Deep Learning Application to Non-Intrusive Load Monitoring

Nguyen Viet Linh, Pablo Arboleya

Department of Electrical Engineering
University of Oviedo
Gijon, Spain
uo254212@uniovi.es, arboleyapablo@uniovi.es

Abstract—Non-intrusive Load Monitoring (NILM) is a technique which accepts the total consumption in a house as an input and computes the estimated demand of individual appliances in that house. All NILM needs is one single meter to record the aggregate signals. With NILM, the users can assess the information of appliances without the intrusion of measurement devices. Such information could help users adapt their energy-usage habit for better saving and facilitate the grid management. This paper presents the disaggregation with recurrent neural network (RNN) in NILM. The Reference Energy Disaggregation Data Set (REDD) has been employed to test the proposed method. The results show the better performance achieved by RNN, compared with the optimization approach. RNN also performs a fair job to test unseen appliances.

Index Terms—Non-intrusive Load Monitoring, Energy Disaggregation, Recurrent Neural Network, Genetic Algorithm, Reference Energy Disaggregation Data Set

I. INTRODUCTION

Firstly introduced in 1992 [1], non-intrusive load monitoring (NILM) is the method to break down the total household demand into the detail consumption of individual appliances. In this method, only a single meter is required to measure the total power demand in a house. The computational techniques are then applied to disaggregate the aggregated signal into individual appliances. The output information can help users manage their energy usage, facilitate the demand response techniques, or detect faulty devices.

The disaggregation methods for NILM could be divided into event-based and non-event-based (event-less) techniques. Event-based approaches capture the changes of appliance states during the switching events [2]. This approach usually requires the high-frequency sampling in order to extract rich features for high accuracy. Some related work extracted those transient features at kHz sampling frequency [3]–[5].

Non-event-based methods do not detect any event transition. Instead, all power states are taken into account. Research based on this approach has studied Hidden Markov Models [6]–[9] for generally modeling appliances or optimization frame [10], [11].

In addition, some researchers also focused on the energy disaggregation with external information. Outside temperature, time of the day, or day of the week are also considered important features [12]. Such information is correlated with individual consumption in an optimization frame.

II. REFERENCE ENERGY DISAGGREGATION DATASET

In 2011, Kolter and Johnson introduced the Reference Energy Disaggregation Dataset (REDD) [13]. The goal is to create a benchmark dataset for evaluating the disaggregation algorithms. REDD covers six houses in Massachusetts, USA, and consists of both main-circuit and individual-circuit levels. At the household level, the main circuits are sampled at 15-kHz in high frequency and 1-Hz in low frequency. At the appliance level, the sub-meters are sampled at 0.3 Hz. One major objective of our work is to create a NILM web server where the inputs are the total consumption of a household, and the outputs are the disaggregated estimations. Several papers have implemented REDD as the main object to test their disaggregation algorithms [14]–[17], which has shown positive outcomes. Since REDD is widely used for benchmarking the NILM algorithms, it is the selected option in this study.

In this work, a simple web interface is initially built for visualizing REDD. When one house is selected, the reading of the main circuit is displayed. The meter of a corresponding appliance will be visualized below as the users select that appliance. Fig. 1 shows the 24-hour record in House 1. This paper only focuses on the low-frequency data as they are in a realistic scenario for the current technology of smart meters. The data from house 1 and 2 are taken into consideration for the disaggregation.

III. PROPOSED METHODS

A. Data Preprocessing

1) Data Filtering: Firstly, a simple median filter is applied for the noise removal of the raw dataset. The selected filter order is 30 in this case. The next step is to threshold the power states of individual appliances. The objective is to cluster the different states for combinatorial optimization (CO) and a binary on/off vector for the manual labeling of recurrent neural network (RNN).

