INDiC: Improved Non intrusive load monitoring using load Division and Calibration

Nipun Batra
Indraprastha Institute of Information Technology
India

Author2
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678-2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—Non-Intrusive appliance load monitoring (NIALM) is the process of disaggregating the overall electricity usage into constituent appliances. In this paper we extend the Combinatorial Optimization (CO) approach for disaggregation, which was originally proposed in the seminal work on NIALM, in following two ways: 1) Breaking the problem into subproblems and reducing the state space; 2) Applying additional constraints backed by sound domain expertise. We evaluate our approach using REDD dataset and show practical problems which need to be solved while dealing with the dataset. We also propose a metric for evaluating NILM, which we believe overcomes many shortcomings of commonly used metrics.**

I. Introduction

- Motivate the importance of energy consumption in building
- Motivate that appliance level information is crucial - detailed feedback and optimized decision making [1]
- Challenges with getting appliance level information - introduce NIALM [2]
- introduce your proposed approach
- Enumerate the contributions

Primary contributions of our work are:

- Fill it up with 2-3 crisp points

Open source implementation of the proposed work is released for comparative analysis with other NIALM approaches as an IPython notebook[1]. We believe this is the first extensive release of a generic NIALM

II. Related work

Non-intrusive appliance load monitoring was first studied by Hart [2] in the early 1990s by examining signatures in aggregated load to indicate activities of appliances. The problem has been well studied in recent years ([3], [4], [5]) and NIALM systems can broadly be divided into two categories based on whether supervised or unsupervised disaggregation methods are used. Supervised learning techniques include optimization-based methods such as integer programming [6] and genetic algorithms [7]. These approaches are compute intensive and appliances with similar or overlapping load signatures are difficult to discern. Other machine learning techniques (such as Artificial Neural Network (ANNs) [8] and Hidden Markov Models (HMM) [9]), have also been shown to work well for the task. Variants of HMMs such as factorial hidden Markov models (FHMMs) [10], additive FHMMs [11], conditional factorial hidden semi-Markov models [12] and difference HMMs [13] have been studied. In factorial HMMs several models evolve independently and in parallel and the observed output is some joint function of all the hidden states; additive HMMs allow emission of a single real-valued (unobserved) output from each HMM and the output is the sum of these HMMs. In difference HMMs [13] each appliance's load is modeled using a graphical model and it is disaggregated from the aggregate power – this process is repeated iteratively until all appliances for which general models are available are disaggregated. It is therefore possible to infer the probability that a change in aggregated power was generated by two consecutive states of an appliance. The applicability of these sophisticated techniques are generally hindered by the difficulty of inference from models with a large number of HMMs. Rahayu et. al. [14] propose a discriminative model for energy disaggregation that predicts the most likely appliance state configuration from aggregated load using nonparametric classification algorithms. They posit that "subset sum" type techniques are not very effective since a large portion of the home energy is not monitored directly. In this paper, we put forward a simple idea – the aggregated load can be further split up by information on mains and knowing which appliance is assigned to which main. This would make the task of disaggregation inherently easier by solving a simpler optimization problem and most of the above sophisticated machine learning based modeling approaches can still be applied to the task[2].

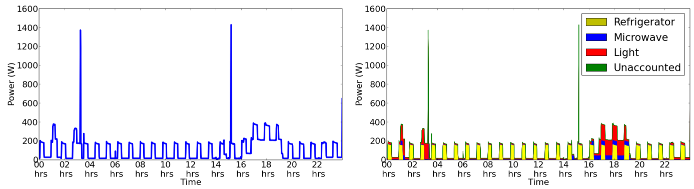
Several datasets have been made publicly available for benchmarking energy disaggregation algorithms including REDD[15], Blued[16], Pecan[17] and Smart*[18]. Our empirical analysis is on the REDD data primarily because this is the most popular data set for evaluating state-of-the-art NIALM approaches.

III. NIALM

A typical NIALM setup involves measuring the power mains with smart meters and individual appliances with appliance

[1] http://www.ipython.org

[2] The use of combinatorial optimization is only for the purpose of illustration and there is clearly no requirement to adhere to this modeling technique alone.



