IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2019, accepted April 22, 2019, date of publication April 29, 2019, date of current version May 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913469

A New Non-Intrusive Load Monitoring Algorithm Based on Event Matching

ZHENGGUANG XU[1], WAN CHEN[1], AND QIFENG WANG[2]
[1]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China
[2]School of Electronical and Information Engineering, Wuhan Donghu University, Wuhan 430212, China

Corresponding author: Qifeng Wang (q.f.wang1978@qq.com)

This work was supported by the Hubei Provincial Natural Science Foundation of China under Grant 2017CFB803.

ABSTRACT Non-intrusive load monitoring is an attractive approach to obtain the energy consumption information without monitoring every device in a building. The main difficulty in the problem is to disaggregate the total power into the power used by specific devices. To solve the problem, this paper proposes an event-based scheme, in which the events corresponding to the appliance on or off are detected and then the events corresponding to the operation of a single device are matched. Finally, the events belonging to a single device are clustered so that the total power is disaggregated into the single device. The experiments are tested on the REDD dataset and the results are compared with the available algorithm, which verifies the validity of the algorithm.

INDEX TERMS Event clustering, event detector, event matching, non-intrusive load monitoring.

I. INTRODUCTION

In today's society, people advocate to save energy. However, for the residents, they only know the total power of the loads rather than the electricity consumption of individual appliances in the home. Knowing specific electricity consumptions is necessary for residents to make better use of electric energy. Non-intrusive load monitoring (NILM) is designed to monitor the electrical circuit that contains a number of devices which switch on and off independently [1]. No access to the individual components is necessary for installing sensors or making measurements. The powers consumed by individual devices in a building are estimated from the measurements of the total power. Utilizing NILM technology, the individual electricity consumption information is obtained only by analyzing the power of the main meter. This can largely save the cost of energy monitoring and facilitate energy conservation.

A good review of NILM is made in [2], where all kinds of NILM approaches are introduced. The NILM approaches could be gathered in two main categories: supervised and unsupervised techniques [2]. The supervised methods require labeled data sets to train the classifier so it would be able to recognize appliance operations from the aggregated load measurement. The supervised methods have shown to perform well for the task of load disaggregation. However, system training requires setting up initial instrumentation, which incurs extra cost and human effort. Compared with the supervised scheme, the unsupervised scheme has lower cost. Therefore, more and more researchers are actively looking to devise unsupervised methods.

The associate editor coordinating the review of this manuscript and approving it for publication was Wen Chen.

According to the sampling rate for the power rate, the unsupervised NILM techniques are different [3]. In this paper, the case of the low-frequency sampling rate is investigated, where the sampling rate is usually as low as 1 to 3 samples per second. Though the information of the low-frequency sampling data is less than that of the high-frequency sampling data, the low sampling scheme requires low-cost hardware. Therefore, it is a more feasible approach, bearing in mind the cost of the solution. In this case, the main work is to detect the appliance state transition (e.g., ON or OFF) from the power measurements. There are two kinds of methods to solve the problem: probabilistic model and event detection.

A widely used probabilistic model for modeling appliance consumption behavior is the hidden Markov model (HMM) [4]: it defines a number of hidden states in which the model can represent the operating condition of the appliance (e.g., ON or OFF states). Different variations of HMM are used such as additive factorial hidden Markov models (FHMMs), additive factorial approximate maximum a-posteriori (AFAMAP) [5], factorial hidden semi-Markov model (FHSMM) [6], and so on. However, the complexity of the HMM models exponentially increases as the number of target appliances increases. Recently, many researchers focus on reducing the computational complexity of the algorithm. Johnson and Willsky used the factorial variant of a hidden semi-Markov model (HSMM) and reduced the number of time indices that need to be considered by running inexpensive change point detection [7]. Zeifman and Roth proposed Viterbi algorithm with Sparse Transitions (VAST) on multiple transition matrices [8]. They took advantage of the sparsity of transitions between appliances' states to simplify the computation. Makonin *et al.* took advantage of sparsity in matrix storage and processing, and presented sparse Viterbi algorithm based on matrix sparsity [9]. They pointed that some states transitions would not make much sense. And we also proposed an event-based nonintrusive load monitoring approach using the simplified Viterbi algorithm [10]. We fully utilize the information contained in the measured aggregated power and simplify the state space by setting thresholds according to the information. Even though all these works are effective in decreasing the complexity of the probabilistic model, the calculation is still very complicated which limits the applicability of this method.

