# A generic data driven approach for low sampling load disaggregation

Kaustav Basu, Vincent Debusschere, Seddik Bacha, Ahmad Hably, Danny van Delft, Geert Jan Dirven

# A generic data driven approach for low sampling load disaggregation

Kaustav Basu[a,b], Vincent Debusschere[a], Seddik Bacha[a], Ahmad Hably[a], Danny Van Delft[b], Geert Jan Dirven[b]

[a] *Univ. Grenoble Alpes, F-38000 Grenoble, France*
[b] *Greeniant B.V, The Hague, Netherlands*

—

*Corresponding author: kaustav07@gmail.com*

**Abstract**

Non-intrusive load monitoring (Nilm) deals with the disaggregation of individual appliances from the total reading at the power meter. This work proposes an industrial scale solution which uses a specific modeling technique for appliance detection, trained and tested on two distinct databases extracted from actual customers readings. The proposed method is tested for different household categories to address its robustness. The validation of the implemented solution is done over a period of one month with a sampling rate of 10 seconds. The results indicate that high energy consuming appliance can be correctly detected (>80 % of accuracy). In addition, general cases of errors are analyzed, paving the way of the next step in the development of a commercial application of the proposed method.

*Keywords:* Non-intrusive load monitoring, load disaggregation, data mining, smart meters, energy efficiency, smart grids.

## 1. Context

### *1.1. Smart meters in the context of smart grids*

Smart meters are getting deployed worldwide on a large scale. This large scale generation of digital energy data mandates a deeper look inside the consumption patterns of different appliances present inside buildings.

In the context of smart grids, insights on the energy consumption patterns through appliances usages could help system operators to better manage the

energy distribution [1, 2], especially regarding the integration of more fluctuating energy sources (i.e. renewables). Strategies commonly employed to that end are usually known as *demand response* [3, 4]. Through these strategies, it is possible to reduce peaks demands by eliminating consumption, or by shifting it to non-peak times. For example, the *time of use pricing* has been successfully employed to this end [2].

From the point of view of energy consumers, load identification can play an important role in the prediction of future usages of particular appliances where the process of historical data collection is made as less intrusive as possible [5]. The current technology of smart meters allow reporting the total electrical consumption of a building. But to further develop energy management strategies, appliances usages insights need to be provided. A forth part system will very likely be used for that purpose, in addition to the end user, the energy provider and the energy distributor.

For that reason, an increasing number of innovative industrial solutions are being under development. The appliances in residential or tertiary buildings could be directly monitored but both the associated costs and inconveniences to the users represent a significant limitation to a correct gross on the market for such implementations.

In that context, this paper focuses on the implementation of a generic solution, based on real-life data acquired in commercial contracts by the start-up *Greeniant B.V.* with which algorithms derived from laboratory developments have been run. The objective is to determine the level of accuracy that can be reached within strict limits of industrial feasibility, privacy concerns and costs.

### *1.2. Data analytics and data usage*

Non-intrusive methods propose an attractive alternative to in-house load monitoring with reduced costs and manual overheads. To achieve this goal, new data analysis mechanisms have to be proposed to inhabitants for their satisfaction and possible energy costs reduction. Note that just a transfer from an analog to a digital system is not good enough for the customers. Indeed, a comprehensive and qualitative data analysis mechanism has to be coupled with the subsequent load management strategies to present enough interest.

Next to the necessary efforts on the data-acquisition side, it is equally important to develop insights in consumers' needs. Typically, an additional context, through analysis on the end-users, would benefit significantly to the

performances of such systems. Customers could be residential consumers, small-business owners, farmers etc. Mapping a daily/weekly routine in a so-called "customer journey map" gives insights in jobs-to-do for these end-users and the pros and cons of using a local energy management system.

### *1.3. From research to industrial development*

As nowadays energy supports most of the indoor activities in buildings, all these jobs-to-do have a relationship to and/or an impact on the overall energy consumption of the buildings. As an example, a customer survey was conducted by *Greeniant B.V.* to question users on their experiences with their equipment. When asked if they would like to know their energy consumption per time used, usually the answer of customers was negative. The same answer came regarding the associated costs as a function of the time of use. However, a typical behavior of the energy management system that would be welcomed as interesting without being too intrusive is to send warnings at particular times, depending on pre-defined thresholds for example. We can easily relate these warnings regarding the energy consumption to similar signals already proposed in a regular household for the daily operation of typical appliances, for example to clean some machine, or to replace some filter, etc.

The results of these surveys show that an adequate accuracy and confidence to support an early-warning system is the key to insure that this system is being widely adopted. A thorough analysis could surface ample applications of given technologies in a unique, tailored made proposition for energy utilities and customers.

In that context, this paper proposes a solution mainly validated on residential buildings but the tools and methodology are fully applicable to any other kind of building (tertiary, etc.). It is to be noted that this work has been conducted as a collaboration between an innovative start-up proposing a solution for energy management in residential households and a research laboratory working, among other topics, on the grid integration of smart buildings as non-conventional loads.

## 2. Problem statement

This work focuses on the power meter readings. The added capacity to automatically monitor and report these measurements make a classical power meter a smart meter. In that context, non-intrusive load monitoring (NILM)

deals with the disaggregation of individual appliances, directly from the total load monitored at the power meter.

If a load curve, considered as a time series $L$ and monitored at a power meter is the sum of three loads whose consumption is respectively the time series $L_1$, $L_2$ and $L_3$, then the task is to determine the state of $L_1$, $L_2$ and $L_3$ individually with the only knowledge of $L$, as summarized in Figure 1. Of course, a way to improve the load identification is to add to the only knowledge of $L$ other useful information, like the weather, historical data, non-technical data, etc.

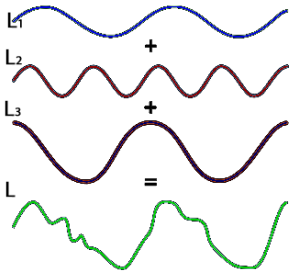



Figure 1: The principle of signal separation, leading to load identification.

The application of that principle in a household will decompose its overall energy consumption into its significant components, as depicted in Figure 2.

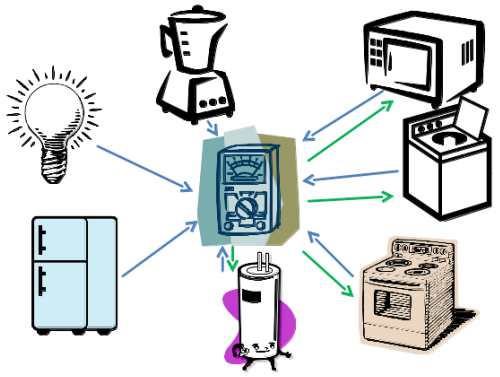


Figure 2: Identification and management of appliances through the power meter.

The only interesting ones are loads consuming a significant part of the overall energy consumption of the building. Indeed, such loads need to be seen either from the grid point of view as a potential flexibility lever, or from

the user point of view, as a potential appliance that could participate in the reduction of their energy consumption, and then directly their energy bill.

