2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA 2017), 10 - 12 March 2017, Penang, Malaysia

Event and Feature Based Electrical Load Disaggregation Using Graph Signal Processing

Kriti Kumar and M. Girish Chandra
TCS Research and Innovation, India
Email: {kriti.kumar, m.gchandra}@tcs.com

***Abstract*—Electrical load disaggregation continues to attract new explorations due to its challenging nature as well as utility. When the loads to be separated are characterized by suitable features, there is a possibility to solve the problem by utilizing the techniques from the emerging area of Graph Signal Processing (GSP). In this paper, we propose a three-staged approach comprising of (i) Event Detection and Clustering (ii) Event Pairing and Feature Extraction and (iii) Load Classification, each of them being pivoted on GSP. For load classification in particular, a robust spectral clustering strategy is appropriately adopted using joint spectrum computed from different features. The efficacy of this novel combination is demonstrated through the results obtained on both public data sets and the simulated active power signals.**

I. INTRODUCTION

Source Separation is one of the important problems, where the objective is to recover the original individual sources from the composite signal, which is a mixture of multiple sources. One application of source separation is Electrical Load Disaggregation or Non-intrusive Load Monitoring (NILM). With the massive deployment of smart meters, increasing environmental concerns and energy consciousness among people, NILM has gained importance in the recent past. Given the aggregate power consumption measurement of a household, NILM detects the individual loads which could be in operation and estimates their consumption. It is more formally stated in Section II A. The load-specific operation information benefits both users and utilities [1]. The consumers can use this information together with the existing energy pricing mechanism for potential energy savings. It also helps utilities towards arriving at effective demand management strategies. Further, network operators can use this information for power planning.

The algorithms for NILM existing in literature can be broadly classified into two categories: (i) state based approach; (ii) event based approach. State based approaches involves modeling the loads using state machines, like Hidden Markov Model (HMM) and its variants [2], [3]. Event based approach involves detecting ON and OFF of loads by analyzing the changes in the power level of the aggregate power consumption. Several methods have been proposed in literature for carrying out edge detection based on steady state and transient changes [4], [5]. Further, the choice of these methods also depends on the sampling frequency of the smart-meter measurement. Based on the different aggregate parameters measured, appropriate features can be extracted from the events and used for load classification. In [6], active power, reactive power, Total Harmonic Distortion, phase angle and spike are considered as features for load classification of a Canadian household. Another work [7], extracts load features from the probability density function of active power signature database of the loads and uses a combination of k-means and Support Vector Machine to perform classification.

Recently, with the success of Graph Signal Processing (GSP) in handling high-dimensional data over irregular domains for different applications like, filtering, clustering, classification, compression, inpainiting etc [8], [9], [10], an attempt is made to combine different techniques emerging in this area to solve the problem of NILM. Only few work discuss GSP application to NILM [11], [12] and that also for low sampled (1 minute) aggregate active power data. The work reported in [11] is a supervised method which is neither state based or event based and carries out disaggregation by applying regularization on the graph. The method uses binary classification of loads which does not give good load detection and consumption accuracy. In [13], the former work is modified appropriately to work on actual power data of loads for improved detection and consumption accuracy. The method in [12] presents an unsupervised event based approach which uses GSP for edge detection, clustering and event pairing for detecting loads. However, this method does not perform load classification and leaves the annotations part to the user to label the loads. A useful way to handle this problem is through appropriate event detection and extracting features with the paired events. In order to clearly differentiate loads, any variability in the signatures is appropriately exploited by defining sensible feature vectors. The whole process also facilitates separation of simultaneously operating loads even when the features are not simply additive, thus enabling a powerful non-linear data (signal) processing strategy.

Extending the work in [12], the relevant portion of which is captured in Section II B, in this paper, a new methodology is proposed by appropriately combining the work in [12] with and an additional module to incorporate the load features for classification. Using the features other than active power alone, within the GSP setting, for electrical load disaggregation, is one of the novelties of this paper. Relevant graphs are constructed using these features and robust clustering is carried out using joint spectrum computation. The said construction is based on the concepts existing in [14], a very recent work where multiple graphs are used for clustering. The complete solution for load disaggregation involving events and features,

978-1-5090-1184-1/17/$31.00

pivoted on the GSP, again, to the best of our knowledge does not exist elsewhere.

