Aalto University
School of Science
Degree programme in Engineering Physics and Mathematics

Energy disaggregation with event-based detection and hierarchical clustering

MS-E2108 Independent research projects in systems analysis
October 17, 2016

Santtu Klemettilä

Advisor: M. Sc. (Tech.) Harri Mäkelin, HMA Analytics
Supervisor: Prof. Ahti Salo

Contents

1 Introduction

Global climate change has triggered a rising concern over energy consumption, which has had an effect on the market in means of energy production and in the increasing share of energy effective electric appliances available. In addition to its ecological significance, reducing energy consumption has an economical perspective to it. This study focuses on energy consumption from a household point of view.

According to the survey by Mettler-Meibom and Wichmann [14], consumers' estimates on energy consumption of specific end uses tend to deviate from their actual values. Energy used for heating tends to be highly underestimated, whereas energy used for appliances, lighting and cooking tends to be overestimated.

Consumers overestimate also the effectiveness of short-term energy conservation methods such as turning off the lights and underestimate long-term solutions such as replacing an inefficient appliance or enhancing a home's insulation [9]. Better knowledge on the actual distribution of energy consumption could help consumers make more meaningful decisions in their efforts on energy conservation.

In this study a method for Non-Intrusive Load Monitoring (NALM or NIALM or NILM) is proposed. The goal in energy disaggregation is to break down total energy usage into energy usage of individual electric appliances. This can be accomplished, for example, by recognizing patterns or by matching a priori information on the data. The non-intrusiveness in NALM means that the electrical infrastructure does not need any additional monitoring gadgets since the data is collected from the total consumption meter only. During the past few decades, numerous methods have been developed for this task.

Another aspect to the benefits of energy disaggregation is the possibility to plan one's energy consumption better with respect to time. The spot price of electricity may vary strongly depending on the time of day, and thus the time of consumption is of high importance. Many households nowadays also have solar panels, which generates income dependent on the spot price. Therefore a household with solar panels may gain double the profit by planning the time of consumption wisely.

Household energy meters provide data primarily for billing reasons, but they can be used for acquiring total energy consumption data for other reasons as well. Nowadays there are special smart meters that gather and send the consumption data forward and enable data collection for personal use.

The data for this study, however, is collected with a smart phone application by observing the blinking lights that appear in most energy consumption meters. The approach is therefore less dependent of the novelty of equipment available, since only a smart phone is needed rather than a high-end smart meter. In contrast to usual setups, the data used in this study does not include any labels for the events nor does it contain any information about the appliance configuration present in the sample household.

The rest of this study is structured as follows: The main principles of NALM and related work is discussed in section 2. The method for collecting data is introduced and the data is discussed in section 3. The proposed method for energy disaggregation is presented in section 4. Finally the results are summarized and further discussion is conducted in section 5.

2 Background

Traditional load monitoring [16] requires complex hardware for data collection. This, however, is seldom possible and is not very scalable, which is why NALM methods have become popular. They require only simple hardware, but the complexity is in the software.

2.1 Non-intrusive Load Monitoring

Hart [6] divides NALM methods into two groups:

- Manual setup: A one-time intrusive period is required to setup the monitoring. The intrusive period allows to recognize individual appliance signatures and thus enables further monitoring without intrusive instruments.
- Automatic setup: Only a priori information about the characteristics of present appliances may be used.

Some studies ([8], [11]) dub methods that require manual setup as supervised, in contrast to methods with automatic setup, which are dubbed as unsupervised. In this study, an automatic setup approach is used, but without any a priori information on the individual appliances.

As proposed by Hart [6], electrical appliances can be divided into three categories:

- Two-state: Appliances like light bulbs or toasters, which are simply just ON or OFF on a constant level of power. This study focuses on monitoring appliances from this category.
- Multistate: Appliances like washing machines and dishwashers, which go through multiple ON states during the time they operate.
- Continuously variable: Appliances like light dimmers and variable-speed tools, which have a continuous range of ON states. These appliances are difficult to monitor, since there are no step changes in the power signal.

