

Blind Non-intrusive Appliance Load Monitoring using Graph-based Signal Processing

Bochao Zhao, Lina Stankovic, and Vladimir Stankovic

Department of Electronic and Electrical Engineering University of Strathclyde, Glasgow, G1 1XW, UK

Email: {b.zhao, lina.stankovic, vladimir.stankovic}@strath.ac.uk.

Abstract—With ongoing massive smart energy metering deployments, disaggregation of household’s total energy consumption down to individual appliances using purely software tools, aka. non-intrusive appliance load monitoring (NALM), has generated increased interest. However, despite the fact that NALM was proposed over 30 years ago, there are still many open challenges. Indeed, the majority of approaches require training and are sensitive to appliance changes requiring regular re-training. In this paper, we tackle this challenge by proposing a “blind” NALM approach that does not require any training. The main idea is to build upon an emerging field of graph-based signal processing to perform adaptive thresholding, signal clustering and feature matching. Using two datasets of active power measurements with 1min and 8sec resolution, we demonstrate the effectiveness of the proposed method using a state-of-the-art NALM approaches as benchmarks.

Keywords—Non-intrusive appliance load monitoring, load disaggregation, graph-based signal processing

I. INTRODUCTION

Energy disaggregation, that is, identifying the individual appliance usage from the total, aggregate load, using purely data analytics, referred to as non-intrusive appliance load monitoring (NALM) [1], can significantly enrich energy feedback. Indeed, up to 20% of reduction in energy consumption is expected via appliance-feedback and specific appliance replacement programs [2]. NALM can support home automation, enable activity recognition [3]-[5] and help appliance retrofit decisions. Energy suppliers can better forecast demand, system operators can monitor the effect of smart grid fluctuations on the residential microgrid, and appliance manufacturers can optimise product design to meet customer usage patterns.

NALM can benefit the wider public if it can operate purely on measurements that are already collected for metering and billing purposes. Hence, low-complexity techniques that work with active power only at resolutions of seconds and minutes [6] are of special interest and could be used with a wide range of current off-the-shelf smart metering devices. However, a downside of current NALM approaches for low-rate data (see [7], [8], [9] literature surveys) is their requirement for long *training* periods and substantial manual effort. To address the challenge of practicality and consumer/industry acceptance, NALM research has focused on unsupervised approaches, that do not require labeled sets for training [10]. These approaches are usually based

on hierarchical clustering or different variants of Hidden Markov Models (HMM)s, such as [11], [12], [13], [14].

Although the above state-based approaches can be *unsupervised*, they use expert knowledge to set a-priori values for each appliance state, and require a training set (usually where appliance operation does not overlap) to build/refine the state models. Since appliance models differ widely between households, using a training set from one house to disaggregate the other has been shown to be ineffective [15], [16]. Unsupervised, time-series approaches, such as [9], either suffer from similar problems or require both active and reactive power and high sampling rates [17].

In this paper, we propose a new, *blind*, low-rate NALM approach that does not require any training. The approach disaggregates any aggregate active power dataset without any prior knowledge. It relies on graph-based signal processing (GSP) [18]. GSP offers an alternative to conventional signal processing and data mining approaches by embedding the structure of signals onto a graph, leading to a powerful scalable and flexible approach suitable to many data mining and signal processing problems, ranging from image denoising and data compression, to classification, biomedical, and environmental data processing (see [18]-[22] and references therein). We note that recently, the latter two authors of this paper propose a GSP-based NALM approach in [23]. However, the approach of [23] is supervised and employs GSP only for data classification based on [20]. The proposed algorithm operates purely on collected aggregate active power data, and uses GSP three times, first to find an optimal event threshold, then to perform clustering, and finally for feature matching. The approach is deterministic and relies only on time-series data without any manual input, except for the final step of testing when unlabelled disaggregated appliances are labelled through expert knowledge.

II. GRAPH-BASED SIGNAL PROCESSING (GSP)

GSP is based on *graph signals* obtained by indexing a dataset by nodes of a graph. The basic idea is to represent a dataset using a graph defined by a set of nodes and a weighted adjacency matrix. Each node in the graph corresponds to an element in the dataset while the adjacency matrix defines all edges in the graph and their weights, where assigned weights reflect the degree of similarity, i.e., correlation, between the nodes.