2) Power States Clustering: k-means is the common clustering algorithm for partitioning the dataset into k clusters from n observations or data points [18]. With the simplicity, the k-means algorithm is employed for
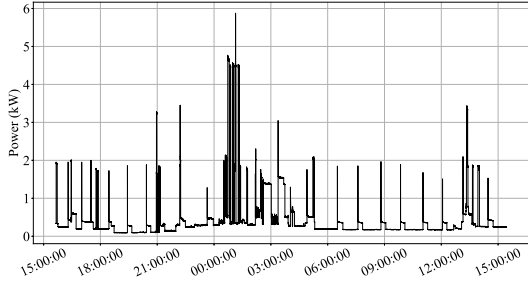
(a) Aggregated power
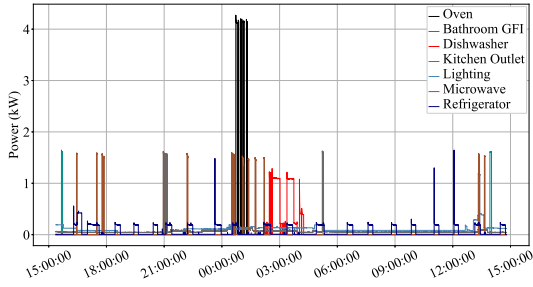
(b) Ground-truth of some individual appliances

Fig. 1. Consumption of house 1 from 18th- 19th April, 2011.

power states identification. The non-off power states of appliances in house 1 and 2 can be summarized in Table I-II, respectively.

TABLE I
NON-OFF POWER STATES IN W OF HOUSE-1 APPLIANCES

Appliances	State 1	State 2	State 3	State 4
Oven	4117.5	-	-	-
Washer Dryer	2657	-	-	-
Microwave	1525	-	-	-
Lighting	45	64	82	-
Kitchen Outlets	22	64	1047	1522
Refrigerator	191	-	-	-
Dishwasher	221	1102	-	-
Bathroom GFI	1596	-	-	-

TABLE II
NON-OFF POWER STATES IN W OF HOUSE-2 APPLIANCES

Appliances	State 1	State 2	State 3
Dishwasher	249	1196	-
Refrigerator	162	-	-
Kitchen Outlets	775	1059	-
Lighting	106	156	289
Microwave	1859	-	-
Stove	407	-	-

B. Combinatorial Optimization

1) Basic Principle: The optimization-based algorithm of disaggregation can be described as follows. The time-series aggregated power of a house is recorded $P = \{P_1, P_2, ..., P_t\}$ where P_t is the kW reading at t^{th} time instant. Assume

that the house has m appliances. The state of kth appliance can be described by a binary vector $X = \{x_1, x_2, ..., x_m\}$. When the kth appliance is on $x_k = 1$; otherwise, $x_k = 0$. The individual consumption of each appliance can be expressed in a vector $p = \{p_1, p_2, ..., p_m\}$. At every t^{th} instant, the combined demand should be equal the sum of all individual demands, as can be expressed:

$$P_t = \sum_{k=1}^{m} p_k x_k(t) + w_t \qquad (1)$$

where w_t is the error.

In supervised learning, the rating power of each appliance should be known. Then, for every iteration of t, we should try to search a variable vector X_t such that w_t is minimized, i.e.

$$\min_{X_t} \left| P_t - \sum_{k=1}^{m} p_k x_k(t) \right| \qquad (2)$$

Assume that the mth appliance has l states. Since any appliance must be in only one state at the tth time instant, the constraint for multi-state appliance can be described:

$$x_{m,1}(t) + x_{m,2}(t) + ... + x_{m,l}(t) \leq 1 \qquad (3)$$

In this work, the genetic algorithm (GA) MATLAB Toolbox is employed to optimize Eq. 2. The selected numbers of generation and population are 100 and 200, respectively.

2) Correction Rule: The result of GA optimization could lead to short-time transition where one load can be in its state for only a few samples. Generally, it is not possible, so a correction rule should be applied to filter the short-time spikes or notches, as follows [11]:

Algorithm 1 Correction Rule

   $X \leftarrow$ *A binary matrix as a result of GA optimization*
   if median(X_t,X_{t-1},…,X_{t-M}) = 0 and X_{t-M} = 1
   then
      $X_{t-M} = 0$
   else if median(X_t,X_{t-1},…,X_{t-M}) = 1 and X_{t-M} = 0
   then
      $X_{t-M} = 1$
   end if

This rule implies that one appliance will remain in its state at least within M samples. In this paper, $M = 5$.

