Post-processing for Event-based Non-intrusive Load Monitoring

[1]Kanghang He, [2]Dusan Jakovetic, [1]Vladimir Stankovic and [1]Lina Stankovic
[1]Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK
[2]Department of Math. and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia

***Abstract*—Most current non-intrusive load monitoring (NILM) algorithms disaggregate one appliance at a time, remove the appliance contribution towards the total load, and then move on to the next appliance. On one hand, this is effective since it avoids multi-class classification, and analytical models for each appliance can be developed independently of other appliances, and thus potentially transferred to unseen houses that have different sets of appliances. On the other hand, however, these methods can significantly under/over estimate the total consumption since they do not minimise the difference between the measured aggregate readings and the sum of estimated individual loads. By considering this difference, we propose a post-processing approach for improving the accuracy of event-based NILM. We pose an optimisation problem to refine the original disaggregation result and propose a heuristic to solve a (combinatorial) boolean quadratic problem through relaxing zero-one constraint sets to compact zero-one intervals. We propose a method to set the regularization term, based on the appliance working power. We demonstrate high performance of the proposed post-processing method compared with the simulated annealing method and original disaggregation results, for three houses in the REFIT dataset using two state-of-the-art event-based NILM methods.**

I. Introduction

Non-intrusive load monitoring (NILM), that is, disaggregating total household/building energy consumption, down to appliance level, using purely software tools, has gained increased interest due to large scale smart meter deployments world wide, and NILM's potential to provide actionable energy feedback and support smart home automation. Consequently, NILM has become a very active research topic [1], [2], [3].

Most current NILM methods disaggregate one appliance at the time, and do not check if the sum of the disaggregated loads is approaching the true measured result (see, for example, [3], [4], [5], [6], [7], [8]). Deviating from traditional NILM approaches (see [1], [9] and reference theirin), [10] uses NILM disaggregation results as a starting point to minimise the difference between the total measured aggregate reading and the sum of disaggregated loads via simulated annealing to avoid significant over/under estimation of the NILM approach. However, the method of [10] is sub-optimal and potentially of high complexity. Having this in mind, in this paper, we propose a generic post-processing algorithm to improve the disaggregation accuracy after conventional event-based NILM is applied. In particular, after disaggregation, we cast an optimisation problem as minimising the distance between the sum of the disaggregated loads and the total measured consumption and add a regularization term to weigh the confidence in the accuracy of the initial disaggregation for each appliance. The resulting optimization problem is a boolean quadratic problem (combinatorial in nature) that is hard to solve exactly; we provide an effective heuristic based on relaxing the zero-one type constraints to the interval-type $[0, 1]$ constraints.

II. Problem formulation

The task of NILM is to estimate individual usage information of each appliance from aggregated meter data. Focusing on the case where the meter measures only active power, the aggregate reading from the meter can be expressed as:

$$P_i = \sum_{m=1}^{M} P_i^m + n_i, \tag{1}$$

where P_i and P_i^m are the total household's power and power of appliance m at time sample i, respectively, and n_i is the noise that includes measurement errors, base-load and all unknown appliances running. The power disaggregation task is now for $i \in [1, N]$ and $m \in [1, M]$ given P_i, to estimate P_i^m, where N is the total number of samples and M is the number of known appliances in the house. Two classes of NILM methods are proposed to address this problem: state-based methods (see [1], [3], [7], [8], [11] and references therein) and event-based methods.

Event-based NILM approaches usually consist of three steps [12]. The first step is event detection: detecting changes in time-series aggregated data (also called edges) due to one or more appliances changing their states. The second step is feature extraction: once events are detected, the electrical features, such as active power, profile between edges, duration, are isolated for each event. The last step is classification and pattern matching: different classification tools are used here to classify the events into pre-defined categories, each corresponding to a known appliance. Various classification tools have been used for event-based NILM [13], [14], [15], such as support vector machines (SVM) [6], neural networks [16], nonnegative tensor factorization [9], k-means [17], decision trees (DT) [4] and Graph signal processing (GSP) classification [10].