Event detection is another available method. The event detector is based on the fact: the change of steady-state active power measurement from a high to low value can identify whether the appliance is being turned on or off. In the literature, several event detection methods have been proposed, in order to characterize the detected events. The cumulative sum (CUSUM) detector in [11], the goodness-of-fit (GOF) detector in [12] and the generalized likelihood ratio (GLR) detector in [13] are three commonly used event detectors. The CUSUM detects the difference power of the detection window and the average power of the background window. The GOF and GLR detectors seek to determine whether the detection window data has the same distribution with the background window data. The common ground of the three algorithms is that the window sizes of the reference set and the current set have to be fixed.

Some event-based NILM approaches have been investigated in [14] and [15]. In [14], a supervised NILM framework is proposed, which consists of the following steps: training, event detection, feature extraction, classification, and energy estimation. After event detection, features (e.g., rising/falling edge, duration) are extracted to classify the events into predefined categories, each corresponding to a known appliance. In this way, the features should be labeled in the training procedure, but the training procedure has high cost in the practical applications. In [15], an unsupervised motif mining approach is proposed to identify recurring events referred to as episodes that are basically the on/off operation for the devices. That temporal motif mining approach consists of six stages: baseline removal, steady states extraction, episode mining and selection, probabilistic sequential mining, time-based motif mining, and device recovery. This approach tries to identify repetitive sequences of power level changes within the range of all events to select episodes that potentially correspond to the operation of a single device in the episode mining stage. After this stage, it selects those most possible episodes by clustering power levels. The core idea in [15] is to match the "On" and "Off" events by three constrain conditions. However, these matching criterions are coarse and the event matching may be wrong in some special cases.

Motivated by the scheme in [15], we propose some new matching conditions and utilize the iterative operation to reduce the matching complexity. Compared with the available schemes in [10], [14], and [15], the proposed scheme has some new features. In [10], the hidden Markov model (HMM) is used to describe the relationship of the events and the simplified Viterbi algorithm is proposed to solve the parameters in the HMM. However, the complexity of the HMM is high, so that the calculation is still complicated. In the proposed scheme, the event matching algorithm is proposed to find the relationship of the "On" and "Off" events, so the calculation process is reduced. In [14], the events are classified according to the event features, which should be labeled by hand. In our method, the events are matched by the event parameters without manual intervention, which is easy to implement. The differences between the motif mining approach [15] and the proposed approach in the paper are that our event matching method utilizes iterative ideas and different matching conditions. We match the events by the fixed matching gap using four matching conditions, and put those unmatched events into the event set for the next iterative step with the increased gap. In this way, the computational complexity of event matching could be reduced.

II. PROPOSED ALGORITHM

Our framework is shown in Fig. 1. The whole algorithm contains event detection, event matching and event clustering. The three modules are described in the following subsections.

FIGURE 1. The proposed algorithm framework.

The main contribution is the design of the event matching algorithm. The aim of the event matching is to match the "On" and "Off" events of the same power appliance exactly. When the start event and the ending event are matched, the start/ending events are corresponding to the On/Off operations of the single device. Therefore, the power consumption of the device could be calculated from the work time and the work power, which is the import of the event matching. However, the complexity of the appliance behavior and the large variance of the measurement readings may affect the matching accuracy. In order to adapt to the complexity of appliance, we examine the pair of three events besides the pair of two events, and propose the start/ending power matching of the event besides the rising/falling power in the matching algorithm. In order to reduce the interference introduced by the measurement noise, we consider the relative error of the power besides the absolute error in the matching criterion. In order to improve the accuracy of the matching algorithm, we also propose an iterative process to match the event. In the description of the algorithm, we will use an example to show how these methods operate in the matching process.

A. EVENT DETECTION

Usually, the appliance state changes are accompanied by power changes and often occur near the maximum or minimum points of the total power curve. Based on this, the new event detection algorithm based on the maximum and minimum points (MMP) is proposed [10]. The MMP algorithm consists of three steps. First, search the extreme points of the aggregated power data. Then, determine the rising interval and falling interval according to the extreme points. In this step, we set a power threshold to find possible events of which the power changes are higher than a fixed value. Lastly, search the rising and falling edges from the rising interval and the falling interval. In order to eliminate the noise interference and slow power changes, we set a noise threshold.

Although some works are done for reducing the noise, a few mistakes may still exist in the event detector, e.g. some events occur but are not detected, and some false events are detected. These mistakes should be dealt rightly by the following event matching algorithm.