C. Recurrent Neural Networks

Artificial Neural Network (ANN) has been the powerful pattern recognizer for the complicated problem sets. Basically, an ANN architecture consists of an input layer, one or multiple hidden layers, and one output layer.

In multilayer feedforward neural network (MFNN), the connection among layers is completely straightforward and open-loop. MFNN has been commonly used for nonlinear classification. The term of classification in this work denotes a mapping from the aggregate power to the binary on/off vectors of one targeted appliance.
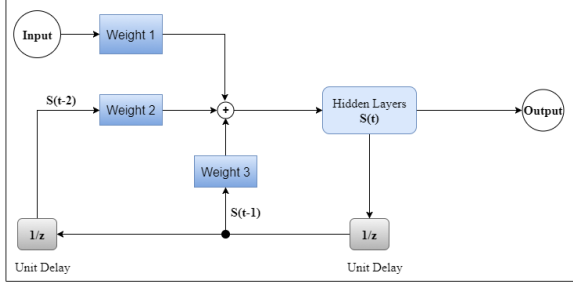
Fig. 2. A simple RNN structure with two delays.

Recurrent Neural Network (RNN) is an advanced version of MFNN where the information is propagated not only forward but also backward in each layer. Fig. 2 exemplifies a RNN structure with two delays in one hidden layer. This means that the values of the time series at time $t-1$ at $t-2$ are stored in the internal memory. At time t, they are simultaneously loaded together with the tth value for processing. In other words, the previous outputs of RNN at $t-1$ and $t-2$ affects the decision at t. With the capability of recognizing the time-series signals, RNN can be the good candidate for energy disaggregation.

In RNN training, tuning hyper-parameters plays a crucial role in optimizing the RNN performance. In MATLAB, such hyper-parameters are configured and adjusted during experiments, as follows:

- Maximum number of epochs: 200
- Number of delays: 5
- Number of hidden layers: 10
- Training function: Levenberg-Marquardt backpropagation (trainlm)

D. Metrics

The disaggregation metrics are followed by the formula of proportion of total energy correctly assigned over a period T [11], [13], [19].

$$\begin{matrix} \text{Total energy} \\ \text{correctly assigned} \end{matrix} \% = 1 - \frac{\sum_{t=1}^{T}\sum_{i=1}^{n}|P_i^* - P_i|}{2\sum_{t=1}^{T}\overline{P_t}} \quad (4)$$

$$\begin{matrix} \text{Relative error in} \\ \text{total energy} \end{matrix} \% = \frac{|E^* - E|}{\max(E^*, E)} \quad (5)$$

Additional metrics to test unseen appliances:

$$\begin{matrix} \text{Energy correctly assigned} \\ \text{per appliance (ECApA)} \end{matrix} \% = 1 - \frac{\sum_{t=1}^{T}|P_i^* - P_i|}{2\sum_{t=1}^{T}P_i} \quad (6)$$

where:

- P_i^*: Estimated power of appliance i^{th} at t.
- P_i: Actual power of appliance i^{th} at t.
- $\overline{P_t}$: Aggregated actual power at t.
- E^*: Total predicted energy.
- E: Total actual energy.

Metrics based on the confusion matrix are also useful to evaluate the disaggregation performance. The general confusion matrix can be described in Table III. Then, the

Total Population	Ground-truth	
	Positive	*Negative*
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

classification metrics could be followed:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

$$\text{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

$$\text{Accuracy} = \frac{TP + TN}{P + N} \quad (10)$$

In order to identify positive and negative labels in RNN, the discrimination threshold should be defined. This threshold can be tuned by receiver operating characteristics (ROC) curve. ROC curve shows the relationship between Recall versus False-positive-rate at various threshold settings. If the area under ROC curve (AUC) is close to 1, the classification is close to the perfection ($0.5 \leqslant \text{AUC} \leqslant 1$). Therefore, the threshold can be tuned to obtain the optimal value where AUC is maximized (optimal operating point).

IV. EXPERIMENTAL RESULTS

A. Combinatorial Optimization (CO)

Fig. 3a and Fig. 4a show the disaggregation performance of combinatorial optimization. It is obvious that CO model could well disaggregate refrigerators and lightings. One remark is that such appliances are regular in operation. In particular, the operation of refrigerators is periodic.

