c_i A_i \ge 0, \forall i \quad (8)$$

where $D_i \in \mathbb{R}^{T\times K_i}$ contains K_i representative time series for load i in T time instants, and $A_i \in \mathbb{R}^{K_i\times N}$ represents the coefficients of these representative features of load i in all the N time series in $\bar{X}$. A_i^j is the jth column of A_i. $K = \Sigma_{i=1}^{n} K_i$. $A = [A_1; A_2; \cdots; A_n] \in \mathbb{R}^{K\times N}$ represents all coefficients. $D = \big[D_1, D_2, \cdots, D_n\big] \in \mathbb{R}^{T\times K}$ represents the dictionary of all loads. $\mathrm{Tr}(\cdot)$ is the trace of a matrix, and $D^\intercal$ is the transpose of D. λ_j, and λ_D are given positive regularization parameters.

The first term $\|\bar{X} - DA\|_F^2$ of $f(A,D)$ measures the approximation error to the recorded data $\bar{X}$ by the learned dictionary D and the corresponding coefficients A. $\Sigma_{j:i\notin y_j}\|A_i^j\|$ is the sum of ℓ_2-norm of the coefficients that correspond to load i over all the training data $\bar{x}^j$ that do not have label i in the partial labels. The intuition is that if load i does not exist in $\bar{x}^j$, then the vector A_i^j shall be all zero. Since non-labeled loads may not exist in a given training data, the second term of $f(A,D)$ promotes the group sparsity of the coefficients of the non-labeled loads. The parameter λ_i depends on load i because the consumption level of loads can be very different. For instance, in the example shown in Figs. 1 and 2, because there are three loads, and all training data are labeled as load 3, the second term of (6) can be written as

$$\lambda_1(\|A_1^1\| + \|A_1^2\| + \|A_1^3\| + \|A_1^4\|) + \lambda_2(\|A_2^1\| + \|A_2^2\| + \|A_2^3\| + \|A_2^4\|).$$

The last term of (6) is an incoherence regularization term such that D_i and D_j are different from each other for any two different loads i and j. To see this, let d^m denote the mth column (dictionary atom) of D. We construct a weight matrix $\Theta \in R^{K\times K}$ with the entry θ_{mp} of the mth row and pth column being value 1 if and only if d^m and d^p are not in the same dictionary D_i for any i and 0 otherwise. The incoherence regularization term is defined as

$$\mathrm{Tr}(D\Theta D^\intercal) = \Sigma_{m=1}^{K}\Sigma_{p=1}^{K}\theta_{mp}(d^m)^\intercal d^p. \quad (9)$$

(9) is zero if and only if d^m and d^p are orthogonal for every pair of atoms that are not in the same dictionary D_i for some i. As discussed in Section II-B, the dictionaries of different loads should be orthogonal to each other to avoid disaggregation error. Minimizing (9) promotes a discriminative dictionary such that the load patterns of different loads are as different as possible.

Besides the normalization constraints of the dictionary atoms, we also impose non-negative constraints on all dictionary atoms in (7). If load i consumes energy, the corresponding coefficients A_i should be non-negative. If load i provides energy, e.g., solar panels, then the corresponding A_i should be non-positive since all the dictionary atoms are non-negative. We define constants $c_i = 1$ for all load i that consumes energy and $c_i = -1$ for all load i that supplies energy. We add constraints $c_i A_i \ge 0$ for all i in (8) to characterize this property.

Let $\hat{D} = [\hat{D}_1, \hat{D}_2, ..., \hat{D}_n]$ and $\hat{A} = [\hat{A}_1; \hat{A}_2; \cdots; \hat{A}_n]$ denote the solution to (6)-(8). Then $\hat{D}_i$ includes the representative features of load i, and $\hat{A}_i$ are the corresponding coefficients. For example, the load i component in the jth training data x^j can be estimated by $\hat{D}_i\hat{A}_i^j$. $\hat{D}$ will be used for online load disaggregation. Note that $\hat{D}$ records the representative features as temporal correlations in T time instants rather than the energy magnitude. It is thus robust to changes that only affect the magnitude rather than the pattern. For instance, $\hat{D}$ remains the same when more PV panels sharing the same characteristics of power consumption are installed.

Note that our proposed approach in (6)-(8) can tolerate a small portion of wrong labels, e.g., $i \in y_j$ when load i does not exist in $\bar{x}^j$. That is because we do not impose hard constraints on the labels. The label information is used in the sparse regularization term in (6), and a certain level of inaccuracy in the regularization term can be tolerated. We also numerically verify this property in Section IV-B. Similarly, our approach can also handle the case that the training samples are fully labeled, i.e., z_j is known for all j, by replacing y_j with z_j in (6). In this case, the sparse regularization term in (6) promotes zero coefficients for load i for any i not in z_j. For example, if $n = 3$, $\bar{x}_j$ contains load 1 and 2 only, i.e., $z_j = \{1, 2\}$, then the the regularization is imposed over A_3^j for load 3. Therefore, our approach can naturally handle a mixture of training data with full labels and partial labels.

