On the Bayesian optimization and robustness of event detection methods in NILM

Leen De Baets, Joeri Ruyssinck, Chris Develder, Tom Dhaene and Dirk Deschrijver

Department of Information Technology, Ghent University - imec
Technologiepark-Zwijnaarde 15, 9052 Ghent
leen.debaets@intec.ugent.be

Abstract

A basic but crucial step to increase efficiency and save energy in residential settings, is to have an accurate view of energy consumption. To monitor residential energy consumption cost-effectively, i.e., without relying on per-device monitoring equipment, non-intrusive load monitoring (NILM) provides an elegant solution. The aim of NILM is to disaggregate the total power consumption (as measured, e.g., by smart meters at the grid connection point of the household) into individual devices' power consumption, using machine learning techniques. An essential building block of NILM is event detection: detecting when appliances are switched on or off. Current state-of-the-art methods face two open issues. First, they are typically not robust to differences in base load power consumption and secondly, they require extensive parameter optimization. In this paper, both problems are addressed. First two novel and robust algorithms are proposed: a modified version of the chi-squared goodness-of-fit (χ^2 GOF) test and an event detection method based on cepstrum smoothing. Then, a workflow using surrogate-based optimization (SBO) to efficiently tune these methods is introduced. Benchmarking on the BLUED dataset shows that both suggested algorithms outperform the standard χ^2 GOF test for traces with a higher base load and that they can be optimized efficiently using SBO.

Keywords: NILM, event detection, χ^2 GOF method, robustness, cepstrum, Surrogate-Based Optimization

1. Introduction

In October 2014, EU leaders agreed upon three key targets for the year 2030 [1]: 1) at least 40% cuts in greenhouse gas emissions, 2) at least 27% share for renewable energy, and 3) at least 27% improvement in energy efficiency. Energy monitoring proves an useful aid for reaching these targets by providing an accurate, detailed view of energy consumption. It helps because: 1) if this information is given to households, studies have shown that they could save up to 12% of electrical energy and thereby reduce the emissions [2] (also useful for non-residential buildings [3]), 2) this information allows us to assess and exploit flexibility of power consumption, which in turn is important for demand response systems that are responsible for an increased penetration of distributed renewable energy sources, 3) energy monitoring is one major prerequisite for energy efficiency measures [4].

In order to achieve the required energy monitoring cost-effectively, i.e., without relying on per-device monitoring equipment, non-intrusive load monitoring (NILM) provides an elegant solution. NILM concerns the analysis of the aggregate power consumption of electric loads in order to identify the existence and consumption profile of each individual appliance. In 1992, Hart was the first to describe an event-based workflow for NILM [5]. It starts with the detection of state transitions (events) and then matches these with unique signatures of appliances using clustering and classification.

This paper focuses on the development of event detection methods. An important property of these methods is their robustness towards differences in the base load (i.e., the background consumption of devices already consuming power when the event to be detected occurs): it is unwanted that the performance decays if high power consuming devices are on. To the best of our knowledge, this property has not yet been thoroughly investigated. This paper shows that the state-of-the-art event detection method, chi-squared goodness-of-fit (χ^2 GOF), lacks this robustness. Two alternative methods robust to changes in the base load are proposed. The first method is an adapted version of the standard χ^2

GOF method [6], which is extended with a voting mechanism [7]. The second robust method is a new method that uses smoothed frequency components to detect events.

A second contribution of our paper concerns parameter optimization of event detection methods. A standard but slow approach is a brute-force exhaustive search that tries out all the possible parameter configurations and selects the best one. In this paper, this process is optimized by introducing Surrogate-Based Optimization (SBO) [8].

The remainder of the paper is structured as follows: in Section 2 a brief overview of related work is introduced, in Section 3 the preprocessing of the input is described. Section 4 describes the state-of-the art NILM event detection method, discusses its lack of robustness towards base load power differences and proposes a robust adaptation, the voting χ^2 GOF method. Section 5 discusses the newly proposed robust event detection method. In Section 6, a SBO algorithm is proposed to identify optimal model parameter configurations for the statistical tests and in Section 7, the performance of the newly presented methods is benchmarked. A conclusion is given in Section 8.

2. Related work

Event detection methods. In 1992, together with the first event-based work-flow for NILM, Hart described an event detection method that relied on monitoring changes in active and reactive power [5]. A better method to detect events in the active power is the Generalized Likelihood Ratio Test (GLRT) [7, 9], which tests if two neighboring windows representing consecutive time frames share a common distribution. The possible presence of an event in two neighboring windows is determined by calculating a decision statistic from the natural log of a ratio of probability density functions in those neighbouring windows. Another method is the χ^2 GOF test. It detects events by assuming, like the GLRT, that two neighbouring windows share a common distribution. A χ^2 test statistic is applied on two neighbouring windows and an event is assumed if the null hypothesis is rejected [6]. This χ^2 GOF test is widely used [6, 9, 10, 11] and reasons to adopt this method are its simplicity and improved performance as reported in other studies [6, 10, 11, 12]. In addition to these heuristics, more computationally expensive machine learning algorithms are available. Hidden Markov Models (HMMs) [15] formulate the problem of detecting events as finding an ideal set of non-overlapping intervals in which the observations are as heterogeneous as possible. These unknown intervals are the hidden states of the HMM. A disadvantage is that the number of states needs be given or must be predicted. Support vector machines (SVMs) [16] fit models on short segments of the signal, all learned simultaneously using a coupling term that forces neighbouring models to be similar. Bayesian methods [17] work by estimating the run length at every data point. The run length represents the time since the last event. The run length can be inferred given 1) an underlying predictive model whose parameters change when an event occurs, and 2) a hazard function which describes how likely an event is, given an observed run length. This hazard function needs to be given, but the model can be created if it is assumed that the data in each segment is independent and identically distributed (i.i.d.) with respect to some distribution. It must be noted that for HMMs and SVMs, the event detection is a side effect of the approach, and not a separate module in the algorithm itself.