B. EVENT MATCHING

The goal of event matching is to identify repetitive sequences of power level changes and select those events that potentially correspond to the operation of a single device. The main procedure of the event matching algorithm is as follows:

(1) Each unmatched rising event is put into the rising set, while each unmatched falling event is put into the falling set. The matching gap n is set to 0, which means the event number between the rising event and the falling event.

(2) Each falling event tries to match the rising event before the falling event with matching gap n using the match conditions. If the events are matched, the event pair will be deleted from the event sets. After the deleting operation, the matching process continues with the gap n until no event pairs are found.

(3) Set $n = n+1$ and repeat step 2. When all the events are paired, the event matching process is completed.

Now we explain the algorithm by an example to show the running process. A typical power time series is shown in Fig. 2. There are three rising edges e_1, e_2 and e_3 and three falling edges e_4, e_5 and e_6 in the unmatched event set. In the first round of the iterative process, the matching algorithm searches the adjacent rising and falling edge, which satisfy the matching criterions. Obviously, e_3 and e_4 are adjacent and satisfy the matching criterion about the rising/falling power.

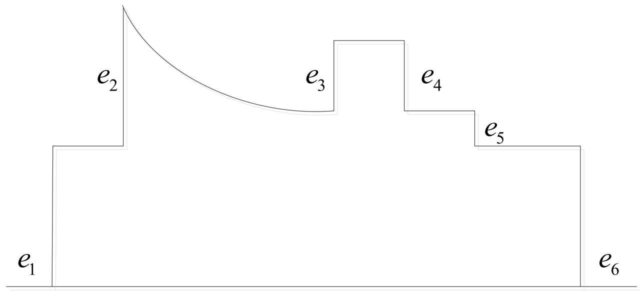

FIGURE 2. **An example of power time series.**

Therefore, the events e_3 and e_4 are matched and deleted from the event set. Then, e_2 and e_5 are adjacent. In the second round, e_2 and e_5 are matched since the matching criterion about the start/end power is satisfied. Finally, e_1 and e_6 are matched in the third round. When all the adjacent edges are matched, the iterative algorithm will try to match the rising/falling edges among which there is one edge. When all the edges are matched, the iterative process is finished. Note that e_2 and e_6 also satisfy the matching criterion about the rising/falling power, but e_2 just matches e_5 in the proposed algorithm, since the adjacent edges are considered firstly in the iterative matching process. In the proposed algorithm, once the falling event matches the adjacent rising event, the event pair will be deleted from the event set and the search range of the remaining events will be reduced. Therefore, the adjacent matching can reduce the computational complexity.

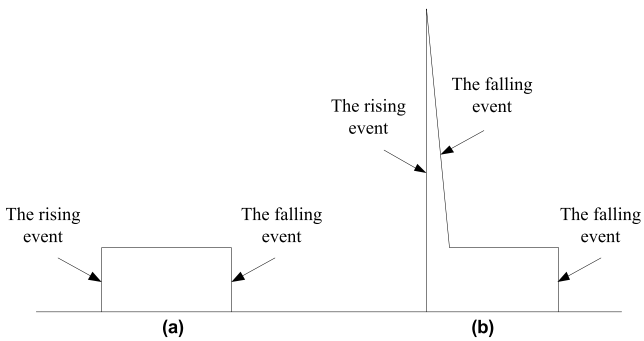

FIGURE 3. **Two cases for the event matching. (a) The pair of two events. (b) The pair of three events.**

Specially, the match conditions can be divided into two cases: two-event matching and three-event matching. These two cases are shown in Fig. 3. The pair of two events shown in Fig. 3(a) is the common case for the application, e.g., ovens, lights, etc. The pair of three events shown in Fig. 3(b) represents the impulse case for the application, e.g., refrigerator. In this case, the appliance has an impulse when the appliance starts and then the power of the appliance falls into the normal level. Therefore, the rising event is always adjacent to the first falling event in the three-event matching.