From the industrial point of view, it is important to identify periods during which the appliances were used. A typical example of support for identification is proposed in Figure 3, directly extracted from the readings of a smart meter, for one full day. The shown data is then just monitored and not processed in the identifier yet. This is a time-line representation and the appliances are emphasized by colors. The blue line represents the total consumption. On that curve, two loads are identified: the clothes dryer and the dish washer.

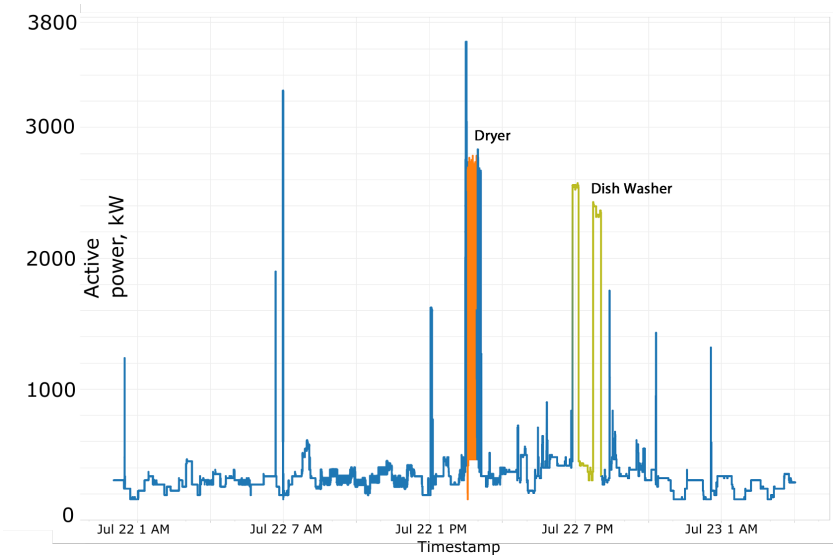



Figure 3: Illustration of periods of usage of appliance (without data-mining involved).

### *2.1. Load identification*

The pioneering work in load disaggregation was published in the early 90's [6]. It proposed to identify appliances by their respective On/Off transitions. It tried to consider power consumption changes (i.e. events) both in the active and reactive power of the signal. An illustration of such On/Off transitions is illustrated in Figure 4.

Methods were then proposed to identify individual appliances from their On/Off transitions. From that time, most of the approaches were event-based and considered a high sampling rate, typically less than one second.

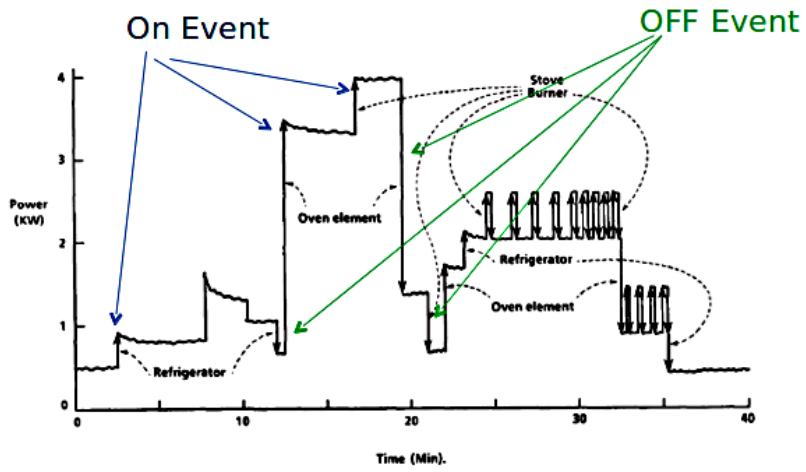



Figure 4: Non-Intrusive Load Monitoring principle [6] and On/Off transitions.

Even though Nilm is considered one field, there are two fundamentally different categories. The main difference between them is in the granularity of the monitored energy data, which drives the use of specific tools and analysis techniques. The two major categories of granularity are defined based on the frequency of monitoring.

- *High sampling rate*: freq.≥50 Hz, up to several kHz (i.e. time ≤ 0.02 s)
- *Low sampling rate*: freq.≤1 Hz (i.e. time ≥ 1 s)

*2.1.1. High sampling rate* Nilm

High-frequency measurements allow the analysis of not just the energy used, but also of the structure of the current-voltage waveform itself [7]. Running different appliances introduces different signatures of disturbances to the pure sine waveform of the theoretical voltage, which can be used to disaggregate them. Getting this high-frequency data requires specialized monitoring equipment such as currents clamps and a lot of computational and communication power to process them.

In that field, [8] proposed a method to generate signatures based on the ranges of the active power, the reactive power and the harmonic content. This kind of approaches typically consists of identifying the steady state (or in some cases the transient state) features of the appliances, and thus to be able to recognized them rapidly enough [9, 10]. Subsequently, these signatures are

matched with earlier learned models using a pattern recognition algorithm [11, 12].

The drawbacks of these approaches are mainly hardware requirement due to high sampling rates and the impracticality of the process being totally non-intrusive [13, 14] and thus widely accepted by customers. Also, these methods do not fit well into smart meters capabilities (which are country dependent). An alternative is a separate device, which has to be installed for training, visualization and communication to the grid. These are major drawbacks, commercially and practically speaking. The load identification at a high sampling rate of all appliances also raise privacy concerns as users activities can be easily detected, interpreted and monitored [15].

*2.1.2. Low sampling rate* NILM

Low-frequency measurements obviously complicate the disaggregation process, but have one major advantage: they are easily available from smart meters from all over the world. Typically, these power meters supply whole-home electricity usage every one second, ten seconds or more. Since all the information of the waveform is lost, and typically only active power is available (no reactive power or harmonic content), different techniques are required for disaggregation.

The major issue at low sampling rates is that low energy consuming devices are difficult to be detected as the switching events are not very prominent. However, high energy consuming appliances, such as a water heater or a washing machine can still be identified with a reasonable precision even at a sampling rate of 15 minutes for example [16, 17]. Another illustration can be found in [18, 19], where the states of high energy consuming appliances, especially the water heater, were identified correctly for consumption readings monitored at a time step of 10 minutes in France.

Considering the constraint of the low sampling rate, the differentiation of the methods is directly dependent on the choice of algorithms. Indeed, algorithms have further been implemented and tested in the field of load monitoring from one to ten seconds. For example, a method generating appliances features using Eigen Vectors and a pattern recognition technique was proposed to match the appliances energy signatures during the testing time [20]. Another method partially disaggregating the total household electricity usage into five loads categories has been proposed at a lower sampling rate in [21] where different sparse coding algorithms are compared and a *discriminative disaggregation sparse coding* algorithm is tested. A feature-

based *support vector machine* classifier accuracy is also mentioned but not presented. The method of [21] is an implementation of the blind source separation problem, which aims at disaggregating mixture of sources into its individual sources. A classic example for this would be the problem of identifying individual speakers in a room having multiple microphones placed in different locations.