This paper focuses on the χ^2 GOF method as event detection method as it is widely used, simple and has good performance [6, 9, 10, 11, 12].

Benchmark dataset. The most commonly used benchmark for event detection is the BLUED dataset consisting of the aggregated active power signal from a family residence in the United States for a whole week [21]. In this dataset the steady-state power consumption never exceeds 500W for phase A and 1500W for phase B. However, it is likely that much higher power values occur in households, e.g., when electrical heaters (easily consuming 1500W) are used. This paper will show that the performance of the χ^2 GOF method decreases rapidly if a base load is added to the power consumption, while the performance of the proposed voting χ^2 GOF and cepstrum method remains the same under similar conditions.

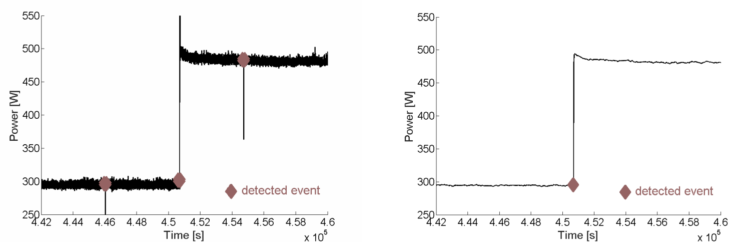
Parameter optimisation. The χ^2 GOF method, the proposed voting χ^2 GOF method and the cepstrum method are parametric and require an optimization step to tune the parameters in order to minimize misdetection rates. This tuning can be done in a supervised way provided that enough data is available. Tuning can also be done in an unsupervised manner requiring a cost function such that the algorithm with the optimal model parameters has the lowest cost [18]. Either way, all model parameter configurations need to be checked (brute force approach) and the amount of possibilities grows with the amount of model parameters and the size of their ranges. A computationally efficient procedure will be introduced in Section 6 to optimize the model parameters in a reduced amount of time compared to the traditional brute force approach.

3. Denoising power signals using median filter

As input for the χ^2 GOF method, the power signal of a household is taken. This section explains the preprocessing needed before for this signal can be used by the event detection method. This preprocessing will also be done for the proposed voting χ^2 GOF and cepstrum method.

Definition of the power signal. A power signal measures the amount of energy consumed per unit time. Thus if an appliance is turned on or off (i.e., an event occurs), the power signal will either increase or decrease. In the Americas and parts of Asia, the maximal power frequency is set to 60 Hz and in the rest of the world it is set to 50 Hz. Note that event detection can also be applied on other signals characterising the events defined by the turning on and off an appliance, e.g., the voltage/current measurements.

Denoising. In [19] it is reported that noise or spikes in the power signal can trigger false detection of transitions, which can significantly hamper the performance of the event detection and thus successful load disambiguation of individ-



(a)

(b)

Figure 1: An example power trace [21] with noise is given in (a). The noise is falsely detected as an event. Figure (b) shows the power trace after applying the median filter with $m = 30$ samples.

ual appliances. Therefore, it is important to remove the noise by preprocessing the data. In digital image processing, a similar preprocessing step is also needed and quite often this is done by the median filter, as it can remove impulsive noise while preserving sharpness of the edges [20]. Each sample in the power signal p_i is replaced by the median of its m neighbours:

$$p_i = median(p_{i-m/2} + p_{i-m/2+1} + ... + p_{i+m/2-1} + x_{i+m/2}) \qquad (1)$$

As an example, Figure 1 shows that the standard χ^2 GOF event detection method identifies noise as an event if the signal is not filtered. It is found that the effectiveness of the median filter depends on the choice of its window size m. Therefore, this model parameter must be optimized (preferable in an efficient manner), as discussed further in Section 6.

4. Voting χ^2 GOF Method

The standard χ^2 GOF method [6] detects events by relying on the fact that the distribution of power values before/after the occurrence of an event are different. To assess this difference, a probabilistic χ^2 test can be used. Assume two consecutive non-overlapping windows $q = (q_1, q_2, \cdots, q_n)$ and $p = (p_1, p_2, \cdots, p_n)$, each containing n data samples from the power signal. Then, an event occurs at the end of window q with a confidence level of $100(1-\alpha)\%$

and $n-1$ degrees of freedom, if

$$l_{GOF} = \sum_{i=1}^{n} \frac{(q_i - p_i)^2}{p_i} > \chi^2_{\alpha,n-1}. \qquad (2)$$