Hart [6] concentrates on low-frequency (1 kHz or slower) data, which practically only enables observation of steady states. Steady state observation means the detection of phases where the power level somewhat stable for a certain minimum amount of time. The phases between steady states are called transients.

Leeb et al. [13] have studied NALM with high-frequency data (over 1 kHz). When observing high-frequency data, is is possible to recognize the transient fingerprints of appliances. Usually the transient phases are so fast that low-frequency observation only enables the detection but not the characterization of such phases. The sharper resolution on transient phases allows for more elaborate sensitivity to individual appliances, since the method is then not reliant only on the power level information.

Yang et al. [17] and Farinaccio and Zmeureanu [3] study the disaggregation problem using detection of ON and OFF events. Farinaccio and Zmeureanu [3] propose a pattern recognition approach, which is the approach of this study as well.

2.2 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models that were initially introduced by Baum and Petrie [2] and have thereafter been used successfully in a wide range of applications, such as speech recognition [15]. They are an extension to Markov models, and consequently states are not observable but rather connect to the system output through a probability distribution.

Ghahramani and Jordan [4] introduced a generalization for HMMs, where the hidden state is factored into multiple state variables. These generalizations are called factorial hidden Markov models (FHMMs). FHMMs have been widely applied in energy disaggregation problems (e.g. [10, 11]). Because of their scalability when discovering multiple independent factors they perform better in modeling time series of multiple appliances than HMMs, which require exponentially many parameters in order to represent all the states [10].

2.3 Deep Neural Networks

Kelly and Knottenbelt [8] present an open source solution for using deep neural networks on energy disaggregation. Their benchmarking against FHMMs and combinatorial optimization shows that their approach yields better results.

Artificial neural networks (ANNs) consist of nodes which represent artificial neurons and edges allow information to flow from one node to another. ANNs have an input layer, an output layer and hidden layers between the two. Each artificial neuron takes in the weighted sum of its inputs and passes the sum through its activation function and thus produces its output. The process of information going from the input layer to the output layer through hidden layers is called a forwards pass. [8]

Setting up a neural network means updating the weights for the connections between the neurons. The goal is to minimize the error between the network output and the expected output. Considering the amount of parameters in the model, plain enumeration of the error surface is usually not an option. Hence algorithms like back propagation must be used. [8]

Because of the amount of parameters, neural nets require a huge amount of training data. In deep learning applications, it is common to produce training data by duplicating and slightly modifying the real training data. In some applications one might even create simulated data. In energy disaggregation it is possible, for example, to artificially create data by randomly combining different appliance outputs. [8]

3 Data

There exist a few popular benchmarking datasets for energy disaggregation like the REDD dataset [12], the UK-DALE dataset [7] and the BLUED dataset [1]. In this study, however, we use a different dataset in order to study the possibilities of the equipment and software used.

The data has been collected from a single household in Espoo, Finland in the summer 2016 by using the mobile application iSmartMeter[1]. The iSmartMeter application tracks the LEDs of a smart meter that blink according to the energy that is used. The power consumption can thereby be calculated by observing the interval between the blinks. For example, the active power LED blinks every 5 seconds. One blink corresponds to one Wh (Watt-hour) of energy, so the power is $\frac{3600s/h}{5s} * 1Wh = 720W$. Since we only know the energy consumption by the resolution of one Watt-hour, we may only calculate the average power between the blinks.

The period of collection is 54 days. The household has a solar panel, which disrupts the consumption data during daylight. The meters do not show, whether the energy flow is inbound or outbound, so there are times, when the solar panels produce more energy than is consumed and the drops in the power signal are actually increments in the power consumption. Filtering the solar panel power out from the data would require further research on its characteristics in regard to weather and time of day, so in this study we concentrate only on the data outside daylight hours.

In addition to the solar power, there seems to have been something interfering with the data collection during the first 18 days and the last 3 days, which makes the data strangely volatile. For this reason, we omit those days from the data as well.

4 Energy Disaggregation

In this section we propose an energy disaggregation method that uses hierarchical clustering to combine the events into groups that could be from one individual appliance. First the data is denoised and then the steady state periods are recognized after which the events are clustered by hierarchical clustering.