Now the match condition for the pair of two events is discussed. Supposing that the start power of the rising event is P^+, the power change of the rising event is ΔP^+, the end power of the falling event is P^-, and the power change of the falling event is ΔP^-, the events are matched if one of the four conditions are satisfied:

$$\begin{aligned} &(1)\ \left|P^+ - P^-\right| < \varepsilon \\ &(2)\ \frac{\left|P^+ - P^-\right|}{\min\left(P^+, P^-\right)} < \eta \\ &(3)\ \left|\Delta P^+ - \Delta P^-\right| < \varepsilon \\ &(4)\ \frac{\left|\Delta P^+ - \Delta P^-\right|}{\min\left(\Delta P^+, \Delta P^-\right)} < \eta \end{aligned} \tag{1}$$

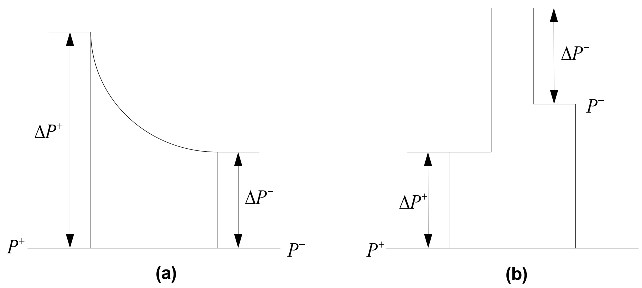

FIGURE 4. Two cases for the matching conditions. (a) Match for the start-end power. (b) Match for the power changes.

The first and second conditions are designed for the case shown in Fig. 4(a), where the power changes of the rising event and the falling event are different since the slow power change after the rising event cannot be detected by the event detector. In this case, the start power and the end power are the same, so the pair of the events represents the behavior of the same appliance. The parameter ε represents the absolute power error and the parameter η represents the relative power error. The absolute error condition is more effective for the low power cases, while the relative error condition is more effective for the high power cases. The third and fourth conditions are designed for the case shown in Fig. 4(b), where a rising event exists between the right pair of rising and falling events. In this case, the start and the end powers are different since another appliance turns on, but the power changes of the appliance in the two events are the same. Therefore, the two events could be matched by the power changes.

The case for the pair of three events is similar. The start power P^+ and the power change ΔP^+ of the rising event are defined in the same way, but the end power P^- is defined as the end power of the second falling event and the power change ΔP^- is defined as the sum of the power changes of the two falling events. In this way, the match condition for the three events is the same as that for the two events. Then, the case shown in Fig.3 (b) could be dealt with by the matching condition in (3).

Actually, there exist some similarities and differences between the event matching algorithm and the event-based algorithm using the simplified Viterbi which exploits the ADFHMM as the load model [10]. Here, we use an abbreviation, ES-ADFHMM, to represent that algorithm. First, both of the two approaches exploit the differential and the aggregated powers. The latter uses both two power data to estimate possible states each time the event happens. Differently, the former uses only one kind of power data each time event happens. For instance, when the case shown in Fig. 4(a) happens, the event matching algorithm only exploits the aggregated power, P^+ and P^-. Detect if they satisfy the condition (1) and (2) listed in formula (1). If satisfy, then the pair of the falling event and the rising event belongs to one appliance. If not, then the match fails. When the case shown in Fig. 4(b) happens, the event matching algorithm just use the differential power, ΔP^+ and ΔP^-. If the two values satisfy the condition (3) and (4) listed in formula (1), the match succeeds. That means the two events belong to one appliance. If not, the match fails. Second, the ES-ADFHMM algorithm uses the Viterbi. It selects several possible states from all combined states instead of giving a certain one each time in the aim to considering the whole time chains. However, once the event matching algorithm finds the falling event, it goes to the matching step and gives a certain result, success matching or failed matching. To a certain extent, the event matching algorithm further simplified the disaggregated work of the ES-ADFHMM.

To illustrate the computational complexity more conveniently, we assume that all the appliances in the house have only two states (ON and OFF). Assuming N is the number of the power sampling points, the computational complexity of the event detector is $O(N)$. Assuming M is the number of the events. and P is the number of the appliances.

For the matching algorithm, the matching complexity of event pair of two should be $O\left(M^2\right)$ in the basically exhaustive search. However, we utilize the adjacent matching and iterative search to reduce the search process. We define the event gap as n, which means the event number between the rising event and the falling event. For three-event matching, the event gap n means the event number between the first falling event and the second falling event, since the rising event and the first falling event are always adjacent. In each iterative process, the falling event matches the rising event with the gap n, and the matching complexity of event pair is $O(M)$. Suppose the iteration number is K, and the matching complexity of event pair is $O(KM)$. Obviously, K is less than M and K is close to M in the worst case. However, K is much less than M in our data experiment.