Towards providing the necessary details for the proposed technique and the associated results, the paper is organized as follows. Section II describes the problem of load disaggregation and gives an overview of the proposed algorithm. Algorithm is presented in detail with all the sub modules implemented using GSP. Subsequently, Section III discusses the results obtained using this algorithm on two different datasets, Reference Energy Disaggregation Data Set (REDD) [15] and Home Energy Simulator (HES) [16]. Finally, Section IV concludes the work.

II. Proposed Methodology for NILM

This section starts with the NILM problem description. Subsequently, an overview of the proposed event and feature based electrical load disaggregation using GSP is presented, describing the individual blocks in more detail.

A. NILM Problem Description

Electrical load disaggregation in the residential scenarios generally involves estimating the individual load operation and consumption from a single point aggregate power measurement of a household. This can be more formally stated as in Eqn. (1) where, given the aggregate power measurement $p(i)$, sampled at time instants $i = 1, ..., N$, it is required to compute the individual loads $p_m(i)$ which could have contributed to that measurement.

$$p(i) = \sum_{m=1}^{M} p_m(i) + n(i) \tag{1}$$

Here, $m \in M$, where M is the total number of loads of interest and $n(i)$ is the measurement noise which also includes the base loads and other loads of the house which are not considered in M.

B. Algorithm Overview

The proposed disaggregation framework uses events to identify the switching activity (ON/OFF) of load operation. Events are characterized by significant changes in the aggregate power measurement denoting the activity of load operation. Once event is detected, event pair matching is done to identify the start and end of the individual events. Subsequently, features are extracted for all event pairs and classification is performed to label the event pairs to specific load operation. The proposed algorithm uses the emerging concepts of GSP to carry out disaggregation which has three stages: (i) Event Detection and Clustering; (ii) Event Pairing and Feature Extraction; and (iii) Load Classification using Spectral Clustering. These stages are described in more detail in the subsequent sections.

1) Event Detection and Clustering: Events describe the switching activity (ON/OFF) of the loads in the aggregate power measurements. Event detection plays a major role for any event based disaggregation method. They are identified as significant power variation $\Delta p(i)$ between adjacent aggregate power measurements where, $\Delta p(i) = p(i + 1) - p(i)$ for $i = 1, ...N - 1$. Usually, fixed or adaptive thresholds are used for event detection. The thresholds used should be chosen such that they are large enough to filter out noise and base load interference and small enough to detect low power loads of interest. This method uses a small fixed threshold T for event detection. So, all $\mid \Delta p(i) \mid > T$ form the set S of events which are considered for clustering in the next stage.

Clustering is performed by exploiting the correlation between data samples. The correlation between the data samples are well represented by a weighted graph. A graph $G = (V, E)$, is represented by a set of vertices V, where $V = \{v_n\}_{n=1}^{N}$ and a set of edges E connecting the vertices. The adjacency matrix $\boldsymbol{A} \in \mathbb{C}^{N \times N}$ defines the connections between vertices using edges with weights representing the correlation between data samples. GSP is used to cluster the events similar to the work in [12]. Let N be the number of elements in set S, a graph of N vertices is constructed based on the elements in S using a Gaussian kernel weighting function expressed as:

$$\boldsymbol{A}(i, j) = exp(-|dist(i, j)|^2/\sigma^2) \tag{2}$$

where, σ is a scaling factor and $dist(i, j)$ is the distance measure between vertices i and j. This distance measure can be either the Physical distance, Euclidean distance or Cosine distance depending on the application. In this paper, Euclidean distance is considered as a distance measure. The scaling factor is chosen such that $\boldsymbol{A}(i, j)$ evaluates to a high value if the values at i, j are similar and less otherwise.