The problem (6)-(8) is nonconvex, and we solve it by alternatively updating the estimates of D and A, denoted by $\hat{D}$ and $\hat{A}$, respectively. In each alternation, we first fix $\hat{D}$ and update A. When fixing $\hat{D}$, the optimization problem is convex in A and can be solved by any off-the-shelf convex solvers. We then fix $\hat{A}$ and update D. Since the problem is still nonconvex in D, we update D using projected gradient descent [53]. There is an inner loop of $k_{\max}$ iterations for a predetermined constant $k_{\max}$. In the kth iteration, we update D using gradient descent, and the step size α_k is determined by "Armijo rule along the projection arc" [54]. This approach

Algorithm 1 Dictionary Learning From Aggregate Data

Input: $\bar{X}$, the regularization parameters $\lambda_i, i = 1, 2, \cdots, n$, the maximum iteration number $t_{\max}$ for the outer loop, the maximum iteration number $k_{\max}$ for the inner loop, the tolerance ε, constants β and σ in $(0, 1)$, and the initial step size α_0.
Randomly select K_l samples with partial labels containing l and normalize them as the initialization of $\hat{D}_l$. Other $\hat{D}_l$'s are randomly samples from the training data. $\hat{A}$ is initialized as a zero matrix. $t = 1$, $\Delta_f = f_{\text{pre}} = f(\hat{A}, \hat{D})$
while $|\Delta_f| \geq \varepsilon$ and $t < t_{\max}$ **do**
  Keep $\hat{D}$ fixed, update $\hat{A}$ by

$$\hat{A} = \arg\min_{A} f(A, \hat{D}) \text{ s.t. } c_i A_i \geq 0, \forall i$$

  $k = 1$;
  While $k < k_{\max}$ **do**
  $\alpha_k = \alpha_{k-1}$
    While α_k satisfies (12):
  $\alpha_k = \alpha_k / \beta$
    end while
  $\alpha_k = \alpha_k \beta$
    While α_k does not satisfy (12)
  $\alpha_k = \alpha_k \beta$
    end while.
  Set $D^{k+1} = [D^k - \alpha_k \nabla f(D)]_+$
  $k = k + 1$;
  end while
  $\hat{D} = D^{k_{\max}}$. Then normalize each column of $\hat{D}$ to be unit norm.
  $\Delta_f = f(\hat{A}, \hat{D}) - f_{\text{pre}}$
  $f_{\text{pre}} = f(\hat{A}, \hat{D})$
  $t = t + 1$
end while
Output: $\hat{D}$ and $\hat{A}$.

can guarantee the sufficient decrease of objective function with a fast searching time of the step size. Specifically, the first-order and second-order gradients of $f(\hat{A}, D)$ with respect to D are

$$\begin{aligned} g_1(D) &= \nabla_D f(\hat{A}, D) \\ &= -2\Sigma_{l=1}^{n}(\bar{X}^{(l)} - D\hat{A}^{(l)})(\hat{A}^{(l)})^\intercal + 2\lambda_D \Theta \end{aligned} \quad (10)$$

$$g_2(D) = \nabla_D^2 f(\hat{A}, D) = 2\Sigma_{l=1}^{n}\hat{A}^{(l)}(\hat{A}^{(l)})^\intercal + 2\lambda_D \Theta. \quad (11)$$

Starting from an initial guess of the step size α_k, one continuously multiplies it by either $1/\beta$ or β for a predetermined β in $(0, 1)$ while checking if the condition (12) still holds.

$$\text{(1-}\sigma\text{)} \left\langle g_1(D^k), D - D^k \right\rangle + \tfrac{1}{2} \left\langle D - D^k, g_2(D^k)(D - D^k) \right\rangle \leq 0 \quad (12)$$

where $D = [D^k - \alpha_k g_1(D^k)]_+$, $[Z]_+$ sets all the negative entries of matrix Z to zero, and $\langle \cdot, \cdot \rangle$ takes the inner product. D^k is the estimate of dictionary in the kth iteration. α_{k-1} can be used as an initial guess of α_k [55], and α_0 is given. After $k_{\max}$ iterations, we normalize the columns to be unit norm and let the result be the current estimate of the dictionary $\hat{D}$.

For initialization, if there exist some data samples containing load l as partial labels, we pick K_l time series to initialize D_l. Otherwise, D_l is initialized through randomly sampling the training data. All coefficients A are initialized to be zero. We repeat the alternating updates until the change of the objective function is less than a fixed threshold ε or the maximum number of steps $t_{\max}$ is achieved. The details of the algorithm are shown in Algorithm 1.

K_l can be selected by experience or approximated by data. For example, let $\bar{X}^{(l)}$ contain all the time series that have l as partial labels. Let $\sigma_i, i = 1, \cdots, T$ be the singular values of $\bar{X}^{(l)}$ in the descending order. K_l is determined by

$$K_l = \arg\min_{r} \frac{\Sigma_{i=1}^{r}\sigma_i}{\Sigma_{i=1}^{T}\sigma_i} \geq \tau, \quad (13)$$

$\tau \in (0, 1)$ is a pre-determined approximation accuracy. If load i is not labeled in any recorded data, one can set K_i to be the average of K_l's of all other loads l.

If τ is small, K_l is small. Each load is presented by only a few dominate patterns, and it is relatively easier to distinguish different loads. However, if τ is too small, the approximation error is large, and D_l does not characterize all the patterns of load l. Therefore, there is a trade-off between disaggregation and representation accuracy when selecting τ. We find in numerical experiments that the performance is not very sensitive to τ. The disaggregation results are similar when τ varies in a range reasonably large.

We remark that if the training data contain the pure measurements of some loads (not necessarily all types of loads), the accuracy of the learned dictionary by our method can be greatly enhanced. In fact, if every training sample is the aggregate measurement, the performance can be poor by any method in the worst case. To see this, consider a simple case that $n = 2$, and each of these loads has only one pattern. If every training sample contains both loads, there is no way to identify which pattern corresponds to which load by any method, and the disaggregation will fail. The advantage of our method is that it does not need to know a prior which training samples measure one load only and can incorporate these data to enhance the dictionary learning accuracy naturally. For example, it handles the case that $n = 2$ and every training sample is labeled as "load 1" due to partial labels. In contrast, conventional dictionary learning methods [21] [35] require pure measurements for all loads with correct labels.

