Title
Virtual Power Sensing Based on a Multiple-Hypothesis Sequential Test

Authors
Kang, Zhaoyi
Zhou, Yuxun
Zhang, Lin
et al.

Publication Date
2014-11-18

Virtual Power Sensing Based on a Multiple-Hypothesis Sequential Test

Zhaoyi Kang[1], Yuxun Zhou[1], Lin Zhang[2] and Costas J. Spanos[1]
[1]Dept. of Electrical Engineering & Computer Sciences, UC Berkeley, Berkeley, CA 94709
[2]Dept. of Electronic Engineering, Tsinghua University, Beijing 100084
[1]{kangzy, yxzhou, spanos}@eecs.berkeley.edu, [2]linzhang@mail.tsinghua.edu.cn

***Abstract*—Virtual-Sensing, which is achieved through the disaggregation of composite power metering signals, is a solution towards achieving fine-grained smart power monitoring. In this work we discuss the challenging issues in Virtual-Sensing, introduce and ultimately combine the Hidden Markov Model and the Edge-based methods. The resulting solution, based on a Multiple-hypothesis Sequential Probability Ratio Test, combines the advantages of the two methods and delivers significant improvement in disaggregation performance. A robust version of the test is also proposed to filter the impulse noise common in real-time monitoring of the plug-in loads power consumption.**

I. Introduction

Buildings account for more than 40% of the total power consumption in the US, and can play a critical role in addressing the current energy and climate issues [1]. Significant effort has been invested in this topic, from benchmarking, to control and monitoring. However, building energy usually depends greatly on occupant behavior, especially at fine-grained metering level, such as plug-in loads, user-controlled lighting, user-adjusted HVAC, for which the traditional commissioning system is far from enough [2].

So-called smart grid technology is proposed as a solution from an informatics perspective, which provides multi-scale monitoring of building power & environmental conditions, based on high-density Wireless Sensor Networks (WSNs) and efficient data processing methods [3]. Large-scale WSNs has been deployed with hundreds of terminal meters to provide long-term monitoring [4]. Demand-response application of plug-loads and personalized lighting systems has also been demonstrated, which shows the great potential of an integrated power control strategy in smart buildings, especially at fine-grained plug-in loads level [5]. Furthermore, fine-grained metering also provides important statistics to understand user behavior and the potential of *smart* living environment [4] [5].

Notwithstanding all the benefits, issues as cost, privacy and network stability arise as the system of fine-grained metering scales up to whole-building level. One solution is to develop power strip level metering as the *ElectriSense* line of products [6]. However, network burden and the cost problem still remain. Another scheme is to use fewer sensor nodes, and *reconstruct* the missing nodes through data mining techniques, which is equivalent to having *virtual* sensors at those nodes. We call this idea *Virtual-Sensing* (VS). VS reduces the number of sensors in the network, and can therefore address the cost, privacy and network issues.

A common thread in VS is *Load Disaggregation* (LD). In LD, we *decode* the streams of individual appliances from a high-level *composite* power stream, as illustrated in Figure 1, therefore to create *virtual* monitoring at low-levels. As we monitor with sparsely instrumented nodes, network issue is alleviated and less intrusiveness is caused at the terminal users [7]. In this work, we will study various existing VS methods and discuss their performance, ultimately proposing a Multiple-Hypothesis Sequential Probability Ratio Test (MSPRT) that shows advantages over the existing methods.

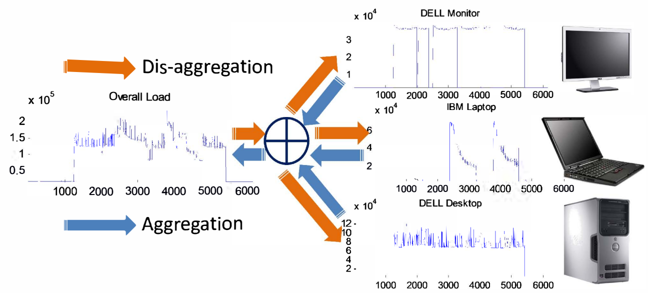

Fig. 1. Virtual-Sensing framework for load disaggregation

This paper is organized as follows. Section II formulates the problem. Section III presents the background. Section IV describes MSPRT in some detail. Section V outlines a complete VS method based on MSPRT, also proposing a robust enhancement to address typical power sensing noise. Finally, Section VI concludes the paper.