On the other hand, the disaggregation of dishwashers is unsuccessful in most cases. The high accuracy scores in dishwasher recognition are due to their high portions of true negative. This means that dishwashers are off for a long time. To be more precise, the CO model is totally unable to detect the ON state of the house-1 dishwasher, where its F1 score returns null.

For the kitchen outlets, the optimization outputs tend to return low accuracy. This is due to the difficulty in clustering their exact power states as their state differences are insignificant. It is noticeable that kitchen outlets always consume some powers from the household networks.

Several outlets which consume high power like bathroom GFI, washer dryer, or microwave could be fairly disaggregated. This is understandable as their non-off power states are quite distinctive. However, we could not state that the disaggregation with CO is successful if the average total energy correctly assigned is only around 50%.
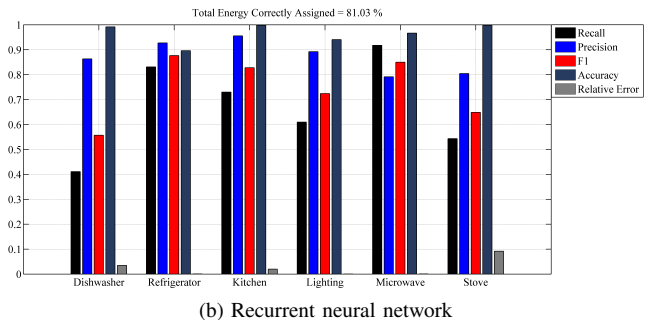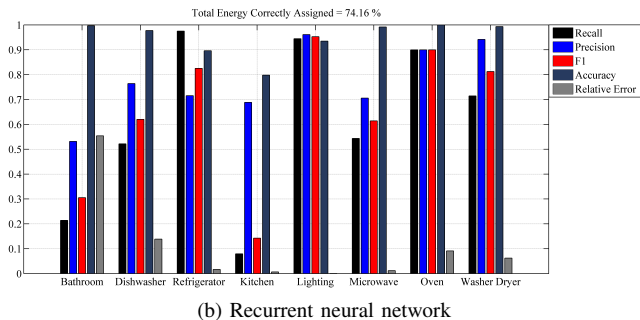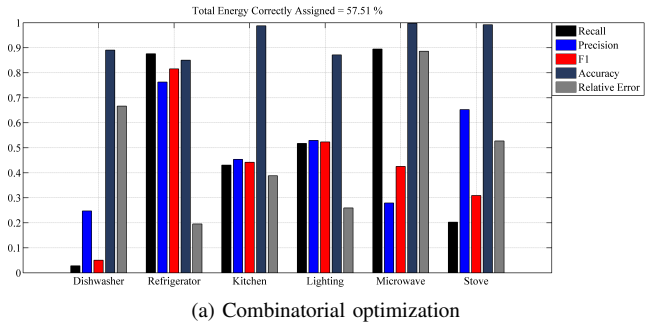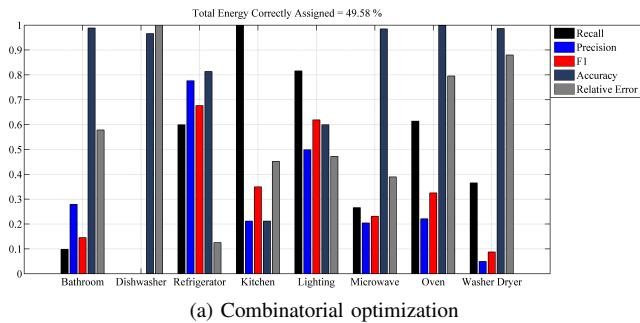
(a) Combinatorial optimization

(b) Recurrent neural network

Fig. 3. Disaggregation results of house 1.

(a) Combinatorial optimization

(b) Recurrent neural network

Fig. 4. Disaggregation results of house 2.