B. Online disaggregation of real-time aggregate measurements

After receiving the aggregate measurements from T time steps in real-time, represented by $x \in \mathbb{R}^T$, one can directly apply the exiting sparse decomposition method in (5) using the learned dictionary $\hat{D} = [\hat{D}_1, \cdots, \hat{D}^n]$ from the offline training by Algorithm 1. Let a_i denote the dictionary coefficients that correspond to $\hat{D}_i$ returned by (5), then load i component in x is $\hat{D}_i a_i$.

Here we show that one can further exploit the repetitive patterns of loads to enhance the disaggregation accuracy. Individual loads such as solar, industrial, and residential loads

usually have daily or weekly patterns. Thus, the combination of loads in x may have appeared in the recorded training data. Instead of disaggregating with respect to $\hat{D}$ as (5), one can disaggregate with respect to some representative load combinations in the training data to enhance the disaggregation accuracy. Specifically, we propose to solve

$$\min_{w\in\mathbb{R}^q} \|x - \hat{D}\tilde{A}w\|_2 + \mu\|w\|_1, \tag{14}$$

where $\tilde{A} = [\tilde{A}_1; \cdots; \tilde{A}_n] \in \mathbb{R}^{K\times q}$ is a submatrix of $\hat{A}$ by selecting q columns of $\hat{A}$ returned by Algorithm 1. The idea is to select q training samples in $\bar{X}$ with representative load combinations and represent x as a sparse combination of these q samples. Let $\hat{w}$ be the solution to (14), then the load i component in x can be estimated by $\hat{D}_i\tilde{A}_i\hat{w}$.

The downsampling scale q is chosen to be around K. Moreover, we uniformly select $\tilde{A}$ from $\hat{A}$ such that the load decomposition coefficients of these q samples are as distinct as possible to cover a large range of possible load combinations. (14) is convex and can be solved efficiently by any off-the-shelf solver. The computational complexity is $\mathcal{O}(q^2T/\varepsilon)$ where ε is the error tolerance.

IV. Experimental Results

A. Experimental Setup

We evaluate our method for disaggregating three loads, two different industrial sites and one solar generation in Sections IV-B and IV-C. We consider both the case that every load is labeled in partial data samples (Section IV-B) and the case that the solar load is not labeled in any data samples (Section IV-C). In Section IV-D, we also disaggregate one industrial load and the solar generation, where the solar is not labeled in the training data. The study in Sections IV-C and IV-D represents estimating BTM solar from aggregate data.

We employ Algorithm 1 to learn the dictionary $\hat{D}$ from the aggregate measurements with partial labels. $\beta = 0.1$, $\alpha_0 = 10^{-6}$, and $\sigma = 0.1$. We adjust the parameters λ_i ($i = 1, 2, 3$) such that the returned coefficient matrix $\hat{A}$ is column-sparse. We observe numerically that the solutions returned by Algorithm 1 are not sensitive to these parameters, and one can select them from a relatively large range. We then solve (14) to disaggregate the real-time measurement. The training time for the following experiments varies from 15 seconds to 30 seconds.[6]

We compare our method with the following methods: the conventional dictionary learning with sparse decomposition (DL-SD) ((3)-(4) for dictionary learning and (5) for load disaggregation), discriminative disaggregation sparse coding (DDSC) algorithm [21], and sum-to-k non-negative matrix factorization (S2K-NMF) [35]. Moreover, since our method improves over DL-SD in both dictionary learning and disaggregation, to further illustrate the improvement, we add a comparison with using Algorithm 1 to learn the dictionary and disaggregating using conventional sparse decomposition in (5). We abbreviate this approach as PropDL-SD. Note that both DDSC and S2K-NMF require recorded measurements of individual loads to obtain the dictionary of each load. To apply DDSC, we utilize the training data that are labeled as load l to learn $\hat{D}_l$, and the regularization coefficient is selected to be 0.01. We randomly select training time series with label l as the dictionary for load l when implementing S2K-NMF. The weight parameter is set to be 0.05. To reduce the influence of the random selection of the dictionaries over the disaggregation performance, the performance of S2K-NMF is averaged over 100 selections.

We employ various metrics to measure the disaggregation error. Root Mean square error (RMSE) is a standard method to compute the estimated errors [35], [56],

$$\text{RMSE}_i = \sqrt{\frac{\Sigma_{j=1}^{N'}\|\hat{x}_i^j - x_i^j\|_2^2}{T\times N'}}. \tag{15}$$

RMSE_i measures the average error of disaggregating load i from N' aggregate testing measurements. $\hat{x}_i^j$ and x_i^j in $\mathbb{R}^T$ represent the estimated and ground-truth values of load i from the jth testing data, respectively. RMSE relies on the actual values of the loads. To compare the performance on different datasets, we also define normalized RMSE (NRMSE) as follows [30],

$$\text{NRMSE}_i = \sqrt{\frac{\Sigma_{j=1}^{N'}\|\hat{x}_i^j - x_i^j\|_2^2}{\Sigma_{j=1}^{N'}\|x_i^j\|_2^2}} \tag{16}$$

References [21], [35] define metrics to measure the average disaggregation performance of all types of loads, assuming the load consumptions are all positive. Since we also consider negative loads such as solar generation, we modify the metric in [21] and define the Total-Error-Rate (TER) as follows,

$$\text{TER} = \frac{\Sigma_{j=1}^{N'}\Sigma_{i=1}^{n}\min(\|\hat{x}_i^j - x_i^j\|_1, \|x_i^j\|_1)}{\Sigma_{j=1}^{N'}\Sigma_{i=1}^{n}\|x_i^j\|_1} \tag{17}$$

TER is always less than 1.