II. Problem Formulation

Let p_t be the observed composite power signal time series from n appliances with t from 1 to T. Let $\mathbf{s}_t$ be the state vector of the n appliances at time t. Our task is to infer $\mathbf{s}_t$ from p_t. $\mathbf{s}_t$ is a vector of n binary variables, one for each appliance, i.e. $\mathbf{s}_t \in \{0,1\}^n$, in which 0 stands for OFF state and 1 for ON state. There are in total 2^n combinations of ON/OFF states.

III. Background and Related Work

Typical solutions to this problem are either based on a Hidden Markov model, or on Edge-based model. In the next section, these approaches will be reviewed and compared.

A. Hidden Markov Model (HMM)

In a Hidden Markov Model (HMM) [8], the transition probability between previous state $\mathbf{s}_{t-1}$ and current state $\mathbf{s}_t$ is given as $\Pi_{\mathbf{s}_{t-1},\mathbf{s}_t} = \Pr(\mathbf{s}_t|\mathbf{s}_{t-1})$. The mean of the composite power signal is the outcome of the hidden state $\mathbf{s}_t$ and its spread is modeled as a Gaussian distribution. In order to infer the status of the individual appliance, we estimate the state at each step, based on Maximum Likelihood Estimation (MLE):

$$\mathbf{s}_t = arg\max_{\mathbf{s}} \Pr(\mathbf{s}_t = \mathbf{s}|\mathbf{p}_{1:T}) \tag{1}$$

Since the search space is 2^n, there will be an exponential explosion w.r.t. n. However, if we assume that only one appliance is switching at each step, the incremental state search space from $\mathbf{s}_{t-1}$ to $\mathbf{s}_t$ is only n. This assumption is reasonable with manually switched devices and a fast power sampling rate in the order of 1 sec/sample. Equ. (1) can be solved by a Viterbi algorithm [8]. We note $\delta_t(\mathbf{s}) = \Pr(\mathbf{s}_t = \mathbf{s}|\mathbf{p}_{1:T})$ as the likelihood function and we use $\mathbf{s}' = \Psi_t(\mathbf{s})$ to store the most likely state at step t given that the state at $t-1$ is $\mathbf{s}$, then we have the relation: i:

$$\delta_t(\mathbf{s}) = \max_{\mathbf{s}'} \delta_{t-1}(\mathbf{s}')\Pi_{\mathbf{s}',\mathbf{s}} \Pr(p_t|\mathbf{s}_t = \mathbf{s}) \tag{2}$$

$$\Psi_t(\mathbf{s}) = arg\max_{\mathbf{s}'} \delta_{t-1}(\mathbf{s}')\Pi_{\mathbf{s}',\mathbf{s}} \tag{3}$$

The above problem is solved sequentially, as first estimate the state at the last step, and then backtrack for the best estimate at each step as $\mathbf{s}_{t-1} = \Psi_t(\mathbf{s}_t)$.

The HMM gives good results and has been used a lot in VS applications. Wang *et al.* [9] treated VS in a convex optimization framework using sparse constraints. [10] solved the HMM by the Extended Viterbi algorithm and considered only the major power consuming appliances. The sampling method is widely used to deal with the exponential explosion issue. In [11] [12] [13] [14], statistical inference of the joint distribution is based on Factorial HMM [15], though most of the sampling methods have computation issues.

However, the standard HMM does not have a good way to handle the fact that states may stay unchanged for a wide range of time intervals. This is significant for our problem, since many appliances, such as a lamp or a monitor, will have very different timing characteristics, while HMM models the duration as a Geometric distribution [12]. Some extensions of HMM have been proposed to address this issue. In [16], the persistence of state (stickiness) is guaranteed by introducing a constraint on the Markov chain model. Whereas in [12], [13], a Hidden Semi-Markov Model is used to model duration statistics. However, in most cases, we need a long training period of time of this model, since ON/OFF of individual devices may not be that frequent.

B. Edge-based Model

An intuitive way to deal with duration modeling is to focus only on the ON/OFF edges, in an approach we will call the Edge-based model. Edge-based model applies a change detection algorithm to track the edges and trace the source based on statistical learning methods [17]. Usually, we track the mean and variance of the power stream over time using an exponential moving average filter as:

$$\beta_0 = \frac{1}{d}\sum_{\tau=t-1}^{t-d} P_\tau \exp\left(-\frac{\tau - t}{\omega}\right) \tag{4}$$

$$\sigma_0^2 = \frac{1}{d}\sum_{\tau=t-1}^{t-d} (P_\tau - \beta_0)^2 \exp\left(-\frac{\tau - t}{\omega}\right) \tag{5}$$

where ω is the decay factor and d the window size. Then, we look at the deviation of the current data point w.r.t. the previous statistics [18]. Edge-based model originates from the early work on Non-Intrusive Appliance Load Monitoring (NIALM) [19]. A review can be found in [7]. Algorithms that are studied for NIALM include Linear Discriminant Classifier [20], Bayes classifier [21], Neural Network [22], etc.

