Developing and Evaluating a Probabilistic Event Detector for Non-Intrusive Load Monitoring

Lucas Pereira
Madeira-ITI / LARSYS
Funchal, Portugal
lucas.pereira@m-iti.org

***Abstract*—In this paper we present and evaluate probabilistic event detection algorithm for Non-Intrusive Load Monitoring. Like the other probabilistic event detectors, this algorithm also calculates the likelihood of a power event happening at each sample of the power signal. However, unlike the previous algorithms that threshold or employ voting schemes on the event likelihood, this algorithm employs a maxima/minima (i.e., the *extrema*) locator algorithm to identify potential power events. The proposed algorithm was evaluated against four public datasets, and its performance was compared to that of other four alternative solutions. The obtained results show that this new algorithm is competitive with the other alternatives in the four datasets. Furthermore, the results also suggest that using an *extrema* locator instead of a voting scheme, increases the performance of one of the state-of-the art algorithms.**

***Index Terms*—NILM, event-based, event detection, algorithm, benchmark**

I. Introduction

Non-Intrusive Load Monitoring (NILM) [1], or more generally single point energy disaggregation, is a promising approach to provide detailed information about the energy consumption of the individual appliances that co-exist in a building's electrical circuit.

To date, NILM research is formally categorized according to two different approaches: i) event-based approaches, which consist of keeping track of every appliance state transition (e.g. TV turning *ON* or *OFF*) by means of event detection and classification, assuming that the system was previously trained [2], [3] and ii) event-less approaches, where no previous knowledge of the existing appliances is assumed and the load disaggregation is performed by means of techniques like Hidden Markov Models [4], [5].

In this paper we focus on the event-based approaches, more particularly in the event detection step. Extensive reviews of the existing approaches for solving the NILM problem using both approaches can be found in [6], [7].

The base assumption of event-based NILM algorithms is that every change in the total power consumption of a building happens as a response to an electric device changing its state, e.g. a hair dryer going from *LOW* to *HIGH*. The approach consists of applying signal processing and machine-learning techniques to measurements in the vicinity of those power changes, hence the crucial importance of the event detection step.

Here we present and thoroughly evaluate a new event detection algorithm. The remaining of this paper is organized as follows: first, we provide a literature review on the state of the art in event detection. Second, we propose a new event detection algorithm, and describe its main characteristics. Third, we describe the evaluation methodology that will be followed to assess and benchmark the performance of the proposed algorithm. Then, we present and discuss the event detection and performance evaluation results, before we concluded and outline future work.

II. Event Detection Algorithms

Event detection is the process of identifying the relevant changes (i.e., that represent appliances changing their state of operation) in the aggregate consumption data. According to the literature in this topic, the different approaches are grouped in three categories [8]: i) expert heuristics; ii) probabilistic models; and iii) matched filters. Next we briefly survey the different proposed approaches.

A. Expert Heuristics

Algorithms under the expert heuristic category are probably the least complex, and follow the basic principle of scanning the time series data looking for changes that are above a certain threshold, as defined by Hart is his seminal work [1].

For example, in [9] the power signal is first filtered to minimize the presence of noise and reduce the chance of false positives. On a second step, the power events are detected by means of computing the absolute differences between two consecutive samples and selecting the indexes where this difference is above a pre-defined threshold. In [10] a similar approach is proposed, yet instead of computing the absolute differences between two consecutive samples, the differences are calculated between the current sample and the sample x seconds before. Moreover, in order to help reduce the number of false positives, an index with absolute value above the pre-defined threshold is only considered a power event if no power event was detected in the last y seconds.

B. Probabilistic Models

Another approach to event detection is by means of probabilistic methods. In this category of detectors, the event detection occurs in two steps, as described below:

978-3-901882-99-9 ©2017 IFIP

In the first step, it is necessary to calculate the chance of an event occurring at each sample of the power signal. This signal is normally referred to as the detection statistic, and is computed by applying either statistical tests (e.g., Generalized Likelihood Ratio (GLR) [11], Goodness-of-Fit (GOF) [12], CUmulative SUM (CUSUM) [13]) or other mathematical functions (e.g., Kernel Fisher Discriminant Analysis (KFDA) [14]), to the power measurements by means of sliding windows.

In the second step the power events are extracted from the resulting detection statistic signal. This is normally done via thresholding, i.e., whenever the detection statistic is above a certain threshold a power event is flagged in the power sample that corresponds to that index [11], [12]. Nevertheless, for the particular case of NILM, more robust strategies have been designed. For instance, in [2] the selection of the power events is done by applying a voting algorithm to the detection statistic signal.

C. Match Filters

In this category of algorithms, power events are detected by correlating a known, or template signal with an unknown signal to detected the presence of the former signal in the later. In other words, match filter event detectors work by trying to find known appliance transients (i.e., templates) in the aggregated consumption signal (i.e., unknown signal) by means of filtering techniques.

To the best of our knowledge, this was first attempted in the NILM domain in [15] and [16] where the authors propose an event detector that attempts to match segments of start-up transients (obtained from training) to the aggregated signal using two transversal filters in sequence. The first filter is used to find the transient shapes in the aggregate signal, and the second filter is used to enforce that the matches correspond to actual transients and not some fortuitous noise [16].

As of today, the match filters category also incorporates those detectors that use filters to transform the power measurements into signals that emphasize potential power events while depreciating the steady stage regions, similarly to what is done in the probabilistic models category. For example, in [17] the authors apply a Hilbert transform to the instantaneous current, sampled at 20 kHz. This is followed by a combination of average and derivation filters on the transformed signal such that only the transitions of interest (i.e., power events) are represented.

Another example of event detection based in filter matching is the work of Baets et al. [18] that apply Cepstrum analysis to the power signal, computed at 60 Hz. The resulting signal is then thresholded such that only the positions where the signal is above a certain value are considered power events.

In this paper we propose the Log Likelihood Ratio Detector with Maxima (LLD-Max) event detection algorithm that as the name suggests falls in the probabilistic category. The LLD-Max algorithm is thoroughly described in the next section.

