Applicability of using time series subsequences to study office plug load appliances

Balaji Kalluri\textsuperscript{a,}\textsuperscript{*}, Andreas Kamilaris\textsuperscript{b}, Sekhar Kondepudi\textsuperscript{a}, Harn Wei Kua\textsuperscript{a}, Kwok Wai Tham\textsuperscript{a}

\textsuperscript{a} Department of Building, National University of Singapore, 119077 Singapore
\textsuperscript{b} Department of Computer Science, University of Cyprus, 20537, Cyprus

A R T I C L E   I N F O

Article history:
Received 25 November 2015
Received in revised form 20 May 2016
Accepted 23 May 2016
Available online 4 June 2016

Keywords:
Office appliances
Office plug load dataset
Subsequence mining
Grammar induction
SAX
Motif discovery
Feature extraction
Characterization
Bag of Rules
Classification

A B S T R A C T

Energy management in offices requires efficient methods (e.g. non-intrusive load monitoring techniques, NILM) to monitor the large number of workstations and office appliances. The key purpose of this study is to ascertain the applicability of using time series subsequence data mining to study and classify the transient operations of typical appliances in an office. The approach involves discovering hidden subsequences (i.e. feature extraction) that are characteristic of individual appliance transient states, using an extension of Symbolic Aggregate approXimation (SAX). Such characteristic features are used to create a repository of rules to help supervised classification of aggregate time series measurements. It is one of the first studies to demonstrate the potential of classifying subsequence features into individual appliances and their states within large aggregate time series data using a “Bag of Rules” approach. The results indicate that distinct, characteristic patterns represent office appliances and their states, in the form of SAX grammar rules. These patterns can then be used for NILM with promising results. This ongoing study demonstrates SAX based time series subsequence mining as a proof-of-concept; not only to discover similarities presented by appliance events but also to demonstrate their applicability to disambiguate aggregate signatures in the context of office NILM.

\textcopyright 2016 Elsevier B.V. All rights reserved.

1. Introduction

Miscellaneous Electrical Loads (MELs) account for more than 20% of primary energy use in commercial buildings [1,2]. Among MELs, there is a sub-class of office appliance (commonly referred as Information Communication and Technology—ICT) plug loads. A recent study in UK estimated the energy usage breakdown of ICT appliances in an office [3] and found that 50% energy goes to PCs while 30% goes to ICT services (data centres, servers, storage etc.) and 20% to combined peripheral and telecom usage (multi-function devices, mobile chargers, phones, speakers etc.). It has also been reported that ICT forms a significant share (about 23%) of energy usage in commercial buildings [4] and this trend is predicted to grow to 36% by 2030 [5]. A detailed review [6] was conducted on field protocols for office appliance audits, measures to improve their energy conservation and techniques to benchmark energy performance of various offices appliances. It was found that monitoring, analyzing and cutting down energy wastage from such appliances in office buildings are keys to improve building energy efficiency. Relevant studies [7,8] outline some best practices in the rational use of ICT appliances in offices. Some examples of such best practices include switching-off desktop PCs and printers at night, adjusting brightness and effective power management of monitors etc., with an estimate of approximately 35% of potential savings. In a study [9] examining the impact of feedback to occupants on their appliance-level energy usage pattern in an office, it was found that occupants and their interaction with office appliances (e.g. desktop PC) play a key role in improving energy conservation in offices. The guide to good practices in operating ICT devices in offices has shown potential energy savings of more than 130,000 kWh annually [7]. The actionable feedback to occupants has shown a potential of 311 kWh of savings from desktop PCs within a small office environment involving 18 occupants [9].

All the aforementioned studies showcase the importance of proper energy management, and this can be achieved only by detailed load monitoring. Non-Intrusive Load Monitoring (NILM) was originally conceived in 90s [10] as a tool to enable the predictive measurement of energy consumption of individual appliances
without the need for individual sub-metering, but several varying approaches have been proposed thereafter. Reviews on state-of-the art approaches for energy disaggregation [11,12] found that most of the initiatives are primarily applicable to residential buildings. An alternative to distributed smart energy metering in offices is by employing virtual sensors [13], however this approach offers only approximations of energy consumption per appliance. A conceptual schematic of a single-point based energy disaggregation in offices is presented in Fig. 1. Ideally, in this scenario, it is desirable to measure load at the sub-meter panel labeled ‘M’ and be able to disaggregate individual workstation loads. However, offices present several identical appliances at each workstation, all operating at the same time in the plug load circuit. Therefore, load disaggregation from such single-point measurement in offices still remains challenging [6]. It underpins the need for intelligent feature extraction and characterization of office plug load appliance events prior to disaggregation. Additionally, it requires a dedicated dataset comprising of energy signature measurements of multiple office appliances usage scenarios. Considering these challenges, the primary purpose of this study is to demonstrate the applicability of using subsequences in the form of context-free grammar rules as features for more effective and accurate NILM in offices. A detailed appliance-level characterization through NILM in offices can be a valuable tool to several stakeholders such as:

- **Facility managers and building owners:** Energy conservation to the level of every plug load can be made possible with real-time appliance level insights.
- **Building occupants:** Precise actionable measures when feedback is provided appropriately to building occupants, can strengthen the value chain of the occupant in driving the energy efficiency programs in buildings [9].
- **Energy auditors:** The reduced instrumentation effort in buildings not only saves the time and cost but also presents less data in a more structured fashion.
- **Utilities:** Knowledge of how the different loads are being used can greatly enhance the value of existing electricity supply chain in terms of efficiency, safety and productivity.