Around the edges, there are several transient features that can be extracted from the active power or the reactive power readings, the latter often having unique harmonic patterns when observed at high enough sampling rates [23]. Such high frequency transients can help distinguish between, for example of a coffee-maker and a chandelier, especially when focusing on their reactive power patterns.

In general, high frequency sampling will also be useful in distinguishing between appliances, since larger data sets, aided by the *Law of Large Numbers* [18], will generally be better for discriminating among different patterns. The obvious tradeoff here is, of course, that higher sampling rates would typically imply higher instrumentation and computational cost, and/or lower instrumentation and computational robustness.

IV. Multi-Hypothesis Sequential Probability Ratio Test (MSPRT)

As we discussed above, edge detection is inherently a hypothesis testing problem [18]: the null hypothesis is H_0 (*no change detected*), and the alternative hypothesis is H_1 (*change detected*). Let $T(x)$ be a test statistic so that if and only if $T(x) > c$ that H_0 is rejected. Next we discuss how this principle can be implemented in our problem.

A. Neyman-Pearson (N-P) Test

Let the probability density and parameter spaces be $f_0(x)$ and θ_0 for H_0, and $f_1(x)$ and θ_1 for H_1. The N-P test ensures that the Uniformly Most Powerful test given certain False Positive (FP) Rate is achieved by using Probability Ratio $T(x) = \frac{f_1(x)}{f_0(x)}$ as test statistic. The test result $\delta(x)$ is:

$$\delta(x) = \begin{cases} 1 & \text{if} \quad T(x) > \lambda \quad \text{i.e. reject } H_0 \\ 0 & \text{if} \quad T(x) < \lambda \quad \text{i.e. do not reject } H_0 \end{cases} \tag{6}$$

where the value of λ is determined from the constraint of FP Rate $\alpha = \Pr(T(x) > \lambda|H_0)$. The optimality of this test motivates us to cast the VS problem into this formulation. However, the power of the N-P test depends on the sample size of the input data x, which limits the performance of the test.

B. Sequential Neyman-Pearson Test

The sample size issue can be solved by the sequential version of the N-P Test, knows as the Sequential Probability Ratio Test (SPRT). In this framework, the likelihood function is incrementally updated after every new sample arrival [24], given $x^n = \{X_1, \cdots, X_n\}$:

$$\mathbf{L}(x^n) = \log\frac{f_1(x^n)}{f_0(x^n)} = \mathbf{L}(x^{n-1}) + \log\frac{f_1(X_n)}{f_0(X_n)} \tag{7}$$

We reject H_0 if $\mathbf{L}(x^n) > \alpha$ and reject H_1 if $\mathbf{L}(x^n) < \beta$, where α and β are two constants. If $\alpha \geq \mathbf{L}(x^n) \geq \beta$, we continue to accept new samples till a decision can be made.

SPRT simulates the way human makes decisions. One makes decision if one has enough confidence and will continue to receive information if not. In SPRT, we do not need to pre-determine the size of the test. Instead, the size is adaptively determined based on the observations. Even better is that SPRT requires fewer samples than standard non-sequential N-P test given the same FP Rate constraint. The expected number of samples for certain FP rate α is given as [25]

$$\mathbf{E}(N|H_0, H_1) \approx \frac{\log(\alpha)}{D(f_0|f_1)} \quad \text{for Sequential} \tag{8}$$

$$\mathbf{E}(N|H_0, H_1) \approx \frac{\log(\alpha)}{C(f_0|f_1)} \quad \text{for Non-sequential} \tag{9}$$

In which $D(f_0|f_1)$ is the Kullback-Leibler (K-L) distance and $C(f_0|f_1)$ the Chernoff distance. For Gaussian variable, the K-L distance is usually greater than Chernoff distance. Therefore, SPRT needs fewer samples to reach a decision.