Since similar elements tend to belong to the same cluster, all elements of S form a graph signal $\boldsymbol{s}$ that does not change rapidly from node to node if the elements are similar. In other words, the signal values of strongly connected nodes do not vary much over the graph and tend to have similar values i.e. the total variation of the signal over the underlying graph is small. Using this notion, clustering is performed in an iterative manner by labelling the first element in S. The graph signal $\boldsymbol{s}$ is defined as:

$$\boldsymbol{s}(1) = \begin{cases} 1, & \text{for } \Delta p(1) > T_1. \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

$\boldsymbol{s}(j) = 0$ for $j > 1$. Here, T_1 is a small threshold (taken to differentiate positive and negative events) which can take value 0. The elements which are similar to $\boldsymbol{s}(1)$ are predicted as a solution to the optimization problem which minimizes the total variation of the signal on a graph expressed as:

$$\min_{\mathbf{s}} \|\boldsymbol{s}^{\boldsymbol{T}} \boldsymbol{L} \boldsymbol{s}\|_2^2 \tag{4}$$

where, $\mathbf{s}$ is the graph signal and $\boldsymbol{L}$ is the graph Laplacian computed as:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{5}$$

where, $\boldsymbol{A}$ is the weighted adjacency matrix defined using Eqn. (2). $\boldsymbol{D}$ is the degree matrix which is a diagonal matrix containing the sum of weighted edges incident on the vertices of the graph along its diagonal, and is expressed as:

$$\boldsymbol{D}(i,i) = \sum_j \boldsymbol{A}(i,j) \tag{6}$$

where, j is the neighbours of vertex i.

Since, $\boldsymbol{s}(1)$ is known and $\boldsymbol{L}$ is a symmetric matrix the closed form solution of this problem is derived by taking the derivative of the smoothness term with respect to the signal $\boldsymbol{s}$ and equating it to zero. This is given as:

$$\boldsymbol{s}^* = \boldsymbol{L}(2:N, 2:N)^{-1}.(-\boldsymbol{s}(1)).\boldsymbol{L}(1, 2:N)^T \tag{7}$$

If $\boldsymbol{s}^*(j) >$ threshold T_1, the event $\Delta p(j)$ and $\Delta p(1)$ are considered similar and are clustered together. These elements are removed from S. In the next iteration, the first element of the remaining events in S is considered for clustering in the similar way from a new graph constructed using Eqn. (2) with the remaining elements in S. This iterative process is carried out till all the elements in S are clustered and S becomes empty. At the end of this process, a number of positive and negative event clusters are obtained.

2) Event Pairing and Feature Extraction: The positive and negative event clusters obtained in the previous stage are paired together to form an event pair corresponding to a possible load operation. The feature matching method described in [12] is adopted to carry out event pair matching using GSP. For any positive event only those negative events are chosen for pairing which have a similar magnitude as the positive event and occur after the positive event in time. Two graphs are constructed using Eqn. (2), one based on magnitude difference and other based on time difference between the positive and negative event. The results obtained by applying regularization over these two graphs individually using Eqn. (7) are combined using Eqn. (8) to obtain a valid event pair.

$$\max_i(\alpha \boldsymbol{s_M}^* + \beta \boldsymbol{s_T}^*) \tag{8}$$

where, $\boldsymbol{s_M}^*$, $\boldsymbol{s_T}^*$ are obtained by applying regularization over the magnitude difference and time difference graph respectively. α and β are heuristically chosen constants and $i = 1, ..., n$ where n is the number of negative event candidates for a particular positive event.

The work presented in [12] for disaggregation was limited to event pair matching. The labelling of the event pairs to the corresponding load operation was not carried out. The novelty of the proposed method lies in extending the work of [12] by carrying out labelling of the event pairs using features unique to load operation. This provides a complete disaggregation solution which is more robust due to inclusion of load features. Restricting to aggregate active power alone, this method uses features like, ON amplitude, OFF amplitude, ON time and portion of the raw power waveform to classify the loads. In this work, the last feature is a count which represents the presence of low power (non-heater) events riding on top of load operation which is a signature of washer. For all event pairs, these features are extracted and passed on to the load classification stage.