Proposed in [10], the computational complexity of the traditional Viterbi could be $O\left(2^P(P+1)M\right)$. The ES-ADFHMM algorithm decreases this computational complexity by simplifying the state space to a certain extent. However, according to the experiment results shown in [10], the calculation of the ES-ADFHMM algorithm is still more complex than the event matching algorithm.

C. EVENT CLUSTERING

In the above step, the event pairs have been found. Each event pair represents the action for one appliance, so that the total power could be disaggregate into the individual power for each appliance. The event clustering is used to cluster the event pairs that are produced by the same appliance, and then the power consumed by the appliance could be calculated. Given the true power of the appliances in a house, the event clustering process is to calculate the power difference between the power change of the event pair and the true power of the appliance one by one. Then the event pair belongs to the appliance that has the minimal power difference. In this way, each event pair could be dispatched to an appliance.

III. EXPERIMENTS ON REDD DATASET

A. EVENT DETECTION RESULTS

Many metrics for measuring the accuracy of event detection have been proposed in the past literatures. In this paper, the metrics described in [13] are used. Some notations are defined as follows. E is the number of events. TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives or misses.

1) TRUE POSITIVE RATE $\hat{\psi}_{\mathrm{Rate}}$

TPR represents true positive rate as follows

$$TPR = \frac{TP}{TP+FN} \in [0,1] \tag{2}$$

TPR represents false positive rate as follows

$$FPR = \frac{FP}{FP+TN} \in [0,1]. \tag{3}$$

For any event detector, the indexes TPR and FPR are contradictory. When we change the parameters of the detectors, TPR becomes better while FPR becomes worse. We wish to find the parameter ψ which makes TPR close to 1 and FPR close to 0, so we define the metric

$$\|(0,1)-(FPR,TPR)\| = \sqrt{FPR^2+(1-TPR)^2}.$$

If ψ makes the metric least, we choose the parameters in the event detector and obtain the least $\hat{\psi}_{Rate}$:

$$\hat{\psi}_{Rate} = \min_{\psi\in\Psi} \|(0,1)-(FPR,TPR)\|^2. \tag{4}$$

2) TRUE POSITIVE PERCENTAGE $\hat{\psi}_{pere}$

In this metric, the percentage of events correctly detected and the ratio of false positives to total number of actual events E are compared. The smaller $\hat{\psi}_{pere}$ is

$$\begin{aligned}\hat{\psi}_{Pere} &= \min_{\psi\in\Psi} \|(0,E)-(FP,TP)\|^2\\ &= \min_{\psi\in\Psi} \|(0,0)-(FP,FN)\|^2\end{aligned} \tag{5}$$

3) TOTAL POWER CHANGE $\hat{\psi}_{\Delta P}$

The first two metrics mentioned above considered only the relative number of true and false positives without taking into account the power. This metric incorporate information about the sum power changes of all the misses and all the false positives. Given an edge e, define its power change ΔP_e as follows:

$$\Delta P_e = \frac{1}{w_3}\sum_{i=e+w_2+1}^{e+w_2+w_3} P(i) - \frac{1}{w_1}\sum_{i=e-w_1}^{e-1} P(i) \tag{6}$$

where w_1, w_2 and w_3 are window length. w_1 and w_3 are used for calculating the pre-and post-event means and w_2 is used to allow a delay for the transient to end and reach a more steady state. It is the difference of the mean of the signal a short time after the event and just before. M is defined as the set of all misses and F is defined as set of all false positives. The total power changes for the misses and false positives are ΔP_M and ΔP_F respectively

$$\Delta P_M = \sum_{m\in M} |\Delta P_m|, \tag{7}$$

$$\Delta P_{FP} = \sum_{f\in F} |\Delta P_f|. \tag{8}$$

Therefore,

$$\hat{\psi}_{\Delta P} = \min_{\psi\in\Psi} \|(\Delta P_{FP},\Delta P_M)\|^2. \tag{9}$$

4) AVERAGE POWER CHANGE $\hat{\psi}_{\Delta\bar{P}}$

This metric aims at minimizing the average power of misses and positives. The average power for the misses $\Delta\bar{P}_M$ and the average power for false positives $\Delta\bar{P}_M$ are described in equations (10) and (11)

$$\Delta\bar{P}_M = \frac{1}{|M|}\sum_{m\in M} |\Delta P_m|, \tag{10}$$

$$\Delta\bar{P}_{FP} = \frac{1}{|F|}\sum_{f\in F} |\Delta P_f|. \tag{11}$$

Therefore,

$$\hat{\psi}_{\Delta\bar{P}} = \min_{\psi\in\Psi} \|(\Delta\bar{P}_{FP},\Delta\bar{P}_M)\|^2. \tag{12}$$

Details about these metrics can be found in [13].