The ability to disambiguate an aggregate energy signature not only helps measure but also manage office plug load appliances from a single point measurement. This is the central idea of office NILM. This study employs Office Plug Load Dataset (OPLD) [14] for the purpose of time series data mining. OPLD was developed using the proposed approach as in Fig. 1 to help improve the understanding of individual appliance states from aggregate signatures [14]. The proposed approach approximates every workstation subcircuit into a dis-(aggregation) point (labeled A in Fig. 1), each associated with three or four dissimilar appliances. To the best of our knowledge, OPLD is the only dataset till date that has been developed with a specific aim to study office appliances for plug load disaggregation. The goal is to facilitate the ability to predict multiple individual plug loads by measuring at a single point. Clearly the incentives of the proposed disaggregation approach especially in offices are manifold. The key contributions of this study are the following.

1. It is the first to demonstrate the applicability of context-free grammar rules through time series subsequence mining for characterizing and classifying individual appliances and their events within aggregate time series office-related data.
2. A novel idea of “Bag of Rules” (BoR) for appliances’ transient events is proposed. Several aggregate time series measurements from OPLD are analysed to develop BoR.
3. A simple classification approach employing BoR for a Multi-Functional printing Device (MFD) is introduced as a proof-of-concept. The classification results suggest the applicability of time series subsequence mining in the context of NILM in offices.

This paper is organized as follows: In Section 2, a brief review on existing plug load datasets and various time series subsequence mining approaches for energy applications are presented. Section 3 describes the feature extraction step towards office load disaggregation. This is proposed using a subsequence mining technique called SAX. Section 4 presents a detailed appliance energy signature characterization using an open source tool. A case study presents the analysis of both an individual office appliance and its aggregate time series measurements from OPLD. In Section 5, the idea of creating a corpus of characteristic appliance features (rules) is illustrated. This is based on the idea of using such “Bag of Rules” repository from Information Retrieval research to perform a supervised classification of unknown energy signatures. Finally, in Sections 6 and 7, the summary of important findings from this study, potential challenges and possible approaches as future steps towards scaling this study are presented.

### 2. Literature review

It is important to first identify the nature of data required and then investigate appropriate methods to solve the NILM problem in offices. Hence, existing plug load datasets are listed in this section, followed by a review to understand the applicability of time series subsequence data mining techniques for energy signature analysis.

#### 2.1. Plug load energy datasets

A plug load dataset is a repository of electricity consumption data of several individual appliances (for example, microwave oven, refrigerator, computers) and their aggregate measurements across multiple levels (such as whole-home or floor-level) using energy meters. In general, such datasets can vary based on building type (e.g. residences, offices or other). Recently, there has been a plethora of publicly available datasets regarding office plug loads, such as the Reference Energy Disaggregation Dataset (REDD) [15], Building-Level Fully labeled Electricity Disaggregation dataset (BLUED) [16], Almanac of Minutely Power Datasets (AMPDs) [17] and Electricity Consumption & Occupancy dataset (ECO) [18]. The existing public plug load datasets were specifically developed for homes and exhibit the following characteristics:
• No or little real presence of office appliances [17,18]
• Absence of data for multiple power management (operational) modes [16]
• Lack of clear description of electronic appliances [17]
• Inappropriate means of labeling and characterizing appliance events [16]
• Lack of appropriate measurement plans specifically for office appliances [15–18]

A more detailed description of the above datasets in terms of the nature of appliances, number of buildings measured, duration of measurement and their actual data collection plans is presented in [14]. The weaknesses of the aforementioned datasets imply that NILM in offices need to be treated differently using a different kind of dataset, such as OPLD, which has been developed specifically for offices, offering the following:

• A repository for energy signatures of several common office workstation plug load appliances. Currently, it encompasses four appliance types: desktop PCs & laptop PCs, monitors and MFD.
• Characterized by low-frequency measurements of both individual appliance and aggregate circuit using metrics such as real power (P), apparent power (S) and current (I).
• Laboratory-based experimental measurements that approximate real usage patterns of ICT appliances in offices.
• Multiple operational modes aiding office appliance characterization across multiple operational states thus enabling improved non-intrusive load disaggregation in office environment.
• Repeated measurements to aid better characterizing appliances and build models.