Given an acquired set of measurements \mathbf{x} , we define a graph $G = \{\mathbf{V}, \mathbf{A}\}$, where a set of nodes \mathbf{V} in the graph corresponds to the acquired measurements, and edges are defined by a weighted adjacency matrix \mathbf{A} . The graph signal is then defined as a map from a set of nodes \mathbf{V} to a set of complex numbers \mathbf{s} , called *graph signal*, where each element s_i is indexed by a node $v_i \in \mathbf{V}$. The adjacency matrix \mathbf{A} defines all edges in the graph and their weights. The values of $A_{i,j}$ are often naturally defined by the physical meaning of the collected data. If that is not the case, a Gaussian kernel weighting function:

$$A_{i,j} = \exp\left\{-\frac{|dist(x_i, x_j)|}{\sigma^2}\right\}, \quad (1)$$

is often used in the literature [20], [21], [23], where $dist(x, y)$ can be, for example, Euclidian distance between x and y . The graphs and signals on graphs defined above can be conveniently used to represent very different data structures, such as time series, images, sensors, tracked objects, social networks, hyperlinked documents etc. [19], [18].

If \mathbf{x} is piecewise smooth in a defined feature space, then $\mathbf{s}^T \mathbf{L} \mathbf{s}$ is generally small, where \mathbf{L} is the graph Laplacian operator [19] given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where \mathbf{D} is a diagonal matrix with nonzero entries $D_{i,i} = \sum_j A_{i,j}$. This graph-signal smoothness prior can effectively be used for regularization, since it can be shown that the Laplacian regularizer $\mathbf{s}^T \mathbf{L} \mathbf{s}$ is a good measure of variation in the signal modulated by weights in \mathbf{A} . Then, we can formulate the smoothness term minimization problem as:

$$\arg \min_{\mathbf{s}} \|\mathbf{s}^T \mathbf{L} \mathbf{s}\|_2^2. \quad (2)$$

If \mathbf{s} is an N -length discrete signal, then \mathbf{L} is an $N \times N$ matrix and [21], [23]:

$$\begin{aligned} \mathbf{s}^T \mathbf{L} \mathbf{s} = & s(1)L(1,1)s(1) + s(1)L(1,2:N)s(2:N) + \\ & s(2:N)^T L(2:N,1)s(1) + \\ & s(2:N)^T L(2:N,2:N)s(2:N), \end{aligned} \quad (3)$$

where $s(1)$ is the first element in \mathbf{s} , $s(a:b) = [s(a), s(a+1), \dots, s(b)]$, $a < b$, and similarly for matrix \mathbf{L} . Note that (3) is the same as in [20], [23], except that we replaced a vector of known samples used for training in supervised classification approaches with a randomly picked sample ($s(1)$). Since \mathbf{D} is a diagonal matrix, \mathbf{L} is also diagonally symmetric. Thus, since the first term in (3) does not affect minimization, we simplify minimization in (2) as:

$$\begin{aligned} \arg \min_{\mathbf{s}} \|\mathbf{s}^T \mathbf{L} \mathbf{s}\|_2^2 = & \arg \min_{s(2:N)} 2s(2:N)^T L(2:N,1)s(1) + \\ & s(2:N)^T L(2:N,2:N)s(2:N). \end{aligned} \quad (4)$$

As an unconstrained quadratic programming problem, this minimization has a closed form solution [21], [24]:

$$\mathbf{s}^* = L(2:N,2:N)^\# (-s(1))L(1,2:N)^T, \quad (5)$$

where $(.)^\#$ denotes the pseudo-inverse matrix. \mathbf{s}^* is the smoothness optimization solution, i.e., a solution that minimizes the total graph variation.

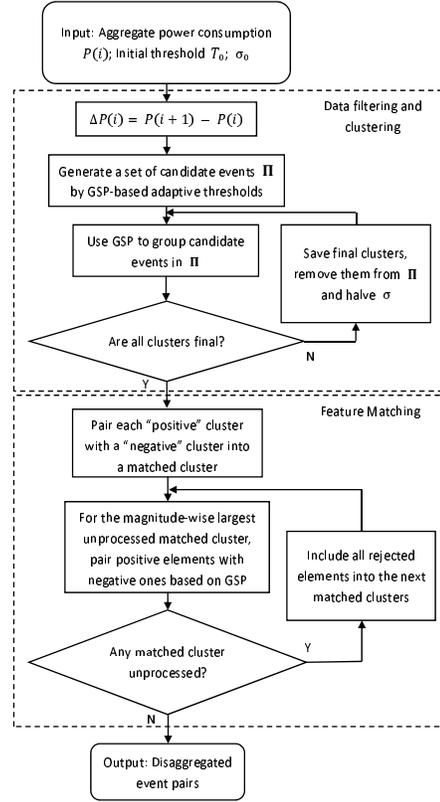


Fig. 1. Flowchart of the proposed algorithm.