4.1 Denoising

As can be seen in figure 1, the data collection method used in this study generates some error measurements, when blinks are either missed or detected twice. To filter out these mismeasurements, we must drop out all values that clearly stand out from their adjacent values.

[1]http://www.ismartmeter.com

Let V be the measurement value vector. First we calculate the difference vector V' for the value vector and the difference vector V'' for the difference vector. We take out the value on index i if $V'_{i-1} > \tau$ and $|V''_{i-1}| > \phi * |V'_{i-1}|$, where τ is the threshold value for the difference and ϕ is the threshold factor for the second difference. We use $\tau = 1000W$ and $\phi = 1.8$. The filtered data is shown in figure 2.

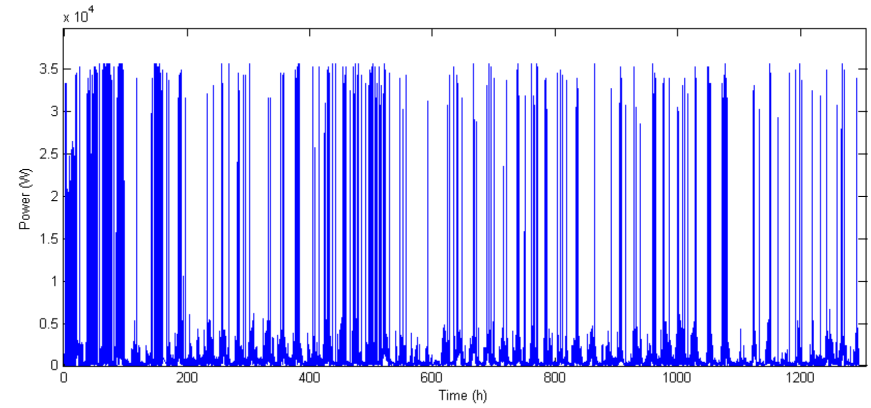

Figure 1: Data before denoising

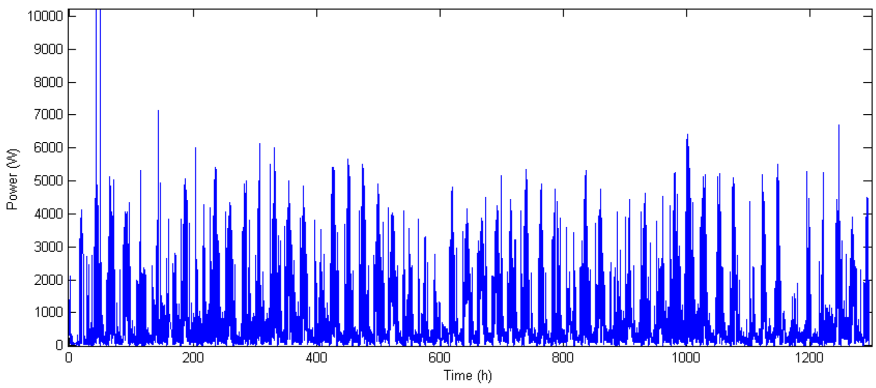

Figure 2: Data after denoising

4.2 Event Recognition

In our approach, the biggest challenge is event recognition. The data is split up into events that consist of a step up followed by a step down in the power level. These events will be further analyzed and clustered in section 4.3.

In order to spot the appliances in the data, we begin by examining the points where power level rises over a threshold value of t_δ and a percentage threshold level of p_δ. For each such point we examine the next n_s points for a beginning of an event. Starting from the first beginning point candidate, we look at the next n_e points for a possible constant power level. The level is constant if a maximum of t_n events deviate from the value of the beginning point at maximum by the threshold deviation t_e. In order to prevent duplicate events from being created on a single true event, subsequent points that surpass the threshold levels may only produce one event.

If a candidate for a constant power level phase is found, we examine the n_b events before the first deviation (i.e. the point where power level rises at least t_δ). These events form the base level in determining the power value for the event. An additional requirement for an event to be acknowledged is that the base level events all be at least t_δ below the mean value of the event values inside the threshold interval. If all the requirements apply, an event is recognized. The power value is determined as the difference of the mean of the event values inside the threshold interval and the mean of the base level events.