B. Recurrent Neural Network (RNN)

1) Test Seen Houses: Fig. 3b and Fig. 4b show the RNN outcomes in the case of seen houses. The training is implemented in one house, and the test is done in the same house but in a different period from the training set. For the same house, RNN obviously outperforms CO on every appliance on F1, accuracy, total assigned and relative error. In particular, RNN is able to detect the on-state of dishwashers while CO is not. Some examples of RNN outputs can be also visualized. It is obvious that RNN well re-constructs the appliance signatures from the aggregated power. This clearly explains why RNN is the potential candidate for disaggregating household appliances.

2) Test Seen Appliances: In the next circumstance, two common appliances: dishwasher, and refrigerator are selected for the unseen-house experiment. The data selection for this scenario can be listed below:

- Dishwasher: Training in house 1, 2, 3, 4 (3 days/house). Test in house 5
- Refrigerator: Training in House 1, 2, 3, 6 (3 days/house). Test in house 5

The purpose is to consider how good the generalization of RNN model is, for the unseen data. Without the prior information of power states, CO could not follow this scenario.

Fig. 6 describes the disaggregation performance in this case study. It is noted that the RNN has not succeeded in generalizing the dishwasher where its metrics are too low. Three days per house is possibly insufficient for the RNN model to learn the information in this case. Another remark is that the power in house 5 is recorded within only two days. If the RNN model aggregates the prediction over a

(a) House 1 on 18/04/2011.

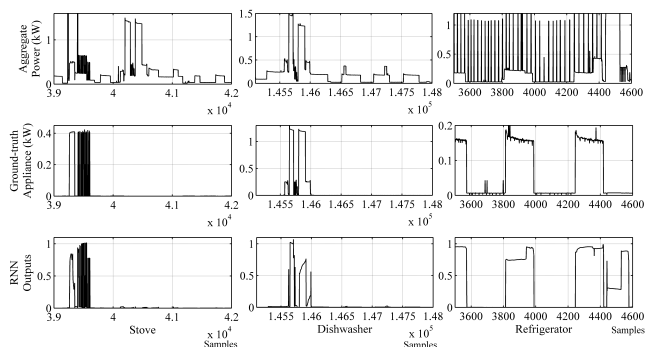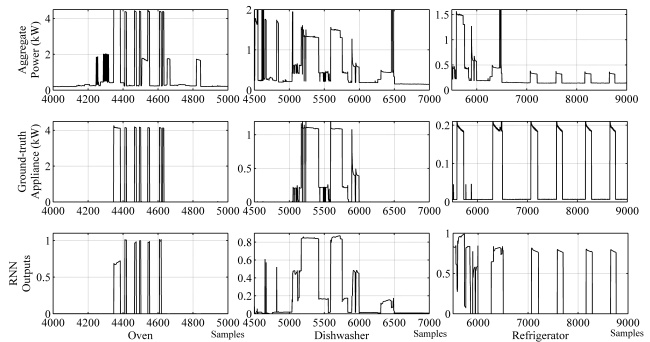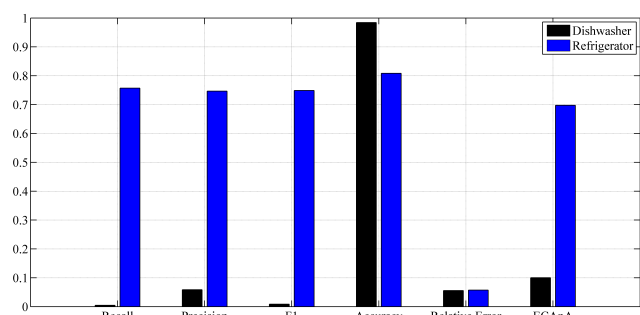(b) House 2 on 18/04/2011 (Stove, Refrigerator) and 02/05/2011 (Dishwasher)
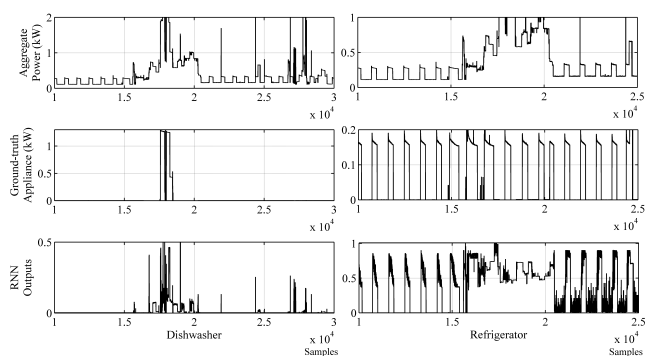
Fig. 5. Some examples of RNN outputs (seen houses).

longer duration (more test dataset), the errors tend to be reduced.