B. Disaggregate three loads and every type of load is labeled in partial training data

1) Datasets: We generate the aggregate measurements of three loads by adding the individual measurements from the solar dataset by the National Renewable Energy Laboratory (NREL) [57] and the data of two different industrial sites in EnerNOC GreenButton Data [58]. The NREL dataset contains the solar measurements in one year with a 5-minute resolution. The EnerNoc dataset has 100 different industrial loads from different sites and also has a 5-minute resolution. We choose a time window of eight hours from 8:00 am to 4:00 pm (the method applies to other time windows as well) in selected 220 days and denoise the raw measurements by Gaussian windows. We choose two different industry loads from two sites as load 1 and load 2. These three loads are scaled to have a similar power level. The daily power consumption of load 1 ranges from 23 MW to 73 MW and load 2 ranges from 39 MW to 74 MW. Load 3 generates power between 4MW and 65 MW. The measurements of these three loads are added together to obtain aggregate measurements.

[6]The codes are available at : https://github.com/mengwang6308lab/Energy-Disaggregation

We first verify that time series of the same type of load indeed share some common patterns on the solar data within an 8-hour time window in one year in dataset [57]. We use (13) to determine the approximate rank K of a matrix to approximate the original data matrix with the desirable accuracy τ. One can see from Table II that the data indeed share a relatively small number of common patterns. For example, a rank-15 matrix can approximate the data with an accuracy of 99%.

Table II: The approximate rank of the solar data of one year with different approximation accuracy τ

τ	99.00%	98.00%	95.00%	90.00%
K	15	12	9	5

To study the influence of the measurements of individual loads on the disaggregation performance, we vary the percentage of the measurements of pure individual loads, denoted by γ. $\gamma = 30\%$ means 30% measurements labeled as load 1 (similarly load 2 and 3) contain pure load 1 (or 2 or 3) and remaining 70% measurements contain other types of loads. The mixture measurements are equally divided for all possible combinations of loads, i.e., all loads, loads 1 and 2, loads 1 and 3 when the label is load 1. Note that γ is for the analysis purpose, and the recovery method does not need γ. We use 120 days of data to generate 360 time series (120 labeled as load i, $i = 1, 2, 3$) for training. We then pick data from another 100 days and generate 300 testing cases in the same way. τ is 0.9 except in Table VI. The downsampling scale q is set as 12, which is much less than N.

Because the training problem is nonconvex, we choose multiple initializations and run Algorithm 1 to obtain multiple pairs of $\hat{D}$ and $\hat{A}$. We pick the best one based on the approximation error of the training data and coefficient sparsity. For $\gamma = 70\%$, 50%, and 30%, the number of initializations are 100, 500, and 5000, respectively. The same number of initializations are applied to PropDL-SD and DL-SD for a fair comparison.

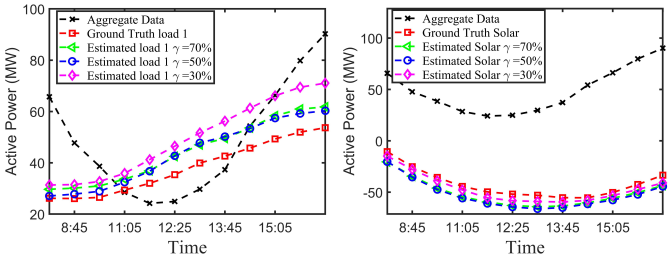


Fig. 3: Disaggregation of three loads (every load type is labeled in partial training data) Left: disaggregation of load 1. Right: disaggregation of load 3 (solar)

2) Disaggregation Performance: Fig. 3 shows the disaggregation results of three loads in one testing aggregate measurement. We only show the results of load 1 and load 3 (solar) due to the space limit. One can see that the estimated individual load components are very close to the ground-truth. When γ is large, the estimation performance improves. That coincides with the intuition that if recorded datasets include more measurements of individual loads, then the learned dictionary is more accurate, and the disaggregation performance is improved. The point to note is that our method performs comparably well when most measurements include unlabeled loads (see the performance when $\gamma = 30\%$).

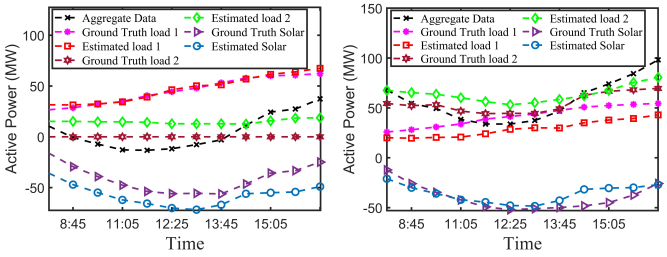


Fig. 4: Disaggregation of three loads (one load not labeled) Left: aggregate data contain load 1 and load 3. Right: aggregate data contain all loads

Table III: Disaggregation accuracy of three loads (every load is labeled in partial training data) when $\gamma = 70\%$