(a) Aggregate Power (b) Disaggregated Power

Fig. 1: Disaggregating a home's electrical mains into 3 appliances: refrigerator, lighting and microwave. Some power is unaccounted as complete information about all appliances is not available.

TABLE I: Terminologies and Functions

Symbol	Meaning
$t \in 1,..T$	Time slice
$n \in 1,..N$	Appliance number
$\theta^n = \{\theta^n_1,...,\theta^n_T\}$	Measured power sequence for n^{th} appliance
$\theta^{M1} = \{\theta^{M1}_1,...,\theta^{M1}_T\}$	Measured power sequence for Mains 1
$\theta^{M2} = \{\theta^{M2}_1,...,\theta^{M2}_T\}$	Measured power sequence for Mains 2
$x = \{x_1,..,x_T\} = \theta^{M1} + \theta^{M2}$	Measured aggregate power sequence
$e = \{e_1,...,e_T\}$	Aggregate noise power sequence
$y^n = \{y^n_1,..y^n_T\}$	Predicted power sequence for n^{th} appliance
$k \in 1,..K$	Appliance power state
$z^n = \{z^n_1,..z^n_T\}$	Appliance state sequence for n^{th} appliance,$z^n_i \in [1,..K]$
$z^n_{t,k} \in 0,1$	Whether n^{th} appliance is in k^{th} state at time t
$\mu^n = \{\mu^n_1,..\mu^n_K\}$	Power draw by n^{th} appliance in k^{th} state
$\theta^n_{k^1,k^2}$	Measured power sequence when n^{th} appliance transitions from k^1 to k^2 state
$Mapping[n] \in M_i$ where $i \in 1,2$	Mapping of n^{th} appliance to i^{th} Mains
$Downsample(s, filter, resolution)$	Function to downsample a timeseries s to a $resolution$ according to specified $filter$
$Preprocess([s^1,..s^n], method)$	Function to ensure that n timeseries start and end at same times and handling missing data using specified $method$
$Sort([s^1,..s^n], parameter, order)$	Function to sort n timeseries according to $parameter$ in specified $order$
$Event\_Detection(s, threshold)$	Function returning magnitude and times of step events in time series whose absolute s value is greater than $threshold$

meters for ground truth. The NIALM problem in supervised setting is formulated as predicting the power sequence for n^{th} appliance, y^n, given the measured power sequence for each appliance θ^n (measured via appliance meters) and the total aggregate power sequence x (measured via smart meters). Remaining terminologies and functions are defined in Table I [15], [13], [2]. Figure 1 shows the process of disaggregation applied to a house mains, whereby the consumption patterns of 3 appliances: refrigerator, lighting and microwave can be seen. It must be highlighted that it is improbable to instrument all the appliances in a home, thus, there will be some unaccounted power, which can also be seen in the figure.

Combinatorial Optimization(CO): At a given time an appliance can only be in a single state which is given by:

$$\sum_{k=1}^{k=K} z^n_{t,k} = 1$$

The power consumption by n^{th} appliance in k^{th} state at time t is given by:

$$\hat{\theta}^n_{t,k} = \sum_{k=1}^{K} z^n_{t,k}\mu^n_k$$

The overall power consumption of all appliances at a given time t is given by:

$$\hat{x}_t = \sum_{n=1}^{N}\sum_{k=1}^{K} z^n_{t,k}\mu^n_k$$

The error in power signal after the load assignment explained above is given by:

$$e_t = |x_t - \sum_{n=1}^{N}\sum_{k=1}^{K} z^n_{t,k}\mu^n_k|$$

Combinatorial optimization tries to find the optimal combination of appliances in different states which will minimize this error term, by the following state assignment scheme:

$$z_t = argmin_{z_t}|x_t - \sum_{n=1}^{N}\sum_{k=1}^{K} z^n_{t,k}\mu^n_k|$$

The corresponding predicted power draw by n^{th} appliance is given by:

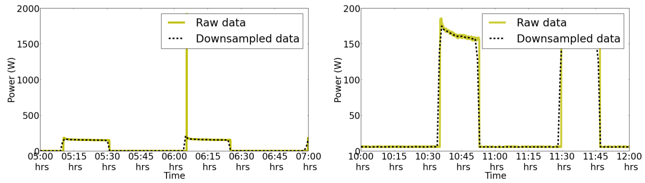
$$y^n = \{\mu^n_{z^n_1},..\mu^n_{z^n_T}\}$$

This optimization problem resembles subset sum problem [19] and is NP-complete. The state space size of this optimization function is K^N, which means it is exponential in the number of appliances. Owing to the exponential nature of the state space and the fact that the algorithm requires all appliances be known, this approach has not been thoroughly studied in the past [13]. We chose to use this approach as a proof of concept of our contributions.