III. Log Likelihood Detector with Maxima

The event detector that we are proposing is inspired in the Log Likelihood Ratio test, that was already used in [11] and [2] to calculate the likelihood of a power event occurring.

However, unlike these two algorithms, that either threshold ([11]) or apply voting schema ([2]) to the event likelihood signal to identify power events, the LLD-Max employs a maxima / minima (i.e., the *extrema*) locator algorithm on the detection statistics output in order to identify potential power events. In other words, for each *extremum* that is found in the detection statistics, a possible power event is signaled in that position.

Our implementation of this event detector consists of two different algorithms. The first, to which we refer to as *Detection Statistic*, is used to calculate the power event likelihood. The second, referred to as *Detection Activation*, is used to extract the power events from the signal generated by the previous algorithm. Next we describe the two algorithms.

A. Detection Statistics

The detection statistics algorithm works with one sliding window (detection statistics window - dsw) that is used to calculate the likelihood of a change of mean happening at a given sample. The dsw is composed of two separate windows, a pre-event (w_0) and a post-event (w_1) window, and for each sample x in the power metric P the detection statistic $ds[x]$ is given by equation 1.

$$ds(x) = \frac{\mu_1 - \mu_0}{\sigma^2} \times \left| P(x) - \frac{\mu_0 + \mu_1}{2} \right| \tag{1}$$

Where μ_0 and μ_1 are the mean of the pre- and post-event windows respectively, σ^2 is the variance of the detection window, and $P(x)$ is the power of the x^{th} sample.

Lastly, the detection statistics signal, $ds(x)$, is adjusted by forcing it to be equal to zero when the absolute difference between μ_0 and μ_1 is below a pre-defined threshold P_{thr}. Ultimately, the likelihood of a power event occurring at sample x is given by the equation 2.

$$ds(x) = \begin{cases} ds(x), & \text{if } |\mu_1 - \mu_0| > P_{thr} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

B. Detection Activation

In theory, the detection activation of the LLD-Max algorithm would work by sliding an extrema locator window (elw) across the values of the detection statistic $ds(x)$ looking for the local maxima and minima. However, to minimize the computational costs, we slide the elw across the absolute values of $ds(x)$, looking only for the local maxima.

The detection activation algorithm consists of one single parameter, the maxima precision M_{pre}. This parameter is used to make the process of finding the maximum values more stable, and is what defines the size of the extrema locator window (elw). More concretely, the length of the elw is equal to twice the maxima precision plus one ($2 \times M_{pre} + 1$). For each shift of the window, the sample in the middle will be

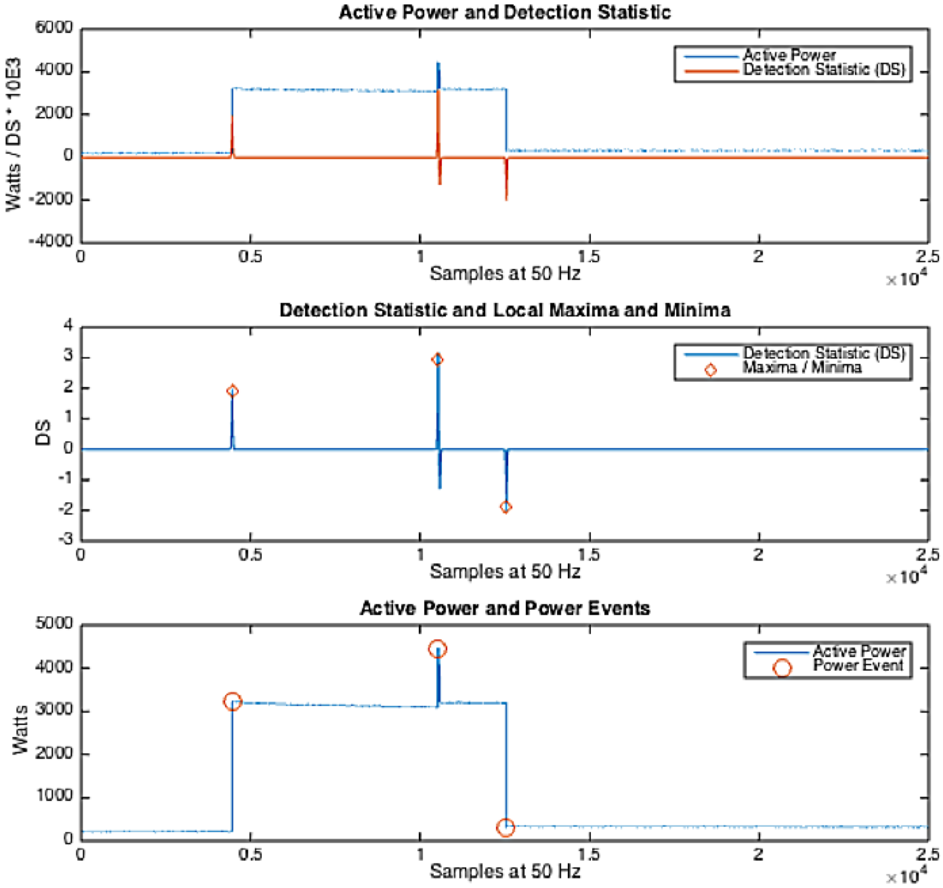


Fig. 1: Illustration of the LLD-Max event detection process. Active power and detection statistics (top), detection statistics and local maxima (center), active power and power events (bottom).

signaled as a power event if its absolute value is larger than the absolute value of all the M_{pre} samples to its left and right. The M_{pre} must be greater of equal to 1.

Overall, the parameter space of this algorithm consists of four adjustable parameters: a pre-event window length (w_0), a post-event window length (w_1), a power threshold (P_{thr}), and the maxima precision M_{pre}. The parameter space of LLD-Max is summarized in table I. It is important to remark that the M_{pre} parameter does not allow the detection of power events when these are separated by less than M_{pre} samples.

In figure 1 we show an illustration of the LLD-Max event detection process. The different parameters were set to the following values: $P_{thr} = 100$ W, $w_{pre} = w_{post} = 50$ samples, and $M_{pre} = 50$ samples.