The previous procedure allows us to determine the beginning points for the events. In order to determine the ending points, we conduct the same procedure, but with the data order inverted.

The next step is to combine the start and end points of the events. Our implementation handles only step changes that do not occur simultaneously. This means that if two or more appliances are either turned on or off simultaneously, the step change is interpreted as having only one source. Such changes could be addressed to the corresponding step changes when followed or preceded by multiple smaller step changes of a summed value same as the change.

4.3 Hierarchical Clustering

Clustering is a task, in which objects are grouped into groups (clusters) so that the objects in one cluster are more similar to one another than with those of another group. Hierarchical clustering is a method which seeks to form a hierarchy of clusters.

In order to cluster the events, we need to standardize the data so that the variables would have a controlled influence on the outcome. Here we standardize the data so that for each variable the mean is zero and the standard deviation is one. The new value z_{ij} for each value v_{ij} can thus be calculated as:

$$z_{ij} = \frac{x_{ij} - m_j}{\sigma j}, \tag{1}$$

where m_j is the sample mean of the variable j and δ_j is the sample standard deviation for the variable j.

There are numerous ways to determine the distance between two measurement points for the clustering method. It depends on the context, which metric is the most suitable. In our case we use the most common metric, Euclidian distance, to determine the distance between two points.

In order to decide which cluster composition is optimal, one must decide on the linkage criteria. There are many ways to determine the distance between groups as well. Some of the most common criteria are single-linkage, complete-linkage, UPGMA and UPGMC. In single-linkage clustering the distance is counted as the minimum distance between any two objects not in the same group, whereas in complete-linkage the maximum distance is taken into account. UPGMA determines the distance between two clusters as the mean distance between points from different groups. In UPGMC the distance is calculated as the distance between the centroids of the two groups.

In this study, we use Ward's linkage as our linkage criteria. Ward's linkage determines distance between two clusters as the increase in the sum of squares of the distances between all objects within the cluster that would be the result of joining the two clusters.

Hierarchical clustering methods are divided into agglomerative and divisive methods. In the agglomerative approach, each object starts its own cluster, which is then paired with another cluster until all the objects are connected. Divisive methods start with one cluster, which is split recursively further down the hierarchy.

In this study we use an agglomerative hierarchical clustering method. In the initial setting every object forms its own cluster. The method proceeds by combining the cluster such that every step is optimal for the cluster composition in means of the linkage criteria.

The results of hierarchical clustering are usually presented in a dendrogram. A dendrogram is a hierarchical binary cluster tree, which shows the linkages between clusters and their corresponding distances. The original clusters are located on the horizontal axis and connected recursively with upside-down

U-links according to the progress of the clustering method. The height of the links represent the distance between the clusters connected.

5 Results

We set the following parameters for the algorithm explained in section 4.2: $t_\delta = 25$, $p_\delta = 5\%$, $n_s = 20$, $n_e = 20$, $t_n = 4$, $t_e = 20$ and $n_b = 2$. Because there is no ground data, there is no good way to determine well-defined evaluation criteria for the algorithm. The calibration is thus conducted based on human evaluation. A few snapshots of the real data and the recognized events are shown in figure 3.

From the snapshots in figure 3 we may conclude that the algorithm recognizes events from the data decently, yet there is room for improvement. Events appearing solo over a steady baseline are recognized easily, but there are difficulties in handling noise and multiple simultaneous events.

The hierarchical clustering produces a dendrogram, which is presented in the figure 4. From the dendrogram we can assess, into how many clusters we should group the events. We note that with four clusters the clusters should be rather distant from each other, which we can verify from figure 5. With nine clusters there are still clear clusters, but from the figure 6 we see that the distance comprises mainly of the distance in the duration dimension.

The clustering figures 5 and 6 show that the household has appliances with different mean power consumption levels. Two main groups in the power level are clearly visible. The event on the higher power level seem to have a somewhat constant duration compared to that of the even on the lower level.