Load division: In the US, homes have 2 electrical mains corresponding to different phases. Many Asian countries have multiple meters per home. Different loads are electrically connected to different mains/meters. Since an NIALM deployment requires monitoring different electrical mains/meters, we leverage the load division to perform disaggregation on these mains separately to improve load disaggregation. We assign N_i different loads to p different mains where: $\sum_1^p N_i = N$ After load assignment to different mains, the state space size for different mains are given by $K^{N_1}..K^p$. This leads to an exponential reduction in state space, which also leads to an exponential decrease in running time. As a practical example, if a home has two mains and 20 loads equally distributed across the two mains, state space size before load division is given by 2^{20} and after load division is 2^{10}. **Where to write that we can also benefit by parallelizing**

IV. INDIC NIALM

In this section we explain our algorithm- Improved Non intrusive load monitoring using load Division and Calibration



(a) Filtering startup transients (b) Filtering voltage fluctuations and oscillations

Fig. 2: Effect of downsampling appliance data

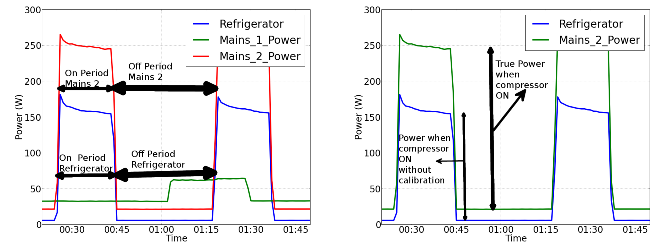
(INDiC). While INDiC can be used with any modeling technique described in Section II we present INDiC-CO (INDiC using Combinatorial optimization as modeling approach). The various steps of INDiC-CO NIALM shown in Algorithm 1 are described below.

Preprocessing: Since multiple data streams (appliances and mains power time series) from varied hardware are measured, it is highly likely that some hardware malfunctions during the data collection process. In the preprocessing step we ensure that all the time series start and end at the same time and handle missing data using techniques such as forward filling (padding).

Downsampling: While performing CO it is desired that transients and fluctuations in the power signal are filtered[2]. The transients occur due to the high starting current of the appliance, whereas the fluctuations are a consequence of minor voltage fluctuations and oscillatory nature of appliances. Figure 2a and Figure 2b show how starting current and voltage fluctuations can be filtered by down sampling. Filters such as mean/median can be used to down sample a time series to a time window, where the value assigned to a time window is the mean/median of values occurring during that time window in the original series.

Assigning Loads to Mains[3]: This step aims to identify the mapping between appliances and mains. Patterns corresponding to appliances having higher peak power are generally easier to extract from main signal, thus we sort the appliance in decreasing order of peak power. Starting from the appliance having highest peak load we compare its power at all times with both mains. If at any time the power of the appliance is greater than either mains, we can assign the appliance to the other mains. However, if we are not able to assign an appliance to mains this way, we find the times when events (absolute change in power is greater than a threshold of 15 Watts) occur in the appliance power series. This should be a subset of times of events occurring in one of the mains to which this appliance is assigned. The threshold of 15 Watts is to ensure that minor voltage fluctuations are not counted as events. After an appliance has been assigned to a mains, its power sequence is subtracted from the corresponding mains to simplify mains assignment for remaining appliances. Figure 3a shows how refrigerator is assigned to mains 2 since during this time interval its power is more than that of mains 1. Also, one may verify it by observing that the events in mains 2 and refrigerator power series occur at same time.