3) Load Classification using Spectral Clustering: Clustering is performed in the spectral domain using GSP based on features extracted in the previous stage. Spectral clustering is a well established technique which is popular for its simple implementation and good performance for graph based methods [17], [18]. This method computes the spectrum of the graph G which is characterized by the eigen-decomposition of the graph Laplacian $\boldsymbol{L}$ which is expressed as:

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1} \tag{9}$$

where, $\boldsymbol{U}$ is $N \times N$ matrix containing of eigenvectors and $\boldsymbol{\Lambda}$ is $N \times N$ diagonal matrix containing eigenvalues of $\boldsymbol{L}$ for G with N vertices. It then transforms the signal on the original vertices to the low dimensional space obtained from the graph spectrum. Clusters can then be obtained using any traditional clustering algorithms like, k-means. The steps involved in spectral clustering using a single graph are detailed in [18].

Since here, four features are present, four graphs are constructed to obtain a robust clustering by appropriately combining the information from individual graphs. The four graphs, G_{ON}, G_{OFF} , G_{ONT}, G_{RAW} are constructed based on ON amplitude, OFF amplitude, ON time and portion of the raw power waveform respectively using Eqn. (2). Since all the feature graphs share with the same set of vertices, these four graphs can be viewed as a single four layer graph. One way to perform clustering using multiple graphs is to combine the information from the individual graphs using methods described in [14] and perform spectral clustering on it. These methods are more prone to scaling effects if the scale of the weights vary across multiple layers. Moreover, all layers are given same importance. The other way is to compute the joint spectrum from multi-layer graphs as described in [14] giving different importance to different layers. The latter method is reported to be more efficient and hence is adopted in our proposal to carry out clustering using multiple graphs.

For two graphs, G_1 and G_2, the joint spectrum is computed using Eqn. (10) as described in [14]:

$$\min_{\boldsymbol{f}_i \in \mathbb{R}^n} \frac{1}{2}\|\boldsymbol{f_i} - \boldsymbol{u}_i^{(1)}\|_2^2 + \lambda \boldsymbol{f}_i^T \boldsymbol{L}^{(2)} \boldsymbol{f}_i \tag{10}$$

where, $\boldsymbol{u}_i^{(1)}$ are the eigenvectors computed from G_1, $\boldsymbol{L}^{(2)}$ is the Laplacian obtained from G_2 and $\boldsymbol{f}_i$ is the joint spectrum obtained from G_1 and G_2 for $i = 1, ...n$. The regularization parameter λ acts as a trade off between the fidelity and smoothness term. The joint spectrum is computed such that $\boldsymbol{f}_i$ is closer to the eigenvectors obtained from G_1, and smooth with respect to G_2. This optimization problem has a closed form solution [14]:

$$\boldsymbol{f}_i^* = \mu(\boldsymbol{L}^{(2)} + \mu I_n)^{-1} \boldsymbol{u}_i^{(1)} \tag{11}$$

where, $\mu = 1/\lambda$. The regularization parameter λ can be tuned to reflect the relative importance of a particular layer in the multi-layer graph structure.

Since it is required to consider four graphs in our scenario, the aforementioned method can be applied. First, the joint spectrum of G_{ON}, G_{OFF} is computed using Eqn. (11) as they contain most distinctive information about the loads. Eigenvectors are obtained using G_{ON} and G_{OFF} is used for the regularization process. Subsequently, G_{ONT} is used for the regularization process on the eigenvector obtained from the joint spectrum calculated above to get the next joint spectrum. In the similar way the last graph G_{RAW} based on the portion of raw load waveform is also added to compute the joint spectrum corresponding to the four graphs. This joint spectrum maximizes the mutual information among different graphs. The first k eigenvectors of the joint spectrum is then used to perform clustering. Simple k-means clustering is performed to arrive at robust clusters corresponding to different loads [18]. Since the typical values of these features are known for the different loads, load classification is carried out by labelling the clusters as different loads.