Let $\alpha_i^{m^*} \in \{0, 1\}$ with $i \in [1, N_E]$ and $m \in [1, M]$ represent results of an event-based NILM method, where N_E is the total number of time instances at which at least one event (event here is a change of state of an appliance) is detected. That is, $\alpha_i^{m^*} = 1$ means that after NILM, it is estimated that the ith event is caused by Appliance m being switched on or off; $\alpha_i^{m^*} = 0$ means that Appliance m was not turned on or off at Event i. For $j \in [1, N]$, let $\Delta P_j = P_{j+1} - P_j$.

Then, according to $\alpha_i^{m^*}$ and corresponding ΔP_j, we can decide whether the detected edge is a rising or falling edge and estimate $S_j^{m^*} \in \{0,1\}$, the state of Appliance m at the sample j, which is 1 if Appliance m is running at time sample $j, j \in [1,N]$, or 0 otherwise. Note that while $\alpha_i^{m^*}$ points to events when the appliance is turned on or off, $S_j^{m^*}$ indicates whether Appliance m is running at time sample j ($S_j^{m^*}=1$) or not ($S_j^{m^*}=0$).

In other words, given the variables $\alpha_i^{m*}, i=1,...,N_E, m=1,...,M$ we can recover the state S_j^{m*} of each appliance $m \in [1,M]$ for all time instances $j \in [1,N]$. For example, let there be only one non-zero α_i^{m*} that corresponds to Appliance $m=1$; let this quantity be α_3^1, i.e., Appliance 1 changes its state during Event 3. Suppose that from ΔP we also learn that this change corresponds to a raising edge. Then, we recover: $S_j^1=0$, for time instances j preceding Event 3, and $S_j^1=1$, for time instances j following Event 3.

Given the average working power $\overline{P^m}$ obtained from training or appliance manual, one can estimate the power consumption for each appliance at each time sample j as $\hat{P}_j^m = \overline{P^m} S_j^{m^*}$. This can further be refined via post-processing described next.

III. Post-processing for Event-based NILM

Before applying post-processing on event-based NILM algorithms we first introduce the fidelity term [9], [10] as:

$$\sum_{j=1}^{N} |P_j - P_j^0 - \sum_{m=1}^{M} S_j^m \overline{P^m}|^2 \tag{2}$$

where $\hat{P_j^m} = S_j^m \overline{P^m}$ is the estimated power consumption of Appliance m at time j and P_j^0 is the estimated base-load.

The fidelity term, given by Eq(2), represents the difference between aggregate power without the base-load, i.e., $P_j - P_j^0$, and the sum of the estimated loads after disaggregation, $\sum_{m=1}^{M} S_j^m \overline{P^m}$. In [18], an approach was introduced that uses the output of a NILM algorithm as a 'prior' for a NILM approach paired with a heuristic optimization scheme. In [10], simulated annealing (SA) is used to refine the disaggregation results from an event-based NILM approach for single-state appliances with high working power by changing the state of disaggregated appliances S_i^m, one at the time, to minimise (2).

Though changing the states of appliances S_i^m to minimise Eq. (2) sounds like a logical step that will inevitably lead to performance improvement, there are several reasons why minimising (2) might not provide more accurate results. First, we cannot distinguish two appliances with similar working powers $\overline{P^m}$ by minimising the fidelity term alone. Secondly, the fluctuations of power values around the mean $\overline{P^m}$ during the appliance operation is ignored. Thirdly, the sum of two or more appliance loads might be close to another load, leading to wrong fidelity minimisation. Finally, noise including measurement errors and unknown appliances is not taken into account. In summary, purely minimising Eq. (2) is not a robust way of disaggregating appliances, and hence it is rarely used alone in practice.