In the NILM context, the problem is undermined as there is only one mixture and a large number of sources. The sources (i.e. appliances or loads) do have separate usage patterns which could be used but the issue using blind source separation is the assumption of no prior information about them. Nevertheless, blind source separation still remains a promising direction of research in this field. Temporal graphical models such as *hidden Markov models* also have been promisingly used in this field as they are a classical method for sequence learning [22]. The problem is to learn the model parameters given the set of observations as input sequence and appliances states as output sequence. But at a very low sampling rate, they seem to have a sensitivity to monitoring noise during the training stage.

Finally, [23] proposed a technique using subtractive clustering and the maximum likelihood classifier. The features used to identify appliances are the power levels and the ON/OFF duration. The power levels are computed using a normal distribution and the ON/OFF duration using a Weibull distribution. For six identified appliances, the average accuracy reaches 86 %.

*2.1.3. The implemented method*

In this work, the technical proposed in [23] is enhanced to take into account the various power levels (or "states") at which appliances operate and also the temporal correlation among them. It is tested for a commercially viable solution considering all the practical challenges encountered during the implementation of that solution. The main objective is to identify as accurately as possible electrical appliances usages from the smart meter monitoring.

The "technical" goal is then to be able to identify individual loads from the aggregate power consumption in a non-intrusive manner. The novelty of the work lies in using smart meter data (active power at a 10 seconds sampling rate) already used in a commercial application in the Netherlands and in proposing a novel set of appliance state features, which are validated on real-life data. Note that, even if it was a hard work, the practical implementation of the solution on a large scale commercial base is intentionally not presented

here, because the information and telecommunication problems have no room in the context of this paper.

### *2.2. Database architecture*

The initial database is provided by *Greeniant B.V.*, which is a smart meter analytics company based in the Netherlands. This database consists of more than three thousand residential customers with the corresponding power meter readings hosted in a cloud based server. The training of the proposed model is built on a subset of the *Greeniant B.V.* database of approximately 20 actual houses, with plugs for various appliances in addition to the power meter of the complete house. For the purpose of validating the proposed technology, four houses were specifically selected and extracted from the initial database. Additional plugs for the appliances power readings were installed in these houses, as they will represent the testing database in our case.

The database used for validation corresponds then to different houses than the one used for testing. They have been chosen based on their various consumption levels, number of inhabitants and surfaces. To sum up, the model is learned once for all houses from the testing database, and is then validated on the four houses of the validation database. Both databases have time series of one year. This is an ambitious validation method for such data mining technique, compared to the literature.

In Figure 5, the average hourly consumption metrics (mean and standard deviation) are shown for the full month of simulation.

We can see in that figure that house number two has the maximum hourly mean and standard deviation, whereas the other houses are clustered in the same area. This observation led us to statistically analyze approximately three thousand households from the database available at *Greeniant B.V.*. The conclusion is that a majority of the houses ($> 80\,\%$) presents a similar profile to houses one, three and four whereas the rest of the houses are similar to the house number two. Therefore, we decided to keep three houses in the first category (houses one, three and four) and only one in the second (house two).

Finally, it is to be emphasized that the testing database is chosen totally independently from the training database to observe the performances of the algorithms in an unknown environment (both databases are also geographically different).

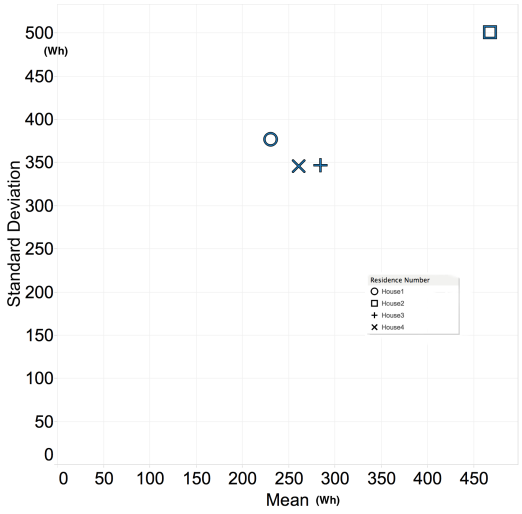



Figure 5: The energy consumption profile of the residences used for validation.

## 3. Methodology

The methodology implemented in this paper can be described as a part-based data driven approach. The power meter readings are time series from which daily overlapping sub-sequences are generated through the use of a time sliding window. Before going further, some definitions are in order.

**Time Series:** Ordered set of $n$ real-valued variables $T = T_1, \ldots, T_n$ associated with a continuous ordered time sequence of size $n$ and defined by its time step $t$.

**Sub-sequence:** For a given time series $T$ of length $n$, a sub-sequence $C_k$ of $T$ is a sampling of length $w \leq n$ of contiguous position from $T$, that is, $C_k = T_k, \ldots, T_{k+(w-1)}$ for $1 \leq k \leq n - (w + 1)$.

**Sliding Time Window:** Given a time series $T$ of length $n$ and a sub-sequence of length $w$, a matrix $M$ of all possible sub-sequences can be built by "sliding a time window" across $T$ and placing sub-sequence $C_k$ in the $k^{th}$ row of $M$. The size of the matrix $M$ is $\{(n-w+1)/J\} \times \{w\}$, with $J$ the "jump" variable that represents the shifting step of the sliding time window.

**Event:** Detection of a variation of power (in one way or the other) greater than a threshold value in the considered time sliding window.

**Element:** A pair of mapped events (a positive one and a negative one).

**Cluster:** A partition of $n$ observations into $K \leq n$ sets that minimizes a distance between elements within clusters.

**Electrical component:** A part of any appliance, responsible of a part of its behavior, monitored through variations in the power meter readings (for example heating elements or motor spin for the washing machine). An illustration of identified electrical components for a washing machine is proposed in Figure 6.

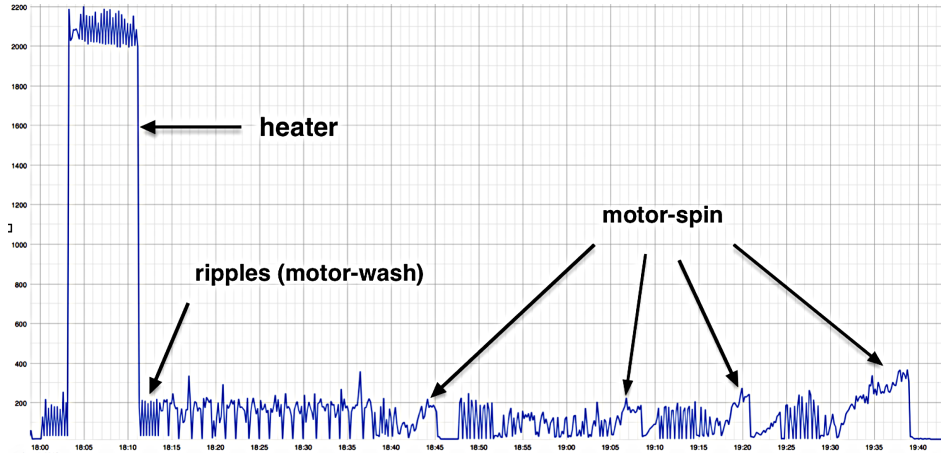



Figure 6: The electrical components as a relation to states of an appliance (here a washing machine).