6 Discussion

The algorithm proposed in this study produces usable results in disaggregating electrical appliance data. The lack of ground truth prevents any use of popular disaggregation methods, although it also causes the validation of the proposed method to remain rather minuscule.

A big problem with the algorithm is its parameters. Labeling the data by hand with the desired level of accuracy would enable better calibration, which could easily be executed even by brute force since the data amount is so small.

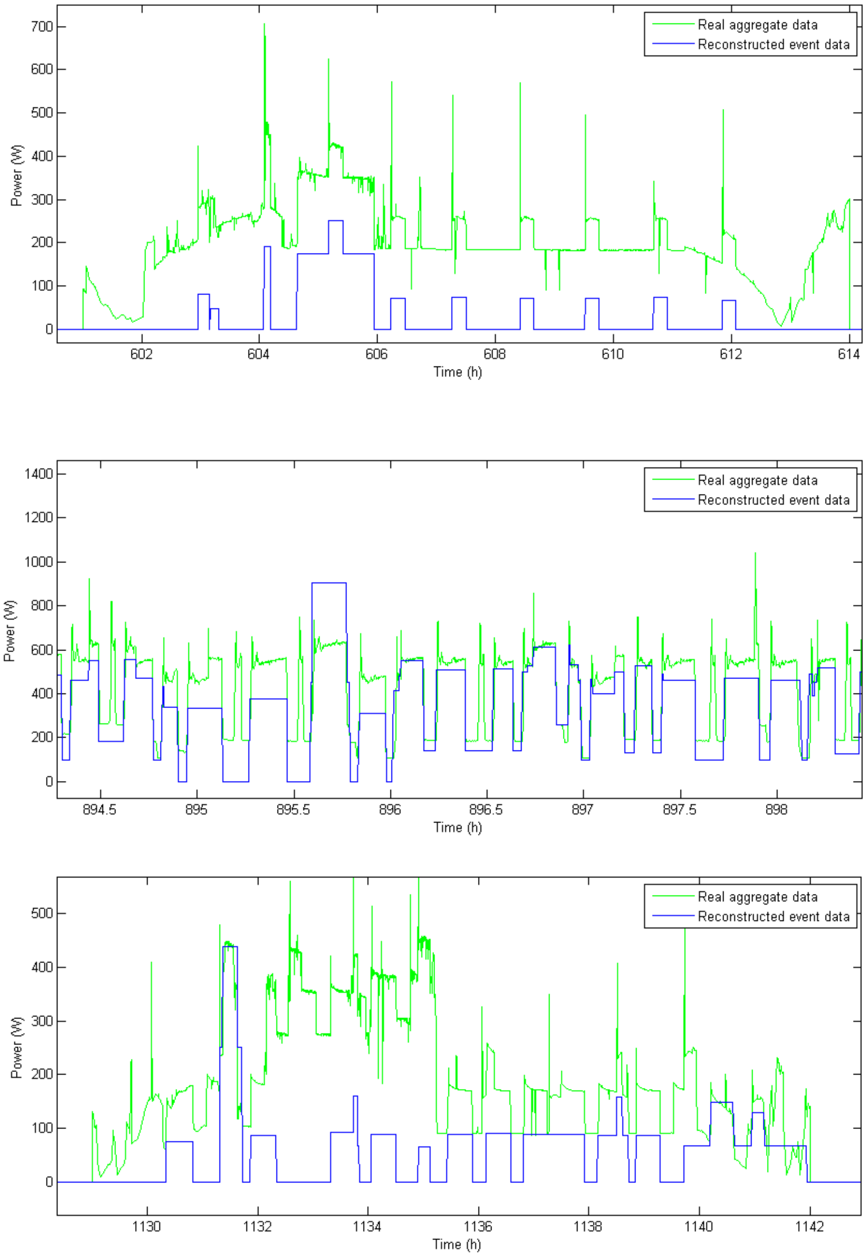

Figure 3: Snapshots showing the real aggregate data and the reconstructed event data that is the aggregate values of the events recognized by the algorithm proposed in this study