To improve the reliability and accuracy of post-processing, we introduce the influence of disaggregation results as regularization to the fidelity term. The objective function now becomes:

$$\min_{S_j^m \in \{0,1\}} \sum_{j=1}^{N} |P_j - P_j^0 - \sum_{m=1}^{M} S_j^m \overline{P^m}|^2 + \sum_{j=1}^{N}\sum_{m=1}^{M} \lambda_m |S_j^m - S_j^{m^*}|^2 \tag{3}$$

where MN optimisation variables, $S_j^m \in \{0,1\}$, $m=1,...,M$, $j=1,...,N$, take values from a discrete set (0 or 1), and λ_m is the weight of the regularization term for Appliance m. The regularization term $|S_j^m - S_j^{m^*}|^2$ in Eq. (3) reflects the difference between states of appliances estimated by optimising the fidelity term and states estimates given by NILM algorithms. Large λ_m means we have more confidence in the results of the original NILM for Appliance m. Small λ_m, on the other hand, means that we have less confidence in the NILM result, and put more weight in minimising the fidelity term. Note that λ_m is appliance dependent, to reflect the case that a NILM method is usually good for disaggregating certain appliances, and bad for others.

To reduce the computational complexity and considering that the event-based NILM algorithm will provide edge detection results α^*, we modify the objective function as:

$$\min_{\alpha_i^m \in \{0,1\}} \sum_{i=1}^{N_E} ||\Delta P_i| - \sum_{m=1}^{M} \alpha_i^m |\overline{\Delta P^m}||^2 + \sum_{i=1}^{N_E}\sum_{m=1}^{M} \lambda_m |\alpha_i^m - \alpha_i^{m^*}|^2 \tag{4}$$

to only optimise for sample i when the events are detected. The minimisation here is with respect to MN_E discrete variables α_i^m taking values 0 or 1.

We also use ΔP, instead of P. Similarly, we use $\Delta \hat{P}_i^m = \overline{\Delta P^m} \alpha_i^m$ to estimate power change of Appliance m at the ith event, where $\overline{\Delta P^m}$ is the average power change of the appliance when it changes states. N_E is the total number of events detected, which is usually much smaller than N, hence post-processing complexity has been reduced. We heuristically find the following choice for tuning parameters λ_m:

$$\lambda_m = \zeta \frac{\theta^2}{\overline{\Delta P^m}^2} \sum_{m=1}^{M} \frac{\beta^2}{|\overline{\Delta P} - \overline{\Delta P^m}|^2}. \tag{5}$$

where $\overline{\Delta P}$ is the average aggregate power change for events detected. As expected, λ_m is inversely proportional to the appliance mean power, which implies that for high loads, we put more weight on the fidelity term, since these loads contribute to the total aggregate the most. Second, being inversely proportional to $|\overline{\Delta P} - \overline{\Delta P^m}|^2$, λ_m depends on all household loads; that is, if Appliance m's mean power value is close to mean power of some other appliance, we rely less on the fidelity term.

Optimisation problems (3) and (4) are (combinatorial) boolean quadratic programs that are known to be very hard to solve exactly, e.g., [19]. To solve efficiently the optimisation problem, we introduce relaxation, that is, instead of being one or zero, S_i^m and α_i^m in Eq. (3) and (4) take soft real-number values in the set [0, 1]. This way, we can convert the minimisation problem in (3) and (4) to a convex optimisation problem, which enables the use of known convex optimisation tools (a problem with convex quadratic cost and (convex) box constraints).

To solve Eq. (4) we use CVX, a package for specifying and solving convex programs [20], [21]. In particular, to solve

Eq.(4), the infeasible path-following algorithm is used [22]. This method always finds a non-negative solution and uses two Newton steps per iteration.

