A Fully Unsupervised Appliance Modelling Framework for NILM

Bo Liu, Wenpeng Luan, *Senior Member, IEEE* , Yixin Yu*, *Senior Member, IEEE*

***Abstract*—Most of the existing Non-Intrusive Load Monitoring (NILM) technologies need to measure the appliances to get the relevant information to build appliance model before they can be put into use, which limits the technological advantages and implementation. In this paper, we proposed a fully unsupervised appliance finite state machine (FSM) modelling framework for NILM. Without intruding into the internal load or requiring any priori knowledge of the appliance model, the framework can automatically establish the FSM model for appliance including complete state set of FSM, topological structure, and related model parameters, only based on load event signatures. It can significantly improve the applicability of NILM technologies, and provide basis for the realization of auto-setup NILM.**

***Keywords*- NILM; FSM; Appliance Modelling**

I. Introduction

Non-intrusive Load Monitoring (NILM) is a novel technique for monitoring the details of load consumption [1]. Building the appliance load model is the basis of implementing NILM. So far, the finite state machine (FSM) model has been commonly used for appliance modelling. There exist two main ways to obtain the FSM model: (1) to obtain a complete set of labeled appliance load signature samples, and then determine the parameters of appliance FSM model by the supervised parameter learning methods [2]; (2) to learn the state set and the state transition topology of the FSM model of the appliance, and then use the unsupervised parameter learning methods to estimate the model parameters from the consumption data of the total load [3]. They all need to intrude into the internal load to get the relevant information used to build appliance model, which limits the applicability of NILM technologies. To this end, researchers have been carrying out researches on completely unsupervised appliance FSM modelling for NILM.

For the fully unsupervised appliance FSM modelling, initially, Hart (in 1992) proposed an overall idea of automatic FSM modelling with the use of *Zero Loop-Sum Constraints* (ZLSC) and *Uniqueness Constraints* (UC) two heuristic rules[1]. Baranski and Voss (in 2004) took the load event sequence as the analysis object, used genetic algorithm (GA) and dynamic programming algorithm to extract the operating state transition sequential patterns of appliances based on the results of event clustering. However, genetic algorithm may not reach the global optimum, and eventually the learned state transition sequential patterns are still needed to be combined into the FSM model [4]. The method proposed by Streubel and Yang (in 2012) for appliance FSM modelling can only use the consumption data generated by a single appliance instead of by the total load [5]. Parson et al (in 2014) used the generic model (considered as a kind of priori knowledge in this paper) of a certain type of appliances to learn the specific load model of the appliances of the same kind inside the total load from the data collected from the target scenes, however, acquiring a generic load model can not avoid the requirement of supervised modelling [6].

In this paper, we proposed a fully unsupervised appliance finite state machine (FSM) modelling framework for NILM. Without intruding into the internal load or requiring any priori knowledge of the appliance model, the framework takes the consumption data of the total load as processing objects, automatically establishes the appliance FSM model including the complete state set of FSM, topological structure, and related model parameters, only according to the load event signatures; so it can completely avoid the supervised modelling and greatly improve the applicability of existing NILM technologies, and lay a foundation for the realization of the auto-setup NILM.

This paper is organized as follows. Firstly, the basic principle and flow chart of the proposed framework are outlined and presented in Section II. Then the implementation methods are described in details in Section III. After that the proposed framework is tested on the real measured dataset to validate its effectiveness in Section IV. Finally, conclusion of this paper and prospects for future work are briefed in Section V.

II. Framework Overview

The fundamental task in building appliance finite state machine (FSM) is to establish the model topology structure (MTS), and determine the model parameters. The MTS can be expressed by the model topology graph (MTG), as shown in Fig.1. In the MTG, the simple cycle is a closed walk with no repetitions of vertices and edges, except for the repetition of the starting and ending vertex [7]. In this paper, the load event sequence generated by any operating state transition process which corresponds to a simple cycle of the MTG whose number of vertices is more than 1 is called simple cycle event sequence (SCES). Building the MTS of an appliance can be completed through detecting and combining all the different SCES patterns that may be generated by the appliance, from the total load consumption data. Based on this, the FSM model parameters can be further derived by the statistical

Manuscript received March 15, 2016.

Bo Liu is with School of Electrical and Automation Engineering, Tianjin University, No.92 Weijin Road, Nankai District, Tianjin 300072, China (e-mail: liubo@tju.edu.cn).