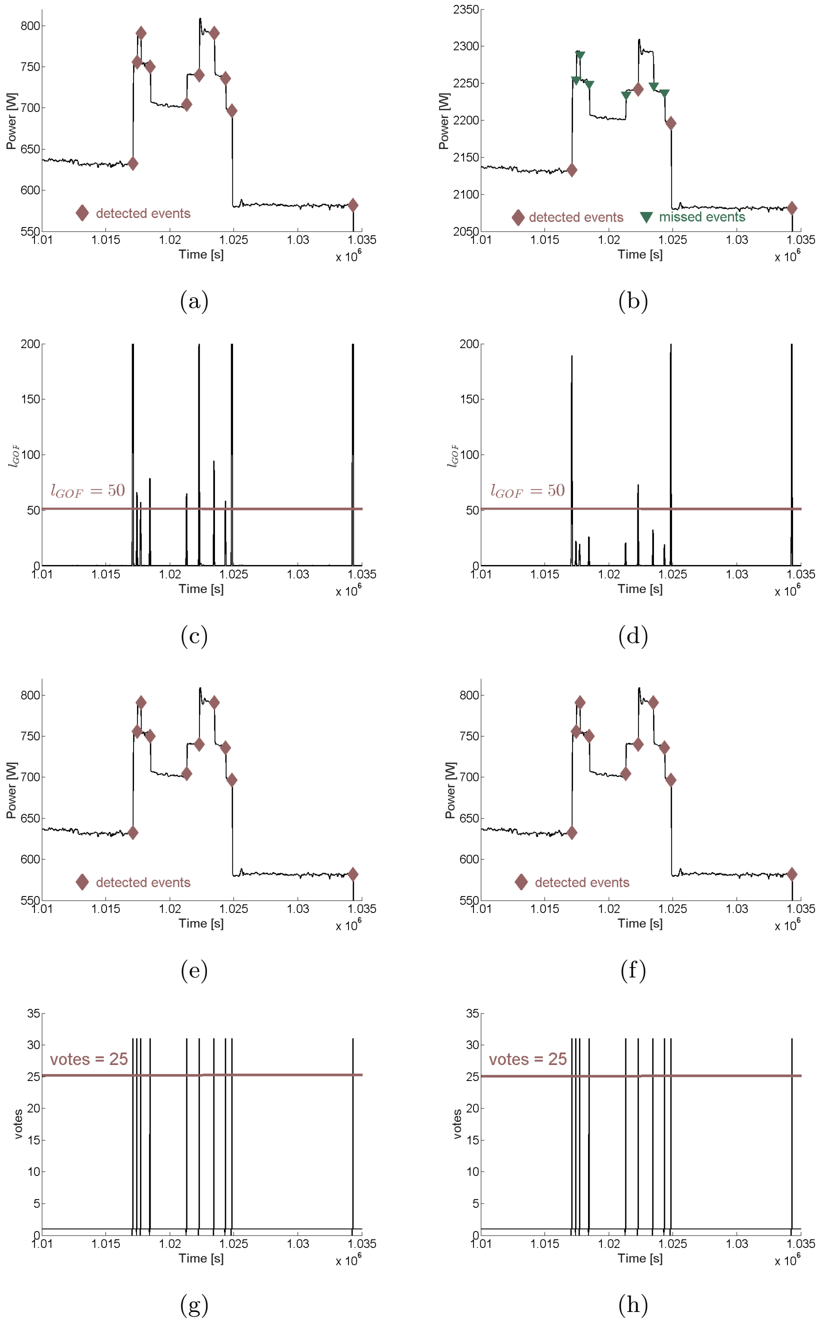
The values of the $\chi^2_{\alpha,n-1}$ distribution can be looked up in a table[1] .

Base line robustness. Although this χ^2 GOF method has been shown to be effective, it is not robust against base load changes. Figures 2 illustrates this. When the power base level is around 600W and an appliance using 50W is switched on (as in Figure 2a), the event is correctly detected (see Figure 2c). However, the event would have been missed if a base load of 1500W is added (see Figure 2b)and 2d). It is seen from equation (2) that events are characterized as a change in power $(q_i - p_i)$ relative to the power signal itself (p_i). Therefore, the method is prone to miss smaller events when the base load of the signal is high. This can lead to poor results, as shown in Figure 2b. As a solution, a voting mechanism (based on the idea presented in [7]) is proposed in this paper to solve the problem.

Improved method. In the voting χ^2 GOF method, the GOF is calculated for each sample in the power signal as given by equation (2). A voting window of length w slides over the resulting time-series of GOF-values and a sample gets a vote if its GOF-value is the highest among all points in the voting window. This results in a maximum of w votes. Each sample receiving at least v_{thr} votes is flagged as an event. As illustrated in the example of Figure 2e and 2f, the voting χ^2 GOF method is able to improve the detection ratios compared to the standard approach. The results section (Section 7) shows the robustness of the voting method against changes in base load in more detail.

Parameter configurations. Both the normal and voting χ^2 GOF method are sensitive to model parameter configurations, i.e., the confidence level α, the window size n, and extra for the voting method: the voting window size w and

[1] `http://sites.stat.psu.edu/~mga/401/tables/Chi-square-table.pdf`



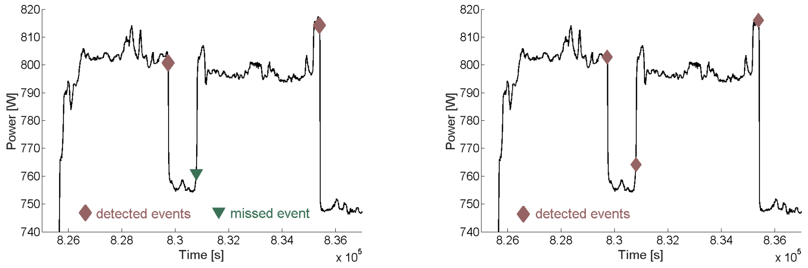
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 2: (a) A sample power trace from [21] and its detected events using the standard χ^2 GOF method with $n = 40$. (c) The corresponding l_{gof} values. (e) The detected events using the voting χ^2 GOF method with $w = 30$ and $v_{thr} = 25$. (g) The corresponding votes. Figures (b), (d), (f) and (h) show the same information for cases where a base load of $1500W$ is added.



(a) (b)

Figure 3: A sample power trace from the BLUED dataset [21], after applying a median filter with $m = 30$, with events detected by χ^2 GOF with $\alpha = 90\%$ for different window sizes: (a) $n = 20$, (b) $n = 40$. In the first case, only two events are detected, while three events are detected when $n = 40$.

voting threshold v_{thr}. In [6] a suggestion is given for determining the window size n. However, small changes in the model parameter configurations can lead to missed events. Figure 3 shows an example for the normal χ^2 GOF method where three events are detected when the window size $n = 40$, but only two events are detected when the window size $n = 20$. It is thus beneficial to optimize the model parameters in an efficient way, which can be done using surrogate based optimization, see Section 6.