III. PROPOSED DISAGGREGATION ALGORITHMS

The proposed algorithm consists of filtering, clustering and feature matching steps as shown in the flow chart in Fig. 1.

A. Data filtering and clustering

The objective of this step is event detection, i.e., detecting appliance switching on/off as well as all state transitions in multi-state appliances, such as washing machine. Let $P(t_i)$ be the real power consumption measurement at time instance t_i , for $i = 1, \dots, n$. Then, $\Delta P(t_i) = P(t_{i+1}) - P(t_i)$, for $i = 1, \dots, n-1$, denotes the power variation signal between adjacent aggregate power readings. Note that each $\Delta P(t_i) > 0$ is defined as an increasing edge and $\Delta P(t_i) < 0$ are decreasing edges. As the interval $t_{i+1} - t_i$, for $i = 1, \dots$, is constant, we simplify the notation as $\Delta P(i) = \Delta P(t_i)$. To avoid detecting stand-by settings, the initial event threshold is set to $T_0 = 10\text{W}$, thus, all $\Delta P(i) \in (-\infty, -T_0) \cup (T_0, \infty)$ will be regarded as candidates for event occurrences [23].

Next, we design a graph where a sample $\Delta P(i)$ that is greater than T_0 (or less than $-T_0$) is associated to a node $v(i)$ of a graph. $A(i, j)$ denotes the weight of an edge from node $v(i)$ to $v(j)$ that depends on the level of correlation between $x_i = \Delta P(i)$ and $x_j = \Delta P(j)$, calculated as in (1) using Euclidian distance measure. We

set $s(1)$ to 1 if $\Delta P(i) > T_0$ and -1 otherwise, and initialise all $s(j) = 0, j > 1$. Then, we perform GSP and calculate (5). If $s^*(j) > q$, where q is a constant, then $\Delta P(j)$ is assigned to the first cluster of events, and removed. We set then $s(1)$ to the first remaining un-clustered element, i.e., the first $\Delta P(j)$ that is less than q , and repeat the procedure by adding $\Delta P(i)$'s to the next cluster of events. The procedure is iterated until all candidate events are clustered.

Let μ_i and σ_i denote, respectively, the mean value and the standard deviation of Cluster C_i . We evaluate the *quality* of cluster C_i by a ratio R_i as: $R_i = |\frac{\mu_i}{\sigma_i}|$. The mean values of the clusters with the largest R_i will determine T_N and T_P , thresholds used for positive and negative edges, respectively. That is, a set of candidate events $\mathbf{\Pi}$ is redefined as: $\mathbf{\Pi} = \Delta P \in (-\infty, T_N) \cup (T_P, \infty)$.

We continue the same GSP-based clustering process as above for all clusters that have $R_i > K$ by halving σ in (1) in each iteration. After each iteration, all resulting clusters with $R_i \leq K$, will be saved as final clusters, passed to the following matching cluster in the Feature Matching step, explained in the next section, and removed from $\mathbf{\Pi}$. The clustering will end when there are no remaining elements in $\mathbf{\Pi}$.

B. Feature matching

Since the final clusters compose several ‘‘positive’’ clusters (comprising increasing power edges) and the same number of ‘‘negative’’ clusters (with decreasing power edges), we pair each ‘‘positive’’ cluster with a ‘‘negative’’ cluster into a merged *matched cluster*, by comparing the absolute mean value of all final clusters. Next, we use GSP to pair each increasing edge in a positive cluster with a decreasing edge in its matched negative cluster, exploiting magnitude differences and time intervals between increasing and decreasing edges as two matching features in the following way.

We start with the magnitude-wise largest matched cluster, that is a matched cluster that has the largest absolute power value. Let \mathbf{C}_P and \mathbf{C}_N denote, respectively, the set of positive and negative events, namely, increasing and decreasing power edges, in the matched cluster. The task is, for each $C_P(t) \in \mathbf{C}_P$ to find an optimal match among all candidates $C_N(t) \in \mathbf{C}_N$. We form a graph by considering only events in \mathbf{C}_N that occur after $C_P(t)$ and before $C_P(t + 1)$. This subset of \mathbf{C}_N will be regarded as a set of candidates, denoted by $\mathbf{\Omega}$. Let $\mathbf{\Omega}_M$ and $\mathbf{\Omega}_T$ represent, respectively, the set of magnitude differences between $C_P(t)$ and each element in $\mathbf{\Omega}$ and the corresponding set of time intervals.