Wenpeng Luan is with China Electric Power Research Institute, Haidian District, Beijing 100192, China (e-mail: luanwenpeng@epri.sgcc.com.cn).

Yixin Yu is the Corresponding Author with School of Electrical and Automation Engineering, Tianjin University, No.92 Weijin Road, Nankai District, Tianjin 300072, China (tel: +86-13602198158; fax: +86-022-27892809; e-mail: yixinyu@tju.edu.cn).

analysis of the data and information of the load events related to each SCES pattern in the appliance MTS.

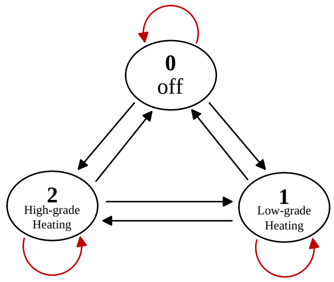

Fig.1. The FSM model topology graph of a three-state hair drier

In practical, each consumer has a certain energy consumption habits, thus the typical SCES pattern of the appliances inside the load can repeat several times in a period of time (a week or even a day). For instance, the start-and-stop cycle of the refrigerator will happen many times in one day. Using the frequent pattern mining techniques, we established a fully unsupervised appliance FSM modelling framework, as shown in Fig.2.

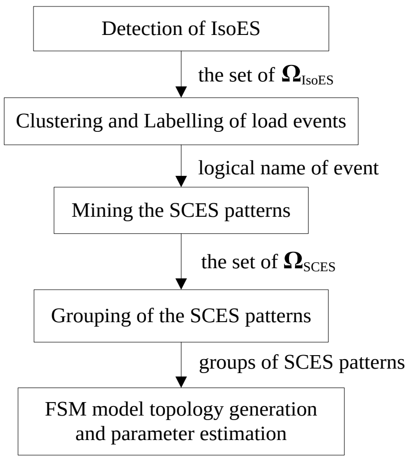

Fig.2. Flow chart of the unsupervised FSM modelling framework

The framework consists of the following steps: (1) In the load event sequence, the isolated load event sequence (IsoES) is detected and recorded in the set $\boldsymbol{\Omega}_{\text{IsoES}}$ (Section III.A will introduce the concept and benefits of IsoES); (2) through the load event clustering, the load events are labeled with logical names prepared for the frequent pattern mining to be carried out in the following step (3); (3) according to the logical name of the load event, the SCES patterns are mined in the set $\boldsymbol{\Omega}_{\text{IsoES}}$ and recorded in the set $\boldsymbol{\Omega}_{\text{SCES}}$; (4) the SCES patterns are grouped so that all SCES patterns in one group belong to one appliance; (5) the topology of the appliance FSM model is generated using the SCES patterns from its group, and the parameters are estimated.

III. Methods

A. Detection of the Isolated Load Event Sequence (IsoES)

As the core of the framework proposed in this paper, SCES pattern mining can be treated as a problem of frequent sequential pattern mining [8], which requires establishing an Event Sequence Database (**ESDB**). This paper takes the isolated load event sequence (IsoES) as an event sequence record in the Database. IsoES is the load event sequence contained in an isolated load window. An isolated load window refers to such an operating period of the total load: ① the operating state of the total load (referred to as load state in the following) before and after this period is the same; ② the total active power at any moment within this period is not less than that at the initial moment of the period. Fig. 3 shows an example of the isolated load window. The past experience shows that the total load can always be divided into a number of isolated load windows. Obviously, any IsoES is composed of several SCESs of several appliances. Therefore, taking IsoES as event sequence record in the **ESDB** can improve the efficiency (by reasonably reducing searching space) and accuracy of the SCES pattern mining.

Fig.3. The power curve of a three-state hair drier

IsoES detection method: Here, we define the load event, whose active power increment is more than zero as positive event, and that less than zero as negative event. Details of the IsoES detection method is as follows:

In the process of load event detection (detected load events are fed into the load event sequence, and initially marked as “unvisited”), once a negative event is encountered, it is marked as initial event for searching. Then forward searching is carried out in the load event sequence, until a positive event is found that the load state before it is the same as the load state after the initial event, and this detected positive event is regarded as searching termination event. At this point, including searching initial/termination event, load events which have been marked “unvisited” between them form an IsoES, and all load events in the new IsoES are marked as “visited”. During the search, if the total active power before (or after) any positive (or negative) event is less than that after the searching initial event, then the searching is abandoned, and start another similar searching from next negative event to be found in the load event sequence.