5. Cepstrum method

The previous sections investigated the power signal in the time domain. Alternatively, an analysis can be performed in the frequency domain using, e.g., cepstrum analysis. Cepstrum analysis was first introduced in 1963 to analyze the echoes within seismic signals produced by earthquakes [22]. Since then, it has proven to be a potent technique in several domains. One application is passive sonar, which involves listening to the environment without sending signals in order to detect objects [23]. Another application is speech recognition [24], where cepstrum analysis has been successfully applied to increase the robustness of various algorithms. In the context of NILM, recent work has demonstrated the usefulness of cepstrum analysis for appliance recognition [25],

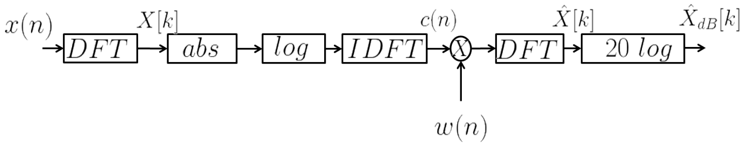


Figure 4: A schematic overview of the transformation from a time signal to spectral smoothed dB-scaled frequency components.

especially when multiple devices are (de)activated simultaneously. Here, cepstrum analysis is used for event detection, rather than for appliance recognition.

Robust Cepstrum Method. When using the cepstrum method, events are detected in the frequency domain where smoothing occurs in the quefrency domain, rather than the time domain. The different steps are outlined in Figure 4. Consider a window x of length n from a power signal p,

$$x = (p_i, p_{i+1}, ..., p_{i+n}) \tag{3}$$

where events need to be detected. First, this window will be converted from the time to the frequency domain, by using the Fourier transform:

$$X[k] = \sum_{j=1}^{n} x[j]\ e^{-2\pi ikj/n}\ ,\ 0 \leq k < n. \tag{4}$$

Then, the inverse Fourier transform is applied to the logarithm of $|X|$, leading to the cepstrum components in the so-called quefrency domain:

$$c[n] = \frac{1}{N} \sum_{k=0}^{N-1} log_{10}(\ |X[k]|\ )e^{2\pi ink/N}\ ,\ 0 \leq n < N. \tag{5}$$

These cepstrum components are smoothed by means of a filter z, after which they are transformed back to frequency components by applying the Fourier transformation:

$$\hat{X}[k] = \sum_{j=1}^{n} z[j]\ c[j]\ e^{-2\pi ikj/n}\ ,\ 0 \leq k < n. \tag{6}$$

The filter z is defined as one minus the Hann window, with a response as visualised in Figure 5:

$$z[j] = 1 - 0.5\ (1 - cos(2\pi j/n))\ ,\ 0 \leq j \leq n. \tag{7}$$

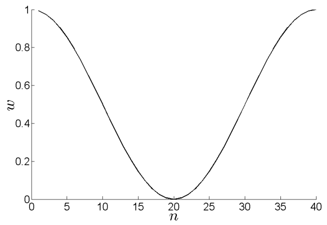


Figure 5: The response of the filter z.

Because the relative difference in values of the components is more informative than the absolute difference, the frequency components are converted to a decibel (dB) scale:

$$\hat{X}_{dB}[k] = 20\ log_{10}(\hat{X}[k]). \tag{8}$$

These components are an informative indicator for the absence or presence of events in the time window. This is illustrated in Figure 6: if an event is present, all the cepstrum smoothed dB scaled frequency components have higher values (see Figure 6b) than when no event is present (see Figure 6d). Finally, it is checked whether all frequency components exceed a chosen threshold τ, and declare an event if the following condition holds:

$$\min_{0 \leq k < N}(\hat{X}_{dB}[k]) > \tau \tag{9}$$

Note that the threshold τ needs to be optimized in order to achieve high event detection ratios. The efficient optimization of this parameter τ (and others indicated previously) is discussed next, in Section 6. The result section (Section 7) shows that this method, just like the voting χ^2 GOF method, is robust against changes in the base load.

6. Efficient surrogate-based model parameter optimization

All the methods described in the previous sections have parameters that need to be optimized in order to achieve high event detection ratios. The total number of model parameter configurations that must be evaluated is very high, see Section 7 for the specific numbers. Rather than reducing the granularity of

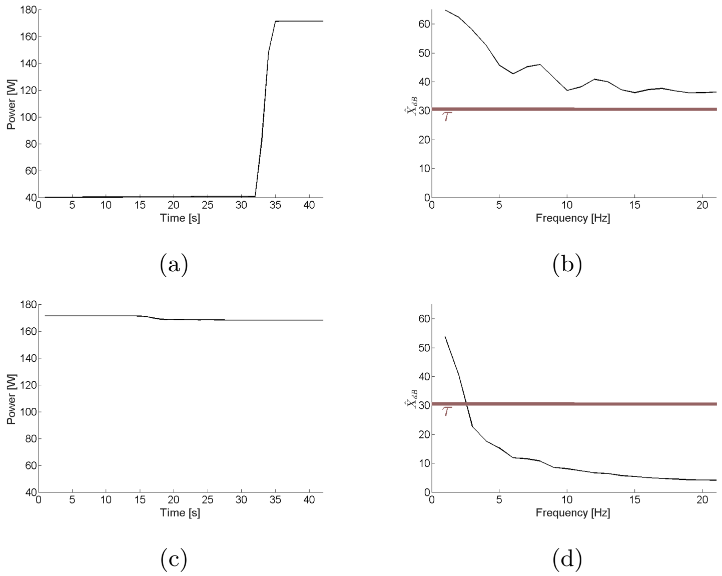


Figure 6: Example of a window with size $n = 40$ of a power trace and the corresponding smoothed frequency components $\hat{X}$ of an event, respectively (a) and (b), and a non-event, respectively (c) and (d).