First, the power detection statistics (ds) is calculated for each power sample. The resulting detection statistics for each sample is depicted on the top of figure 1 (scaled by a factor of 10^3 for better visualization). Next, the *extrema* locator algorithm, with a tolerance of 50 samples, is applied to the detection statistic signal. In this example, two local maxima and one local minimum are found, as it can be observed from the representation in the center of figure 1. Finally, in the third step, the *extrema* indexes in the detection statistic signal are mapped in the power signal, and the respective power events are extracted as shown in the bottom of figure 1.

TABLE I: Parameter space for the LLD-Max detector

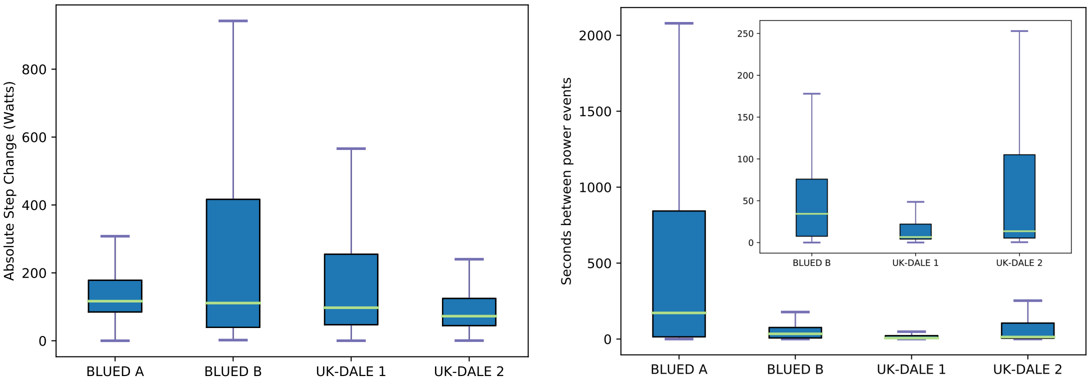
Parameter	Symbol
Pre-event window size	w_0
Post-event window size	w_1
Power Threshold	P_{thr}
Maxima Precision	M_{pre}

IV. Performance Evaluation Methodology

In this section we describe the methodology that will be followed to assess and benchmark the performance of the LLD-Max algorithm. We start by presenting the datasets, the benchmark algorithms and the performance metrics that will be used in this evaluation. We then describe the parameter sweep that was executed in each of the algorithm to gather the event detection data used in this experiment. Finally, we describe the process of calculating the different performance metrics.

A. Datasets

In this work we use data from two public NILM datasets, namely the BLUED [19] and UK-DALE [20]. More con-



(a) (b)

Fig. 2: Summary of datasets: (a): absolute active power change, and (b) seconds between power events.

cretely, use use the Phases A and B from BLUED, and one week of data from UK-DALE's Houses 1 and 2, that are both single-phase.

BLUED was labeled by the dataset creators, and the details can be found in the respective publications. As for the UK-DALE, we had to label the data ourselves. To this end, we followed the semi-automatic labelling approach presented in [21]. To state more concretely, our approach was three-fold: i) we applied an expert heuristic detector to the daily aggregate data (at 50Hz), ii) we manually removed the false positives, and iii) we manually added the missing labels.

Finally, it is important to remark that we are not keeping track of which appliance triggered each power change, and that, for the sake of consistency with BLUED, we only considered the power events with a minimum absolute power change of 30 Watts.

Figure 2 summarizes the four datasets in terms of absolute power change (a), and number of seconds between power events (b). As it can be observed, the minimum absolute power change is the same in each dataset. As for the number of seconds between power events, it is possible to see that in BLUED A, the power events are much more sparse than in the others. Ultimately, these two quantities are very related with the event detection performance, as we will see in the remaining of this paper.

B. Benchmark Algorithms

In this work we will benchmark our algorithm against the heuristic detector presented in Meehan 2014, [10] and the LLR probabilistic detector presented in [2].

TABLE II: Event detection algorithms used in this work

Algorithm	Symbol
Expert Heuristic Detector	EH
Log Likelihood Ratio with Voting	LLR-Vote
Log Likelihood Ratio with Maxima	LLR-Max
Log Likelihood Detector with Maxima	**LLD-Max**
Log Likelihood Detector with Voting	LLD-Vote

Furthermore, to further understand how our algorithm compares with other probabilistic alternatives, we will also evaluate the LLD-Max detection activation algorithm in combination with the voting algorithm of the LLR detector, and vice-versa.

To summarize, in table II we list five event detection algorithms that are used in this work. For additional details please refer to the respective publications.

C. Performance Metrics

Regarding the performance metrics we use the Precision (P), Recall (R) and three variations of the $F_\beta - measure$ ($F_{0.5}$, F_1, and F_2).

In the event detection problem, P reports the fraction of power events that were correctly detected among all the detected events, whereas R reports on the fraction of existing power events that were found by the event detector algorithm.

The $F_\beta - measure$ is a metric that balances P and R, and is reported in terms of the weighted harmonic mean of these two metrics. β is a weighting factor, and is used to attach β times as much importance to R as to P. For example, if $\beta = 2$, R is twice as important as P, whereas if $\beta = 0.5$, P is twice as important as R. Finally, when $\beta = 1$, R and P have the same weight.

D. Parameter Sweep

In order to gain deeper insights on the nature and structure of the data that is produced by the event detection algorithms we first performed a parameter sweep. A parameter sweep refers to a controlled variation of a number of parameters in a particular algorithm (i.e., structural changes) and provides insights into how the different parameters affect the final results.