Here, the load state is described by the two-dimensional vector combining the active power and reactive power. In accordance with the described method, load event sequences within a period of time for a household can be divided into a number of IsoESs, as shown in Fig.4, the "square" and "dot" is the start and stop positions of IsoES respectively. For example, four IsoESs are contained in the box, according to the labelling sequence in the process of detection, they are: < E_2, E_3 >, < E_4, E_5, E_6 >, < E_7, E_8 > and < E_1, E_9>.

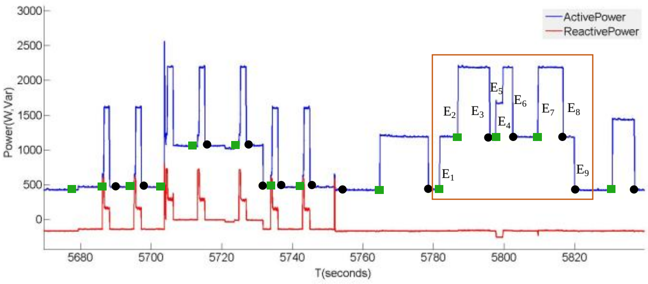

Fig.4. Results of the IsoES detection in the total load

Further, in this paper the set composed of all detected IsoESs during a period of time is denoted as $\mathbf{\Omega}_{\text{IsoES}}$, and the set composed of all load events in all the detected IsoESs is denoted as $\mathbf{\Omega}_{\text{E}}$.

B. Clustering and labelling of load events

As described in section III-A, IsoES in the $\mathbf{\Omega}_{\text{IsoES}}$ set can serve as the event sequence record in the **ESDB**. Before starting the pattern mining, we also need to label each load event in each IsoES with a unique logical name, and it is required that all load events generated by a specific appliance while experiencing the same state transition process have the same logical name, so that the content of **ESDB** can reflect the existence of the SCES patterns. For this purpose, cluster analysis techniques were applied to divide $\mathbf{\Omega}_{\text{E}}$ into different load event clusters, and the logical name of the load event is labeled by the label of the cluster it belongs to. Since the number of the load event type is unknown, theoretically, any cluster analysis methods not requiring the cluster number as the input parameter can be applied to this framework. The mean-shift clustering algorithm is adopted in this paper [9].

Before the load event clustering, there is a need to describe or represent load event with appropriate signatures. In order to improve the accuracy of the load event clustering, this paper uses vectors (donated as $\boldsymbol{\beta}$) combining the fundamental and high order harmonic components (3, 5, 7) of active power and reactive power increments before and after the event to represent load events.

C. Mining the SCES patterns

As the conventional representation of event sequence in the field of sequence pattern mining, the example of **ESDB** can be simply shown in Table 1. For each IsoES, a unique corresponding sequence of (e_{ij}, t_{ij}) structured by the load event information is stored in the database, where, e_{ij} represents the type label of the j-th load event in the i-th record of IsoES, and t_{ij} is the timestamp when load event e_{ij} occurred. For instance, e_{ij} =6 in the Table 1, "6" denotes the logical name of load event provided by the clustering analysis in Section III-B.

Frequent Event Sequence (FES) patterns can be obtained through performing frequent sequence mining on **ESDB**, which form a set denoted as $\mathbf{\Omega}_{\text{FES}}$. In this paper, the classic GSP (*Generalized Sequential Pattern*) algorithm is adopted for frequent sequence mining [8]. The GSP algorithm is based on the theory of Apriori. It firstly generates a longer candidate sequence pattern via bottom-up procedure based on a shorter sequence pattern, and then calculates the support of the candidate sequence pattern (namely, the frequency that sequence pattern occurs), and finally makes comparison with support threshold to determine whether the candidate is a frequent sequence pattern. For instance, if the support threshold value is 2 (the actual value is generally greater than this), then the bolded and underlined event sequence whose length is 3 shown in Table 1 is a frequent sequence <**6**,**1**,**2**>, corresponding sequence of cluster center is <$\boldsymbol{\beta}_6$, $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$>.