Method	Proposed	PropDL-SD	DL-SD	S2K-NMF	DDSC
RMSE_1	5.19	8.59	14.04	12.07	23.04
RMSE_2	5.62	7.15	11.05	9.82	25.39
RMSE_3	4.25	5.91	7.67	9.69	15.45
NRMSE_1	14.81%	24.50%	40.05%	34.43%	65.73%
NRMSE_2	15.04%	19.14%	29.56%	26.28%	67.92%
NRMSE_3	13.70%	19.05%	24.72%	31.25%	49.80%
TER	9.00%	11.98%	19.04%	18.04%	37.87%

Tables III-V show the comparison of our proposed method with four other methods mentioned in Section IV-A. All the results are averaged over 100 test cases. Our method (abbreviated by "proposed") has the best disaggregation performance among all the methods, and the corresponding disaggregation error is significantly lower than DL-SD, S2K-NMF, and DDSC. Moreover, the improvement of PropDL-SD over DL-SD results from our proposed dictionary learning method in (6)-(8). The improvement of "Proposed" over PropDL-SD results from our proposed disaggregation method in (14). When γ increases, the disaggregation performance improves.

3) Impact of parameter selections: In Algorithm 1, the number of dictionary atoms K_l is estimated by τ in (13).

Table IV: Disaggregation accuracy of three loads (every load is labeled in partial training data) when $\gamma = 50\%$

Method	Proposed	PropDL-SD	DL-SD	S2K-NMF	DDSC
RMSE_1	7.52	9.33	14.22	16.26	23.31
RMSE_2	6.16	7.55	15.50	14.24	24.76
RMSE_3	7.30	10.04	20.29	15.62	18.51
NRMSE_1	19.50%	24.20%	36.88%	42.17%	60.45%
NRMSE_2	15.08%	18.47%	37.92%	34.84%	60.56%
NRMSE_3	21.65%	29.78%	60.19%	46.34%	54.90%
TER	10.84%	12.30%	25.56%	26.93%	39.24%

Table V: Disaggregation accuracy of three loads (every load is labeled in partial training data) when $\gamma = 30\%$

Method	Proposed	PropDL-SD	DL-SD	S2K-NMF	DDSC
RMSE_1	11.46	12.72	24.90	20.23	23.67
RMSE_2	9.05	11.10	17.41	21.46	23.38
RMSE_3	12.06	16.37	22.13	22.38	23.93
NRMSE_1	27.30%	30.30%	59.34%	48.21%	56.41%
NRMSE_2	20.68%	25.36%	39.77%	49.01%	53.40%
NRMSE_3	32.95%	44.73%	60.44%	61.13%	65.36%
TER	15.15%	17.88%	38.83%	37.08%	46.05%

Table VI: The influences of τ on the selected rank K_1, K_2, K_3 and the corresponding TER when $\gamma = 70\%$

τ	0.96	0.93	0.9	0.87	0.84	0.78
K_1	5	4	3	3	2	2
K_2	6	5	4	3	3	2
K_3	7	5	4	4	3	3
TER	10.77%	10.67%	9.00%	10.24%	11.66%	14.76%

Table VII: The influence of λ_D on TER when $\gamma = 30\%$

λ_D	0	400	800	1200	1600
TER	20.05%	19.01%	17.70%	16.78%	16.43%
λ_D	2000	2400	2800	3200	3600
TER	16.06%	16.07%	15.60%	15.15%	15.24%

Table VIII: The influences of λ_1 on TER when $\gamma = 50\%$

λ_1	0	0.000001	0.00001	0.0001	0.001	0.01
TER	23.75%	20.46%	15.77%	14.08%	11.25%	16.63%

We learn various dictionaries by varying τ and compare the TER of the disaggregation results in Table VI. One can see that the result is not sensitive when τ changes from 0.96 to 0.87. When τ is less than 0.84, two or three dictionary atoms are not sufficient to represent corresponding loads, and the performance degrades.

Table VII shows the impact of varying λ_D while fixing other parameters. One can see that adding an incoherence penalty indeed reduces the testing error by about 5%, and λ_D can be selected from a wide range.

We also investigate the influence of the column-sparsity regularization. Table VIII shows the disaggregation result of varying λ_1 while fixing both λ_2 and λ_3 as 0.0003. One can see that a nonzero λ_1 decreases the disaggregation error. That indicates the effectiveness of column-sparsity regularization. Moreover, λ_1 can be selected in a wide range.

4) Impact of inaccurate labels: We next study the influence of inaccurate labels on the disaggregation performance. We consider the case that $\gamma = 70\%$. We vary the percentage of wrong labels, denoted by β. Specifically, $\beta = 10\%$ means that 10% of data labeled as load 1 are actually pure load 2 measurements; 10% of data labeled as load 2 are actually pure load 1 measurements; 10% of data labeled as load 3 are actually pure load 1 measurements. The disaggregation results are shown in Table XI. Compared with the first column of Table XI, one can see that the performance degrades only slightly when β increases. Our method is robust to a small portion of wrong labels and still obtains reasonable results.

Table IX: Disaggregation of three loads (solar load not labeled) when $\gamma = 50\%$

Method	Proposed	Prop-15min	PropDL-SD	DL-SD
$RMSE_1$	8.27	9.30	10.38	21.32
$RMSE_2$	10.09	11.76	11.15	47.72
$RMSE_3$	8.52	9.39	12.55	44.71
$NRMSE_1$	19.12%	21.62%	23.99%	49.27%
$NRMSE_2$	22.64%	26.47%	25.02%	107.07%
$NRMSE_3$	33.05%	36.44%	48.69%	173.40%
TER	15.47%	17.53%	19.55%	43.44%

Table X: Disaggregation of two loads (solar load not labeled) when $\gamma = 50\%$

Method	Proposed	Proposed 15min	Proposed-SD	DL-SD
$RMSE_1$	4.65	4.66	4.85	15.01
$RMSE_2$	4.54	4.56	4.81	15.00
$NRMSE_1$	17.39%	17.44%	18.16%	56.16%
$NRMSE_2$	45.63%	45.86%	48.31%	150.67%
TER	15.90%	17.76%	19.25%	45.48%

Table XI: The disaggregation results of three loads when $\gamma = 70\%$ with different percentage of wrong labels.