In this particular case we decided to set the power threshold (P_{thr}) to 30 Watts, since this is the minimum power change for which there are labeled events in any of the four datasets. Also, we used the real power signal, since it is probably the most widely used power measurement in event detection literature. Tables III, IV, and V show the parameters that were changed

TABLE III: Parameter ranges for the EH event detector

Parameter	Min	Max	Increment
G_0	0	5	1 (second)
w_0	1	5	1 (second)
w_1	1	5	1 (second)
T_{elap}	0	5	0.5 (seconds)
E_{edge}	1, 0.5F_s, F_s		
Total Models:		4950	

TABLE IV: Parameter ranges for the LLR-Vote and LLD-Vote event detectors

Parameter	Min	Max	Increment
w_0	0.5	5	0.5 (seconds)
w_1	0.5	5	0.5 (seconds)
w_v	0	5	0.5 (seconds)
v_{thr}	5	*	15 (votes)
Total Models:		9500 (50 Hz), 11000 (60 Hz)	

TABLE V: Parameter ranges for the LLR-Max and LLD-Max event detectors

Parameter	Min	Max	Increment
w_0	0.5	5	0.5 (seconds)
w_1	0.5	5	0.5 (seconds)
M_{pre}	0.5	5	0.5 (seconds)
Total Models:		1000	

in each algorithm, the respective values, and the number of produced models. Next we briefly describe each parameter. For additional details, please refer to the respective publications.

In tables III, IV, and V w_0 and w_1 are the pre- and post-event window sizes, respectively.

In the table III, G_0 is the number of samples before the sample under test, T_{elap} is the minimum number of samples before last power event, and E_{edge} is the index of the sample inside the power event window after it is extracted from the data. In table IV, w_v is the length of voting window, and v_{thr} is the minimum number of votes that are necessary to trigger an event.

Finally, it is important to remark that the number of models for the LLR-Vote and LLD-Vote algorithms varies with the sampling rate of the dataset. For example, in a 60 Hz dataset for each half-a-second increment it is always possible to increment twice the voting threshold by fifteen samples, which is not always true in the case of 50 Hz datasets.

E. Metrics Calculation

In this step we compute the performance metrics for each of the models returned by the parameter sweep. To do this, we first count the number true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) for each model. This is done by comparing the events triggered by each model with the true events in the corresponding dataset (i.e., the ground-truth). To accomplish this, we define a tolerance interval in which the detected events must fall in order to be considered correct detections. The detection interval is defined by equation 3 and is based on the ground-truth position (GT), and tolerance (Tol) value that was added to account for eventual ambiguity when defining where an event occurs during the labeling process. [22].

$$\Omega = [GT - Tol, GT + Tol] \tag{3}$$

In previous work on this topic [22] the authors varied this parameter from one to six seconds (in one-second steps) and found that no improvements were observed with more than three seconds. Consequently, we decided to set this parameter to range between zero and three seconds with variable steps, as defined by the set τ in equation 4, where F_s is the sampling rate of the dataset.

$$\tau = \{0, 1, 5, 15, F_s, 1.5 \times F_s, 2 \times F_s, 2.5 \times F_s, 3 \times F_s\} \tag{4}$$

Regarding the process of creating the confusion matrix, we developed an custom algorithm that given a list of detected events and another one with the ground-truth data, works as follows:

For each ground-truth event, if there are detections that fall within the interval Ω given by equation 3, the event that is closer to the ground-truth position (in absolute distance) or the one that was detected first (in the case of equidistant detections) is considered a TP, whereas the others must be compared with the next ground-truth event. Otherwise, if no events are detected within the specified interval, a FN is added. Next, the detected events that do not fall within any of the possible intervals Ω (one per each ground-truth event) are considered FP. Lastly, when all the detected and ground-truth events have been processed, the TN are calculated by subtracting the TP, FN and FP from the number of samples in the dataset, i.e., all the positions where an event could have happened.

V. Results and Discussion

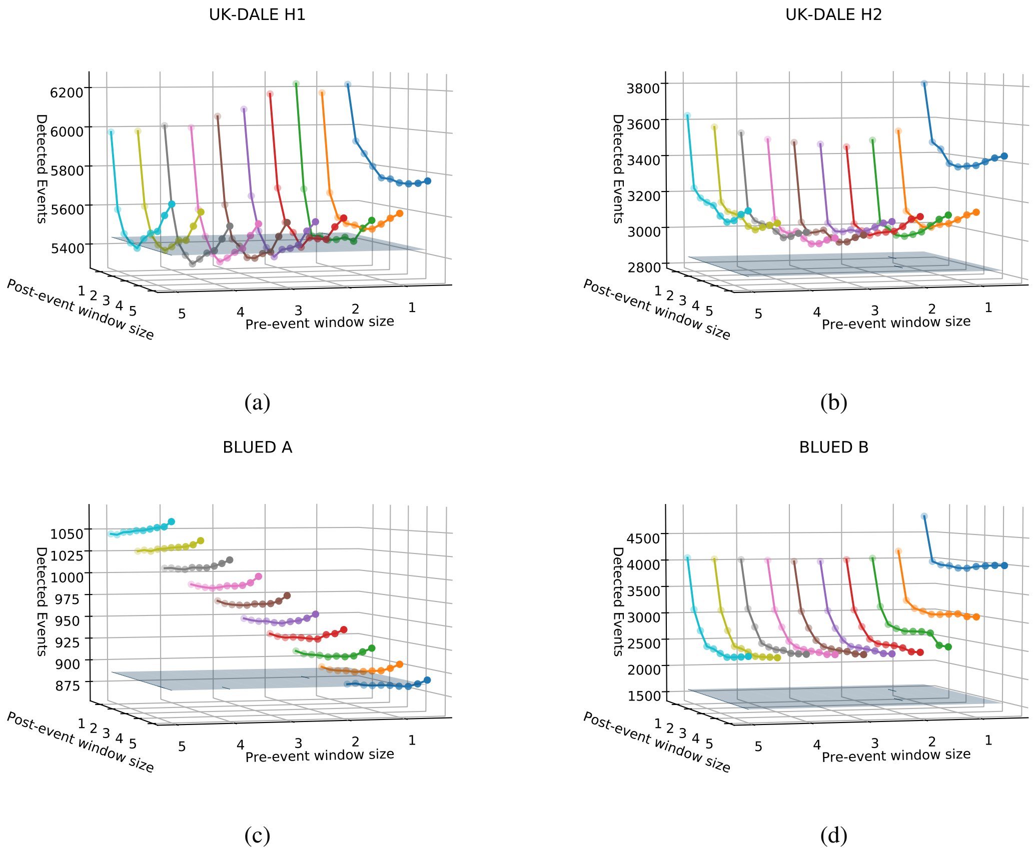
In this section we present and discuss the different results of this experiment. In sub-section V-A we analyze how the different parameters affect the detection results. In sub-section V-B we show how the detection tolerance parameter affects the overall performance of the algorithm. In sub-section V-C we give results for how the different detectors perform with respect to the selected performance metrics on the four datasets. Finally, in sub-section V-D we give results for how our event detector compares with the other four alternatives across the four datasets.