the model parameter ranges, surrogate-based optimization is proposed, which can significantly speed up the process. It is chosen to adopt Surrogate-Based Optimization (SBO) [8], which assumes that smooth changes in the model parameter configurations will lead to smooth changes in detection ratios. Under that condition, an exhaustive search of the overall model parameter space is not required to find the optimum solution. Rather than computing the results for all possible model parameters, a surrogate model of the optimization objective function (the utility function) is generated that is locally accurate in the regions of interest (the optimum). SBO makes use of one of the most popular sequential sampling concepts, namely, the Expected Improvement (EI) measure for optimization with locally accurate surrogate models [8]. The EI measure aims to maximize the utility function by guiding the sequential selection of appropriate model parameter configurations into the direction where the optimum solution is most likely to be found, using Bayesian methods. Once the algorithm discov-

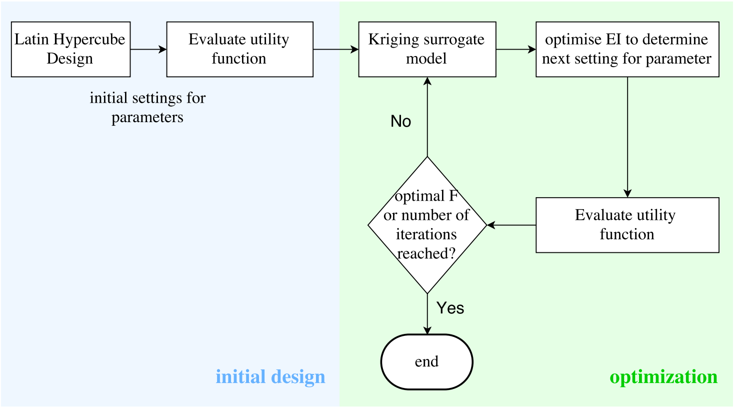

Latin Hypercube Design
Evaluate utility function
Kriging surrogate model
optimise EI to determine next setting for parameter
initial settings for parameters
No
optimal F or number of iterations reached?
Evaluate utility function
Yes
end
initial design
optimization

Figure 7: A flow chart explaining SBO.

ers a configuration that is sufficiently close to the optimum, the optimization terminates and the final solution is returned. As a result, only part of all model parameter configurations need to be tested to obtain the optimal solution. SBO has already been applied successfully in other research areas, like e.g., wireless communication [26], electromagnetics [8], and microwave filter design [27].

The different steps of the algorithm are summarized in Figure 7. SBO requires a unified utility function that needs to be maximized (i.e., the F-measure as explained further). First, a limited set of calculations are performed on this utility function such that the model parameter space is well sampled to create the initial design. Then, a Kriging surrogate model is built that is sequentially updated with additional configurations as suggested by the EI infill criterion. The EI infill criterion effectively balances between enhancing the global accuracy of the surrogate model (exploration) and improving its accuracy near the optimal solution found so far (exploitation). As the algorithm proceeds, the search is guided towards the optimal solution while limiting the amount of possible configurations for the model parameters. As soon as a satisfactory result is found, the optimization is terminated and the best solution is returned.

Definition of the utility function. To quantify the performance of event detection methods, the harmonic mean of precision and recall (also known as the

F-measure) is used as suggested in [28]. If the considered model parameters are $\mathbf{g}$, it is defined as:

$$F(\mathbf{g}) = 2 \cdot \frac{\mathit{precision}(\mathbf{g}) \cdot \mathit{recall}(\mathbf{g})}{\mathit{precision}(\mathbf{g}) + \mathit{recall}(\mathbf{g})} \tag{10}$$

$$\mathit{precision}(\mathbf{g}) = \frac{\mathit{TP}(\mathbf{g})}{\mathit{TP}(\mathbf{g}) + \mathit{FP}(\mathbf{g})} \tag{11}$$

$$\mathit{recall}(\mathbf{g}) = \frac{\mathit{TP}(\mathbf{g})}{TP(\mathbf{g}) + \mathit{FN}(\mathbf{g})} \tag{12}$$

where *precision* is the fraction of detected events that are true and *recall* is the fraction of true events that are detected, *TP* are the true-positives (correctly predicted events), *FP* are the false-positives (incorrectly predicted events), *FN* are the false-negatives (undetected events). The goal of the optimization procedure is to choose the model parameters $\mathbf{g}$ in such a way that the utility function is maximized.

Evaluation of the initial configurations for model parameters. First, a limited number of configurations for model parameter $\mathbf{g}$ are evaluated using Equation (11) and (12) to determine corresponding values of precision and recall. To this end, an optimized Latin Hypercube Design (LHD) was used because of its space-filling properties [29]. Next, the corresponding F measures are calculated. This leads to the configurations:

$$S = \{(\mathbf{g}_k, F(\mathbf{g}_k)), k = 1 \dots K\}. \tag{13}$$

Generation of a Kriging surrogate model. With the calculated F-measures, a Kriging model is built. Kriging models are part of a broader class of approximation methods, called Gaussian Processes (GP), and have a particular importance in SBO. While traditional approximation methods predict only a single function value, GP methods can predict the uncertainty of a function value as the realization of a normally distributed random variable $Y(\mathbf{g}) \sim N(\mu(\mathbf{g}), \sigma(\mathbf{g}))$ where $\mu(\mathbf{g})$ denotes the predicted value ($\mu(\mathbf{g}) \sim F(\mathbf{g})$) and $\sigma(\mathbf{g})$ denotes the prediction variance. This property is exploited by the EI infill criterion to guide the sequential sampling, as shown in the next section. More details about Kriging can be found in literature, e.g., [30].