III. Results and Discussion

The proposed method was tested with public dataset REDD and simulated dataset obtained using HES for performance evaluation. Here, T = 100W was used as a fixed threshold for edge detection for detection of high power loads. The clustering and event pair matching was done by constructing graphs using σ = 0.5 and 1 respectively. The scaling factor was computed empirically for the dataset considered (refer [13] and the references therein for more details) depending on the permissible variance allowed for the events. The four features based on ON amplitude, OFF amplitude, ON time and portion of the raw power waveform were extracted and the graphs are constructed for these features. The value of σ = 10 for G_{ON}, G_{OFF} and σ = 1 for G_{ONT}, G_{RAW} was considered. The joint spectrum of the four graphs was computed and spectral clustering was performed for load classification for both the datasets. It was observed that the clustering performance was limited by the difference in the operating power levels of the loads considered. As the difference became small the disaggregation ability was lost as detailed [12]. Hence only loads with different operating power levels were considered here for disaggregation. Loads with multiple states like, Dishwasher, Dryer etc. were considered as multiple instances of the same load.

Four commonly used loads namely, Refrigerator, Stove, Dishwasher and Microwave were considered from REDD House 2. The aggregate power measurement was taken as a sum of individual load power sampled at 3 seconds. Three features namely, ON amplitude, OFF amplitude and ON time were used to carry out disaggregation. For the loads considered, the fourth feature based on raw power waveform did not improve the clustering results and hence was not used. The value of λ was tuned to give more importance to ON and OFF amplitudes than ON time duration. This was done to account for loads like Microwave which can have different operating time at different instants. Spectral clustering was carried out and four clusters (k = 4) were obtained using k-means algorithm. The results obtained after extensive experimentation for these four loads are summarized in Table I. Figure 1 presents the disaggregation results obtained with House 2 data of REDD.

The algorithm was also tested with simulated data from HES sampled at 10 seconds. Four loads namely, Washer, Refrigerator, Microwave and Dishwasher were considered for disaggregation. The aggregate power measurement contained base loads along with other loads of interest. Here, all four features were used for carrying out load classification, since, the information from the raw power waveform helped in identifying Washer operation. The value of λ was tuned to give more importance to ON, OFF amplitudes and raw power waveform than ON time duration. Spectral clustering was performed to obtain four clusters corresponding to the four loads. Figure 2 presents the disaggregation results obtained using HES data. Sometimes there were misclassification of loads as the inclusion of one feature would nullify the contribution of another feature. For example, as seen in Fig. 2, one cycle of Dishwasher is misclassified as Washer due to the conflict of same ON time duration. The same ON time duration dominates over raw power waveform feature which is a characteristic of washer and leads to misclassification. This is because the raw power feature is considered last in computing the joint spectrum. Using extensive simulations, load detection accuracy $\geq 90\%$ was achieved for all the loads considered in this dataset.

The disaggregation results obtained from the proposed algorithm using REDD House 2 can be compared with that obtained from other state of the art GSP based techniques [11], [12] sampled at 1 minute. It was observed that including the features and using a higher sampling frequency significantly improved the disaggregation accuracy. For the same four loads, the F_M score of all loads were considerably increased to those available in [12]. Thus, with the availability of high sampled data, more features can be identified and these features can further enhance the disaggregation ability of the loads using this method.

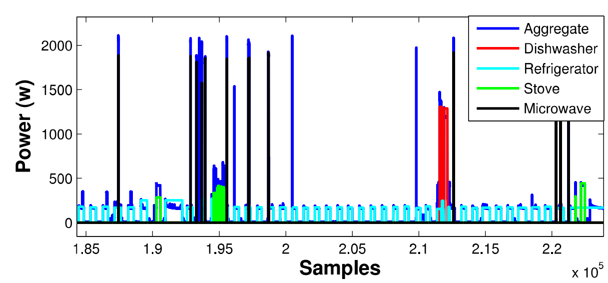

Fig. 1. Disaggregation results with 3 second sampled REDD for House 2

IV. Conclusion

Using the emerging concepts in the area of GSP, in this paper a novel methodology for load disaggregation using features is presented. Extending the previous GSP based solution, using load features and appropriately combining with clustering using multiple graphs a new solution is developed