Table1: Example of ESDB used in the proposed framework

ID of Event sequence	Event Sequence
1	< (**6**,t_{11}), (**1**,t_{12}), (3,t_{13}), (**2**,t_{14}), (5,t_{15}) >
2	< (3,t_{21}), (4,t_{22}), (7,t_{23}), (5,t_{24}) >
3	< (**6**,t_{31}), (8,t_{32}), (**1**,t_{33}), (10,t_{34}), (**2**,t_{35}) >
…	…
k	< (**6**,t_{k1}), (**1**,t_{k2}), (9,t_{k3}), (15,t_{k4}), (**2**,t_{k5}) >
…	…

However, the mined event sequence patterns in set $\mathbf{\Omega}_{\text{FES}}$ aren't all SCES patterns, therefore, according to the cluster center sequence corresponding to the FES pattern, we use the constraint of $\boldsymbol{\beta}$-ZLSC to filter out all the SCES patterns from $\mathbf{\Omega}_{\text{FES}}$, which results in the set $\mathbf{\Omega}_{\text{SCES}}$. Here, $\boldsymbol{\beta}$-ZLSC takes higher order of harmonic power into consideration in addition to the total power which is originally suggested by Hart [1], where the vector $\boldsymbol{\beta}$, representing the vector describing the load event as mentioned in Section III-B, is treated as a prefix to show the difference.

D. Grouping of the SCES patterns

For multi-state appliances such as microwave oven and hair dryer, they can produce more than one SCES patterns. Therefore, it is necessary to divide the set $\mathbf{\Omega}_{\text{SCES}}$ into different SCES pattern groups based on the correlation between patterns, making each SCES pattern group associate with one specific appliance. In this paper, the **Event Correlation Rule** is used to group the SCES patterns, which means that, considering the fact that the load events produced by different appliances are different, if different SCES patterns share the same type of load event(s), they are considered to be generated by the same appliance and can be grouped into one group.

E. FSM model topology generation and parameter estimation

To achieve the appliance FSM modelling, it requires combining the elements of acquired group of the SCES patterns into the FSM model topology and estimating related parameters, for which we proposed an algorithm called the **Incremental Topology Generation and Parameter Estimation (ITGPE)**. The details are as follows.

1) Randomly select a SCES pattern in the group to form a single-cycle MTG whose edges correspond to the load events in the SCES pattern. The reference powers of the edges are initially marked by the active/reactive power increments of the cluster centers corresponding to type of the load events, and then they are fine-tuned to satisfy the ZLSC constraint with their sum(can be called “residual power”) equally shared with each event of the SCES (same as follows). After that randomly assign a vertex in the cycle as zero-state vertex, whose reference power is set to 0, while the reference powers of the other vertices can be marked by summing the reference powers of the edges along one path from the zero-state vertex to the vertex to be marked (same as follows).

2) Randomly select a next SCES pattern, and in the formed MTG, add new edges and the vertices associated with new load event type introduced by the new SCES pattern. Mark the initial reference powers for the new edges as above and then tune them with that of the marked edges fixed (same as follows). After that mark the reference powers for the new vertices Repeat this procedure until all SCES patterns in the group are accessed.

3) Assign the vertex with the minimum reference power in the MTG to be the new zero-state vertex, subtract the reference power of the new zero-state vertex from that of all the vertices, and mark all vertices whose reference power is not zero as different logic states.

4) Based on the established FSM model topology, the acquired reference power of the vertex is taken as the power mean parameter of the corresponding operating state of the appliance and the power variance parameter can be estimated by the sum of the power variance of the events directly related to it that can be derived from the clustering result.

If necessary, the probability parameters of state transition of FSM model can be estimated by the timestamp record in the **ESDB** of all the events generated by the appliance[10][11].