We form two graphs: (i) the first one with $A(i, j)$ in (1) calculated using $\mathbf{\Omega}_M$ and $s(1)$ set to be the average value of the magnitude differences between each element in \mathbf{C}_P and the corresponding candidates in \mathbf{C}_N ; (ii) with $A(i, j)$ in (1) calculated using $\mathbf{\Omega}_T$ and $s(1)$ set to be the median of the time intervals between each element in \mathbf{C}_P and the corresponding candidates in \mathbf{C}_N (instead of the mean value to reduce the influence the extreme scenarios). In both cases the nodes are indexed by $\mathbf{\Omega}$. In this way we obtain two

solutions from (5), which we denote as vectors: ω_M^* and ω_T^* . Then we formulate an optimization problem for feature matching as:

$$\arg \max_i (\omega^*(i)) = \arg \max_i (\alpha \omega_M^*(i) + \beta \omega_T^*(i)), \quad (6)$$

where for $i = 1, \dots, m$, m is number of candidates, that is the length of ω_M^* and ω_T^* , and α and β are heuristically chosen, with $(\alpha + \beta = 1)$ to tradeoff magnitude difference and time. The solution of (6) provides the optimal decreasing edge for a given increasing edge. After pairing all events in a matched cluster, the rejected events will be included into the next matched cluster, i.e., a magnitude-wise smaller cluster. See Fig. 1.

Finally, the above disaggregated appliance clusters are labelled by comparing the disaggregated signature with a database of signatures available for that particular household, which could be done, e.g., via Dynamic Time Warping [25] or cross-correlation. The database of signatures is populated via a time diary, i.e., the signature is extracted at the timestamp the householder switches on and off the appliances in their house, preferably with no other appliance operating simultaneously.

IV. RESULTS AND DISCUSSION

We demonstrate the performance of the proposed algorithm using active power readings from two datasets: (1) the REDD public datasets [15] (<http://redd.csail.mit.edu/>) of US houses down-sampled to 1 minute; (2) the UK REFIT dataset (<http://www.refit-smarthomes.org>) with measurements collected every 8 seconds. We set K for evaluating cluster quality R_i to 10% and the initial Gaussian Kernel weighting function scaling factor σ_0 to 20 to avoid over-clustering. $\alpha = \beta = 0.5$, and $q = 0.98$ to cluster only highly correlated samples.

The evaluation metrics used are adapted from [23], [11], *Precision (PR)*, *Recall (RE)* and *F-Measure (F_M)*. As in [11], we separate true positives (*TP*) into two cases, accurate true positive (*ATP*) and inaccurate true positive (*ITP*). *ATP* presents the correct claim the detected appliance was running and the corresponding events are correctly named, while, *ITP* represents the correct claim the detected appliance was running but the corresponding events are wrongly named. False positives (*FP*) denote a wrong claim that the detected appliance was running, and false negatives (*FN*) indicate that the appliance operation was not detected. Then:

$$PR = ATP / (ATP + FP) \quad (7)$$

$$RE = ATP / (ATP + ITP + FN) \quad (8)$$

$$F_M = 2 \cdot (PR \cdot RE) / (PR + RE), \quad (9)$$

PR represents the accuracy of event detection, thus a lower *FP* leads to a higher *PR*. *RE* denotes the strength of events detection and clustering, thus lower *FN* and *ITP* lead to a higher *RE* is. *F_M* balances *PR* and *RE*.

In order to enable like-for-like comparison with [23], we use REDD Houses 2 and 6. The results are shown in Tables I and II. House 2 contains 6 appliances, one appliance is ignored as it was rarely used, i.e., two runs in two weeks. House 6 contains 9 appliances. Approximately two weeks worth consecutive data is used in this experiment. Note that all data is used for testing - there is no training in the proposed method.

TABLE I. PERFORMANCE OF THE PROPOSED APPROACH FOR HOUSE 2 FROM THE REDD DATASET.

Appliance	ATP	ITP	FP	FN	PR	RE	F_M
Microwave	81	6	3	3	0.98	0.9	0.94
Toaster	151	19	54	40	0.74	0.72	0.73
Stove	17	1	84	20	0.17	0.45	0.25
Fridge	574	23	141	149	0.8	0.77	0.78
DW	18	2	9	0	0.67	0.9	0.77

TABLE II. PERFORMANCE OF THE PROPOSED APPROACH FOR HOUSE 6 FROM THE REDD DATASET.