The NILM method can be broadly segmented in a view phases, each helping on various aspects of assessing data [15]. These phases can be converted in equivalent stages implemented in the data mining associated with the monitoring recorded at the smart meter level. In this work the data flow follows these stages, visualized in Figure 7.

1. Database pre-processing;
2. Elements generation from events detection and mapping;
3. Clustering of elements based on their power levels;
4. Feature generation based on power and components clustering;
5. Building the model of the classifier (based on a *support vector machine*);
6. Testing of the built model on an unseen database;
7. Post-processing calculation (start-time, duration and energy).

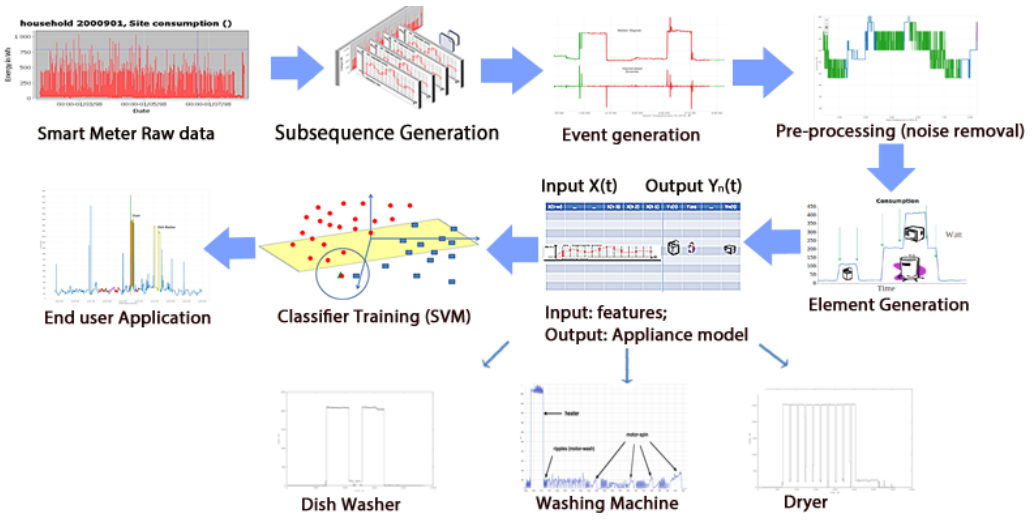



Figure 7: Overall data-processing and analytic pipeline.

### *3.1. Database pre-processing*

As the algorithms are operating on real-life data, measurement errors are present in the database. At the pre-processing stage, these "noises" in the data, such as data spikes and missing data, are processed and automatically removed. It is necessary to remove them as they clearly interfere with elements detection. For illustration, in Figure 8, the spikes occurring in an extract of the database are highlighted. The spikes are removed using a median filter and the missing data suppressed using a predefined threshold. Not to influence results, the appliance identification during the badly monitored period is not considered for training.

### *3.2. Event detection and mapping for elements generation*

During the training phase, an event is registered when the variation in the power level is superior to a threshold manually set for the selected database. As illustrated in Figure 9, a low power fluctuation can be seen in the power meter data, whose fluctuations vary around 32 W. For that reason, in this work, the threshold has been set experimentally to 35 W. This minimum threshold reduces the complexity of events detection and errors due to minor fluctuations.

The power variation can be expressed by the following equation:

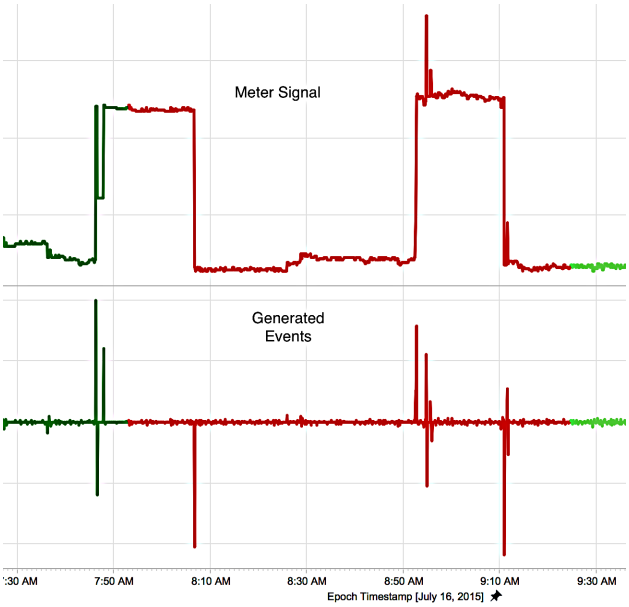



Figure 8: Spikes occurring in the power meter readings and their corresponding errors triggered in the events detection.

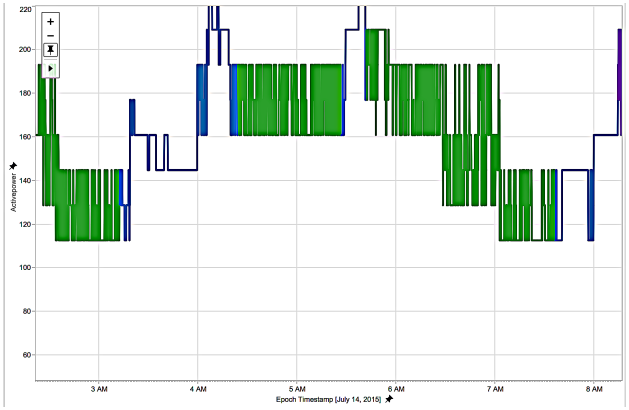



Figure 9: Fluctuations in the monitored power meter data.

$$\left.\begin{array}{l}\Delta(t_i) = P(t_i) - P(t_i - 1) \\ \Delta(t_i) > Threshold\end{array}\right\} \quad t_i \in [0, w] \tag{1}$$

where $w$ is the size of the sliding-window.

The events generated with a ten seconds time step power meter readings is proposed in Figure 10. It can be clearly seen that the generated events correspond to various appliance states, either for the same appliance, or even multiple ones. The dark blue color corresponds to "no appliance identified" and the other colors corresponds to "detected appliances" (one color per appliance).

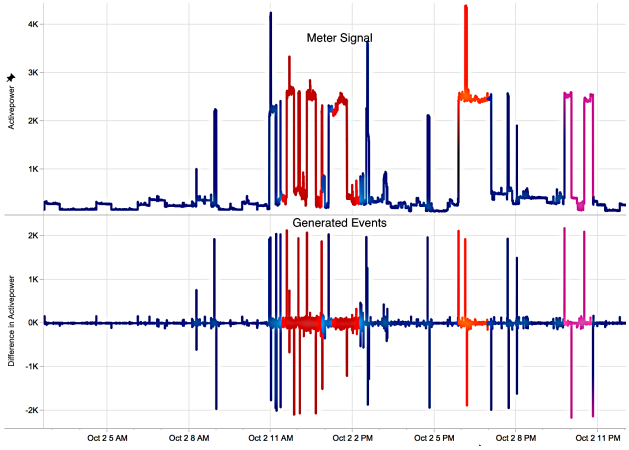



Figure 10: Events detected in the power meter readings. The washing machine is in red, a clothes dryer appears in orange and a dish washer in purple.