C. MSPRT

Now we move on to multiple-hypothesis test. If we have one null hypothesis and k alternative hypotheses, from [18], we should compare one hypothesis with all the other choices. Suppose that the k^{th} hypothesis has a prior π_k, we can write the posterior probability of the k^{th} hypothesis as:

$$p_n^k = \frac{\pi_k \prod_{i=1}^n f_k(X_i)}{\sum_{j=0}^K \pi_j \prod_{i=1}^n f_j(X_i)} \tag{10}$$

For computation purpose, we use its inverse as the test statistic. Decision is made towards the k^{th} hypothesis if the threshold corresponding to the k^{th} hypothesis, which is noted as χ_k, is exceeded. Otherwise, more data are sampled:

$$F_1^n(k) = \frac{1}{p_n^k} < \chi_k \tag{11}$$

The algorithm works as in Figure 2(a), in which $F_1^n(1)$ exceeds the threshold, whereas $F_1^n(2)$ goes to the opposite direction. The threshold for the k^{th} hypothesis is calculated as $\chi_k = \alpha \left(\delta_k \sum_j \frac{\pi_j}{\delta_j}\right)^{-1}$, in which $\delta_k = \min_{j \neq k} D(f_k|f_j)$ [2]. The number of samples we need to reach a decision is:

$$n = \inf \{n \geq 1, F_1^n(k) < \chi_k, \forall k\} \tag{12}$$

The second issue is to locate the edge efficiently. Usually, exact edge location is not known *a priori*. If we assume the edge is at time τ. Then, the accumulation of the probability ratio function in Equ. (12) will start from τ, and the number of sample N will be τ dependent:

$$N(\tau) = \inf \{t \geq 1, S_\tau^t(k) < A_k, \forall k\} \tag{13}$$

As we have discussed before, the functions $F_\tau^t(k)$ will only move toward threshold when its hypothesis is the truth. Thus, if a guess is ahead of the true location, the function will move away from threshold for a while; whereas if the guess is behind the true location, the function will have a late hit to the threshold, as shown in Figure 2(b). Therefore, the exact location will be determined by the function that *firstly* hit the threshold, as:

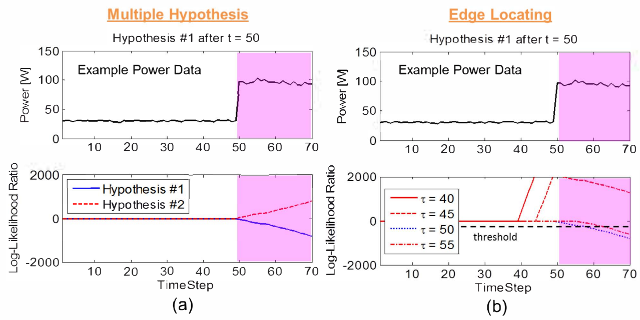

Fig. 2. Demonstration of MSPRT: (a) Log-likelihood function evolution; (b) Edge positioning

$$n = \inf_\tau N(\tau) \tag{14}$$

For Gaussian distribution, the density decays very quickly for outliers. This is not preferable from a numerical standpoint. The log-likelihood function is more promising. Thus, the original formulation is modified as follows:

$$\begin{aligned} N(\tau) &= \inf_{t \geq 1} \left\{F_\tau^t(k) < \chi_k, \forall k\right\} \\ &\approx \inf_{t \geq 1} \left\{ \max_{j \neq k} \sum_{i=t-\tau}^{t} \log \frac{f_j(X_i)}{f_k(X_i)} < \log \frac{\chi_k}{k}, \forall k \right\} \\ &\approx \inf_{t \geq 1} \left\{ \sum_{i=t-\tau}^{t} \max_{j \neq k} \log \frac{f_j(X_i)}{f_k(X_i)} < \log \frac{\chi_k}{k}, \forall k \right\} \end{aligned} \tag{15}$$

The first approximation is to relax the left side of the inequality and transform it into log-likelihood ratio, while the second puts the maximum inside the sum and takes the maximum at each step, hence will make the test robust to noisy data (i.e. "spikes" that frequently appear in power stream data).

The MSPRT originates from the Edge-based model. However, by sequentially considering the density function, MSPRT borrows ideas from the probabilistic HMM and it appears that it combines some of their advantages.

V. MSPRT Virtual-Sensing

The k-hypotheses in MSPRT can be used to test the status of k different appliances. By sequentially applying MSPRT to the power stream, we can find the right hypothesis, hence the right switching appliance. Thus, MSPRT can be used in VS applications. We will discuss this in this section, and compare MSPRT with the HMM and the Edge-based model.

It is also worth noted that to use MSPRT in VS application, we need to know in advance the appliance profiles that connect to the sensor node. This is usually done by learning from a period of ground-truth data. Apart from that, MSPTR doesn't ask for extra parameters than what we need for HMM and edge detection. For situation in which some appliance can have multiple states, the states can be still transformed into separate appliances, which is a similar problem as before.