2.2. Time series subsequence mining of energy signatures

Time series subsequences, refers to frequently occurring patterns within time series data [20]. Several diverse applications of subsequence mining can be found in literature [19]. A related terminology called time series motifs refers to recurring patterns of shorter duration within large temporal data. Mining motifs in time series data has been applied successfully in a variety of domains [21]. This section presents a brief summary of relevant studies in the context of energy data.

The approach of time series data employing subsequence mining was first applied in customer segmentation and energy consumption forecasting [22]. The experimental results suggested that the performance of consumption forecasting using symbolic representation of smart meter data was comparable to real values. In another study [23], a conceptual framework called PowerSAX was proposed, to reduce data volume while retaining all relevant features (e.g. appliance events) in data streams. Another approach [24] exploited temporal sequences corresponding to state change events in order to discover motifs in appliance energy data. Disaggregation based on motifs exhibits effectiveness even if the appliance count in the circuit increases. Another related work proposed the use of a variant of time series subsequence mining called shapelets [25]. The idea of shapelets is to extract discriminative features within large time series residential energy data [16]. The shapelets-based features have shown reasonably good classification of low-power appliance events from aggregate energy data being limited however only to binary ON/OFF modes.

In summary, based on the aforementioned literature, the following observations are made in favor of time series subsequence (motif) mining as a suitable approach for studying office appliances:

• Ease of characterization: Modeling the changes in appliance operational modes can be observed as discriminative features in time series, based on the changes in power consumption (signature) pattern [23]. Motifs can help model such a change in appliance operational modes exhibited as signature changes. For example, using subsequences as characteristic features has much potential in differentiating between instances of different appliances [25]. A built-in SAX discretization engine together with a grammar induction algorithm and an intuitive user interface in GrammarViz2.0 [26] makes characterization simpler.
• Ability to model recurring features: Time series subsequence mining can identify recurring signature patterns in time series data not merely by the shape of the signature pattern but also based on the frequency of its occurrence within the large time series. For example, one potential criterion is the recurring nature of appliance state changes exhibited as characteristic signatures [23]. Additionally, highly recurring motifs within the dataset may influence classification accuracy [24]. OPLD also exhibits repeated measurements of appliance operation as recurring signature changes (i.e. features). The case study of MFD presented in Section 44 demonstrates the ability of motifs to model such recurring appliance features.
• Data dimensionality reduction: Mining large time series datasets is often tedious unless such large dimensional, continuous range of numeric values can be compressed to finite set of simpler features. The power of symbolic aggregation that enables motifs discovery offers such numerosity reduction [22,23]. For example, PowerSAX [23] achieves energy data size reduction by simply representing raw power data as symbols. Feature extraction presented in Section 3 demonstrates this property.

3. Subsequence feature extraction

Feature extraction is an important step in energy disaggregation [10]. Identifying appliances and their operational states from an aggregate energy signature is critical. Our proposed approach is based on feature extraction from subsequences of energy signatures by using SAX. SAX is a novel technique that integrates both discretization and symbolization of time series. A large collection of symbolized subsequences are then analyzed to extract similar features (otherwise called grammar rules) using GrammarViz2.0 [26]. The purpose of GrammarViz2.0 is triple-fold. Firstly, it helps discretize and symbolize the time series data using built-in SAX engine. Secondly, it helps discover the repeated similarities in subsequences within the time series data using a grammar induction algorithm. Finally, it helps extract and visualize corresponding symbolized grammar rules based on similarity of rules. Such symbolized grammar rules are expected to aid in characterizing and classifying appliances and their transient operating states from aggregate energy signatures. The results of subsequence characterization and classification are presented later in Sections 4 and 5 respectively.

3.1. Discretization and symbolization using SAX

SAX is the first symbolic representation of time series that permits both dimensionality reduction and indexing using lower-bounding distance measure [27]. This technique has been around for almost 10 years now and its utility has been demonstrated across several application domains such as medical, imaging etc. [28]. However, the application of SAX to the load disaggregation problem is limited [25]. This study is one of the first to harness the benefits of SAX in the context of office NILM. Some of these benefits are listed below:

3.1.1. Discretization of time series

It allows a time series energy measurements of arbitrary length n to be transformed into a string of arbitrary length w, (typically, w = n). This is the notion of dimensionality reduction stated earlier
in Section 2.2. This discretization property of SAX is unique in the sense that it first transforms the raw time series into piecewise aggregate approximation (PAA) representation. At first, input data is discretized into PAA segments of relatively smaller dimension (e.g., \( w = 8 \)) (see the example subsequence in Fig. 2). The resulting discretized PAA segments are later symbolized using an alphabet to further generate a grammar rule to enable subsequence (motif) discovery. Prior to these steps, SAX normalizes any input time series data \( (S) \) over its mean \( (M) \) and standard deviation \( (SD) \) using Eq. (1).

\[
S_i^* = \frac{S_i - M}{SD}
\]

Normalizing time series data is required as a priori step to compare true signature similarity between subsequences [29]. An example of a normalized subsequence \( (S_i^*) \) from a time series (labeled in blue in Fig. 2) is dimensionless as observed from its \( y \)-axis.