Expected Improvement infill criteria. Once a Kriging model is built, the EI measure determines the optimum location of the next infill point $\mathbf{g}$ that contains the parameter configurations used to configure the event detection method. First, the EI quantifies the Probability of Improvement (PoI), the amount of improvement that is expected to occur when a certain configuration is explored as compared to the optimal value found so far. The EI is calculated by considering every possible improvement over the current best value F_{max}, multiplied with the associated likelihood. If $\phi(.)$ denotes the probability density function of a random variable, then the EI can be written in integral form as follows [31]:

$$E[I(\mathbf{g})] = \int_{F_{max}}^{\infty} I(\mathbf{g}) \cdot \phi(Y(\mathbf{g}))\ dY, \tag{14}$$

where the improvement $I(\mathbf{g})$ of $Y(\mathbf{g})$ over F_{max} is defined as

$$I(\mathbf{g}) = max(Y(\mathbf{g} - F_{max}), 0). \tag{15}$$

A graphical illustration of the EI concept is given in Figure 8 where one model parameter is optimized. Note that the EI function (14) corresponds to the first moment of the shaded area in Figure 8. Once a configuration of $\mathbf{g}$ is found for which the $E[I(\mathbf{g})]$ is maximal, its corresponding F-measure is calculated and added as a new data sample to the set S. Based on the additional information, the Kriging model is rebuilt and the process is repeated until a satisfactory solution is found, i.e., until the maximum is reached (in our case $F = 1$), or when the maximum number of iterations is exceeded.

An implementation of the SBO routine is available in the Surrogate Modelling (SUMO) Toolbox [32, 33] (available online[2]).

7. Results and discussion

Dataset. In this section the robustness against different baseload levels of the proposed methods is tested on the BLUED dataset [21]. The aggregated power

[2] `http://sumo.intec.ugent.be`

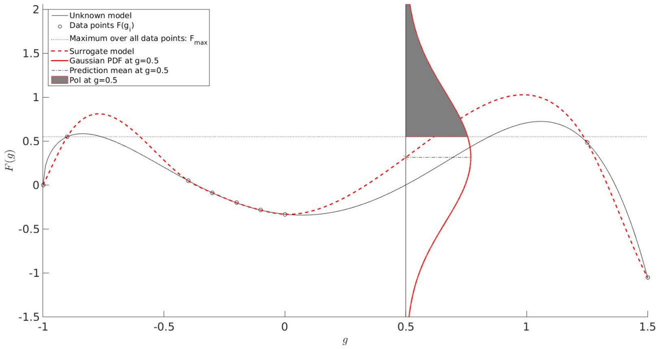


Figure 8: A graphical illustration of expected improvement: a surrogate model (dashed line) is constructed based on some data points (circles) of an unknown function $F(\mathbf{g})$. For each point the surrogate model predicts a Gaussian probability density function (PDF). An example of such a PDF is drawn at $g = 0.5$.

signal sampled at 60Hz from a family residence in the United States for a whole week is considered. Every state transition of each appliance is manually labeled, providing the ground truth. The considered house has a two-phase power consumption, where 904 transitions are recorded in phase A and 1578 in phase B. Each phase has its own properties, e.g., phase B is more noisy than phase A. For that reason, phase A and B are optimized and tested separately. For each method, the data is passed through a median filter, as explained in Section 3.

Cross validation. Performance is evaluated on 20% of the data, whereas the remaining 80% is used for training. Performance is reported averaged over 10 runs (each with a random 20% test split). For training, 5-fold cross validation is used on the other 80% to set the optimal parameter values. The overall set-up is summarized in Figure 9. Note that for the division, the trace of an entire day is taken as a whole unit.

Trained model parameters. The model parameters and the ranges under consideration are listed in Tables 1, 2 and 3 for the standard χ^2 GOF method, the voting χ^2 GOF method and the cepstrum method. As the power frequency is 60 Hz, this means there is a sample every 0.02 seconds. Consequently, when the

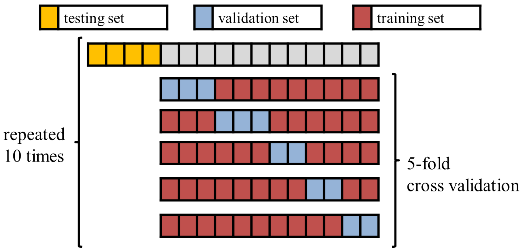


Figure 9: Schematic overview of the optimization procedure.

window size n varies from 1 to 100, it covers a time window from 0.02 seconds to 2 seconds. The total number of model parameter configurations to be evaluated is very high: the choices listed in Table 1-3 amount to 50000 $(= 100 * 100 * 5)$, 10000000 $(= 100 * 100 * 100 * 10)$ and 500000 $(= 100 * 100 * 50)$ for the standard χ^2 GOF method, the voting χ^2 GOF method, and the cepstrum method respectively. Since the approximate running time of the algorithms is 214, 1.635 and 529 seconds, this results in an overall optimization time of 124, 189.236, and 3.061 days respectively on a modern dual core machine. This makes a brute-force optimization of the model parameters practically infeasible. Therefore, training is done using SBO, as explained in Section 6. To begin with, $K = 10$ configurations for the model parameters are evaluated and used to build the Kriging model. Based on this model, one new configuration for the parameters is proposed and evaluated with the F-measure. The result is used to update the Kriging model. This is done 90 times, resulting in 100 evaluated configurations. Adding more iterations did not prove to be useful in practice, as the F-measure did not significantly improve and the suggested parameter configurations were all in the same neighbourhood. A good (but possibly local) optimum is thus found.