% of wrong labels (β)	0%	5%	10%	15%
$RMSE_1$	5.19	6.76	7.62	7.04
$RMSE_2$	5.62	5.88	6.10	7.13
$RMSE_3$	4.25	6.12	6.30	9.19
$NRMSE_1$	14.81%	19.29%	21.73%	20.08%
$NRMSE_2$	15.04%	15.73%	16.31%	19.09%
$NRMSE_3$	13.70%	19.73%	20.31%	29.63%
TER	9.00%	10.26%	11.22%	13.36%

C. Disaggregate three loads and one load is not labeled

1) Datasets: The training and testing data are obtained by removing all the samples with the label "load 3" from the datasets used in Section IV-B. Then load 3 is not labeled in any training data. This is more challenging to disaggregate than that in Section IV-B. To study the dependence of our method on the time resolution of datasets, we also downsample all the training and testing data from 5 minutes per sample to 15 minutes per sample and evaluate our method on these cases separately. The number of initializations is 8000.

2) Disaggregation Performance: Fig. 4 shows the disaggregation results of three loads when load 3 is not labeled in any training data. The left figure corresponds to a testing time series that contains load 1 and load 3. The right figure corresponds to a testing time series that contains all loads. One can see that the estimated consumptions of individual loads are close to the corresponding ground-truth values.

Table IX shows the comparison with DL-SD and PropDL-SD. Because the solar load is not labeled, S2K-NMF and DDSC do not apply here. The performance of our method degrades slightly compared with that in Table IV. That is because it is challenging to detect the existence of solar and learn its patterns. Still, our method is able to disaggregate the solar generation with a reasonable accuracy. The existing methods either fail (S2K-NMF and DDSC) or have much more significant error (DL-SD). Table IX also compares our method on 5-minute resolution and 15-minute resolution datasets. The performance degrades slightly when the time resolution becomes lower. Therefore, our method is not sensitive to the time resolution.

D. Disaggregate two loads and one load is not labeled

1) Dataset: We have the aggregate measurements of industrial loads (a steel factory) and solar loads at one substation from Lianyungang, Jiangsu, China for one year in 2017 at a 5-minute resolution. That industrial load exists all the time and the power consumption is significantly higher than the output of the solar panels. Since the ground-truth disaggregation of these two loads are unknown, we pick a window of 8 hours

from 8 pm to 4 am when only the industrial load exists and add this industrial load data to the solar data in [57] to obtain the aggregate data. The time window for the solar data is from 8:00 am to 4:00 pm. Specifically, we pick six typical solar patterns from [57]. Given a time series of the industrial load, we randomly select three solar patterns, multiply each pattern by a random value uniformly selected from 0.1 to 0.9, and add them together to obtain the aggregate measurements. The industrial load ranges from 9 to 45 MW, and the solar generation lies in the range of 5-30 MW. The aggregate power consumption is always positive at any time. 120 time series are used for learning, and another 120 time series are used for testing. $\gamma = 50\%$. τ is set as 0.9. The number of initializations is 2000.

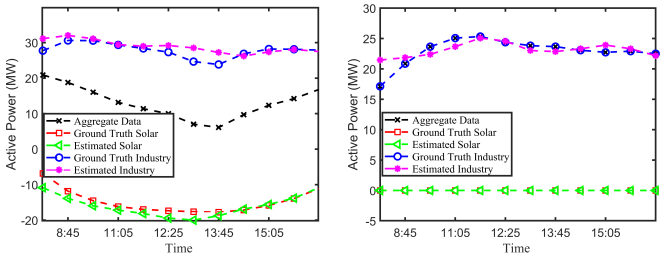


Fig. 5: The disaggregation results of industrial load only (left) and mixed loads (right)

Table X shows the average disaggregation performance on the testing data. S2K-NMF and DDSC do not apply here. Similar to the results in Table IX on three loads with unlabeled solar generation, here our proposed method also outperforms PropDL-SD and DL-SD and is robust to different time resolutions of the datasets.

Fig. 5 shows two examples of load disaggregation. The time window is 8 am to 4pm as that of the solar data. The left case only has the industrial load, and the estimated solar generation is close to zero. The right case has both the industrial load and the solar generation in the aggregate measurements, and the disaggregation results are close to the ground-truth values.

V. Conclusions

Load disaggregation at substations is an increasingly important problem for system planning and real-time operation with the integration of the renewable energy. The measurements at a substation are highly aggregated, and the existence of certain loads such as solar generations during a specific time is sometimes unknown. This paper formulated the problem of load disaggregation with partially labeled aggregate data and proposed a real-time disaggregation method that includes offline dictionary learning and online load decomposition. Our method can disaggregate behind-the-meter solar generations without any model of the solar generation and achieve a small disaggregation error.

Based on the proposed framework, we are interested to disaggregate more loads of other types, such as wind power, electric vehicles, and residential loads, in the future. The energy storage systems in distribution systems may further challenge the disaggregation problem, which is also beneficial to investigate. We are also investigating the case that the partial labels contain errors, and the disaggregation method should correct the wrong labels when separating the loads.