A. Detector Parameters

As we saw above, the LLD-Max detector has four tunable parameters. Here, we discuss three of them and their effects on the event detection process.

The first parameters we consider are the pre- and post-event window lengths, w_0 and w_1, respectively. These determine the number of samples to be used to calculate the mean and variance of the power signal before and after the sample for which we want to calculate the detection statistics.

Figure 3 shows, for each combination tested on the four datasets, the mean number of events returned by all detectors holding these parameters constant, while allowing the others



(a)

(b)

(c)

(d)

Fig. 3: Mean number of events returned by all detectors when changing only the pre- and post-event window lengths

to vary. In this figure, we see that in BLUED A dataset (figure 3c) the behavior of the detectors is slightly different from that of the other three datasets.

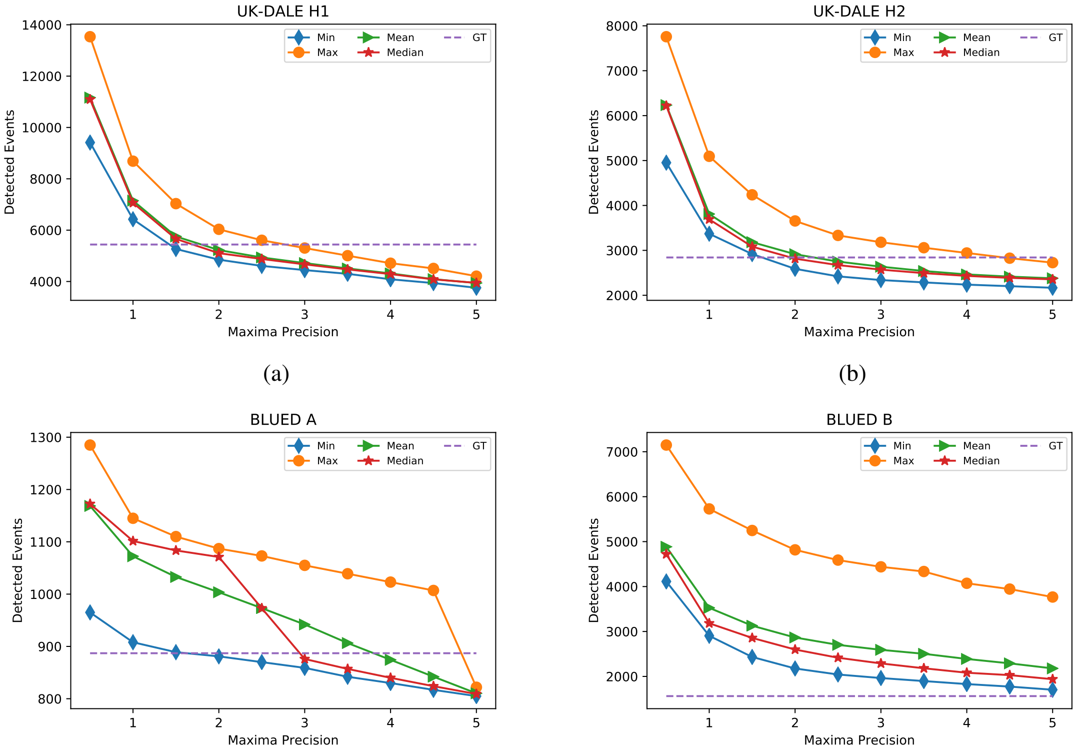
In UK-DALE 1, UK-DALE 2 and BLUED B, small values for w_0 and w_1 clearly have more detected events. This happens because any small variation in the data present in the pre- and post-event windows, will change the mean and variance, thus producing a change in the detection statistics. On the contrary, as the size of these windows increase, the mean and variance become less sensitive to noise in the data and consequent stabilization of the detection statistics. Regarding BLUED A, the reason for the consistent number of detected events for small values of w_0 and w_1, is the fact that this is a very stable dataset due to the low number of appliances. Consequently, there will be very little variation in the mean and variance of smaller windows.

These results also highlight that if w_0 and w_1 increase too much, the number of detection will tend to increase as well. The reason for this is that as the size of the two windows increase, it is possible that the power changes from previous or future events are represented in the mean and variance, which may result in an increase or decrease of the detection statistics. This effect is particularly evident in UK-DALE 1 and UK-DALE 2, that has it can be seen from table **??**, have a large number of events separated by less than 11 seconds (i.e., w_0 and w_1 set to 5 seconds). Here again, BLUED A shows a consistent number of detection, this time because of the high dispersion of the power events (e.g., 75% of the power events are separated by at least 18 seconds).

Finally, it is also noteworthy the fact that in UK-DALE 1, in some situations the mean number of detected events is lower than the number of real power events. The main reason for this is the fact that 10% of the events in this dataset have an absolute power change of about 30 Watts, meaning that in some situations it is possible that the detection statistics will be set to zero in equation 2.

Next we consider the length of the maxima precision window, $Mpre$. This is the number of samples (to the left and right) that must be lower than the actual detection statistic value, such that it can be considered a power event. Figure 4 shows for each of the four datasets, the number of detected events returned by all detectors when holding this parameter constant, while allowing the others to change.