After the above post-processing method is applied, and a solution $\alpha_i^{m*} \in [0,1]$ to the relaxed version of problem (4) is obtained, we replace $\alpha_i^{m^*}$ with $\alpha_{i,final}^m = 1$ if the optimal result obtained is larger than a pre-set threshold 0.5, and $\alpha_{i,final}^m = 0$, otherwise. In other words, we project the solution back to the discrete set $\{0,1\}$. We re-estimate the consumed power of each appliance using new $\alpha_{i,final}^m$ as $\alpha_{i,final}^m \Delta\overline{P^m}$.

IV. Results

To demonstrate effectiveness of the proposed post-processing method, we apply it to the output of two event-based NILM approaches, namely GSP [10] and DT [4]. We use the REFIT dataset [23], which contains active power measurements collected at every 8 seconds. The REFIT dataset contains data from 20 households in the UK monitored over a period of 2 years. In each house, only up to 9 appliances were sub-metered, hence there are many unknown appliances. We pick one month data which is April 2014 from three houses to test the performance of our proposed post-processing approach comparing the results with the SA post-processing used in [10] and the original disaggregation results [10], [4]. We use a previous month recordings for training. We heuristically find the scaling factors for Eq. (5) as $\theta \approx 500$, $\beta = 20$, $\zeta \approx 1000$.

We use F_M defined as $PR = TP/(TP+FP)$, $RE = TP/(TP+FN)$, $F_M = 2(PR*RE)/(PR+RE)$ as the evaluation metrics. F_M is adapted from [24] and has been used in many previous NILM papers to assess the appliance classification accuracy, where true positive (TP) stands for the number of edges we detected correctly, false positive (FP) is the number of edges we detected that do not actually exist and false negative (FN) indicates the number of edges of appliance that are not detected. High precision (PR) represents accurate detection with less FP, and high recall (RE) represents a higher number of events detected with less FN. F_M is the balance of PR and RE and is good at representing the event detecting accuracy, but has no information on energy disaggregation accuracy. Thus, we also use Accuracy ($Acc.$) [25] defined as:

$$Acc. = 1 - \frac{\sum_{j=1}^{N} |\hat{P}_j^m - P_j^m|}{2\sum_{j=1}^{N} P_j^m}. \quad (6)$$

During the simulations we found that the minimisation of Eq. (4) is solved by the proposed method using less than 300 iterations. In SA [10], fidelity term is optimised sample by sample to improve calculation efficiency, and each sample needs more than 300 iterations to coverage to the minimum. In Table I, we compare the execution time of the proposed method and SA for three REFIT houses. One month testing data contains 1546, 3341, and 2968 events detected, respectively, for Houses 2, 6, and 17. It can be seen from Table I that the proposed method converges to a minimum much faster than the SA.

Next, we compare the accuracy of the post-processing results. We use DW to label dishwasher, MW for microwave and WM for washing machine.

TABLE I
Comparison of execution time (sec/sample) between Simulated Annealing (SA) based post-processing and the proposed post-processing NILM for three REFIT houses.

	SA	Proposed
House2	0.23	0.008
House6	0.17	0.003
House17	0.15	0.002

TABLE II
$Acc.$ between original GSP [10] and DT [4], SA based post-processing and the proposed post-processing NILM for REFIT House2.

Appliances		Kettle	Toaster	DW	MW
GSP	NILM	0.61	0.53	0.78	0.71
	NILM+SA	0.79	0.68	-	0.71
	NILM+Proposed	0.81	0.74	0.79	0.83
DT	NILM	0.63	0.44	0.70	0.72
	NILM+SA	0.74	0.54	-	0.73
	NILM+Proposed	0.79	0.67	0.70	0.80

TABLE III
F_M between original GSP and DT, SA based post-processing and the proposed post-processing NILM for REFIT House2.