### 3.1.2. Symbolizing subsequences

The resulting discretized energy signature (i.e., numeric PAA data segments) is encoded using symbols. Fig. 2 illustrates how a sample input time series data from OPLD of length \( n = 64 \) is transformed into a SAX string of length 8, using an alphabet of size 4 \( (a, b, c, d) \). The example string of length 8 in Fig. 2 is interspersed with multiple small color-coded flat line segments. They represent the equivalent PAA segments obtained through SAX discretization step. It is worth to note that SAX breaks-up the entire range of normalized time series data both vertically and horizontally.

For the purpose of illustration, the normalized input subsequence is divided into 8 equal sections horizontally based on the SAX window size parameter, \( w = 8 \). This is followed by dividing the variations in amplitude specified by three horizontal lines in grey, in the next step. The topmost of the three horizontal boundary lines starts somewhere between 0.5 and 1, whereas the middle line starts at 0 and the bottom-most starts between \(-0.5 \) and \(-1 \) respectively. In the example case presented in Fig. 2, the input alphabet size parameter, \( a = 4 \) ensures that the time series is divided into PAA segments that fall under one of the four regions bounded by the horizontal lines using symbols labeled \( a, b, c \) and \( d \). Thus, normalized input time series subsequence of 64 time units is approximated using a string (i.e., \( "abdcbbbc" \)) of 8 characters/symbols, as shown in Fig. 2. This is the process of symbolizing subsequences. In general, the input time series data is much longer. For example, in the case of MFD, the ground-truth data in OPLD is approximately 12,000 time units (for uniquely representing some operation/state). However, we are only interested in discovering multiple subsequence motifs which typically last for about 15–30 s within the input time series data. This transient appliance operational duration might vary between appliances and operational modes. Hence, the idea of a sliding window-based time series discretization is necessary to extract all possible subsequences from within the input time series. This will be followed by mining for repeated patterns of strings using SAX to discover the recurring appliance features within the measured data. The detailed implementation of SAX algorithm [27] and its numerous applications are available in [19].

### 3.2. Grammar induction and motif visualization

The next step is to extract grammar rules. Grammar rules are a set of recursive string patterns found within the input time series that are used to represent characteristic appliance events. The process of inferring these rules intelligently within the symbolic string data is called grammar induction. It is desirable to have this induction algorithm built into a tool that helps discover automatically characteristic appliance motifs of variable length without having to know their exact length or position. Additionally, the visual representation of the motifs’ structure along with their induced grammar rule is helpful, for offering preliminary insights into repeated patterns in subsequences within input time series appliance measurements.

This study employs GrammarViz2.0\(^1\) an interactive subsequence motif mining tool. It implements all the three previously discussed ideas (symbolic representation of energy signatures, grammar rule induction and motif visualization of extracted features). Sequitur [30] is an algorithm built into GrammarViz2.0 that can efficiently discover recurring string patterns corresponding to hidden appliance features and hierarchical structure within a long time series data. It also outputs a relatively small set of interpretable grammar rules given a time series energy measurement. Such rules can be used to discover similarities within time series data to better aid characterizing appliances’ transient behaviors. Other tools such as VizTree [31] also help in visualizing hidden motifs given an input time series data. In our work, we have selected GrammarViz2.0 in characterizing OPLD for the following reasons:

- It offers visual analysis of large time series data in short time. This helps to quickly mine similarity in subsequences intuitively in a few steps.
- The integration of Sequitur helps to exploit the hierarchical structure within the long time series data. Thus, it helps to discover similar grammar rules of variable length.
- The visual mining of similarity in time series subsequences is aided through simple and intuitive color-coding schemas.
- The pan and zoom features along the time series data additionally help in visualizing the identified subsequence signatures.
- An intuitive summary of the similarity in signatures represented by distinct grammar rules is presented in the form of a table.

Further feature (grammar rules) extraction from the input time series is performed in conjunction with appliance events from labeled ground-truth data. The graphical terminal window of GrammarViz2.0 in Fig. 3 shows four distinct parts of the tool to help analyze time series data. They are the following:

\(^{1}\) Source code and demo available: https://github.com/GrammarViz2/grammarviz2_src.
1. Data display window: This part of the tool (labeled as 1 in Fig. 3) helps to visualize the input time series data loaded for analysis (e.g., an individual MFD measurement from OPLD).