As expected, when this parameter increases, the number of detection decreases. This is particularly evident in UK-DALE 1 and UK-DALE 2, where it is possible to observe that with a precision of 2 or more seconds the average number of detected events is lower than the real number of events. On the contrary, a small precision value will trigger an extremely high number of power events, since there is a high chance that detection



(a)

(b)

(c)

(d)

Fig. 4: Mean number of events returned by all detectors when changing only the maxima precision

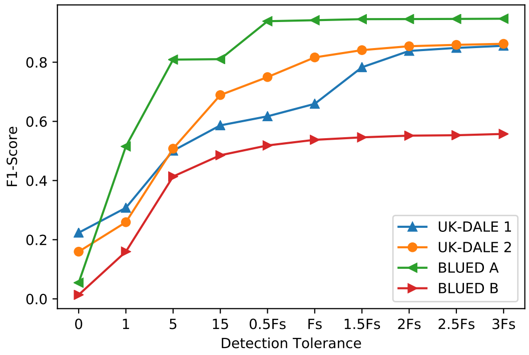


Fig. 5: Mean F_1-Score, for each dataset, under the different detection tolerance values.

statistic values above zero will be considered *extrema* values.

Also noteworthy, is the fact that in BLUED B the number of detected events is higher than the number of real power events independently of the detection tolerance. Ultimately, this means that this dataset is very prone to false positives when using this particular algorithm.

B. Detection Tolerance

The detection tolerance refers to the number of samples that we allow a detected event to deviate from the ground truth location of the event and still be counted as a correct detection (true positive). For example, if the detection tolerance is n samples, then an event will be considered a correct detection if it is reported within n samples prior to or following the actual location of the ground truth event. This means that there are effectively $2 \times n + 1$ positions in which a detection may occur for that event to be considered as correctly detected.

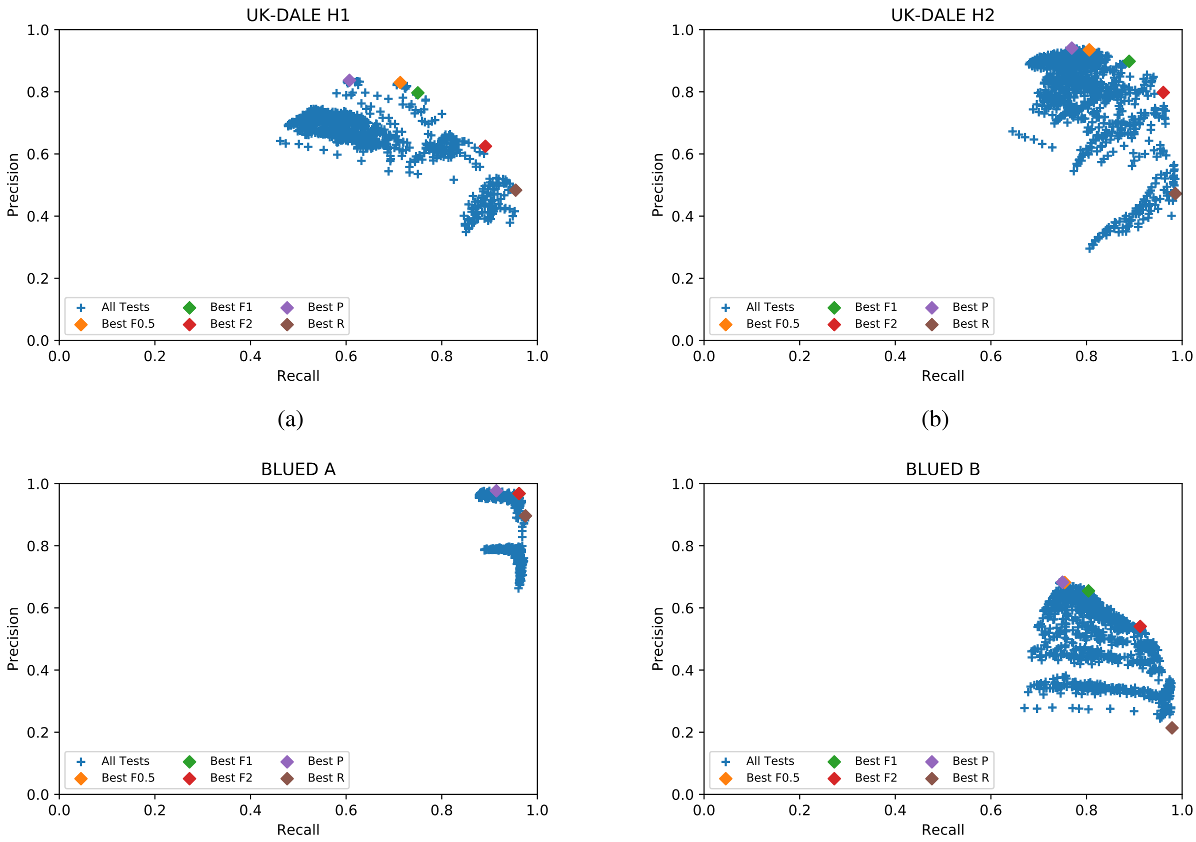
Figure 5 shows the mean F_1-Score, for each dataset, under the different detection tolerances. We see from the chart that after a tolerance value of F_s, (i.e., 1 second), the detection results are not significantly affected by the tolerance value. Again here, BLUED A shows a different behavior, with the F_1-score stabilizing just after a tolerance value of 5 samples. As for the smaller tolerance values, there is an evident increase in the performance with the increase in the tolerance, in particular from 1 to 5 samples.

C. Event Detection

In this section, we give results for how the different detectors perform with respect to the selected performance metrics on the four datasets.

Figure 6 shows the obtained results in each datasets based on P and R, as well as the best models according to $F_{0.5}$, F_1, and F_2 scores. Figure 7 shows the distribution of all the results in each dataset according to the F_1-score.

The first general observation is that the performance of the event detection varies considerably with the data, with BLUED



(a) (b)

(c) (d)

Fig. 6: Obtained results in each dataset based on P and R, and the best models according to the $F_{0.5}$, F_1, and F_2 scores

A clearly outperforming the remaining datasets. This variation is particularly evident in figure 7. A one-way ANOVA statistical test for comparing the Top-100 models (based on the F_1-Score) across the four datasets shows that for $\alpha < 0.05$ there are statistically significant differences between the datasets ($F = 4049.92$, $p = 0.0$). A follow up *ad-hoc* Tukey test shows that these differences are significant between all the four datasets. The same is true for the other four performance metrics, suggesting that the event detection results are heavily dependent of the dataset, as it was previously mentioned by the authors of [23].