[3]While our approach is fine tuned for 2 Mains it can be easily extended to support further load division



(a) Assigning refrigerator to Mains 2 since refrigerator power > Mains 1 power and events in Mains 2 and refrigerator occur at same time (b) Calibrating refrigerator power consumption

Fig. 3: Calibrating and assigning refrigerator to Mains 2

Clustering: We use prior knowledge about number of states (K) of an appliance to cluster its power draw into different states. We chose kmeans++[22] as our clustering approach owing to its scalability and the improvements it offers over kmeans in terms of speed and accuracy.

Appliance Power Calibration: Power measured by appliance level meters may need calibration due to the following reasons:

- Difference in measurement of different measurement instrument (diagram showing CC, ZWave, etc) [21]
- Fluctuation in voltage causes power to fluctuate as well
- Missing meta data, whether real or apparent power is being measured at appliance level

In comparison to appliance data, mains data is usually measured using better precision hardware. Thus, we keep mains data as a reference and calibrate appliance data against it. Since in OFF states (state 1) appliance power consumption is almost zero, it does not require any calibration. We find out times when appliance transitions from a lower state to a higher state. During these times, we find the ratio of the magnitude of power change occurring in the assigned mains and the appliance. This ratio serves as a corrective multiplicative factor for a particular state of an appliance. Cluster centroids obtained in the previous step are multiplied by this factor to obtain calibrated cluster centroids.

Combinatorial optimization: Combinatorial optimization is now performed separately for both mains as per the description in Section III.

V. Evaluation

We use Reference Energy Disaggregation Data Set (REDD) [15] for validating our algorithms. This dataset contains power data for mains (2 phases) as well as appliances from 6 homes in Boston area collected in the summer of 2011. The data is made available as raw, high frequency (sampled at 15 KHz) and low frequency (Mains at 1 Hz, appliances at 0.3 Hz). Considering the practical implications of residential smart meter installation, we believe that low frequency data represents the most realistic scenario and thus we use this data for analysis.

Algorithm 1: INDiC-CO

Input: $x, \theta^n, \theta^{M_1}, \theta^{M_2}$
Output: y^n, μ_k^n

Preprocessing

$\theta^1, ..\theta^n, \theta^{M_1}, \theta^{M_2} \leftarrow Preprocess([\theta^1, ..\theta^n, \theta^{M_1}, \theta^{M_2}], forward\ fill)$

Downsampling

for $n \in 1, N$ **do**
  $\theta^n \leftarrow Downsample(\theta^n, mean, resolution)$
$\theta^{M_1} \leftarrow Downsample(\theta^{M_1}, mean, resolution)$
$\theta^{M_2} \leftarrow Downsample(\theta^{M_2}, mean, resolution)$
$s \leftarrow Sort([\theta^1, ..\theta^n], peak\ power)$

Appliance to Mains mapping

for $Appliance\ n \in s$ **do**
  if $\theta_t^n > \theta_t^{M_1}\ for\ any\ t \in 1, T$ **then**
    $Mapping[n] = M_2$
  else if $\theta_t^n > \theta_t^{M_2}\ for\ any\ t \in 1, T$ **then**
    $Mapping[n] = M_1$
  else
    if $Event\_Detection(\theta^n, threshold).Times \subset Event\_Detection(\theta^{M_1}, threshold).Times$ **then**
      $Mapping[n] = M_1$
    else
      $Mapping[n] = M_2$
  $\theta^{Mapping[n]} \leftarrow \theta^{Mapping[n]} - \theta^n$

Divide data into train and test set

Clustering on train set

for $n \in 1, N$ **do**
  $\mu_k^n \leftarrow Cluster(\theta^n, K, clustering\_algorithm)\ for\ k \in 1, K$

Calibration on train set

for $n \in 1, N$ **do**
  for $k \in 2, K$ **do**
    $\mu_k^n \leftarrow \frac{\mu_k^n * Step\_Changes(\theta_{k-1,k}^n, threshold).Magnitude}{Step\_Changes(\theta_{k-1,k}^{Mapping[n]}, threshold).Magnitude}$