Appliance	ATP	ITP	FP	FN	PR	RE	F_M
Microwave	10	0	3	0	0.77	1	0.87
Toaster	4	1	3	3	0.57	0.5	0.53
Stove	7	5	3	2	0.7	0.5	0.58
Fridge	439	8	56	132	0.89	0.76	0.82
DW	26	6	61	5	0.3	0.7	0.42
Heater	3	0	56	3	0.05	0.5	0.09
AC	44	9	0	1	1	0.81	0.9
Light	7	6	7	12	0.5	0.28	0.36
Unknown	146	6	56	65	0.72	0.67	0.69

TABLE III. PERFORMANCE OF THE PROPOSED APPROACH FOR HOUSE 8 FROM THE REFIT DATASET.

Appliance	ATP	ITP	FP	FN	PR	RE	F_M
Microwave	7	10	0	3	1	0.35	0.52
Toaster	4	1	2	1	0.67	0.67	0.67
Kettle	39	7	6	2	0.87	0.81	0.84
Fridge	18	0	2	0	0.9	1	0.95
Freezer	54	16	180	24	0.23	0.57	0.32
TV	4	0	180	6	0.02	0.4	0.04
WM	3	1	8	0	0.27	0.75	0.4

From Table I, most appliances are disaggregated with accuracy $> 75\%$. The worse performance is achieved for the stove. The poor performance for the heater and light in House 6, shown in Table II, is due to the questionable ground truth data collected using sub-metering for these two appliances.

Next, we present the results for the REFIT dataset. The results from REFIT House 8 for 7 available appliances are shown in Table III. It can be seen that the proposed training-less approach showed very good performance for the kettle and refrigerator, but poor results for the TV, caused by TV being grouped in the same cluster as the freezer, because TV and freezer have close active power range. Note that, our approach shows average performance (across all appliances) of $F_M = 0.49$, which is better than the HMM-based approach in [9] which shows $F_M = 0.46$ for the same house.

Then, we compare our results F_{M_U} to those of the

TABLE IV. AVERAGED PERFORMANCE OF THREE NALM APPROACHES FOR HOUSES 2 AND 6 FROM THE REDD DATASET.

Appliance	House2			House6		
	F_{M_U}	F_{M_S}	F_{M_H}	F_{M_U}	F_{M_S}	F_{M_H}
Microwave	0.94	0.26	0.47	0.87	0.92	0
Toaster	0.73	0.59	0.68	0.53	1	0
Stove	0.25	0.41	0.21	0.58	1	0
Fridge	0.78	0.63	0.9	0.82	0.54	0.88
DW	0.77	0.56	0.04	0.42	-	-
Heater	-	-	-	0.09	0.11	0.03
AC	-	-	-	0.9	0.49	0.12
Light	-	-	-	0.36	-	-
Unknown	-	-	-	0.69	-	-

supervised GSP-based approach in [23] F_{M_S} and the results using unsupervised HMM-based method of [12] F_{M_H} as reported in [23]. All three methods are tested using the same data, while the two benchmark methods require additional data for training [23]. The results for the three NALM approaches for both houses are shown in Table IV. The HMM-based approach is the best for the refrigerator disaggregation. In fact, HMMs are known to perform well for events with regular cycles, such as refrigerator that operate usually alone during the night intervals when good statistics can be collected to build a state transition models [12], [25]. Apart from the refrigerator, both GSP-based approaches perform significantly better than the HMM-based approach. The proposed unsupervised GSP-based approach performs, on average, as well as the supervised one of [23], but without the training and supervised labelling overhead.

V. CONCLUSION AND FUTURE WORK

In this paper, we build on the emerging GSP concepts to develop a novel, *blind*, unsupervised low-rate NALM approach. Our approach does not require any training, and has lower complexity than traditional approaches, such as FHMM based. We use GSP in the proposed algorithm for solving different optimization problems. Based on the results from two datasets, our unsupervised GSP-based NALM approach performs as well as the supervised GSP-based NALM approach of [23] outperforming an unsupervised HMM-based method.

This paper has further demonstrated the potential of GSP for power disaggregation. Future work will include enhancing the algorithm robustness for dealing with event malposition and appliances' simultaneous operation, improving performance by averting over-clustering, and developing real-time applications.