Another observation is that, even within the different datasets there is a considerable variation in the obtained results. The exception here is BLUED A, that shows more consistent results, with high P and R values. However, at this stage we are starting to believe that performance evaluation with BLUED A tend to produce over-optimistic performance results.

Regarding the best models for each dataset, R shows that it is possible to find models that are able to detect most of the power events. Yet, the the highest R values come at the cost of very low P values (i.e, a high number of FP). These are known as liberal models, since the tendency is to trigger power events even when the confidence is low. P on the other hand, shows that models with low numbers of FP come at an expense of also low number of TP. These are known as conservative models, since they only trigger power events when the confidence is high.

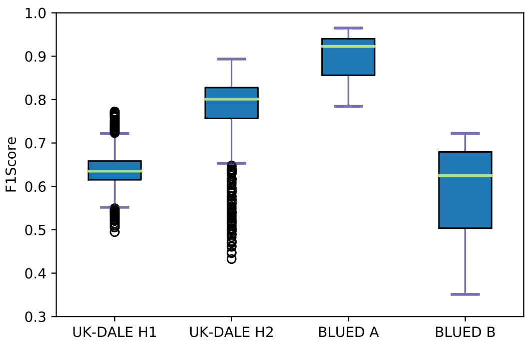
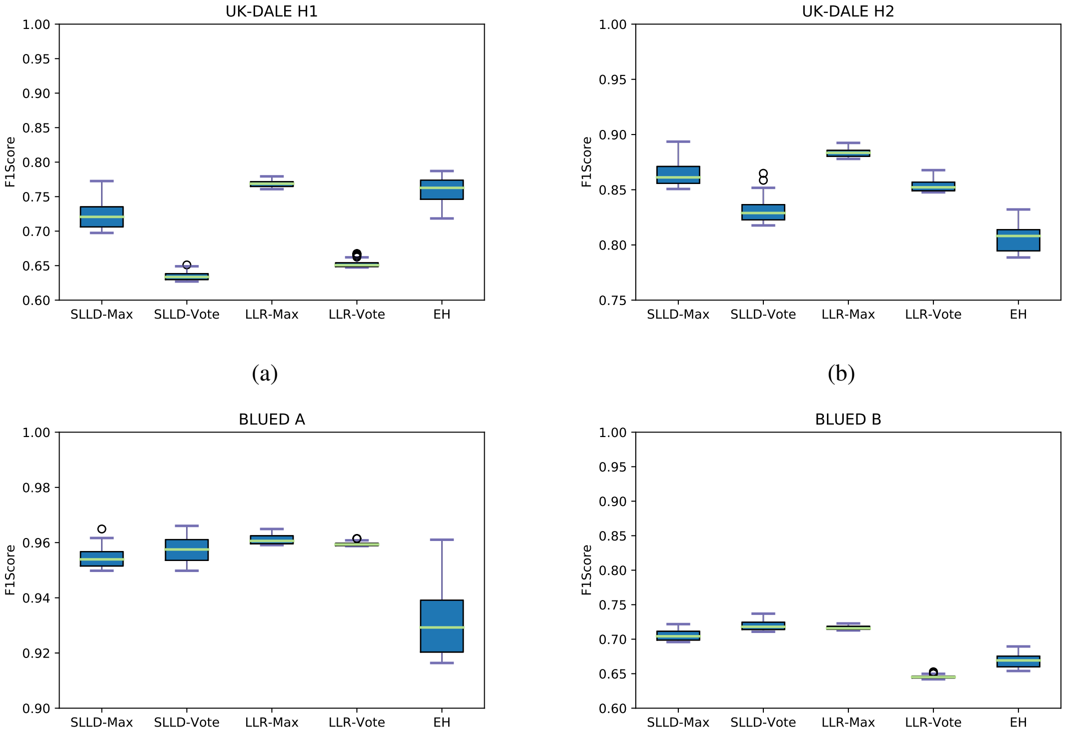


Fig. 7: Distribution of all the results in each dataset according to the F_1-score

With concern to the F_β-score, it is possible to observe that $F_{0.5}$ selects the models with the highest P for the best R. I.e., favours first a low number of FP, while attempting to keep the



(a)

(b)

(c)

(d)

Fig. 8: Distribution of the Top-100 results for each algorithms across the four datasets, according to the F_1-Score

number of FN low as well. On the other hand, F_2, selects the model with the highest R for the best P. I.e., favours first a low number of FN, while attempting to keep the number of FP also low.

Finally, it is possible to observe that the models selected by F_1, tend to be closer the models selected by P and $F_{0.5}$. However, it is not necessarily true that the selected models have P higher than R. The reason for this lies withing the mathematical formulation of this metric, in particular the fact that the harmonic mean tends strongly to the smallest value. In other words, the value of F_1 is highly influenced by the value of the lowest "parent" metric, be it P or R.

D. Benchmark

In this section, we give results for how our proposed event detectors compares to other alternatives. Figure 8 shows the distribution of the Top 100 detection results for each of the five detection algorithms across the four datasets, according to the F_1-Score. Please note that the Y-axis was individually scaled for better visualization.

To understand if your algorithm is significantly better than the others, we run the one-way ANOVA statistical tests with the five algorithms across the four datasets. This did not show any statistical significant differences with either of the five performance metrics. However, from the histograms in figure 8 it is possible to observe that our algorithm is very competitive when compared with the original versions of LLR and the EH.

A second observation is that combining the *extrema* algorithm with the LLR detection statistics algorithm (i.e., LLR-Max) provides better results than the actual LLD-Max algorithm. The reason for this is related with the fact the LLR uses the $\ln$ function in the detection statistics equation, to increase its value when there are considerable differences between the standard deviations of the pre- and post-event windows, hence improving the *extrema* look-up results.

On the opposite direction, combining the Voting algorithm with the LLD detection statistics (i.e., LLD-Vote) decreases the performance. This we believe, suggests that the voting algorithm is not very competitive, since it is very sensitive to the number of votes threshold that needs to be set in advance. I.e., a small voting threshold may results in a high number TP, but at a cost of a high number of FP. On the other hand, a high threshold tends to reduce the number detections, and hence the FP and possibly the TP.