2. SAX input parameter selection fields: These fields (labeled as 2 in Fig. 3) help the user to select appropriate input parameters for time series subsequence mining using SAX. This is done based on the domain knowledge. Characterizing time series subsequences from OPLD requires reasonable level of prior knowledge about appliance usage in different modes. For example, in the case of MFD, an approximate estimate of the duration of each operational mode (e.g., copy, print and scan) would be helpful. This approach is clearly not scalable for analyzing hundreds or thousands of time series at once. However, [32] reported a scalable approach to making SAX parameter estimation using Dividing RECTangles (DIRECT) optimization algorithm. The application of this algorithm with SAX was found effective across diverse range of time series datasets.

3. Grammar rule search table: The table (labeled as 3 in Fig. 3) helps to identify and extract potential features i.e. grammar rules and their characteristics from a list of all possible subsequences inferred by Sequitur within the input time series. The descriptors such as mean motif length, range of lengths of subsequences, frequency of their occurrences, and hierarchical representation of rules allow the user to make informed decision on the selection of motifs.

4. Normalized rule subsequence window: The similarity between subsequence features represented by unique grammar rules is plotted in this window (labeled 4 in Fig. 3). The time series subsequences presented in this window are dimensionless and normalized, in order to compare the similarity between temporal signatures.

4. Subsequence characterization: a case study of MFD

The subsequence feature extraction leads to characterization of appliances and their events. Understanding appliances and their events from temporal energy signatures is the natural step towards load disaggregation. The time series measurements of both ground-truth appliances (i.e., an individual MFD) and their aggregates, which exhibit transient signature patterns, are presented as a case study for subsequence characterization in this section. The choice of a MFD for this demonstration is meaningful, since the appliance has multiple operating modes, each with distinct signature patterns across each state – both in amplitude and duration. Individual energy signatures (with labeled events) of an instance of MFD and its aggregate circuit signature from OPLD are analyzed. The aim of this study is to identify the characteristic motifs corresponding to distinct appliance operations and to extract their grammar rules (features) using GrammarViz2.0.

4.1. Using individual appliance signatures

The ground truth measurement from a single appliance instance P2 (i.e., MFD-Samsung SCX4833FR) from OPLD across its three most common operational states (i.e., copy, scan, and print) is analyzed. In order to discover the hidden signatures relevant to the MFD states, a sample time series comprising of data across multiple states in sequence is used as input for analysis. This input is considered as ground truth because of the prior knowledge of each episode corresponding to appliance events from the labeled dataset (i.e., OPLD). The temporal signature similarity in hidden subsequences and their corresponding unique features from input data are also clearly demonstrated.

The ground-truth time series measurement of an MFD instance P2 from OPLD is applied as input to GrammarViz2.0. The input time series data employed for analyzing P2 in this example case is approximately 11,500 time units (seconds) in length as seen from Fig. 4(a). A first look at this input time series data indicates that there are few fairly stable appliance states represented by flat episodes of approximate zero values. These episodes are interspersed with high-frequency transient episodes that appear as irregularities of varying amplitude and duration throughout the input data. However, a closer examination reveals characteristic patterns of transient states that lie hidden within. An example of one such characteristic subsequence pattern within a scan episode is presented in Fig. 4(b). It is evident that several spikes of about 600W lasting for a duration of 15–16s each are recurring within this scan episode in the input data. There exists some similarity in this spike train which is captured by a characteristic grammar rule using GrammarViz2.0. This similarity in the subsequences is highlighted in Fig. 4(b). Similarly, all transient episodes in the input time
series data presented in Fig. 4(a) are expected to contain characteristic features of target appliance (P2) states.

It should be noted that such characteristic features for each appliance state are distinct (not shown in Fig. 4). This would become clearer later in this section from the characteristic grammar rules summarized for each appliance state.

A more comprehensive view of one input time series data of MFD-P2 from OPLD is presented in Fig. 5. The labels corresponding to solid line, short-dashed line and long-dashed line in Fig. 5(a) highlight the distinct episodes of MFD in copy, scan and print operational states respectively. This is the labeled ground truth input data. Note that these labeled transient episodes within input data encapsulate several hidden, recurring characteristics features similar to the ones previously presented for P2-scan in Fig. 4(b).

Understanding the nature of measurements presented by OPLD, the input time series is expected to contain five episodes of copy, two episodes of scan and two episodes of print [14]. Further, each of these episodes typically lasts for about 540 s and contains nine non-overlapping cycles of each appliance operation. Therefore, the expected total count of motifs in the input time series data corresponding to copy, scan and print states are 45, 18 and 18 respectively. This is an important milestone in characterizing energy signatures within OPLD.

As discussed in Section 3, with reasonable prior knowledge about the length of the desired appliance states, the input SAX parameters for the tool are chosen. This is the first step after loading the input time series to be analyzed. The choice of input SAX parameters is different for each operational state. The next step in the analysis of input data is to discover similar subsequences within target appliance episodes such as copy, scan and print individually. This is performed using GrammarViz2.0 as previously illustrated in Fig. 3.