V. CONCLUSION

This paper has compared two methods in energy disaggregation algorithms. One is the eventless-based combina-

(a) Disaggregation metrics.

(b) RNN outputs on 19/04/2011.

Fig. 6. Results of RNN in unseen dishwasher and refrigerator in house 5.

torial optimization (CO) and the other is the event-based recurrent neural network (RNN). The outcomes conclude that CO mostly falls behind RNN in those case studies, and RNN could work well for the unseen case studies. However, some improvement is considerably needed to enhance the RNN performance for the multi-state appliances.

The CO model has been unable to disaggregate the dishwasher. This is the fundamental issue of non-unique solutions in optimization frame as one power state of an appliance can be equal to the sum of power states from other appliances. Additionally, as the non-event method, CO could not describe the physical characteristics of multi-state appliances.

In RNN, the training process is computationally burdensome, which requires several days up to months to achieve good performance. There is the possibility that training has to be re-done if the end-users buy or replace new appliances. This could frequently occur depending on their usage habit and how well the NILM models can generalize the unseen houses. Due to the limitation of our computer systems, only a small days per house can be used for training. The more the training data are provided, the better the network outputs are produced. Overall, RNN requires the powerful computer hardware. The current technology of embedded systems could not possibly put the complex RNNs into real-world applications. At this moment, one possible solution is to give the customer the off-line feedback.

REFERENCES

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

[2] 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.

[3] H. H. Chang, K. L. Chen, Y. P. Tsai, and W. J. Lee, "A new measurement method for power signatures of nonintrusive demand monitoring and load identification," *IEEE Transactions on Industry Applications*, vol. 48, no. 2, pp. 764–771, March 2012.

[4] M.-S. Tsai and Y.-H. Lin, "Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation," *Applied Energy*, vol. 96, pp. 55 – 73, 2012, smart Grids. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261911007240

[5] Y. H. Lin and M. S. Tsai, "Development of an improved time-frequency analysis-based nonintrusive load monitor for load demand identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 6, pp. 1470–1483, June 2014.

[6] SIAM, Ed., *Unsupervised Disaggregation of Low Frequency Power Measurements*, 2011. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972818.64

[7] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proceedings of theTwenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, July 2012, pp. 356–362. [Online]. Available: https://eprints.soton.ac.uk/336812/

[8] S. Makonin, F. Popowich, I. V. Baji, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, Nov 2016.

[9] K. A. Agyeman, S. Han, and S. Han, "Real-time recognition nonintrusive electrical appliance monitoring algorithm for a residential building energy management system," *Energies*, vol. 8, no. 9, pp. 9029–9048, 2015. [Online]. Available: http://www.mdpi.com/1996-1073/8/9/9029

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

[11] M. Z. A. Bhotto, S. Makonin, and I. V. Baji, "Load disaggregation based on aided linear integer programming," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 792–796, July 2017.

[12] M. Wytock and J. Z. Kolter, "Contextually supervised source separation with application to energy disaggregation," in *AAAI*, 2014.

[13] J. Z. Kolter and M. J. Johnson, "REDD: A Public Data Set for Energy Disaggregation Research," in *SustKDD Workshop on Data Mining Applications in Sustainability*, 2011.

[14] 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, Sept 2013.

[15] T. Hassan, F. Javed, and N. Arshad, "An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 870–878, March 2014.

[16] 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.

[17] Z. Guo, Z. J. Wang, and A. Kashani, "Home appliance load modeling from aggregated smart meter data," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 254–262, Jan 2015.

[18] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theor.*, vol. 28, no. 2, pp. 129–137, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.1982.1056489

[19] J. Kelly and W. J. Knottenbelt, "Neural NILM: deep neural networks applied to energy disaggregation," *CoRR*, vol. abs/1507.06594, 2015. [Online]. Available: http://arxiv.org/abs/1507.06594