In (4) the collection Ψ is the parameter set and $\hat{\psi}_{Rate}$ is the value of $\|(0,1)-(FPR,TPR)\|^2$, which measures the proximity of FPR to 0 and TPR to 1. The other metrics $\hat{\psi}_{Pere}$, $\hat{\psi}_{\Delta P}$ and $\hat{\psi}_{\Delta\bar{P}}$ are defined in the similar way. Obviously, for the same detector the parameter ψ may be different in (4), (5), (9) and (12), so we just choose some proper ψ for tradeoff in the four metrics. In this paper, four event detection algorithms are tested on one week data of house1 in REDD dataset [16]. The selected parameters ψ for each detector are shown in Table 1.

Using metrics mentioned above to evaluate and compare the results of four detectors. Test four algorithms on main meter and use the real events obtained from sub meters as a reference. The number of miss and false detections are given in Table 2. Table 3-6 depict the results of these detectors. As can be seen from these tables, the proposed detector, labeled as MMP detector, shows the best performance.

TABLE 1. Parameter settings.

Algorithm	Parameters
CUSUM	$\lambda_1 = 0.8, \lambda_2 = 2, \Delta_{\min} = 50, T_{\max} = 9, N_m = 2, N_d = 4$
GLR	$w_{vote} = 4, w_{pre} = w_{post} = 2, P_{thr} = 50, v_{thr} = 2$
GOF	$n = 3, \alpha = 0.05$
MMP	$G_p = 50, G_n = 20$

TABLE 2. Statistical test results.

Algorithm	Events	FN	FP
CUSUM	782	25	94
GLR	744	28	98
GOF	892	23	144
MMP	861	22	82

TABLE 3. True positive rate.

Algorithm	TPR	$FPR(10^{-3})$	$\hat{\psi}_{Rate}$
CUSUM	0.96423	0.74425	0.03577
GLR	0.95994	0.77592	0.04007
GOF	0.9671	1.1401	0.03292
MMP	0.96853	0.64924	0.03148

TABLE 4. True positive percentage.

Algorithm	FN	FP	$\hat{\psi}_{Pere}$
CUSUM	25	94	97.268
GLR	28	98	101.92
GOF	23	144	145.83
MMP	22	82	84.900

TABLE 5. Total power change.

Algorithm	ΔP_M	ΔP_F	$\hat{\psi}_{\Delta P}$
CUSUM	5705.5	3677.3	6787.9
GLR	11521	5072.5	12588
GOF	7505.8	10940	13268
MMP	1627.9	95.12	1630.7

B. ENERGY DISAGGREGATION RESULTS

The experiments are conducted on the low frequency data from the REDD dataset [16]. We focus on House 1 since it

TABLE 6. Average power change.

Algorithm	$\Delta\bar{P}_M$	$\Delta\bar{P}_F$	$\hat{\psi}_{\Delta\bar{P}}$
CUSUM	228.22	39.12	231.55
GLR	411.46	51.76	414.70
GOF	326.34	75.975	335.07
MMP	73.996	1.16	74.005

is also used in [15]. In all, there are 18 devices but 4 of them are seldom used; and, thus the remaining 14 devices can be disaggregated by the proposed methods. The parameters of the proposed algorithm is $G_p = 50$, $G_n = 20$, $\varepsilon = 30$ and $\eta = 0.1$. The proposed event matching algorithm is compared with the motif mining algorithm in [15], and the metrics for measuring the accuracy of the power disaggregating are Precision, Recall and F-Measure, which are defined as follows

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F - Measure = \frac{2Precison \times Recall}{Precision + Recall} \tag{15}$$

Table 7 lists the results of the comparison between the proposed event matching algorithm and the motif mining algorithm of which the result we quote is in [15]. For high power consumption devices, such as oven1&2, microwave, bathroomgfi, kitchen outlet1 and washdryer2, the F-Measure index of the proposed algorithm is larger than 0.7. In nearly all the F-Measure of the appliance, the proposed algorithm performs better than the motif mining algorithm. Only for high frequency devices, such as the refrigerator, motif mining performs better than the proposed algorithm, but the difference is small. Compared with the constraint of the motif mining approach, power state changes in a device are assumed to be greater than a support threshold, the event matching method is more comprehensive. For example, when using the motif mining approach, if a support threshold of 0:1 is used, the episode (1000,-850,-90) will get disqualified, and then the events in this episode may be neglected in later work. This may cause many events to be ignored. While in the propose event matching approach, we consider all unmatched events in the unmatched event set for each matching process to consider as many detected events as possible. This makes our approach more accurate.