The output time series corresponding to P2-copy, P2-scan and P2-print obtained through SAX subsequence characterization are presented in Figs. 5(b)–(d). The highlighted sections correspond to characteristic signatures discovered by the tool. A closer look at the highlighted episode in Fig. 5(b) visually indicates the quality of the characteristic copy feature. Additionally, it also indicates that the discovered features are found not only across the desired copy episodes but also in fewer counts outside. The same is also observed for P2-scan, as seen in Fig. 5(c). The motif discovery corresponding to P2-print labeled in Fig. 5(d) presents some sparsity of motifs within the desired episodes.

In order to develop a repository of characteristic appliance features for further classification, mere visual motif discovery is insufficient. A quantitative representation of these characteristic features is required. This is accomplished through distinct grammar rules extracted using GrammarViz2.0. The characteristics of such features for desired appliance’s (P2) transient states are summarized in Table 1.

From Table 1, it can be observed that the inferred grammar rule corresponding to a characteristic feature for MFD-P2 in copy state is [eebb-debb]. Note that this grammar rule corresponds to highlighted motifs previously presented in Fig. 5(b). It is found that this characteristic feature is observed on 28 instances within the input data, against the expected frequency of 45. The expected frequency of motifs is described in brackets. This indicates a measure of the quality of motifs discovered using this approach. Moreover, this grammar rule presents a mean motif length = 23 using input SAX parameters: window size (w) = 22, PAA size = 4, alphabet size (a) = 6. Similarly, the inferred grammar rule for MFD-P2 in print state from Fig. 5(c) is [bbbb-cfbb] with mean motif length = 19 obtained using SAX parameters: w = 18, PAA = 4, a = 6; whereas grammar rule for MFD-P2 in print state from Fig. 5(d) is [debb-edbb] with mean motif length = 28 obtained using SAX parameters: w = 26, PAA = 4, a = 6.

It can be seen that the frequency of occurrence of grammar rules for P2-scan is 20 as against its expected count of 18 whereas for P2-print it is 12 out of its expected count of 18. By combining visualization of features from Fig. 5 and the frequency of their occurrences from Table 1, it can be observed that true positive features for copy, scan and print are 27, 16 and 11 respectively. This suggests characteristic feature extraction measure of about 60%, 88% and 61% for MFD-P2 in copy, scan and print states respectively. The analysis indicates the measure of confidence of applying these characteristic grammar rules as potential features for NILM based on single time series measurement from OPLD. However, the real classification model is based on features extracted from multiple individual appliance and aggregate circuit measurements. A more compreh
hensive summary of the characterization results from aggregate signatures is presented later in Section 4.2.

In summary, the subsequence mining of an individual ground truth appliance instance (i.e. MFD-P2) presented in this section, suggests the following:

- The characteristic feature candidates discovered for each transient operation i.e. copy, scan and print are clearly distinct. This is represented by different grammar rules.
- These distinct grammar rules strongly indicate the similarity between signatures of hidden subsequences across multiple desired labeled episodes.
- Across all transient states, the use of grammar rules in conjunction with labeled ground truth signatures indicate the true quality of features extracted. This indicates the ability of the proposed technique to help quickly characterize hidden transient appliance states given a long time series data and little prior knowledge of appliance operation.
- The technique employed is promising in recognizing and characterizing the transient appliance states. This is indicated by the feature extraction measure of 60%, 88.8% and 61.1% for copy, scan and print states of the example appliance instance (i.e. MFD-P2) respectively.
- The candidate characteristic features for copy and print were relatively weaker than scan in the MFD example. This is likely due to the relative occurrences of these motifs outside their desired episodes and signature similarity between copy and print operations.
- The mean length suggests that the extracted features within each episode are of variable length in time. However, Sequitur-based grammar rule induction technique built into GrammarViz2.0 is intelligent in combining and discovering approximately similar subsequences using distinct grammar rules.
- From the extracted features (i.e. grammar rules) it can be seen that characterizing appliance transient states does not require complex data pattern. The SAX based time-series subsequence mining offers the ability to uniquely represent each operation using relatively fewer symbols. This demonstrates the ability of SAX dimensionality reduction.

4.2. Using aggregate appliance signatures

Additionally, the analysis of aggregate time series measurements when an MFD is operating together with two other office appliances such as a desktop PC and a monitor in a sub-circuit is considered. The previous case study of MFD-P2 presented three distinct characteristic features from ground truth appliance data corresponding to copy, scan and print operations. The analysis of discovering such individual appliance features from several aggregate measurements is the objective of the analysis presented here. This helps in faithfully characterizing the MFD’s transient states given a single-point measurement in the context of office plug
load disaggregation. Such characterization further helps building a repository of rules (i.e. features) to facilitate classification as discussed in Section 5. Several aggregate measurements of MFD-P2 in operation with multiple instances of desktop PCs (e.g. D1, D2, D3) and monitors (e.g. M1, M2, M3) from OPLD are analyzed using GrammarViz2.0. Similar to individual MFD data, the aggregate appliance data is also supervised through ground truth labeled episodes obtained from OPLD. The knowledge of time series episodes help in estimating the accuracy of extracted features in faithfully classifying aggregate signature using the proposed technique.