Appliances		Kettle	Toaster	DW	MW
GSP	NILM	0.80	0.60	0.75	0.79
	NILM+SA	0.81	0.64	-	0.80
	NILM+Proposed	0.92	0.71	0.80	0.85
DT	NILM	0.76	0.58	0.74	0.81
	NILM+SA	0.81	0.64	-	0.81
	NILM+Proposed	0.88	0.69	0.78	0.83

Tables II and III display the comparison of $Acc.$ and F_M between the proposed post-processing method, SA and original disaggregation results using GSP and DT NILM algorithm for House 2 in the REFIT dataset. It is clear that both SA and the proposed method improved the disaggregation result for all listed appliances with respect to the original NILM result. The proposed methods is also better than SA for the mentioned appliances.

Fridge has small mean $\overline{P^m}$, thus noise including measurement error, unknown appliances, and fluctuations from other high power loads will easily cause error when minimising the fidelity term. For that reason, the SA provides results with extremely low accuracy, so in [10], SA is not applied for refrigerator. Our proposed method avoids this problem via λ_m, the weight of regularization term, as in (5), leading to results very close to NILM algorithms, thus not included in the table.

SA purely optimises (2) and provides poor results for WM and DW in House 2. So the SA post-processing in [10] is not applied for these two appliances.

TABLE IV
$Acc.$ between original GSP and DT, SA based post-processing and the proposed post-processing NILM for REFIT House6.

Appliances		Kettle	Toaster	DW	MW
GSP	NILM	0.82	0.40	0.48	0.62
	NILM+SA	0.84	0.52	0.55	0.70
	NILM+Proposed	0.88	0.59	0.53	0.78
DT	NILM	0.76	0.45	0.43	0.61
	NILM+SA	0.80	0.55	0.52	0.69
	NILM+Proposed	0.83	0.55	0.53	0.74

Tables IV and V show the comparison of $Acc.$ and F_M for House 6 in the REFIT dataset. Fridge again is not included for the same reasons as in House 2. For appliances listed in the tables, both the proposed and SA post-processing methods

TABLE V
F_M BETWEEN ORIGINAL GSP, SA BASED POST-PROCESSING AND THE PROPOSED POST-PROCESSING NILM FOR REFIT HOUSE6.

Appliances		Kettle	Toaster	DW	MW
GSP	NILM	0.95	0.63	0.34	0.78
	NILM+SA	0.95	0.75	0.52	0.80
	NILM+Proposed	0.95	0.80	0.61	0.80
DT	NILM	0.91	0.58	0.41	0.81
	NILM+SA	0.91	0.64	0.46	0.80
	NILM+Proposed	0.93	0.67	0.49	0.81

show good improvement compared with NILM only, i.e., with no post-processing.

TABLE VI
$Acc.$ BETWEEN ORIGINAL GSP AND DT, SA BASED POST-PROCESSING AND THE PROPOSED POST-PROCESSING NILM FOR REFIT HOUSE17.

Appliances		Kettle	MW	WM
GSP	NILM	0.77	0.61	0.30
	NILM+SA	0.78	0.65	0.45
	Proposed	0.83	0.73	0.47
DT	NILM	0.69	0.65	0.36
	NILM+SA	0.75	0.69	0.44
	NILM+Proposed	0.79	0.74	0.45

TABLE VII
F_M BETWEEN ORIGINAL GSP, SA BASED POST-PROCESSING AND THE PROPOSED POST-PROCESSING NILM FOR REFIT HOUSE17.

Appliances		Kettle	MW	WM
GSP	NILM	0.93	0.70	0.56
	NILM+SA	0.94	0.74	0.62
	NILM+Proposed	0.93	0.78	0.62
DT	NILM	0.88	0.73	0.59
	NILM+SA	0.90	0.78	0.63
	NILM+Proposed	0.92	0.80	0.61

Tables VI and VII show results for House 17 in the REFIT dataset. For the tested appliances, the improvements for both SA and the proposed method are obvious. Except for kettle, which is accurately disaggregated with the original algorithm, the results for other two appliances show significant improvements due to post-processing, where again the proposed method outperforms SA [10].