References

[1] F. Ding and B. Mather, "On distributed pv hosting capacity estimation, sensitivity study, and improvement," *IEEE Trans. Sustain. Energy*, vol. 8, no. 3, pp. 1010–1020, 2017.

[2] B. Chen, C. Chen, J. Wang, and K. L. Butler-Purry, "Sequential service restoration for unbalanced distribution systems and microgrids," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1507–1520, 2018.

[3] K. Baker, A. Bernstein, E. Dall'Anese, and C. Zhao, "Network-cognizant voltage droop control for distribution grids," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2098–2108, 2018.

[4] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, "Data-driven probabilistic net load forecasting with high penetration of behind-the-meter pv," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3255–3264, 2018.

[5] F. Wang, K. Li, C. Liu, Z. Mi, M. Shafie-Khah, and J. P. S. Catalão, "Synchronous pattern matching principle-based residential demand response baseline estimation: Mechanism analysis and approach description," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6972–6985, 2018.

[6] J. Alonso-Montesinos, F. Batlles, and J. Bosch, "Beam, diffuse and global solar irradiance estimation with satellite imagery," *Energy Convers. Manag.*, vol. 105, pp. 1205–1212, Nov. 2015.

[7] D. Chen and D. Irwin, "Sundance: Black-box behind-the-meter solar disaggregation," in *Proc. 18th International Conf. Future Energy Systems*, ser. e-Energy '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 45–55.

[8] P. Gotseff, J. Cale, M. Baggu, D. Narang, and K. Carroll, "Accurate power prediction of spatially distributed pv systems using localized irradiance measurements," in *Proc. IEEE PES General Meeting Conf. Expo.* IEEE, 2014, pp. 1–5.

[9] E. C. Kara *et al.*, "Disaggregating solar generation from feeder-level measurements," *Sustain. Energy Grids.*, vol. 13, pp. 112–121, Nov. 2018.

[10] F. Kabir, N. Yu, W. Yao, R. Yang, and Y. Zhang, "Estimation of behind-the-meter solar generation by integrating physical with statistical models," in *2019 IEEE International Conf. Communications, Control, and Computing Technologies for Smart Grids*. IEEE, 2019, pp. 1–6.

[11] M. Tabone, S. Kiliccote, and E. C. Kara, "Disaggregating solar generation behind individual meters in real time," in *Proc. 5th Conf. Systems for Built Environments*. ACM, 2018, pp. 43–52.

[12] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, "Data-driven probabilistic net load forecasting with high penetration of behind-the-meter pv," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3255–3264, 2017.

[13] Q. Zhang, J. Zhang, and C. Guo, "Photovoltaic plant metering monitoring model and its calibration and parameter assessment," in *Proc. IEEE Power and Energy Society General Meeting*, 2012, pp. 1–7.

[14] C. Dinesh, S. Welikala, Y. Liyanage, M. P. B. Ekanayake, R. I. Godaliyadda, and J. Ekanayake, "Non-intrusive load monitoring under residential solar power influx," *Applied Energy*, vol. 205, pp. 1068 – 1080, 2017.

[15] F. Wang, K. Li, X. Wang, L. Jiang, J. Ren, Z. Mi, M. Shafie-khah, and P. Catalão, João, "A distributed pv system capacity estimation approach based on support vector machine with customer net load curve features," *Energies*, vol. 11, no. 7, p. 1750, 2018.

[16] F. Sossan, L. Nespoli, V. Medici, and M. Paolone, "Unsupervised disaggregation of photovoltaic production from composite power flow measurements of heterogeneous prosumers," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3904–3913, 2018.

[17] H. Shaker, H. Zareipour, and D. Wood, "A data-driven approach for estimating the power generation of invisible solar sites," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2466–2476, 2016.

[18] F. Bu, K. Dehghanpour, Y. Yuan, Z. Wang, and Y. Zhang, "A data-driven game-theoretic approach for behind-the-meter pv generation disaggregation," *IEEE Trans. Power Syst.*, pp. 1–1, 2020.

[19] G. S. Ledva, L. Balzano, and J. L. Mathieu, "Real-time energy disaggregation of a distribution feeder's demand using online learning," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4730–4740, Jan. 2018.

[20] S. Wang, R. Li, A. Evans, and F. Li, "Regional nonintrusive load monitoring for low voltage substations and distributed energy resources," *Applied Energy*, vol. 260, p. 114225, 2020.

[21] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1153–1161.

[22] S. Singh and A. Majumdar, "Analysis co-sparse coding for energy disaggregation," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 462–470, 2017.

[23] Z. Guo, Z. J. Wang, and A. Kashani, "Home appliance load modeling from aggregated smart meter data," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 254–262, 2014.

[24] H. Ahmadi and J. R. Martı, "Load decomposition at smart meters level using eigenloads approach," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3425–3436, 2015.

[25] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale-and context-aware convolutional non-intrusive load monitoring," *IEEE Trans. Power Syst.*, 2019.

[26] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1739–1747, 2016.

[27] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[28] J. M. Gillis, S. M. Alshareef, and W. G. Morsi, "Nonintrusive load monitoring using wavelet design and machine learning," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 320–328, 2015.

[29] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 26–40, 2016.

[30] M. Zhong, N. Goddard, and C. Sutton, "Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3590–3598.

[31] S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in *Proc. Conf. Mach. Learn. Appli.*, vol. 2, 2012, pp. 515–520.