Within the time sliding window, a first list of consecutive positive events is sorted then a list of consecutive negative events that come after the last positive one is also sorted. Negative events from the second list are considered consecutively and matched with a positive event from the first list. The matched events are removed from both lists and stored as a new element for the considered time sliding window. The process continues as an iteration process to build a library of generated elements per time sliding window.

### *3.3. Clustering based on power levels*

The *K-means* clustering method is used to partition a set of observations into a set of clusters where each object belongs to the cluster with the nearest

average value. It effectively partitions the data into *Voronoi* cells which is a way of dividing the space of the database into a given number of regions [24]. Considering a set of observations ($x_1$, $x_2$, ..., $x_n$), where each observation is a $d$-dimensional real vector, the *K-means* clustering aims to partition the $n$ observations into $K \leq n$ sets $S = \{S_1, S_2, \ldots, S_K\}$ that minimize distances between elements within clusters. The definition of the distance uses in this paper is expressed as follows:

$$\arg\min_{S} \sum_{i=1}^{K} \sum_{x_j \in S_i} ||x_j - \mu_i||^2 \tag{2}$$

where $\mu_i$ is the average value of the points contained in the set $S_i$.

In the classical methods, the number of partitions $K$ had to be provided but many variants have evolved to overcome this restriction. The *X-means* algorithm extends the *K-means* principle by an improved structuring part. Indeed, the clusters are split in better defined sectors. Also, the input vector is normalized using the *min-max* algorithm.

In this work, the *X-means* algorithm is used to segment the power levels into clusters. This is an important step as physically it broadly corresponds to differentiate the appliances present in the building. An illustration of identified clusters, based on their power level is proposed in Figure 11 for a washing machine.

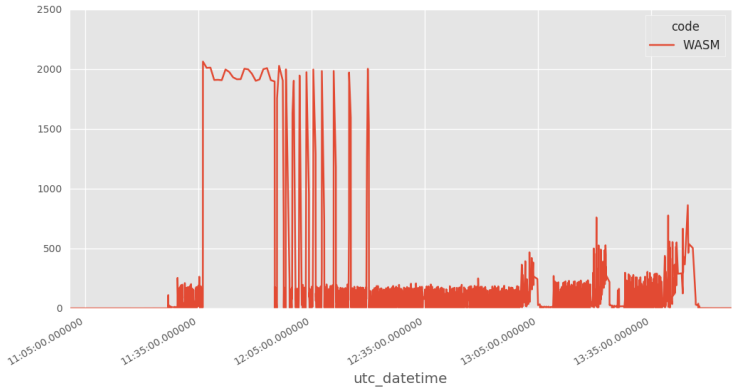
The comparison with Figure 10 shows how the identified clusters roughly correspond to some of the electrical components of the washing machine, like the heater, ripples due to washing cycles and various spins.

Two other illustrations of typical signatures of major appliances in actual houses of our database are proposed in Figure 12 for the clothes dryer and the dish washer

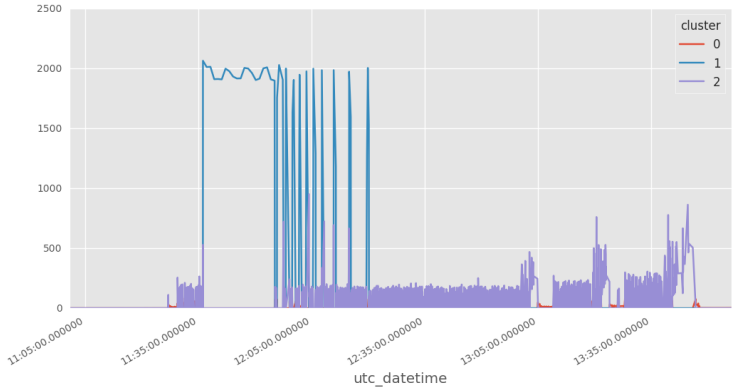
### *3.4. Feature generation*

The previously exposed appliance-wise power levels are used as features during the generation step to establish the characteristics of each cluster. Indeed, the part-based model is built on a specific knowledge of the appliances we are willing to disaggregate, which is defined as features. Such features are related to power levels, their corresponding duration, their number of occurrences within the time-window, etc.

During this stage of the algorithm, the features associated with the elements (i.e. paired events) are computed within clusters. As clusters are

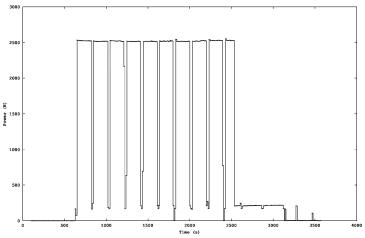



(a) Raw data.

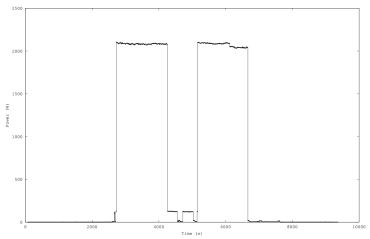



(b) Data with cluster.

Figure 11: Electric signature of a washing machine and clustering of its electrical components.



(a) Clothes Dryer.



(b) Dish Washer.

Figure 12: Electric signature of typical appliances in the houses of the database.

defined based on a power level, features will vary in significance and value from on cluster to the other. The time difference between each part is also computed. Except for "mean and standard deviation of the power level of each cluster", the following features are generated for each element within a cluster and a time sliding window.

1. Mean and standard deviation of the power level of each cluster.
2. Mean and standard deviation of the duration of each element in each cluster.
3. For similar power levels and duration:
   (a) Mean and standard deviation of the On duration of an element (i.e. duration between a positive event and the next negative event) within each cluster in the considered time sliding window.
   (b) Number of occurrences of elements within the time sliding window.
   (c) Average time difference between the starting time of successive generated elements in the clusters (the start time is the first occurrence of the considered element).

For the features based on similar power levels and duration, the calculus is done on elements belonging to the same cluster of a sliding window. First, the mean of the elements (power and duration) is calculated, then only elements within $\pm 5\,\%$ of these average values are considered. This is primarily a filtering operation, because for many appliances, the elements within a cluster exhibit similar characteristics (in power levels and duration).

Assuming all clusters are represented in each sliding window, the number of training instances is then the product between the number of sliding windows and the number of considered clusters.

As previously stated, the power level based clustering roughly corresponds to various states or electrical components of multi-states appliances. It allows taking into consideration the temporal relations between the various states (as a part-based model). As various appliances are used in temporal sequences, this temporal correlation is taken into consideration with the presented part-based model.

### *3.5. Training of the part-based model parameters*

#### *3.5.1. Training instances*

The generated elements correspond to various states of a multi-state appliance. The technique of features generation is part-based on relations

among the elements as a function of the time. Indeed, the part-based model utilizes the temporal correlation between the various elements which can correspond to the time duration between appliances usages.

The models are built per appliance. An appliance is identified to be used in a particular day if it starts before the end of that day (24:00). In addition, four hours of time overlap are added at the end of the considered day for the cross-over cases. Indeed, we have experimentally measured that the average maximum duration of any appliances usages in the available database lies around four hours. That duration has been also set for the time sliding window in the algorithm.