V. Conclusion

In this paper, a post-processing method is introduced to help improve accuracy of event-based NILM algorithms. The performance of the proposed method is compared with simulated annealing in [10] using three houses from the REFIT dataset and two state-of-the-art event-based NILM methods. The proposed method has better performance than SA, the post-processing of [10], and lower processing time.

The proposed methodology involves as an intermediate step a heuristic to solve a (combinatorial) boolean quadratic problem through relaxing zero-one constraint sets to compact zero-one intervals. Future work will include a branch-and-bound algorithm and alternative convex relaxation methods including semidefinite programming (SDP)-based relaxations.

VI. Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 734331.

Research of D. Jakovetic is also supported by the Serbian Ministry of Education, Science, and Technological Development, Grant no. 174030

References

[1] A. Zoha, A. Gluhak, I. Muhammad, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[2] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Tras. Consumer Electronics*, vol. 57, 2011.

[3] O. Parson, S. Ghosh, M. J. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types.," 2012.

[4] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2014, pp. 535–540.

[5] S. Giri and M. Berges, "An energy estimation framework for event-based methods in non-intrusive load monitoring," *Energy Conversion and Management*, vol. 90, no. Supplement C, pp. 488 – 498, 2015.

[6] G. Lin, S. Lee, J. Y. Hsu, and W. Jih, "Applying power meters for appliance recognition on the electric panel," in *2010 5th IEEE Conf. Industrial Electronics and App.*, June 2010, pp. 2254–2259.

[7] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, Neil D. Lawrence and Mark A. Girolami, Eds., 2012, vol. 22, pp. 1472–1482.

[8] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. the 2011 SIAM Int. Conf. Data Mining*. SIAM, 2011, pp. 747–758.

[9] M. Figueiredo, B. Ribeiro, and A. de Almeida, "Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home," *IEEE Trans. Instrum. and Measurement*, vol. 63, no. 2, pp. 364–373, Feb 2014.

[10] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, in press, 2017.

[11] D. Egarter, V. P. Bhuvana, and W. Elmenreich, "Paldi: Online load disaggregation via particle filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 467–477, Feb 2015.

[12] K. D. Anderson, M. E. Bergs, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event detection for non intrusive load monitoring," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct 2012, pp. 3312–3317.

[13] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Enhancing electricity audits in residential buildings with nonintrusive load monitoring," *J. Industrial Ecology*, vol. 14, no. 5, pp. 844–858, 2010.

[14] H. Goncalves, A. Ocneanu, M. Berges, and RH Fan, "Unsupervised disaggregation of appliances using aggregated consumption data," in *The 1st KDD Workshop Data Mining App. in Sust. (SustKDD)*, 2011.

[15] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Learning systems for electric consumption of buildings," in *Computing in Civil Engineering (2009)*, pp. 1–10. 2009.

[16] D. Srinivasan, W. S. Ng, and A. C. Liew, "Neural-network-based signature recognition for harmonic source identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, Jan 2006.

[17] H. Altrabalsi, V. Stankovic, J. Liao, and L. Stankovic, "Low-complexity energy disaggregation using appliance load modelling," *AIMS Energy*, vol. 4, no. 1, pp. 884–905, 2016.

[18] T. Jaakkola and J. Z. Kolter, "Approximate inference in additive factorial hmms with application to energy disaggregation," .

[19] Ma Luo, Z., W.-K., A. Man-Cho So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Sig. Proc. Magazine*, May 2010.

[20] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[21] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," *Recent advances in learning and control*, pp. 95–110, 2008.

[22] P. Tseng, "An infeasible path-following method for monotone complementarity problems," *SIAM J. Optimization*, vol. 7, no. 2, pp. 386–402, 1997.

[23] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Nature Scientific Data*, vol. 4, pp. 160122, 01 2017.

[24] D. L. Olson and D. Delen, "Advanced data mining techniques," Springer, 2008.

[25] J.Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," *In Proc. Workshop Data Mining App. Sust. (SIGKDD)*, vol. 25, pp. 59–62, Jan 2011.