Combinatorial optimization on test set

Solve combinatorial optimization as described in Section III
return y^n, μ_k^n

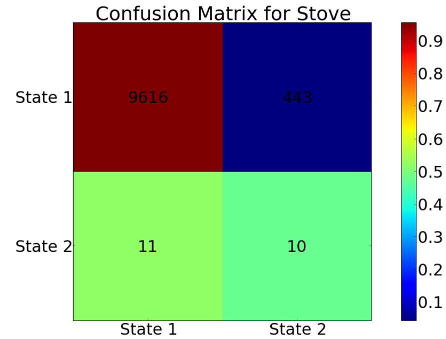


Fig. 4: Confusion Matrix showing predicted state accuracy for Stove

A. Evaluation Metric

Commonly used metrics such as accuracy, sensitivity and specificity can be misleading when applied to NIALM. We illustrate this with the help of a confusion matrix whose [m,n] entry in indicates the number of times appliance in state m is predicted to be in state n. It can be seen from the confusion matrix in Figure 4 that since stove is mostly in state 1 (Off), accuracy will be largely decided by accuracy for this state. This will overshadow the accuracy for state 2 (On) which is unintended. Thus, we use the following metrics which have been used in the past work [13], [15]:

Mean Normalized Error (MNE%): Normalized error in the energy assigned to an appliance n over time period T, given by:

$$MNE(n) = \frac{|\sum_{t=1}^{T} \theta_t^n - \sum_{t=1}^{T} y_t^n|}{\sum_{t=1}^{T} \theta_t^n} \tag{1}$$

above metric will give a 0 MNE for cases such as follows: $y^n = [0, 10, 0, 10]$ and $\theta^n = [10, 0, 10, 0]$, which can be misleading. Thus, we propose a modified Mean Normalized Error metric further referred as MNE which is sensitive to such cases.

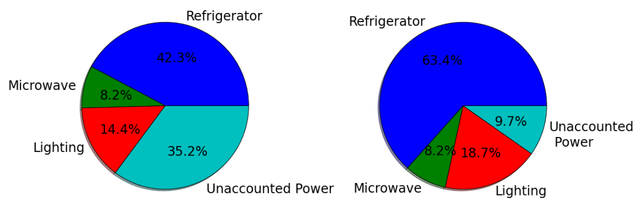
$$MNE(n) = \frac{\sum_{t=1}^{T} |\theta_t^n - y_t^n|}{\sum_{t=1}^{T} \theta_t^n} \tag{2}$$

Since it is a known fact that $|\sum a - \sum b| \leq \sum |a - b|$ where a and b are vectors containing floating point numbers, our results might look worse than if the original definition of MNE were used.

RMS Error (RE Watts): RMS error in power assignment to an appliance n per time slice t given by:

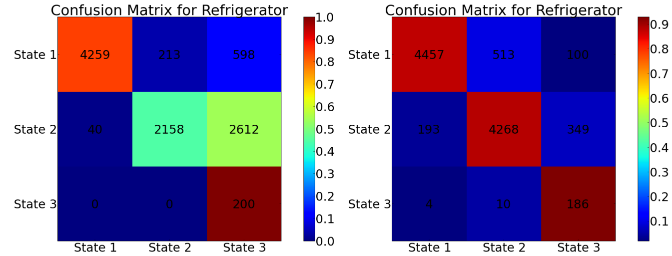
$$RE(n) = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\theta_t^n - y_t^n)^2} \tag{3}$$

Both these quantities represent error, the lesser they are the better is the prediction.



(a) Without applying calibration (b) After applying calibration

Fig. 5: Mains 2 Break down by load



(a) Without applying DiCaCo (b) After applying DiCaCo

Fig. 6: Confusion Matrix for refrigerator disaggregation

B. Empirical Analysis

We performed empirical analysis on REDD dataset Home 2, which consists of 11 channels (including 2 mains and 9 appliances). We believe that the same analysis can be easily repeated across multiple homes. We preprocessed the data and downsampled it to one minute using mean filter. Since we had two weeks of clean data, we used the first week as the train set and the second week as the test set. We used INDiC-CO algorithm described earlier to assign loads to different mains and calibrate appliance data. Table II shows the assignment of loads to different mains. Further this table also shows the learnt power states of these appliance via k-means++ clustering. Refrigerator and lighting showed significant difference in power states post calibration. Based on prior experience and appliance circuity, we believe that since only these two appliances needed calibration, it may be a case that the appliance level monitor measured real power instead of apparent power. These loads constitute a major chunk of Mains 2 power. Figure 5 shows the reduction in unassigned power due to calibrating these two appliances. Two appliances - washer dryer and disposal are ignored from the analysis since washer dryer had a peak power consumption of 8 W (which means it was Off throughout) and the contribution of disposal to overall power was less than 0.1 %.