The subsequent characterization procedure similar to the one presented in Section 4.1 for ground-truth MFD data is carried out for each aggregate signature. The results of analysis of 20 such aggregate time series measurements using GrammarViz2.0 are summarized in Table 2. It should be noted that each row represents the summary of characteristics of features (i.e. grammar rules) extracted by analyzing an aggregate time series data from OPLD. Similar to the individual MFD data previously presented in Fig. 5, each of these aggregate data are approximately 11,500 time units (seconds) in length. The results presented in Table 2 are categorized based on the three hidden individual appliance operations such as MFD-P2 in *copy, scan* and *print*. These are the desired individual appliance features of interest that lie hidden in the aggregate measurements. Being able to extract consistent characteristic features of individual appliance from the aggregate signature is the goal of classification in the context of load disaggregation. For each of these operations, three attributes such as grammar rule, percentage of positives (SP) and percentage of false positives (SFP) are determined during characterization. These are summarized in Table 2.

Eqs. (2) and (3) describe the usefulness of each grammar rule (i.e. characteristic feature) in determining their accuracy. Further, these accuracy metrics (i.e. %P and SFP) also help to ascertain the characteristic rules for each transient operation.

\[
\text{%Positives} = \frac{\text{Observed total number of occurrences of the rule in entire time series}}{\text{Expected total number of occurrences of the rule in desired episodes}} \times 100(2)
\]

\[
\text{%False Positives} = \frac{\text{Observed total number of occurrences of the rule outside desired episodes}}{\text{Observed total number of occurrences of the rule in entire time series}} \times 100(3)
\]

The following are some key observations from Table 2 that are worth summarizing,

- The grammar rule [ebbb-debb] under P2-COPY is clearly the most prominent characteristic rule (feature) to represent the desired appliance (i.e. MFD-P2) in *COPY operation*. This distinct rule is observed within 18 out of 20 aggregate time series data. In most of these cases, the performance is reasonably better with false positive representation under 10%. Additionally, on an average, this characteristic feature is observed slightly less than 50% (with maximum at 64.44%) of the expected occurrences within the input aggregate data.

- Similarly, for the case of recognizing MFD-P2 in print operation, the most prominent subsequence rule is found to be [debb-eddb] under P2-PRINT. Therefore, [debb-eddb] is considered the characteristic rule (feature) for MFD-P2 in *PRINT operation*. This rule is found promising on 15 out of 20 cases. This characteristic rule performs reasonably well with overall mean occurrences above 80% (with maximum at 100%) and false positives close to 30%. Furthermore, it can be seen that the following rules [dbbb-eebb] and [debb-eebb] are found prominent in other cases. These rules are similar to the most prominent rule [debb-eddb]. This similarity between rules can be understood from the SAX representation of their subsequences. From the above rules, it can be observed that the underlined symbols within the grammar rule representation presents only one (e.g. ‘e’ in eebb) or two symbol change (e.g. ‘f’ and ‘e’ in dfbb-eebb) when compared with the most prominent rule [debb-eddb]. This one-symbol difference between the rules represents a subtle signature shift in time within the subsequence.

- However, the case of recognizing P2 in scan operation within the aggregate time series data is little different from above two cases. Under P2-SCAN, there are clearly two prominent subsequence features represented by distinct grammar rules [bbff-cffb] and [bbbf-bccf]. Each of these is observed on 7 and 6 instances out of 20 cases respectively. However, the accuracy metrics for the rule [bbbf-bccf] present a mean positives of about 91% out of which 19% are false positives, whereas the same for the rule [bbbf-bccf] is about 80% and 17% respectively. The quantitative comparison of such metrics suggests that P2-SCAN is relatively better represented using the former rule i.e. [bbbf-bccf]. Additionally, considering the characteristic feature for P2-SCAN previously presented in Table 1 in conjunction strongly indicates [bbbf-bccf] as the characteristic rule for MFD-P2 in *SCAN operation*. Besides the rule labeled [bbbf-bccf] is also found to be quite similar to the characteristic rule [bbbf-bccf] from SAX representation point of view as discussed above.

- Besides all the above three characteristic rules that represent the transient MFD operations, there are some distinctly dissimilar rules that are also observed. For example, [ecdb-eddb] and [edcc-eddb] under P2-COPY, [bcff-bbcfb], [cbff-cfbf] and [bcff-cfbf] under P2-SCAN, and [edcb-fccb], [bcff-cfbf-deb] and [cbff-deb-ecdb] under P2-PRINT. Apparently, each of these rules also represents various MFD-P2 operations within the input aggregate time series data. However, all these rules contribute to the repository of appliance rules used for classification discussed later in Section 5. The results of classification presented in Section 5 might suggest the utility of such rules.