[32] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé, and M. Fournier, "Approach in nonintrusive type i load monitoring using subtractive clustering," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 812–821, Aug. 2015.

[33] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific data*, vol. 2, March 2015, Art. no. 150007 .

[34] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *32nd AAAI Conf. Artificial Intelligence*, 2018.

[35] A. Rahimpour, H. Qi, D. Fugate, and T. Kuruganti, "Non-intrusive energy disaggregation using non-negative matrix factorization with sum-to-k constraint," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4430–4441, April 2017.

[36] E. Elhamifar and S. Sastry, "Energy disaggregation via learning'powerlets' and sparse coding," in *Proc. 29th AAAI Conf. Artificial Intelligence*, 2015, pp. 629–635.

[37] S. Singh and A. Majumdar, "Deep sparse coding for non–intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4669–4678, Feb. 2017.

[38] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[39] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 175–188, Jan. 2014.

[40] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, Oct. 2006.

[41] W. Gao, L. Wang, Z.-H. Zhou *et al.*, "Risk minimization in the presence of label noise," in *30th AAAI Conference on Artificial Intelligence*, 2016.

[42] O. Dekel and O. Shamir, "Good learners for evil teachers," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 233–240.

[43] P. Shamsi, M. Marsousi, H. Xie, W. Fries, and C. Shaffer, "Dictionary learning for short-term prediction of solar pv production," in *2015 IEEE Power & Energy Society General Meeting*. IEEE, 2015, pp. 1–5.

[44] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, April 2009.

[45] Q. Zhang and B. Li, "Discriminative k-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2691–2698.

[46] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 457–464.

[47] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[48] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Sep. 2012.

[49] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1697–1704.

[50] X. He, L. Chu, R. C. Qiu, Q. Ai, Z. Ling, and J. Zhang, "Invisible units detection and estimation based on random matrix theory," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 1846–1855, 2019.

[51] X. Zhang and S. Grijalva, "A data-driven approach for detection and estimation of residential pv installations," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2477–2485, 2016.

[52] S. Hosur and D. Duan, "Subspace-driven output-only based change-point detection in power systems," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1068–1076, 2018.

[53] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

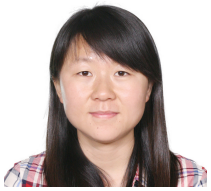
[54] D. P. Bertsekas, "Nonlinear programming. athena scientific," *Belmont, MA*, 1999.

[55] C.-J. Lin and J. J. Moré, "Newton's method for large bound-constrained optimization problems," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1100–1127, 1999.

[56] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *26th AAAI Conf. Artificial Intelligence*, 2012.

[57] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, "The wind integration national dataset (wind) toolkit," *Appl. Energy*, vol. 151, pp. 355–366, July 2015.

[58] D. T. Nguyen, M. Negnevitsky, and M. De Groot, "Pool-based demand response exchange—concept and modeling," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1677–1685, 2010.



Wenting Li (S'16) received the B.E. degree from Harbin Institute of Technology, Heilongjiang, China, in 2013.

She is pursuing the Ph.D. degree in electrical engineering at Rensselaer Polytechnic Institute, Troy, NY. Her research interests include high-dimensional data analytics, feature extraction, application of machine/deep learning methods to identify and locate events in power grids.

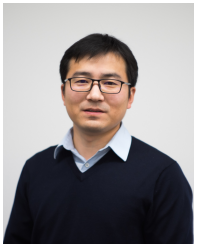
Ming Yi (S'17) received the B.E. degree in automation from Harbin Engineering University, Harbin, China, in 2016, and the M.S. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2018, respectively.

He is currently a Ph.D. student in Rensselaer Polytechnic Institute, Troy, NY, USA. His research interests include signal processing, machine learning, power systems monitoring, and high dimensional data analysis.

Meng Wang (M'12) received B.S. and M.S. degrees from Tsinghua University, China, in 2005 and 2007, respectively. She received the Ph.D. degree from Cornell University, Ithaca, NY, USA, in 2012.

She is an Assistant Professor in the department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute, Troy, NY, USA. Her research interests include high-dimensional data analytics, machine learning, power systems monitoring, and synchrophasor technologies.

Yishen Wang (S'13–M'17) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2011, the M.S. and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 2013 and 2017, respectively. He is currently a Senior Research Engineer of the AI System Analytics Group with GEIRI North America, San Jose, CA, USA. His research interests include power system economics and operation, energy storage, load modeling, AI and PMU data analytics.



Di Shi (M'12-SM'17) received the Ph.D. degree in electrical engineering from Arizona State University. He is currently the Director of Fundamental RD Center and Department Head of the AI System Analytics Group, Global Energy Interconnection Research Institute North America (GEIRINA), San Jose, CA, USA. Prior to joining GEIRINA, He was a Research Staff Member with NEC Laboratories America. His research interests include PMU data analytics, AI, energy storage systems, IoT for power systems, and renewable integration. He is an Editor of the IEEE Transactions on Smart Grid and the IEEE Power Engineering Letters.

Zhiwei Wang (M'16-SM'18) received the B.S. and M.S. degrees in electrical engineering from Southeast University, Nanjing, China, in 1988 and 1991, respectively. He is President of GEIRI North America, San Jose, CA, USA. Prior to this assignment, he served as President of State Grid US Representative Office, New York City, from 2013 to 2015, and President of State Grid Wuxi Electric Power Supply Company from 2012-2013. His research interests include power system operation and control, relay protection, power system planning, and WAMS.