To show the significance of load division and calibration, we considered 4 possible cases: i) no calibration, no load division; ii) no calibration, load division; iii) calibration, no load division; iv) calibration, load division (INDiC-CO). These results are presented in Table III. For the overall dataset, it can be seen that MNE reduces from 187% to 39%, RE reduces from 478 W to 168 W, after applying INDiC-CO. All appliances show reduction in MNE and RE after applying INDiC-CO. However, there is significant improvement in correctly predicting refrigerator and lighting. Figure 6 shows the confusion matrix for refrigerator prediction pre and post applying INDiC-CO. It can be seen that after applying INDiC-CO there is a vast improvement in predicting refrigerator's state 1 and 2.

VI. Conclusion and Future Work

The conclusion goes here. We also provide mains load assignment of all 6 homes from REDD to further the research in this direction.

- Applying model on noisy datasets
- 2 D CO (when Real and Reactive Power are known)
- Factoring in Time of Day etc.
- Factoring in Appliance Correlation
- Factor in switch continuity, essentially leads to Factorial HMM
- Distributed NILM
- Adaptive Learning

TABLE II: Mains assignment and appliance states power

Appliance	Mains	States Power (W)					
		Pre calibration			Post calibration		
Refrigerator	2	7	162	423	9	210	423[4]
Microwave	2	10	832	1730	10	832	1730
Lighting	2	9	96	156	10	110	178
Dishwasher	1	0	256	1195	0	256	1195
Stove	1	0	374	-	0	374	-
Kitchen	1	5	727	-	5	727	-
Kitchen 2	1	1	204	1032	1	204	1032

TABLE III: MNE and RE with and without INDiC-CO

	Without calibration				With calibration			
	Without load division		With load division		Without load division		With load division	
Appliance	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %
Refrigerator	136	109	71	32	130	95	59	21
Microwave	102	98	97	110	104	97	96	109
Lighting	51	164	48	195	44	83	38	60
Dishwasher	406	2947	63	100	377	2517	63	100
Stove	77	1191	36	281	75	1118	36	281
Kitchen	64	182	58	168	69	196	58	168
Kitchen 2	95	267	91	117	92	230	91	117
Overall	478	187	161	58	450	157	168	39

Acknowledgment

The authors would like to thank TCS Research and Development for supporting the first author through PhD. fellowship. We would also like to thank NSF- DEITy for funding the project.

References

[1] S. Darby, "The effectiveness of feedback on energy consumption," *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, 2006.

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

[3] K. Carrie Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, 2012.

[4] 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.

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

[6] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, "Nonintrusive appliance load monitoring based on integer programming," in *SICE Annual Conference, 2008*, 2008, pp. 2742–2747.

[7] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in nialm systems," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, 2004, pp. 3462–3468 vol.4.

[8] A. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, 2010, pp. 1–9.

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

[10] Z. Ghahramani, M. I. Jordan, and P. Smyth, "Factorial hidden markov models," in *Machine Learning*. MIT Press, 1997.

[11] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," *Journal of Machine Learning Research - Proceedings Track*, vol. 22, pp. 1472–1482, 2012.

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

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

[14] D. Rahayu, B. Narayanaswamy, S. Krishnaswamy, C. Labbe, and D. Seetharam, "Learning to be energy-wise: Discriminative methods for load disaggregation," in *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on*, 2012, pp. 1–4.

[15] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, 2011, pp. 1–6.

[16] A. Filip, "Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research."

[17] C. Holcomb, "Pecan street inc.: A test-bed for nilm," in *International Workshop on Non-Intrusive Load Monitoring*, 2007, pp. 271–288.

[18] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," in *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.

[19] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[20] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.

[21] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Training load monitoring algorithms on highly sub-metered home electricity consumption data," *Tsinghua Science & Technology*, vol. 13, pp. 406–411, 2008.

[22] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.