IV. Experiments

The proposed framework is tested with the consumption data for 24 hours from a real household on Saturday and the power data was sampled at 1Hz. Firstly, using the method suggested in Section III-A, the number of IsoESs detected is 76. Here the event detection method proposed by Hart is adopted and the power change “tolerance” is set as 50W [1]. According to Section III-B, the results of load event clustering are shown in Table 2. As only the event cluster whose number of elements is greater than the support threshold (here set to be 3 for the duration of the dataset sampled is relatively short) is meaningful for frequent pattern mining, only the clusters with more than 3 elements are listed in the table, and △P represents the active power increment of the cluster center. Based on the load event clustering, 6 SCES patterns are detected; the results are shown in Table 3. Furthermore, according to Section III-D, No. 3 and No. 6 SCES patterns belong to one three-state appliance. Finally, according to the **ITGPE** algorithm in Section III-E, 5 appliance FSM models are established. Take the No.3 appliance “Hair drier” in Table 4 as instance, first, a single-cycle MTG corresponding to the No.3 SCES <***12***,***3***> in Table 3 is established, after that the No.6 SCES <***10***,***5***,***3***> introduce two new edges and one new vertex eventually forming the final MTG. Then the related parameters including the mean(μ) and standard deviation(σ) of the active power and reactive power of each state are estimated as shown in Table 4.

Table 2: The results of load event clustering

Cluster label	△P(W)	No. of elements	Cluster label	△P(W)	No. of elements
1	-3101	5	***9***	-585	18
2	1145	25	***10***	496	4
3	-1009	15	***12***	1011	13
4	781	4	***14***	-762	4
5	513	4	***15***	609	18
8	-1130	24	***17***	3142	5

Table 3: The results of SCES pattern mining

ID of SCES pattern	SCES (cluster label, see Table 2)			Frequency of SCES pattern
1	***2***	***8***	—	23
2	***4***	***14***	—	4
3	***12***	***3***	—	12
4	***15***	***9***	—	16
5	***17***	***1***	—	5
6	***10***	***5***	***3***	4

Table 4: The results of FSM model parameters

ID of appliance	μ(W)	σ(W)	Appliance name
1	{**0**,1138/298}	{**0**,22/23}	Microwave oven
2	{**0**,772/1}	{**0**,11/2}	Electric oven
3	{**0**,496/-56,1009/-4}	{**0**,9/19,19/3}	Hair drier
4	{**0**,597/147}	{**0**,21/8}	Vacuum cleaner
5	{**0**,3122/-8}	{**0**,13/11}	Electric Heater

V. Conclusion

In this paper, we proposed a fully unsupervised appliance finite state machine (FSM) modelling framework for NILM. The realization of different function modules is described, and the effectiveness of the proposed framework is validated by experiments in a real household. Without intruding into the internal load or requiring any priori knowledge of the appliance model, the framework can automatically establish the topological structure of appliance FSM model and estimate the associated model parameters, so it can improve the applicability of the existing NILM technologies, and lay a foundation for the realization of the auto-setup NILM.

The framework proposed in this paper will be tested and analyzed on more measured datasets, and the methods and techniques used by different functional modules of the framework will be continuously optimized.

References

[1] Hart G.W, Nonintrusive Appliance Load Monitoring [J]，Proceeding of IEEE, 1992, 80(12): 1870-1891.

[2] Giri S, Bergés M. An energy estimation framework for event-based methods in Non-Intrusive Load Monitoring[J]. Energy Conversion and Management, 2015, 90: 488-498.

[3] Kim H, Marwah M, Arlitt M F, et al. Unsupervised Disaggregation of Low Frequency Power Measurements [C]. SDM. 2011,11:747-758.

[4] Michael Baranski，Jürgen Voss. Detecting Patterns of Appliances from Total Load Data Using a Dynamic Programming Approach[C]. Fourth IEEE International Conference on Data Mining，2004：327-330.

[5] Streubel R, Yang B. Identification of electrical appliances via analysis of power consumption[C] Universities Power Engineering Conference (UPEC), 2012 47th International. IEEE, 2012: 1-6.

[6] Parson O, Ghosh S, Weal M, et al. An unsupervised training method for non-intrusive appliance load monitoring[J]. Artificial Intelligence, 2014, 217: 1-19.

[7] West D B. Introduction to graph theory[M]. Upper Saddle River: Prentice hall, 2001.

[8] Srikant R, Agrawal R. Mining sequential patterns: Generalizations and performance improvements[M]. Springer Berlin Heidelberg, 1996.

[9] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002, 24(5): 603-619.

[10] Rabiner L R. A tutorial on hidden Markov models and selected applications in speech recognition[J]. Proceedings of the IEEE, 1989, 77(2): 257-286.

[11] Vidal E, Thollard F, De La Higuera C, et al. Probabilistic finite-state machines-part I[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005, 27(7): 1013-1025.