All training instances are generated using the time sliding window (typically with a window jump of 30 minutes). The output (the state of the appliances) is taken to be On if the appliance is 100 % present in the considered time sliding window. It means that, during training, the state of the appliance is On during a time window of the starting and ending times of that appliance are within the time window.

Once the features are generated for each time sliding window, they become training instances for the classifier.

*3.5.2. Classification*

Once the training instances are generated, the task is to learn a classifier function which can best map the input to the output. For that purpose, we used the *support vector machine* algorithm, Svm.

The Svm algorithm is a powerful tool for data classification, well described in [25]. The first major step of a Svm classification is to build a decision plane that separates a set of objects with different class memberships. It guarantees the best function to distinguish between members of classes by maximizing the margin between them. The hyper-planes maximizing these margins allow the best generalization abilities and thus the best classification performances on the training database.

The first step of this procedure requires finding the solution of the following optimization problem:

$$\min_{\mathrm{w},b,\xi} \left( \frac{1}{2} \mathrm{w}^T \mathrm{w} + C \sum_{i=1}^{l} \xi_i \right) \tag{3}$$

$$\text{subject to } \begin{cases} y_i \left( \mathrm{w}^T \phi(x_i) + b \right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \tag{4}$$

With $l$ the total number of sub-sequences, w the normal vector of the hyper-plane, $b$ the offset of the hyper-plane, $C$ the penalty parameter of the error term $\xi$ and $\phi$ the kernel function.

The second major step is to choose the kernel function of the algorithm. The *Radial Basis function* is preferred over others in this work.

For two groups $i$ and $j$, the training vectors $x_i$ and $x_j$ are mapped to a higher dimensional space by the kernel function $\phi$ defined as:

$$\mathrm{K} \rightarrow \begin{cases} \mathrm{K}(x_i, x_j) & \equiv \phi(x_i)^T \phi(x_j) \\ \mathrm{K}(x_i, x_j) & = \exp\left(-\gamma ||x_i - x_j||^2\right); \gamma > 0 \end{cases} \tag{5}$$

Where $\gamma$ is a parameter of the kernel. In our work, a grid-search has been conducted on the parameters $C$ et $\gamma$ using cross-validation.

To conclude on the training part and the parameters definition, the *sequential minimal optimization* [26] implementation of [27] is used in this work with a grid search for parameter optimization during training.

### *3.6. Additional step*

Additional steps are implemented to the disaggregation algorithm in order to make its usage proper for an industrial application. Here are some of features developed during this work.

- A communication layer to fetch data from the smart meter.
- An update tool to read and write the latest meter reading with the correct time stamps and consumption values for any customer.
- An implementation of the time-stamps threshold (configurable and similar to the power level threshold) to better identify similar events occurring in different houses.

- A configurable data creation tool, for too long gap between measurements in power meter readings (missing data).
- A tool to automatically identify an appliance if there is no changes in the power readings during a configurable time period.
- A tool to merge events if power variations are below a configurable threshold.

## 4. Performances of the load identification

In this section, the measures used to evaluate the performance of the developed methodology are first discussed, followed by the observed performances. Finally, the results are analyzed, allowing visualizing and identifying the primary sources of error in the presented algorithm.

### *4.1. Measures*

Given a database of labeled instances, supervised machine learning algorithms seek for hypothesis leading them to correctly predict the class of future unlabeled instances. In order to compare structures of predictors, indicators are needed, that will propose a quantitative way of assessing the classifier performances. While comparing these indicators values, the best predictor can be found for a given appliance.

As the objective is not defined between being the most efficient for just one appliance, or being on average efficient for all appliances, we introduce the *confusion matrix* to compare the algorithms configurations [28].

A confusion matrix contains information about the actual and the predicted results obtained by a classification system. The performances of such systems are commonly evaluated using the data contained in this confusion matrix. Table 1 shows the confusion matrix for a two-class classifier. The classes that can be predicted are "positive" or "negative" instances, which in this case signify that the appliance consumes or does not consume energy.

For this study, the entries defined in the confusion matrix reported in Table 1 have the following meaning:

**a** is the number of correct predictions where an instance is negative,

**b** is the number of incorrect predictions where an instance is positive,

**c** is the number of incorrect of predictions where an instance negative,

Table 1: Confusion matrix definition.

| | | Predicted | |
|---|---|---|---|
| | | Negative | Positive |
| Actual | Negative | a | b |
| | Positive | c | d |

**d** is the number of correct predictions where an instance is positive.

Several standard terms have been defined for this two class's matrix:

**The accuracy** is the proportion of the total number of predictions that were correct. It is determined using the ratio $A_C = (a + d)/(a + b + c + d)$. It is the percentage of cases where the predicted energy state (On or Off) is correct for an appliance.

**The true positive rate (recall)** is the proportion of positive cases that were correctly identified, as calculated using the equation $T_P = d/(c + d)$. $T_P$ represents the ratio between the predicted positives states of the appliances (On) and the total number of correct positives states of the appliances.

**The precision** is the proportion of the predicted positive cases that were correct, as calculated using the equation $P = d/(b + d)$. $P$ represents the fraction of the positives states (On) of the appliances correctly predicted.

We chose to consider the appliance identification to be correct when the start time is around thirty minutes of the actual start time and for the correct appliance.

*4.2. Disaggregation performance*

In Table 2, the house-wise disaggregation performance is shown for the major appliances present in a typical Dutch residence, as depicted in our database. Note that a washing machine is present in all the houses and an electric oven in just one as in most residences it runs on gas.

It can be observed from the results that the algorithm is better performing for the washing machine and the dish washer (>80 %). The performances

Table 2: Appliance classification performance, Accuracy ("N.A." for "non available").

| Appliances | Houses | | | |
|---|---|---|---|---|
| | House 1 | House 2 | House 3 | House 4 |
| Washing machine | 1.00 | 0.80 | 0.88 | 0.87 |
| Clothes Drier | N.A. | 0.40 | 0.88 | N.A. |
| Electric oven | N.A. | 0.50 | N.A. | N.A. |
| Dish Washer | 0.71 | 0.87 | N.A. | 0.80 |

reduce significantly for the clothes dryer and more particularly the electric oven. It should be noted that it is easier to have a 80 % accuracy for a device that is On once a week (just never forecast it On) than for a device that is On 50 % of the time. In that case, the recall is more useful than the accuracy to assess the performance [29].

A global view of the performances of the implemented solution is proposed in Figure 13. In that figure, the appliances power ranges that can be realistically disaggregated based on the household consumption is shown. High power appliances in low energy consuming houses can be expected to yield higher performances. Thus, the ratio between the global consumption of the house and the one from a particular appliance represent a significant criterion for knowing in advance the ease of identification through the developed algorithm. Indeed, a very low consuming appliance in a household having a high consumption level will be very likely to be lost in the measurement noises. The only possibility to identify that particular load will be through its consumption pattern.

As a similar result, it has been observed that the performance is correlated to the number of appliances being present in the house, as the signal of appliances other than the one which is being identified can disturb the identification, even if the algorithm is able to forecast several appliances simultaneously.