Test use cases. To find out if each method is robust against changes in the base load of the power signal, the three methods are applied in three different use cases. In all cases, the data is preprocessed with a median filter, and a base load of respectively 0W, 1500W and 3000W is added to the power signal. In practice, such high base load conditions arise when multiple high-power devices

standard χ^2 GOF				
name		range	optimization time / configurations	
window median filter	m	$[1, 100]$	brute force	124 days / 50000
window event detection	n	$[1, 100]$	SBO	5.9 hours / 100
confidence level	α	$\{90, 95, 97.5,$ $99, 99.9\}$		

Table 1: On the left side are the present model parameters **g** and range for the standard χ^2 GOF event detection. On the right side are the optimization time for the model parameters and the amount of parameter configurations needed to be checked.

voting χ^2 GOF				
name		range	optimization time / configurations	
window median filter	m	$[1, 100]$	brute force	189.2 days / 10000000
window event detection	n	$[1, 100]$	SBO	45 hours / 100
window voting system	w	$[1, 100]$		
voting threshold	v_{thr}	$w * [0.1, 1]$		

Table 2: On the left side are the present model parameters **g** and range for the voting χ^2 GOF event detection. On the right side are the optimization time for the model parameters and the amount of parameter configurations needed to be checked.

are operating in the background, such as electrical heaters (that can easily consume 1500W). Considering the results of these use cases, a conclusion can be made about each method's robustness.

Robustness of standard χ^2 GOF method. The results of the standard χ^2 GOF method when applied on the first three use cases are given in Figure 10, showing the spread of the F-measure caused by running the 5-fold cross validation ten times. When no offset is added, the performance for phase A is almost perfect ($F \approx 0.98$) and for phase B the performance is $F \approx 0.80$. However, this changes when the base load is increased by adding an offset to the signal. The F-measure keeps dropping as the offset increases, indicating that the method is not robust

cepstrum				
name		range	optimization time / configurations	
window median filter	m	$[1, 100]$	brute force	3.1 days / 500000
window event detection	n	$[1, 100]$	SBO	14 hours / 100
threshold	τ	$[1, 50]$		

Table 3: On the left side are the present model parameters $\mathbf{g}$ with their respective abbreviation and range for the cepstrum method. On the right side are the optimization time for the model parameters and the amount of parameter configurations needed to be checked.

against higher base loads.

Robustness of voting χ^2 GOF method. The results of the voting χ^2 GOF method when applied on the first three use cases are given in Figure 10. As can be seen, the voting χ^2 GOF method gives comparable results to the standard χ^2 GOF method for phase A and B when no offset is added. When offsets are added to the signal, the F-measure remains the same, indicating the robustness of the voting χ^2 GOF method. This in contrast to the standard χ^2 GOF method.

Robustness of cepstrum method. The results of the cepstrum method when applied on the first three use cases can be found in Figure 10. The F-measure for phase A when no offset is added to the base load, is the same as the previous methods ($F \approx 0.98$). For phase B, it is $F \approx 0.81$, which is a bit higher than the previous methods. When an offset is added to the base load, the F-measure remains the same ($F \approx 0.81$), just like the voting χ^2 GOF method, indicating the robustness of the cepstrum method.

Timing improvement due to SBO. Comparing the running time of SBO and the brute-force approach, it is found that for the standard χ^2 GOF method it is reduced from approximately 124 days to 5.9 hours, for the voting χ^2 GOF method from 189.2 days to 45 hours, and for the cepstrum method from 3.1 days to 14 hours, resulting in a speed up factor of approximately 500, 100000 and 5000 respectively. This is caused by the fact that the amount of evaluated parameter

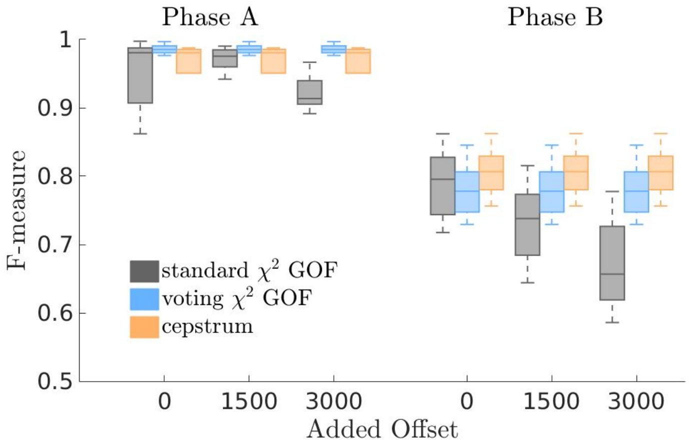


Figure 10: The F-measure when detecting events with the standard χ^2 GOF, voting χ^2 GOF and cepstrum method.

configurations is reduced from 5000, 10000000 and 500000 (for the standard χ^2 GOF method, the voting χ^2 GOF method, and the cepstrum method respectively) to 100 for all methods while maintaining a good F-measure.

8. Conclusion

Two event detection methods have been proposed, namely (1) the voting χ^2 GOF method, and (2) the cepstrum method. Each method is robust against base load differences compared to the standard χ^2 GOF method. For example, when a base load of 3000W (which corresponds to the power consumption of two typical electrical heaters) is added to the power signal, compared to the standard χ^2 GOF method, the voting χ^2 GOF method leads to a performance increase of $7-12\%$ in terms of F-measure, while Cepstrum reaches $7-15\%$ larger F-measure values.