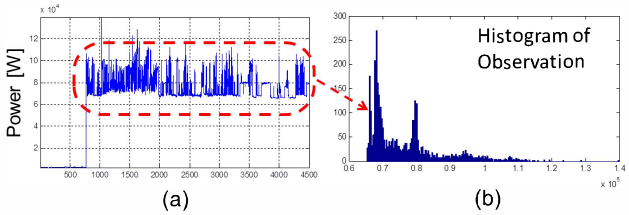

Fig. 3. Impulse Noise in Measured Power Data

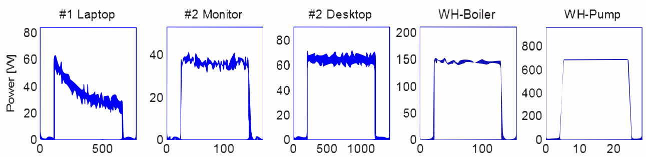

Fig. 4. Simulated power pattern for five devices

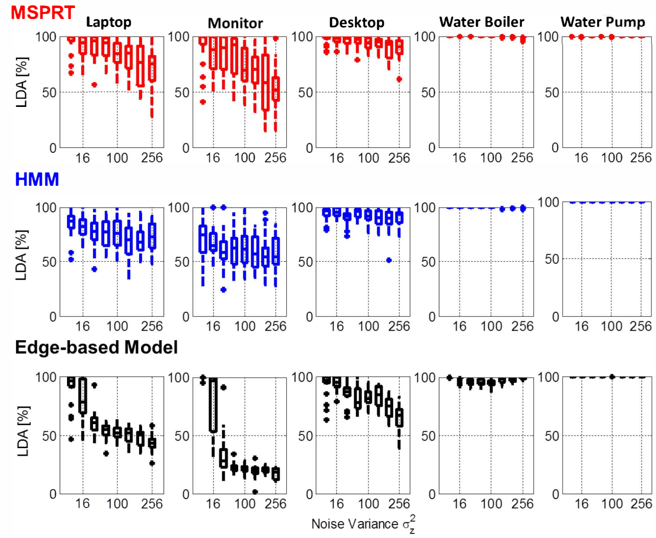

Fig. 5. Monte Carlo Simulated LDA results for the five appliances as a function of Gaussian noise amplitude, under the three models

A. Simulated Power Stream

We use semi-simulated power stream in our analysis. Firstly, we collect a set of real data from measurement. Several meters have been deployed in 550 *Cory Hall* at UC Berkeley collecting power streams of plug-in loads, and some examples can be found in Figure 3(a). Each appliance has its characteristic profile, and some of them are non-stationary, which presents an additional challenge in data modeling. Moreover, some appliances, such as a laptop computer, have a non-stationary pattern, which can be addressed by data post-processing. Noise of the profile is also characterized in Figure 3(b), in which we see effects similar to mixture model. This kind of noise can be modeled by background Gaussian noise plus impulse noise, which will be discussed later.

The data collected from measurement has limited stochasticity and we decide to add white Gaussian noise and/or impulse noise to introduce randomness. By tuning the noise parameters, we can benchmark the potential performance limit of different methods. We performed 50 *Monte Carlo* simulations at each setting of parameters.

There are two evaluation criterions: One is the Detection Error Rate (DER), which is the gap between the detected and the true number of edges, i.e. DER $= \frac{n_{detect}-n_{true}}{n_{true}}$. Another is Load Dis-aggregation Accuracy (LDA). Previously Prec $= \frac{\text{TP}}{\text{TP+FP}}$ (TP: true positive rate, FP: type I error) and Rec $= \frac{\text{TP}}{\text{TP+FN}}$ (FN: type II error) are used to evaluate LDA. In this work we apply a more general term LDA $= \frac{2\text{Prec}\times\text{Rec}}{\text{Prec+Rec}}$. So, the efficacy of the various methods will be judged in terms of achieving low absolute DER and high LDA values.

In our simulation, we include one desktop computer, one computer monitor and one laptop computer, which are the most common appliance in an office building. We also include a water heater with a pump for water filtering. The patterns for the five appliances are shown in Figure 4. Note that non-stationary time series is also considered here, e.g. in the left figure. Non-stationarity definitely bring about extra challenge, and in this work, it is handled by considering the dynamic time series model.