Finally, some use-cases scenarios that can be provided based on the appliance power level and the household consumption is shown in Figure 14. It emphasizes the fact that the activity level (i.e. appliances used) and the average electricity consumption play an important role in electricity load

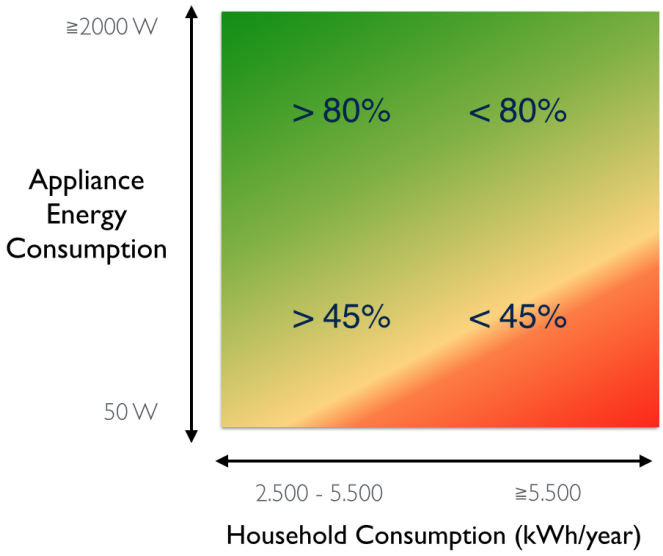



Figure 13: Capability assessment of the disaggregation algorithm.

dis-aggregation capabilities and thus its reliability. This is an important observation, as the current capabilities will be able to operate correctly with a sizable population.

To conclude on the results section, we can say that this method is able to reach a similar accuracy as the one proposed in the literature review, based on a ten seconds time step analysis. The difference here is that the context of implementation of that method is very demanding regarding code development, implementation and industrial robustness. Therefore, the performances of the proposed method are considered to be very good in the context of the collaboration on actual customers data.

### *4.3. Discussion regarding identification errors*

In this section, the major causes of errors are discussed, which include errors in measures and inter-appliances conflicts. One possible reason, but not considered in this work, is the potential presence of a previously unobserved appliance. The algorithm being based on a training set, the accuracy of the results decreases considerably in that case. The recording of households meta-data (appliances present in the houses) could be used to reduce miss-labelling among appliances and thereby increase the performances, at a

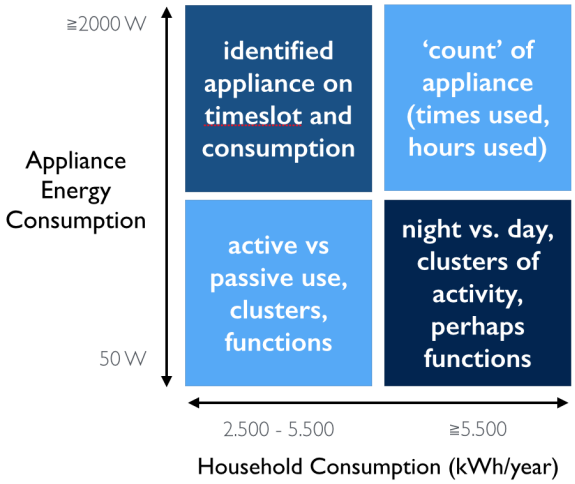



Figure 14: The disaggregation analytics service road-map.

higher installation cost, and also a potential breach in the privacy of inhabitants.

*4.3.1. Power measurements errors*

Due to measuring errors, there are sometimes two steps instead of one for an On/Off event.These two steps cause the algorithm to disregard that event. Most of the false negatives we have identified during testing can be attributed to this phenomenon. A parametric merge may be included in the data pre-processing step to deal with this condition. However, this problem was not often observed in the testing database. This problem is illustrated in Figure 15.

*4.3.2. Invertible appliances conflicts*

Inverting appliances, i.e. appliances having multiple On/Off cycles during the course of an operating cycle, can usually be primarily distinguished based on their power levels. For example, we can name in that category electric ovens, clothes dryers, hair dryers, etc.

As the proposed algorithm considers both the power level and the duration, a better disaggregation capability was expected. For example, the electric oven is in some cases getting miss-labeled as a clothes dryer. This occurs because both the appliances have similar power ranges and are mostly resistive as components.

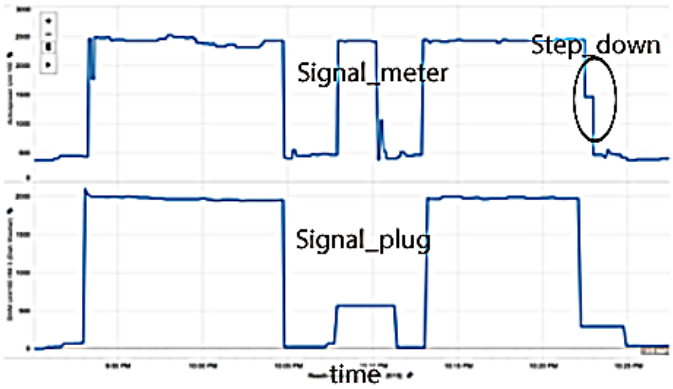



Figure 15: Two steps during an On/Off transition.

It is to be noted that this observation is different from the one seen in [23] where the electric oven power consumption is around 4 kW. A practical explanation comes from the fact that the electric oven may operate at different power levels depending on its usage and size.

Although one may not be able to detect the exact type of the load in such case, it is still quite useful to be able to recognize the load as a member of a known family. Another solution is to add a post-processing module which can identify a clothes dryer only after the use of a washing machine for example.

*4.3.3. Similar appliances conflicts*

The heating blocks of both the washing machine and dish washer have similar signatures. The main difference can only be observed at a low power, for example with inverting On/Off ripples present in the washing cycles. But in some cases, due to noises in the signal measurement, the data contains phases where a washer-like rippling can be observed. In such cases, the dish washer is being identified as two washing machines in the considered time sliding window. This is illustrated in Figure 16 .

This is a significant issue that needs to be addressed either at the pre-processing or at the post-processing stage.

## 5. Conclusion

In this article, a part-based learning model is trained, validated on real data in the context of an industrial application for smart houses and energy efficiency and various sources of practical errors are discussed.

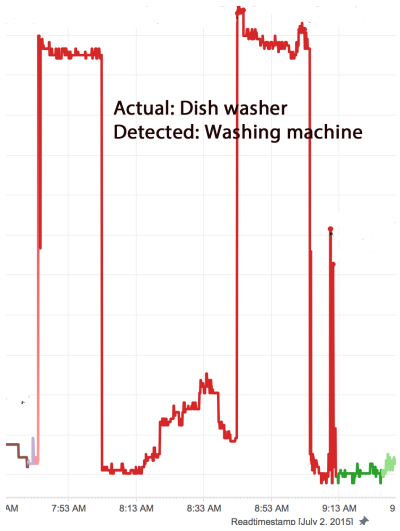



Figure 16: Illustration of an identification conflict. A washing machine was detected where a dish washer is in fact measured.