In order to obtain optimal parameter configurations of these methods, a workflow using surrogate-based optimization is proposed. Timing results confirm that the parameter optimization process can be sped up: in our experiments there is a speed up with a factor up to 100000 between the standard brute force and the surrogate-based optimization.

References

[1] European Commission, *Climate Strategies & Targets*, http://ec.europa.eu/clima/policies/strategies/2030/index_en.htm (Visited August 2016)

[2] Ehrhardt-Martinez, K., et al., *Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities.*, Washington, D.C.: American Council for an Energy-Efficient Economy; 2010.

[3] Iyer, S. R., et al., *Energy disaggregation analysis of a supermarket chain using a facility-model.*, Energy and Buildings 97, 2015, 65-76.

[4] Armel, K. C., et al., *Is disaggregation the holy grail of energy efficiency? The case of electricity.* Energy Policy 52, 2013, 213-234.

[5] Hart, G. W., *Nonintrusive appliance load monitoring.*, Proceedings of the IEEE 80.12, 1992, 1870-1891.

[6] Jin, Y., et al., *A time-frequency approach for event detection in non-intrusive load monitoring*, SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2011, 80501U-80501U).

[7] Anderson, K., et al., *Event detection for non intrusive load monitoring.*, 38th Annual Conference on IEEE Industrial Electronics Society. IECON, 2012, 3312-3317.

[8] Couckuyt, I, et al., *Surrogate-based infill optimization applied to electromagnetic problems.*, International Journal of RF and Microwave ComputerAided Engineering 20.5, 2010, 492-501.

[9] Berges, M., et al., *"Learning systems for electric consumption of buildings."*, ASCI international workshop on computing in civil engineering (38), 2009.

[10] Yang, C. C., et al., *Comparative study of event detection methods for non-intrusive appliance load monitoring.*, Energy Procedia 61, 2014, 1840-1843.

[11] Yang, C. C., et al., *A systematic approach to ON-OFF event detection and clustering analysis of non-intrusive appliance load monitoring.*, Frontiers in Energy 9.2, 2015, 231-237.

[12] Wong, Y. F., et al., *Recent approaches to non-intrusive load monitoring techniques in residential settings.*, Computational Intelligence Applications In Proceedings of the IEEE Symposium on the Smart Grid (CIASG) , 2013, 79-79.

[13] Trung, K. N., et al. , *Event Detection and Disaggregation Algorithms for NIALM System.*, Proceedings of 2nd International Non-Intrusive Load Monitoring (NILM) Workshop. 2014.

[14] Kazmi, A. H., et al. , *Real-Time Event Detection for Energy Data Streams.*, Ambient Intelligence, 2014. 221-225.

[15] Luong, T. M., et al., *Hidden Markov Model Applications in Change-Point Analysis.*, arXiv preprint arXiv:1212.1778, 2012.

[16] Grinblat, G. L., et al., *Abrupt change detection with one-class time-adaptive support vector machines.*, Expert Systems with Applications 40.18. 2013, 7242-7249.

[17] Gu, W., et al., *Fast Change Point Detection for electricity market analysis.*, Proceedings of the IEEE International Conference on Big Data. IEEE, 2013, 50-57.

[18] Volpi, M., et al., *Unsupervised change detection with kernels.*, IEEE Geoscience and Remote Sensing Letters 9.6, 2012, 1026-1030.

[19] Wu, T., and Srivastava, M., *Low-cost appliance state sensing for energy disaggregation.*, Proceedings of the 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, 2012,53-55.

[20] Pitas, I., *Nonlinear digital filters*, Principles and Applications, 1990.

[21] Anderson, K., et al., *BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research*, Proceedings of the 2nd KDD workshop on data mining applications in sustainability (SustKDD), 2012,1-5.

[22] Borgert, B. P., et al. , *The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum and saphe craking.*, Proceedings of the On Time Series Analysis, 1963, 209-243.

[23] Kiran, R.,et al., *Application of Cepstrum in Passive Sonar.*, International Journal of Engineering Research and Applications, 2012, 1919-1924.

[24] Hirsch, H.-G., and Pearce, D., *The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions.*, Proceedings of the Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop, 2000.

[25] Kong, Seongbae, et al., *Home appliance load disaggregation using cepstrum-smoothing-based method.*, IEEE Transactions on Consumer Electronics 61.1, 2015, 24-30.

[26] Mehari, M. T., et al., *Efficient global optimization of multi-parameter network problems on wireless testbeds.*, Ad Hoc Networks 29, 2015, 15-31.

[27] Singh, P., et al., *Accurate hotspot localization by sampling the near-field pattern of electronic devices.*, IEEE Transactions on Electromagnetic Compatibility 55.6, 2013, 1365-1368.

[28] Makonin, S., and Popowich, F., *Nonintrusive load monitoring (NILM) performance evaluation.*, Energy Efficiency 8.4, 2015, 809-814

[29] Joseph, V. R., and Hung, Y., *Orthogonal-maximin Latin hypercube designs.*, Statistica Sinica, 2008, 171-186.

[30] Couckuyt, I., et al. , *Kriging, co-kriging and space mapping for microwave circuit modeling.*, Proceedings of the 41st European. IEEE Microwave Conference (EuMC), 2011, 444-447.

[31] Jones, D. R., *A taxonomy of global optimization methods based on response surfaces.*, Journal of global optimization 21.4, 2001, 345-383.

[32] Gorissen, D., et al., *A surrogate modeling and adaptive sampling toolbox for computer based design.*, The Journal of Machine Learning Research 11, 2010, 2051-2055.

[33] van der Herten, J., et al. , *Adaptive classification under computational budget constraints using sequential data gathering.*, Advances in Engineering Software, 2016, 137-146