B. Gaussian Additive Noise

There are two groups of study in this section. In the first group we only consider Gaussian random noise, and the data is modeled as $p_t = h(\mathbf{s}_t) + z_t$ with $h(\mathbf{s}_t)$ being the state-dependent clean signal, and z_t being the Gaussian noise with variance σ_z^2. The impact of noise is investigated by tuning σ_z^2 from 1 to 256, based on the measurements. The state duration is modeled as Gamma distributed [11], and we assume that at each step, at most one appliance switches.

There are three methods under study in this section, the MSPRT, the HMM and the Edge-based model. The simulation results for the three methods summarized by showing the LDA in Figure 5, and the DER in Figure 6.

In terms of LDA for the laptop and monitor, there is a drop in LDA above certain noise level for the Edge-based model. The reason is that for fixed sample detection, the expected number of samples needed is following Equ. (9). If this number is over the test sample size (which increases as the level of noise increases), then the changes could be missed. MSPRT adaptively learns the test samples size and HMM tunes itself by introducing state transitions. Thus, they do not have the abrupt drop in LDA, as in Figure 5, though MSPRT is slightly better.

In terms of DER, MSPRT is the best since the state changes only after the edge detected and the sample size can be self-tuned. The edge-based model suffers a sudden increase of DER at high noise levels because it is non-sequential, whereas HMM is worse in DER compared to MSPRT since the state "stickiness" is not well modeled in HMM.

C. Impulse Additive Noise

In the second group, we study the impact of the impulse noise. Here, we model the data as $p_t = h(\mathbf{s}_t) + z_t + \lambda w_t$, where w_t is the impulse noise term with variance $\sigma_w^2 \gg \sigma_z^2$, and $\lambda \in 0, 1$ is a Bernoulli process that models the impulse noise probability. We investigate the impact of impulse noise by varying noise variance σ_w^2 as well as the Bernoulli process

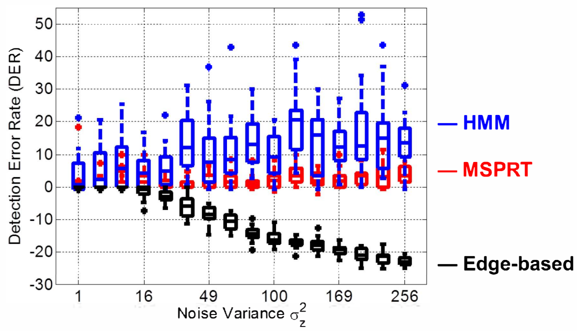

Fig. 6. Monte Carlo simulated DER as a function of Gausian noise amplitude for the three methods under study

probability $\Pr(\lambda = 1)$. Based on our measured data, we set the range of σ_w^2 from 50^2 to 150^2, and set $\Pr(\lambda = 1)$ to be from 0.02 to 0.5. We only focus on MSPRT and HMM here since they give better average performance.

The LDA and DER of MSPRT and HMM are shown in Figure 9 and Figure 10. They have similar performance in terms of LDA and, MSPRT, not surprisingly, has better DER than HMM. However, even for MSPRT, the DER goes beyond 100% as impulse noise level increases.

It is well known that tests assuming a Gaussian distribution are sensitive to outliers or impulses [18]. In the presence of impulse noise, both MSPRT and HMM suffer from degradation caused by the outliers. Therefore, it is necessary to introduce a robust model. This is found to be most convenient for MSPRT as described next.

D. Robust MSPRT

There are several distributions that can model data sets that have either longer than Gaussian tails or are skewed. Examples include the student t-distribution or the Gamma distribution. In this work, inspired by the Huber Robust Loss Function [26], we introduce a robust distribution that has quadratic decay in its main body and linear decay towards its tails. Assuming, without loss of generality, that the data is zero-centered and standardized ($y = \frac{x}{\sigma_k}$):

$$\log f_k \cong -\frac{y^2}{2}\mathbb{1}\{|y| \le \xi\} - \frac{|y| + \xi^2 - \xi}{2}\mathbb{1}\{|y| > \xi\} \quad (16)$$

The normalization coefficient of f_k can be obtained:

$$C = 2\sigma_k \left\{ \sqrt{2\pi}\left(\Phi(\xi) - 0.5\right) + 2e^{-\frac{1}{2}\xi^2} \right\} \propto \sigma_k \quad (17)$$

In which $\Phi(\xi)$ is the cumulative density function (CDF) of the standard Normal Distribution. Thus, the log-likelihood function can be written similar to the Gaussian case ($y_{k(j)} = \frac{x}{\sigma_{k(j)}}$) as $\log \frac{f_j}{f_k}$.