We also compare this event matching algorithm with the ES-ADFHMM algorithm [10]. In our experiments, both of the algorithms are applied to all house in the REDD dataset. We use the same power data of each house and the same rule to select loads as the ES-ADFHMM algorithm. For the loads with low power levels or the loads infrequently used, we do

TABLE 7. Comparing the proposed algorithm against motif mining on the REDD dataset.

device	Power (W)	Event matching Precision	Event matching Recall	Event matching F-Measure	Motif mining Precision	Motif mining Recall	Motif mining F-Measure
oven 1& 2	4000	0.7810	0.9999	0.8770	0.9297	0.5209	0.6677
refrigerator	193	0.7201	0.9844	0.8318	0.9759	0.7368	0.8396
dishwasher	1113; 900; 400; 200	0.6159	0.5483	0.5801	0.9786	0.2858	0.4423
kOutlets3	100; 60	0.2701	0.3553	0.3069	0.1487	0.0318	0.0524
light3	282 90	0.3231	0.9429	0.4813	0.5768	0.1349	0.2187
washdryer1	466; 50	0.3504	0.8669	0.4991	0.1789	0.1236	0.1462
microwave	1527	0.5806	0.9983	0.7342	0.8035	0.3799	0.5158
bathroomgfi	1605	0.9588	0.9963	0.9771	0.5199	0.6815	0.5898
kOutlet1	1076	0.9406	0.9998	0.9693	0.9320	0.6997	0.7993
kOutlet2	1535	0.4105	0.9960	0.5813	0.2233	0.6261	0.3292
light1	64	0.3100	0.9659	0.4694	0.6199	0.1963	0.2981
light2	53	0.2758	0.9510	0.4275	0.2603	0.1404	0.1824
washdryer2	2711	0.9154	1.0000	0.9558	0.9563	0.8305	0.889
mean		0.5733	0.8927	0.6685	0.6234	0.4145	0.4593

TABLE 8. Comparing the proposed algorithm against ES-ADFHMM on the REDD dataset.

House	Event matching Precision	Event matching Recall	Event matching F-Measure	Event matching Run time(s)	ES-ADFHMM Precision	ES-ADFHMM Recall	ES-ADFHMM F-Measure	ES-ADFHMM Run time(s)
1	0.5733	0.8927	0.6685	0.1270	0.7088	0.9226	0.7919	448.3412
2	0.7251	0.9301	0.8065	0.0358	0.7272	0.9057	0.7775	11.6577
3	0.6323	0.8353	0.6479	0.0614	0.7856	0.8606	0.8091	259.6612
4	0.6999	0.7890	0.7338	0.1284	0.6461	0.8776	0.7403	45.0362
5	0.6192	0.8825	0.6941	0.0084	0.7716	0.8824	0.8067	14.0568
6	0.6716	0.9689	0.7484	0.0284	0.8458	0.9607	0.8911	12.421
mean	0.6536	0.8831	0.7165	0.0671	0.7475	0.9016	0.8028	131.8624

not consider. For example, in house1, 4 devices (channel number: 13, 14, 16, 19) are seldom used, so we do not include them in the decomposition work. In house 2, the power of the washer dryer (channel 7) is almost lower than 10W, so we only consider the other 8 devices. Similarly, we also select part of loads in the other houses for experiments according to the above rules.

The results for each house are given in Table 8. Besides the mean value of F-Measure, Precision and Recall, we also give the run time of the two algorithms. As seen from Table 8, in house 1, 3, 5, 6, the event matching algorithm performs worse than the ES-ADFHMM. This is because the ES-ADFHMM considers the whole time period instead of only the current state transition, so it shows better performance than the event matching algorithm. In house 2 and house 4, the results of both approaches are close. This is because the combinations of loads are different in those houses, and the events produced by loads are different as well. So for the event matching algorithm, its performance varies in different load environments. But mean value of F-Measure is higher than 0.65 in all houses.