The proposed method is an industrial application at ten seconds sampling showing the pros and cons of using load disaggregation in residences. The existing literature is based on acquired database but not actual customers, which is a challenge in itself. That challenge has been addressed with a part-based model using power level clustering which is missing in existing literature, to the best of our knowledge.

The identification results highlight the fact that performances of load disaggregation decrease as we move from the laboratory setting to the real residences. This may happen due to errors at various stages: data acquisition, inter-appliance similarity, presence of previously unobserved appliances, etc.

Nevertheless, the results indicate that the proposed NILM algorithm is directly useful for the majority of the houses in the client database of the start-up *Greeniant B.V.* (accuracy $> 80\,\%$) with a sufficient usability to consider it for an industrial application.

Finally, it can be noted that this algorithm has been effectively implemented in a commercial context during the time of publication of this article.

## 6. Acknowledgment

This work has been a joint collaboration between the Grenoble Electrical Engineering laboratory, G2ELAB, in France and the smart meter analytics

enterprise *Greeniant B.V.* operating from the Netherlands.


[1] T. Strasser, F. Andrén, M. Merdan, A. Prostejovsky, Review of trends and challenges in smart grids: An automation point of view, in: Industrial Applications of Holonic and Multi-Agent Systems, Vol. 8062 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 1–12.

[2] P. Palensky, D. Dietrich, Demand side management: Demand response, intelligent energy systems, and smart loads, Industrial Informatics, IEEE Transactions on 7 (3) (2011) 381–388.

[3] D. He, W. Lin, N. Liu, R. Harley, T. Habetler, Incorporating non-intrusive load monitoring into building level demand response, Smart Grid, IEEE Transactions on 4 (4) (2013) 1870–1877. doi:10.1109/TSG.2013.2258180.

[4] P. Siano, Demand response and smart grids - a survey, Renewable and Sustainable Energy Reviews 30 (1) (2014) 461–478. doi:10.1016/j.rser.2013.10.022.

[5] K. Basu, L. Hawarah, N. Arghira, H. Joumaa, S. Ploix, A prediction system for home appliance usage, Energy and Buildings 67 (1) (2013) 668–679.

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

[7] T. Hassan, F. J. N., Arshad, An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring, Smart Grid, IEEE Transactions on 5 (2) (2014) 870–878. doi:10.1109/TSG.2013.2271282.

[8] M. Dong, P. Meira, W. Xu, C. Chung, Non-intrusive signature extraction for major residential loads, Smart Grid, IEEE Transactions on 4 (3) (2013) 1421–1430. doi:10.1109/TSG.2013.2245926.

[9] M. Zeifman, R. Kurt, Nonintrusive appliance load monitoring: Review and outlook, Consumer Electronics, IEEE Transactions on 57 (1) (2011) 76–84.

[10] J. Li, N. Allinson, Building recognition using local oriented features, Industrial Informatics, IEEE Transactions on 9 (3) (2013) 1697–1704. doi:10.1109/TII.2013.2245910.

[11] M. Berges, E. Goldman, H. Matthews, L. Soibelman, Enhancing electricity audits in residential buildings with nonintrusive load monitoring, Journal of Industrial Ecology 14 (5) (2010) 844–858. doi:10.1111/j.1530-9290.2010.00280.x.

[12] T. Bier, D. Benyoucef, D. Abdeslam, J. Merckle, P. Kleinf, Smart meter systems measurements for the verification of the detection & classification algorithms, in: Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE, IEEE, 2013, pp. 5000–5005.

[13] R. Fernandes, I. D. Silva, M. Oleskovicz, Load profile identification interface for consumer online monitoring purposes in smart grids, Industrial Informatics, IEEE Transactions on 9 (2013) 1507–1517.

[14] L. Norford, S. Leeb, Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms, Energy and Buildings 24 (1) (1996) 51–64.

[15] B. Birt, G. Newsham, I. Beausoleil-Morrison, M. Armstrong, N. Saldanha, I. Rowlands, Disaggregating categories of electrical energy end-use from whole-house hourly data, Energy and Buildings 50 (2012) 93–102.

[16] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis, R. Cepeda, Privacy for smart meters: Towards undetectable appliance load signatures, in: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010, pp. 232–237. doi:10.1109/SMARTGRID.2010.5622047.

[17] A. Prudenzi, A neuron nets based procedure for identifying domestic appliances pattern-of-use from energy recordings at meter panel, in: Power Engineering Society Winter Meeting, IEEE, Vol. 2, 2002, pp. 941–946. doi:10.1109/PESW.2002.985144.

[18] K. Basu, V. Debusschere, S. Bacha, U. Maulik, S. Bondyopadhyay, Nonintrusive load monitoring: A temporal multilabel classification ap-

proach, Industrial Informatics, IEEE Transactions on 11 (1) (2015) 262–270. doi:10.1109/TII.2014.2361288.

[19] K. Basu, V. Debusschere, A. Douzal-Chouakria, S. Bacha, Time series distance-based methods for non-intrusive load monitoring in residential buildings, Energy and Buildings 96 (2015) 109–117. doi:http://dx.doi.org/10.1016/j.enbuild.2015.03.021.
URL http://www.sciencedirect.com/science/article/pii/S0378778815002133

[20] H. Ahmadi, J. Marti, Load decomposition at smart meters level using eigenloads approach, Power Systems, IEEE Transactions on 30 (6) (2015) 3425–3436. doi:10.1109/TPWRS.2014.2388193.

[21] J. Kolter, S. Batra, A. Ng, Energy disaggregation via disciminative sparce coding, Advances in Neural Information Processing Systems 23 (7) (2010) 1153–1161.

[22] O. Parson, S. Ghosh, M. Weal, A. Rogers, Using hidden markov models for iterative non-intrusive appliance monitoring, in: Neural Information Processing Systems workshop on Machine Learning for Sustainability, 2011, pp. 1–4.
URL http://eprints.soton.ac.uk/272990/1/MLSUST2011.pdf

[23] N. Henao, K. Agbossou, S. Kelouwani, Y. Dube, M. Fournier, Approach in nonintrusive type i load monitoring using subtractive clustering, Smart Grid, IEEE Transactions on 99 (2015) 1949–3053. doi:10.1109/TSG.2015.2462719.

[24] J. Hartigan, M. Wong, Algorithm as 136: A k-means clustering algorithm, Applied statistics (1979) 100–108.

[25] T. Onoda, G. Rätsch, K. Müller, Applying support vector machines and boosting to a non-intrusive monitoring system for household electric appliances with inverters, in: Second ICSC Symposium on Neural Computation, Berlin, Germany, 2000, pp. 261–267.

[26] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schoelkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998,

pp. 185–208.
URL http://research.microsoft.com/\~jplatt/smo.html

[27] M. L. Project, Weka (Accessed in January 2017).
URL http://www.cs.waikato.ac.nz/ml/index.html

[28] R. Kohavi, F. Provost, Guest editors' introduction: On applied research in machine learning, in: Machine learning, Vol. 30, Springer, 1998, pp. 127–132.

[29] K. Basu, V. Debusschere, S. Bacha, Load identification from power recordings at meter panel in residential households, in: Electrical Machines (ICEM), International Conference on, 2012, pp. 2098–2104.