On a final note, we should also mention the very good results obtained with BLUED A, for each of the five algorithms ($\tilde{F_1} > 0.9$). Furthermore, if we consider only the best result (excluding the outliers) we can see that ($\tilde{F_1} > 0.96$). This we believe is another indicator of the fact that the phase A of BLUED tends to produce very optimistic evaluations.

VI. Conclusion and Future Work

In this paper we presented and thoroughly evaluated the LLD-Max event detection algorithm.

In the first part of the evaluation, we showed how the different parameter combinations will affect the algorithm output. The, we compared the obtained performance across four datasets and observed that the performance evidences some considerable variations across them, independently of the performance metric being used.

Then, in order to compare our algorithm with other alternatives, we produced a benchmark between the LLD-Max and four other detection algorithms. Our results suggest that LLD-Max is very competitive in all four datasets, despite the fact that it was not possible to find statistically significant differences between the proposed algorithms.

Perhaps, even more noteworthy, we have learned that using our method to extract the power events from the detection statistics signal improves the results of the LLR algorithm used in the benchmark. Ultimately, this new combination, (LLR-Max), outperformed the LLD-Max in the four datasets, which implies the importance of a "robust" detection statistics equations.

Regarding future work, we believe that it would be important to continue improving our detection activation algorithm. For example, our results show that if M_{pre} is set to more than two seconds it starts to miss too many power events. On the other hand, if it set to less than 2 seconds it might trigger to many false detection. Against this background, future work should look at ways to avoid relying so much on how this parameter is set when looking for the *(*extrema) values in the detection statistics signal.

Finally, despite the fact that we managed to benchmark our algorithm with four alternative approaches, we believe that in order to truly assess the competitiveness of the LLD-Max (and other algorithms), we should also introduce event detectors from the match filters category, and possibly additional datasets.

Acknowledgements

This research was supported by the FCT doctoral grant SFRH/DB/77856/2011 and project UID/EEA/50009/2013.

References

[1] G. Hart, “Prototype Nonintrusive Appliance Load Monitor,” MIT Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report, Tech. Rep., Sep. 1985.

[2] M. Berges, E. Goldman, H. Matthews, L. Soibelman, and K. Anderson, “User-Centered Nonintrusive Electricity Load Monitoring for Residential Buildings,” *Journal of Computing in Civil Engineering*, vol. 25, no. 6, pp. 471–480, 2011.

[3] J. Alcalá, J. Ureña, Á. Hernández, and D. Gualda, “Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor,” *IEEE Transactions on Instrumentation and Measurement*, vol. PP, no. 99, pp. 1–12, 2017.

[4] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “Non-intrusive load monitoring using prior models of general appliance types,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, Jul. 2012, pp. 356–362. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4809

[5] S. Makonin, F. Popowich, I. V. Bajić, 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.

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

[7] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, “Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey,” *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, Dec. 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/12/16838

[8] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. F. Moura, “Event detection for Non Intrusive load monitoring,” in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct. 2012, pp. 3312–3317.

[9] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, “Leveraging smart meter data to recognize home appliances,” in *2012 IEEE International Conference on Pervasive Computing and Communications*, Mar. 2012, pp. 190–197.

[10] P. Meehan, C. McArdle, and S. Daniels, “An Efficient, Scalable Time-Frequency Method for Tracking Energy Usage of Domestic Appliances Using a Two-Step Classification Algorithm,” *Energies*, vol. 7, no. 11, pp. 7041–7066, Oct. 2014. [Online]. Available: http://www.mdpi.com/1996-1073/7/11/7041

[11] D. Luo, L. K. Norford, S. B. Leeb, and S. R. Shaw, “Monitoring HVAC Equipment Electrical Loads from a Centralized Location Methods and Field Test Results,” *ASHRAE Transactions*, vol. 108, pp. 841 – 857, 2002.

[12] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman, “Robust adaptive event detection in non-intrusive load monitoring for energy aware smart facilities,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4340–4343.

[13] K. T. Nguyen, E. Dekneuvel, B. Nicoll, O. Zammit, C. N. Van, and G. Jacquemod, “Event Detection and Disaggregation Algorithms for NIALM System,” Jun. 2014.

[14] B. Wild, K. S. Barsim, and B. Yang, “A new unsupervised event detector for non-intrusive load monitoring,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 73–77.

[15] 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*, vol. 10, no. 3, pp. 1200–1210, Jul. 1995.

[16] L. K. Norford and S. B. Leeb, “Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms,” *Energy and Buildings*, vol. 24, no. 1, pp. 51–64, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378778895009582

[17] J. M. Alcalá, J. Ureña, and Á. Hernándes, “Event-based detector for non-intrusive load monitoring based on the Hilbert Transform,” in *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Sep. 2014, pp. 1–4.

[18] L. De Baets, J. Ruyssinck, D. Deschrijver, and T. Dhaene, “Event detection in NILM using Cepstrum smoothing,” in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016, pp. 1–4. [Online]. Available: http://hdl.handle.net/1854/LU-7242713

[19] 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 *2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, Beijing, China, Aug. 2012. [Online]. Available: http://www.marioberges.com/SustKDD12/

[20] J. Kelly and W. Knottenbelt, “The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes,” *Scientific Data*, vol. 2, p. 150007, Mar. 2015. [Online]. Available: http://www.nature.com/articles/sdata20157

[21] L. Pereira and N. J. Nunes, “Semi-automatic labeling for public non-intrusive load monitoring datasets,” in *Proceedings of the fourth IFIP Conference on Sustainable Internet and ICT for Sustainability*. IEEE / IFIP, Apr. 2015, pp. 1–4.

[22] K. Anderson, “Non-Intrusive Load Monitoring: Disaggregation of Energy by Unsupervised Power Consumption Clustering,” PhD, Carnegie Mellon University, Pittsburgh, PA, USA, Dec. [Online]. Available: http://repository.cmu.edu/dissertations/507

[23] N. Czarnek, K. Morton, L. Collins, R. Newell, and K. Bradbury, “Performance comparison framework for energy disaggregation systems,” in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2015, pp. 446–452.