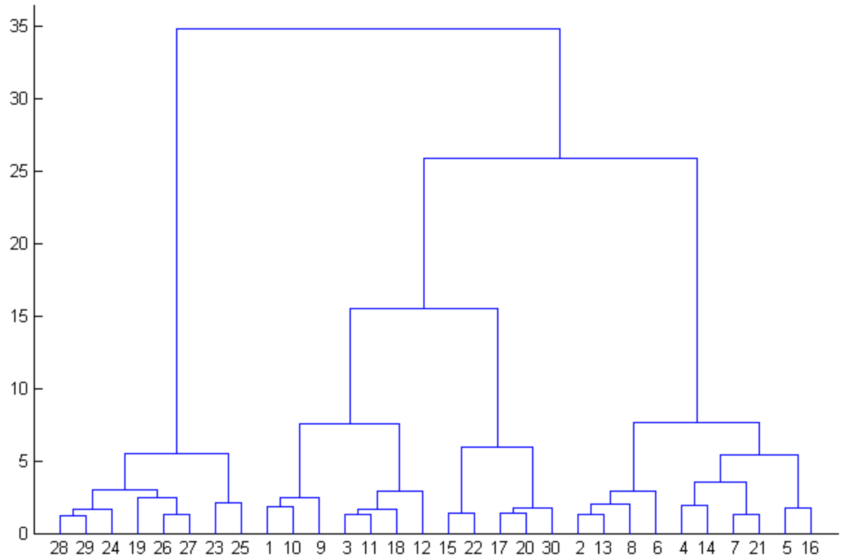

Figure 4: Dendrogram for the hierarchical clustering.

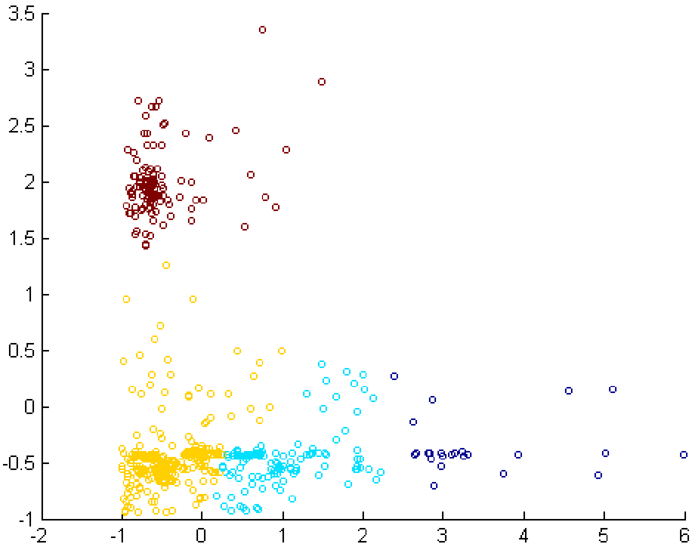

Figure 5: Data grouped into 4 clusters according to the preprocessed variables. The horizontal axis represents the event duration variable and the vertical represents the power level variable.

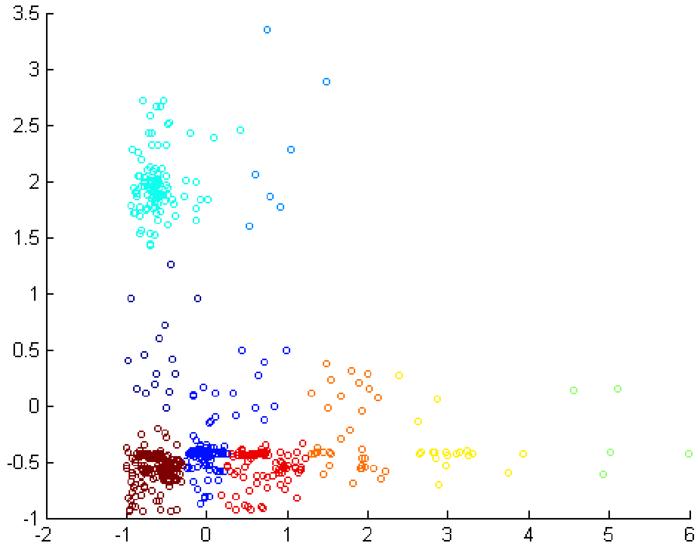

Figure 6: Data grouped into 9 clusters according to the preprocessed variables. The horizontal axis represents the event duration variable and the vertical represents the power level variable.