A demonstration of the Robust MSPRT (R-MSPRT) is shown in Figure 7. From Figure 7(a), we see that R-MSPRT is less sensitive to impulse noise. However, from Figure 7(b), we show that R-MSPRT is at the same time less likely to detect *true* changes, though in normal setting. R-MSPRT is responsive enough with respect to ON/OFF switches.

The LDA and DER of the Robust MSPRT method are compared with the plain MSPRT and the HMM in Figures 8-10, and we only focus on the first three appliances in Figure 5. The R-MSPRT gives better LDA compared with the other two methods, and also shows much better DER. Actually, R-MSPRT has DER consistently below 5% and does not suffer much degradation as noise variance increases. This is due to the introduction of a noise which is robust to large deviation. It should be noted that R-MSPRT has similar computation complexity to the ordinary MSPRT.

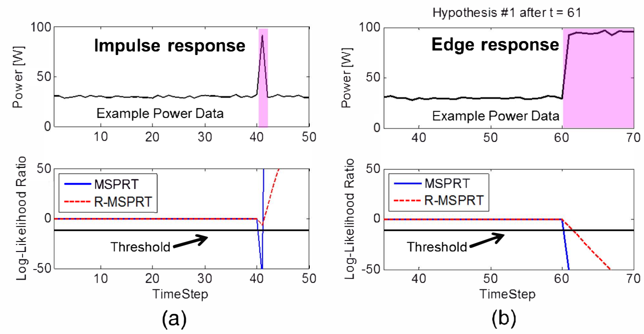

Fig. 7. Demonstration of RMSPRT: (a) Impulse noise response and (b) True edge response

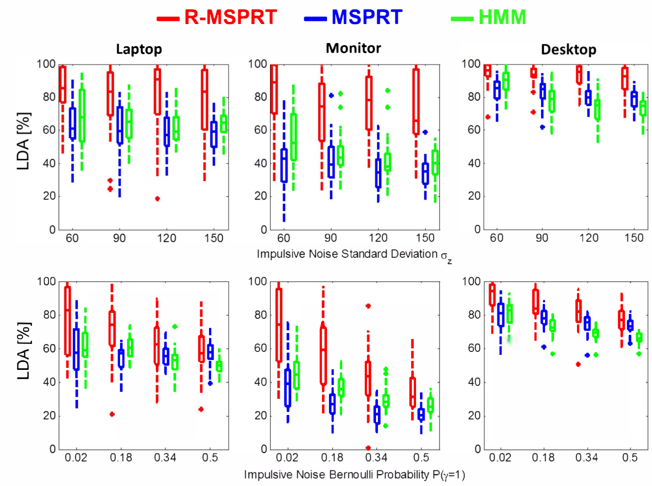

Fig. 8. LDA as a function of impulse noise amplitude and impulse Bernoulli probability for the first three appliances under study, using Monte Carlo simulated data

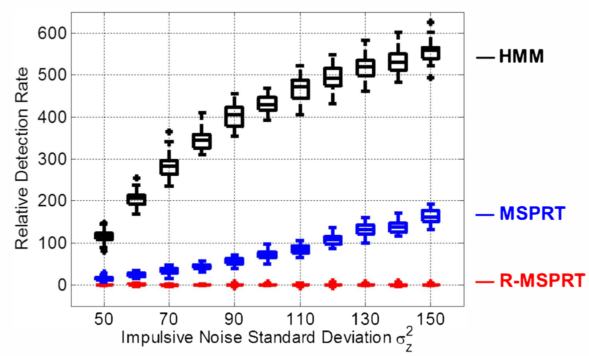

Fig. 9. Monte Carlo simulated DER as a function of Bernouli noise probability showing the efficacy of the Robust noise model

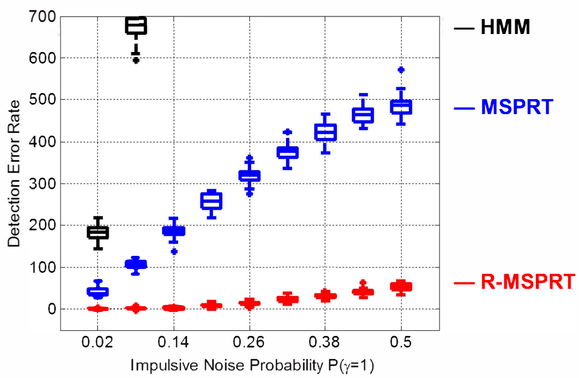

Fig. 10. Monte Carlo simulatede DER as a function of Bernouli noise amplitude showing the efficacy of the Robust noise model.