Particularly, we also record the run time of the two algorithms. Due to the configuration problem of the computer itself, there is a certain error between the time we record and the actual running time. But we use the same computer with the same configuration and exploit the same data to run the two algorithms, so it is still proper to use the run time of them to compare their computational complexity. For the ES-ADFHMM, the time it costs is related to the number of loads and the events occurred. And if the power levels of several loads are close, the ES-ADFHMM need to consider more states within the same threshold when event happens, and then it costs more time. There are more loads in house 1 and house 3, so the run time is longer than the other houses. As for the event matching method, it only gives one matching result when any event occurs. So it costs much less time than the ES-ADFHMM. And because the event matching costs very little time in the decomposition work, it is more easily influenced by the configuration of the computer. Shown in Table 8, the proposed event matching algorithm achieves the load disaggregation goal much faster than the ES-ADFHMM. That proves this approach actually has relatively low computational complexity when compared with the ES-ADFHMM.

IV. CONCLUSION

An intuitive event matching approach to disaggregate the energy has been investigated. The proposed algorithm performs well relative to the available disaggregation based on event detection. And it has low computational complexity and high disaggregation accuracy, so that this algorithm could be used in the practical appliance.

REFERENCES

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.
[2] 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. 16838–16866, Dec. 2012.
[3] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Trans. Consum. Electron*, vol. 57, no. 1, pp. 76–84, Feb. 2011.
[4] T. Zia, D. Bruckner, and A. Zaidi, "A hidden Markov model based procedure for identifying household electric loads," in *Proc. 37th Annu. Conf. IEEE Ind. Electron. Soc.*, Melbourne, VIC, Australia, Nov. 2011, pp. 3218–3223.
[5] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proc. Int. Conf. Artif. Intell. Statist.*, Canary, Spain, Mar. 2012, pp. 1472–1482.
[6] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. Soc. Ind. Appl. Math. Conf. Data Mining*, Mesa, AZ, USA, Apr. 2011, pp. 747–758.
[7] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-Markov models," *J. Mach. Learn. Research*, vol. 14, no. 1, pp. 673–701, Feb. 2013.
[8] M. Zeifman and K. Roth, "Viterbi algorithm with sparse transitions (VAST) for nonintrusive load monitoring," in *Proc. IEEE Symp. Comput. Intell. Appl. Smart Grid (CIASG)*, Paris, France, Apr. 2011, pp. 1–8.
[9] S. Makonin, F. Popowich, I. V. Baji, B. Gill, and L. Bartram, "Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2575–2585, Nov. 2016.
[10] T. Lu, Z. Xu, and B. Huang, "An event-based nonintrusive load monitoring approach: Using the simplified Viterbi algorithm," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 54–61, Oct./Dec. 2017.
[11] S. R. Mohanty, A. K. Pradhan, and A. Routray, "A cumulative sum-based fault detector for power system relaying application," *IEEE Trans. Power Del.*, vol. 23, no. 1, pp. 79–86, Jan. 2007.
[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 *Proc. 2011 IEEE Int. Conf. Acoust., Speech Signal Process.*, Prague, Czech Republic, May 2011, pp. 4340–4343.
[13] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event detection for non intrusive load monitoring," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc.* Montreal, QC, Canada, Oct. 2012, pp. 3312–3317.
[14] S. Giri and M. Bergés, "An energy estimation framework for event-based methods in non-intrusive load monitoring," *Energy Convers. Manage.*, vol. 90, pp. 488–498, Jan. 2015.
[15] H. Shao, M. Marwah, and N. Ramakrishnan, "A temporal motif mining approach to unsupervised energy disaggregation," in *Proc. 1st Int. Workshop Non-Intrusive Load Monitor.*, Pittsburgh, PA, USA, May 2012, pp. 1327–1333.
[16] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. SustKDD Workshop Data Mining Appl. Sustainability*, San Diego, CA, USA, Aug. 2011, pp. 59–62.

ZHENGGUANG XU received the bachelor's degree in communication engineering, the master's degree in communication and information system, and the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology, China, in 2004, 2006, and 2009, respectively, where he is currently an Associate Professor with the School of Electronic Information and Communications. His research interests include signal processing, wireless communication, and data analysis.

WAN CHEN received the bachelor's degree in communication engineering from the Huazhong University of Science and Technology, China, in 2018, where she is currently pursuing the master's degree with the School of Electronic Information and Communications. Her current research interests include big data and signal processing.

QIFENG WANG received the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology, China, in 2016. He is currently with the School of Electronical and Information Engineering, Wuhan Donghu University, China. His current research interests include signal processing, parameter estimation, communication system simulation, and cognitive radio.