ACKNOWLEDGMENT

This work is supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) projects REFIT EP/K002708, under the Transforming Energy Demand in Buildings through Digital Innovation (BuildTEDDI) funding programme. The authors would like to thank J. Kolter and M. Johnson for making the REDD database available.

REFERENCES

- [1] G. W. Hart, "Nonintrusive Appliance Load Data Acquisition Method", *MIT Energy Laboratory Technical Report*, Sept. 1984.
- [2] K. C. Arnel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? The case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.
- [3] J. Liao, L. Stankovic, and V. Stankovic, "Detecting household activity patterns from smart meter data," *IE-2014 10th IEEE International Conference on Intelligent Environments*, Shanghai, China, July 2014.
- [4] C. Wilson, L. Stankovic, V. Stankovic, J. Liao, M. Coleman, R. Hauxwell-Baldwin, T. Kane, S. Firth, and T. Hassan, "Identifying the time profile of everyday activities in the home using smart meter data," in *Proc. ECEEE Summer Study on Energy Efficiency*, Toulon/Hyeres, France, June 2015.
- [5] L. Stankovic, V. Stankovic, J. Liao, C. Wilsaon, R. Hauxwell-Baldwin, and M. Coleman, "Understanding domestic appliance use through their linkages to common activities," in *Proc. EEDAL-2015 8th Int. Conf. Energy Efficiency in Domestic Appliances and Lighting*, Lucerne-Horw, Switzerland, Aug. 2015.
- [6] "Smart Metering Equipment Technical Specifications Version 2," *Department of Energy and Climate Change, UK, Tech. Rep.*, 2013.
- [7] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Trans. Consumer Electronics*, vol. 57, pp. 76–84, Feb. 2011.
- [8] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, pp. 16838–16866, Dec. 2012.
- [9] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *Proc. SmartGridComm-2014 IEEE International Conference on Smart Grid Communications*, Venice, Italy, November 2014.
- [10] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "NILM redux: The case for emphasizing applications over accuracy," *2nd International Non-intrusive Appliance Load Monitoring Workshop*, Austin, TX, June 2014.
- [11] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. 11th SIAM Int. Conf. Data Mining*, Mesa, AZ, April 2011.
- [12] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proc. the 26th Conf. Artificial Intelligence (AAAI-12)*, Toronto, CA, July 2012.
- [13] J. Kolter, and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *J. Machine Learning*, vol. 22, pp. 1472–1482, 2012.
- [14] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric Hidden Semi-Markov Models," *J. Machine Learning Research (JMLR)*, vol. 14, pp. 673–701, Feb. 2013.
- [15] J. Kolter, and M. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop Data Mining App. Sustainability (SIGKDD)*, San Diego, CA, 2011.
- [16] H. Altrabalsi, J. Liao, L. Stankovic, and V. Stankovic, "A low-complexity energy disaggregation method: Performance and robustness," *SSCI-2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid*, Orlando, FL, December 2014.
- [17] K. S. Barsim, R. Streubel, and B. Yang, "An approach for unsupervised non-intrusive load monitoring of residential appliances," in *Proceedings of the 2nd International Workshop on Non-Intrusive Load Monitoring*, 2014.
- [18] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. 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.
- [19] A. Sandryhaila and M.F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, pp. 1644–1656, July 2013.
- [20] A. Sandryhaila and J. Moura, "Classification via regularization on graphs," in *Proc. Symposium on Graph Signal Processing in IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, TX, December 2013.
- [21] C. Yang, Y. Mao, G. Cheung, V. Stankovic, and K. Chan, "Graph-based depth video denoising and event detection for sleep monitoring," in *Proc. MMSP-2014 IEEE International Workshop on Multimedia Signal Processing*, Jakarta, Indonesia, Sept. 2014.
- [22] C. Yang, G. Cheung, and V. Stankovic, "Estimating heart rate via depth video motion tracking," in *Proc. ICME-2015 Intl. Conf. Multimedia and Expo.*, Torino, Italy, July 2015.
- [23] V. Stankovic, J. Liao, and L. Stankovic, "A graph-based signal processing approach for low-rate energy disaggregation," *IEEE Symposium Series on Computational Intelligence (SSCI)*, Orlando, FL, December 2014.
- [24] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, 2004.
- [25] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Power disaggregation for low-sampling rate data," *2nd International Non-intrusive Appliance Load Monitoring Workshop*, Austin, TX, June 2014.