E. Issues and Discussion

A problem of R-MSPRT is that when the observed data is ambiguous, we might have to process a lot of samples to satisfy the confidence requirement. A decision can be made before we reach certain number of samples by a truncated SPRT [25], which could be subject for a future study.

VI. Conclusion

In this work, the Virtual-Sensing (VS), as a solution to fine-grained power load monitoring, is studied comprehensively. The challenging issues in VS are discussed and two existing state-of-art methods are studied. A Multiple-Hypothesis based method called MSPRT is proposed, which combines the advantages of the two existing methods.

Simulated power signals are used to evaluate the methods with better stochasticity, and LDA and DER are chosen as criterion. A comparative study is done in this context, in which MSPRT shows significant improvement in the disaggregation performance w.r.t. the existing methods. Moreover, a robust version of MSPRT based on robust probability density function is proposed to filter out the impulse noise that is common in real-time plug-in loads monitoring, and demonstrates excellent performance compared with standard MSPRT and other existing methods.

Acknowledgment

This research is funded by the Republic of SingaporeâĂŹs National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

References

[1] S. Chu and A. Majumdar, "Opportunities and challenges for a sustainable energy future," *nature*, vol. 488, no. 7411, pp. 294–303, 2012.

[2] W. Kastner *et al.*, "Communication systems for building automation and control," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1178–1203, 2005.

[3] H. Karl and A. Willig, *Protocols and architectures for wireless sensor networks*. Wiley-Interscience, 2007.

[4] S. Lanzisera *et al.*, "Data network equipment energy use and savings potential in buildings," *Energy Efficiency*, vol. 5, no. 2, pp. 149–162, 2012.

[5] T. Weng *et al.*, "Managing plug-loads for demand response within buildings," in *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. ACM, 2011, pp. 13–18.

[6] S. Gupta *et al.*, "Electrisense: single-point sensing using emi for electrical event detection and classification in the home," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 2010, pp. 139–148.

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

[8] Y. Ephraim and W. Roberts, "Revisiting autoregressive hidden markov modeling of speech signals," *Signal Processing Letters, IEEE*, vol. 12, no. 2, pp. 166–169, 2005.

[9] Y. Wang *et al.*, "Tracking states of massive electrical appliances by lightweight metering and sequence decoding," in *Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data*. ACM, 2012, pp. 34–42.

[10] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Nonintrusive load monitoring using prior models of general appliance types," in *26th AAAI Conference on Artificial Intelligence*, 2012.

[11] H. Kim, "Unsupervised disaggregation of low frequency power measurements," Ph.D. dissertation, University of Illinois, 2012.

[12] J. Kolter and M. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, 2011.

[13] J. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *International Conference on Artificial Intelligence and Statistics*, 2012.

[14] M. Johnson and A. Willsky, "Bayesian nonparametric hidden semi-markov models," *arXiv preprint arXiv:1203.1365*, 2012.

[15] Z. Ghahramani and M. Jordan, "Factorial hidden markov models," *Machine learning*, vol. 29, no. 2, pp. 245–273, 1997.

[16] E. Fox *et al.*, "An hdp-hmm for systems with state persistence," in *Proc. International Conference on Machine Learning*, vol. 2. IEEE Press Piscataway, NJ, 2008.

[17] M. Basseville and I. Nikiforov, *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, NJ, 1993, vol. 104.

[18] P. Bickel and K. Doksum, "Mathematical statistics, volume i," 2001.

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

[20] M. Dong *et al.*, "An event window based load monitoring technique for smart meters," *Smart Grid, IEEE Transactions on*, vol. 3, no. 2, pp. 787–796, 2012.

[21] M. Durling *et al.*, "Cognitive electric power meter," Feb. 18 2009, eP Patent 2,026,299.

[22] J. Duan *et al.*, "Neural network approach for estimation of load composition," in *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, vol. 5. IEEE, 2004, pp. V–988.

[23] J. Liang, S. Ng, G. Kendall, and J. Cheng, "Load signature studyâĂŤpart i: Basic concept, structure, and methodology," *Power Delivery, IEEE Transactions on*, vol. 25, no. 2, pp. 551–560, 2010.

[24] C. Baum and V. Veeravalli, "A sequential procedure for multihypothesis testing," *Information Theory, IEEE Transactions on*, vol. 40, no. 6, 1994.

[25] H. Poor, "An introduction to signal detection and estimation," *New York, Springer-Verlag, 1988, 559 p.*, vol. 1, 1988.

[26] P. Huber, *Robust statistical procedures*. SIAM, 1996, vol. 68.