The calibration would require some metric for determining the accuracy of the recognition process. But even with better calibration there is little guarantee on the robustness of the algorithm over different applications.

The algorithm lacks the ability to detect events that start (or end) simultaneously but end (or start) at a different time. Such events are not rare in the data used in this study. One could split up an on or off transient, when two or more nearby transients would sum up to the complement of its value. It would, however, require a more robust method for detecting the transients. With the current method we have to allow a wide margin in deviation between the on and off transients, which would become a problem when splitting the transients and matching them with smaller ones.

The proposed event detection algorithm does not take into account the time dimension as such, but rather just handles data through indices. This approach makes it harder to handle data with floating frequency. In this case the data frequency rises along with the power level, which is why high power levels are not that well identified by the algorithm. Of course the data used in this study could practically be transformed into having a constant frequency.

The frequency problem does not apply only to the algorithm but also for the data collection method, which seems to create some noise around the signal on high power levels. On a level of 3600 W a smart meter blinks once a second, so the blink of light is somewhat too slow a sign to observe. Thus some kind of smoothing should probably be conducted on the data. This could be done simply by the moving average method, for example.

The figures 5 and 6 clearly show an entire group of outliers in the data. In practice the main interest in energy conservation is in the appliances that consume most energy, but given the goal of this study, it would be worth examining the data after ruling out the high power events. The same result could be obtained, of course, by determining a metric that would emphasize the differences in lower power levels. Also, the weight on differences in the event durations could be assessed better. It is not even that important a factor in recognizing an appliance, since you may well use the same appliance (e.g. TV) for 5 minutes or 5 hours.

The data used in this study contains output power of a solar panel, which is hard to filter out, especially because of the power data getting inverted when the solar power exceeds consumed power. Because one of the motivations of this study is the possibility to optimize energy consumption in regard to solar panel output, separating the signal would be useful. Furthermore even forecasting the solar power in the system with the help of weather forecast data could be possible. Gupta et al. [5] have a patent claim on applications related to solar energy disaggregation.

References

[1] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges. BLUED: a fully labeled public dataset for Event-Based Non-Intrusive load monitoring research. In *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, Beijing, China, August 2012.

[2] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37 (6):1554–1563, 1966.

[3] Linda Farinaccio and Radu Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245–259, 1999.

[4] Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273, 1997.

[5] A. Gupta, Y. He, V. Garud, H.T. Cheng, and R. Mohan. Applications of non-intrusive load monitoring and solar energy disaggregation, July 24 2014. URL https://www.google.com/patents/US20140207298. US Patent App. 14/159,058.

[6] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[7] Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. 2(150007), 2015. doi: 10.1038/sdata.2015.7.

[8] Jack Kelly and William Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pages 55–64. ACM, 2015.

[9] Willett Kempton and Laura Montgomery. Folk quantification of energy. *Energy*, 7(10):817–827, 1982.

[10] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han. Unsupervised disaggregation of low frequency power measurements. In *SDM*, volume 11, pages 747–758. SIAM, 2011.

[11] J. Z. Kolter and T. S. Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *AISTATS*, volume 22, pages 1472–1482, 2012.

[12] 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.

[13] S. B. Leeb, S. R. Shaw, and J. L. Kirtley. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Transactions on Power Delivery*, 10(3):1200–1210, Jul 1995. ISSN 0885-8977. doi: 10.1109/61.400897.

[14] B. Mettler-Meibom and B. Wichmann. *Informationsstand und Einstellung als Verhaltensregulative*, pages 73–88. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982.

[15] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[16] T.D. Stickels and L.C. Markel. *Residential load control and metering equipment: costs and capabilities.* Electric Power Research Institute, 1987.

[17] C. C. Yang, C. S. Soh, and V. V. Yap. A systematic approach to on-off event detection and clustering analysis of non-intrusive appliance load monitoring. *Frontiers in Energy*, 9(2):231–237, 2015.