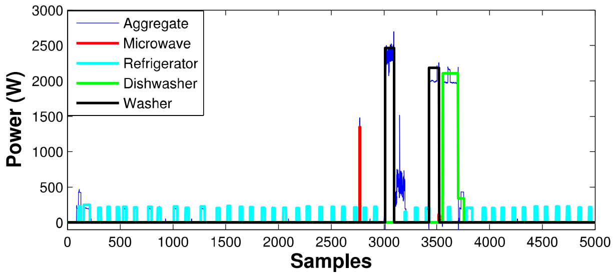

Fig. 2. Disaggregation results with 10 second sampled HES data

TABLE I
DISAGGREGATION RESULTS WITH REDD HOUSE 2

Load	**TP**	**FP**	**FN**	**PR**	**RE**	F_M
Microwave	60	0	3	1	0.95	0.97
Refrigerator	238	56	14	0.80	0.94	0.86
Dishwasher	9	1	0	0.9	1	0.94
Stove	22	11	0	0.66	1	0.79

and studied through extensive experimentation. Promising results are observed on different dataset considered in terms of load disaggregation accuracy. Inclusion of additional features, like reactive power, THD, transients can further increase the disaggregation capability of the proposed method. Since this method is event based it is directly applicable to the realistic scenario where the random switching of loads can be conveniently handled using robust event detection. Needless to say, this event-driven technique is very useful in many other scenarios involving recognition tasks.

REFERENCES

[1] A. Zoha, A. Gluhak, M.A. Imran, and Rajasegarar S., “Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey,” *Sensors 2012, 12*, pp. 16838–16866, 2012.

[2] Y. F. Wong, Y. Ahmet ekerciolu, T. Drummond, and V. S. Wong, “Recent approaches to non-intrusive load monitoring techniques in residential settings,” in *IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, April 2013, pp. 73–79.

[3] T. Zia, D. Bruckner, and A. Zaidi, “A hidden markov model based procedure for identifying household electric loads,” in *37th Annual Conference on IEEE Industrial Electronics Society (IECON)*, November 2011, pp. 3218–3223.

[4] Mario Berges, Ethan Goldman, H. Scott Matthews, Lucio Soibelman, and Kyle Anderson, “User-centered nonintrusive electricity load monitoring for residential buildings,” *Journal of Computing in Civil Engineering*, vol. 25, no. 6, pp. 471–480, 2011.

[5] M. Zeifman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, February 2011.

[6] M. Dong, P. C. M. Meira, W. Xu, and C. Y. Chung, “Non-intrusive signature extraction for major residential loads,” *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1421–1430, September 2013.

[7] H. Altrabalsi, J. Liao, L. Stankovic, and V. Stankovic, “A low-complexity energy disaggregation method: Performance and robustness,” in *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*, Dec 2014, pp. 1–8.

[8] D. I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[9] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, April 2013.

[10] S. Chen, A. Sandryhaila, G. Lederman, Z. Wang, J. M. F. Moura, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovaevi, “Signal inpainting on graphs via total variation minimization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8267–8271.

[11] V. Stankovic, J. Liao, and L. Stankovic, “A graph-based signal processing approach for low-rate energy disaggregation,” in *IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, December 2014, pp. 81–87.

[12] B. Zhao, L. Stankovic, and V. Stankovic, “On a training-less solution for non-intrusive appliance load monitoring using graph signal processing,” *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[13] Kriti Kumar, Rahul Sinha, M. Girish Chandra, and Naveen Kumar Thokala, “Data-driven electrical load disaggregation using graph signal processing,” in *13th International IEEE India Conference - INDICON*, Dec 2016.

[14] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov, “Clustering with multi-layer graphs: A spectral perspective,” *IEEE Trans. Signal Processing*, vol. 60, no. 11, pp. 5820–5831, 2012.

[15] J. Zico Kolter and Matthew J. Johnson, “REDD: A Public Data Set for Energy Disaggregation Research,” in *SustKDD Workshop on Data Mining Applications in Sustainability*, 2011.

[16] Krishnan Srinivasarengan, Y. G. Goutam, and M. Girish Chandra, “Home energy simulation for non-intrusive load monitoring applications,” in *ACM International Workshop on Engineering Simulations for Cyber-Physical Systems (ES4CPS)*, New York, USA, March 2014, pp. 9:9–9:12.

[17] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, “On spectral clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., pp. 849–856. MIT Press, 2002.

[18] Nicolas Tremblay, Gilles Puy, Rémi Gribonval, and Pierre Vandergheynst, “Compressive spectral clustering,” in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1–15.