- It can be argued that part of the characteristic rules for P2-

COPY and P2-PRINT partly share similarities in their signature (i.e. “debb”) and therefore might result in a possible misclassification. However, understanding the SAX representation of time series clearly demonstrates that this part of the characteristic rule represents a part of the signature that appears on the early part of print signature while the same appears on the latter part of copy signature. Thus, these two rules, although they appear similar, they are strikingly distinct in the context of subsequence mining. This idea, when intelligently integrated into the classification model based on signature similarities, can largely disambiguate such similar features, thus improving classification results.

Finally, the Wilcoxon signed rank test is performed to test the statistical significance of grammar rules (features) presented in Table 2. It can be seen that the larger the difference between %Positives and %FalsePositives metrics, the more appropriately the rule serves as a feature in supervised classification. The Wilcoxon signed rank test reveals that the negative sum of ranks equals zero, which is less than the positive sum of ranks (i.e. 210) for all three events (i.e. P2 in copy, scan and print). This infers that the critical value for sample size (N = 20) for a two-tailed test at p ≤ 0.01 is 37. Clearly, the test statistic (i.e. negative sum = 0) is lesser than the critical value (i.e. 37). This implies that the mean of both %P and SFP are statistically different in the population, indicat-
ing that using these grammar rules as features would be promising in classifying time series subsequences and are not misleading.

In summary, the detailed characterization of both individual and aggregate time series data within OPLD, using an example case of MFD, helps us draw promising insights that could potentially assist supervised classification. The characteristic grammar rules extracted from aggregate time series data are consistent with the candidate characteristic features obtained from individual MFD-P2 data. This is a good indication of the potential of SAX subsequence mining in NILM. Several possible rules for an appliance operation suggest that there are clearly many potential subsequence features representing them. However, the accuracy metrics (e.g., %P and %FP) define the characteristic features representative of each target appliance and its operating state, given an unknown signature.

5. Supervised classification approach

After extracting discriminating appliance features and characterizing them from the time-series dataset, then the classification procedure follows. Given an unknown energy signature obtained from a single-point measurement in a circuit, the classification aims at disaggregating the load based on the individual appliance states that it is composed of. In other words, classification is analogous to disambiguating aggregate appliance signatures in the context of NILM. A simple approach to classify any given unknown appliance signature is presented in this section. Towards this effect, potential features (i.e., grammar rules) previously extracted and characterized from several aggregate energy signatures are collected. The Bag of Rules (BoR) – an idea similar to the one presented in Ref. [33] – has been used to aid supervised classification. The notion of such collection of rules or features has been widely employed in text mining and information retrieval research [32,34,35]. This is followed by discussion on using a Naïve approach to classifying unknown energy signatures (i.e., subsequence rule) into the underlying (known) hidden appliances and their operational states, from the knowledge of the BoR. This section demonstrates the applicability of time series subsequence mining as proof-of-concept for NILM in offices.

5.1. Bag of rules repository

Considering the example of MFD as presented in Section 4, a Bag of Rules (BoR) for MFD is created by carefully analyzing the characteristic features i.e. grammar rules presented previously in Table 2. There were in total 15 distinct rules discovered from Table 2. It is worth noting that such BoR for MFD is obtained by analyzing multiple aggregate time series data from OPLD in conjunction with candidate rules obtained from ground truth MFD data. An example of such a simplified BoR representation is presented in Table 3.

Typically, this BoR has three parts, namely the event class type, the grammar rule(s) and its count. In the example case of MFD presented in Table 3, there are clearly three distinct classes namely copy, print, and scan. The task of classifying an unknown aggregate signature among the three classes is presented in this example BoR. It can be seen that all candidate rules with three parts (e.g. [bbfcb-bcbf-cbcbf]), [bbfcb-bcbf-cbcbf] and [bcbf-cbcbf-cbcbf] in Table 2 are discarded in developing BoR representation in Table 3. These rules are noticeably dissimilar from characteristic rules of individual appliances (e.g. MFD) with only two parts. This reduces the total rule count within BoR to 13, out of which 12 rules are different while one rule is common between the scan and print class. The rules highlighted within BoR in Table 3 indicate the characteristic grammar rule for each operation.

As discussed in Section 4, the characteristic feature of any class indicates the relative frequency of its occurrences among other features representing the same class. This is evident from Table 3 across every class. Such BoR represents the nature of the databases used by potential NILM classification models [32,34] based on subsequence features. The BoR is dimensionally reduced from a large time series dataset into a compact grammar rule repository.
This kind of repository can help in supervised classification of any unknown “query” feature from aggregate signature.

5.2. Naïve classification approach

An example of such a supervised classification using BoR is illustrated using a Naïve classification approach. This demonstrates the applicability of subsequence-based energy signature classification approach in the context of sub-circuit office NILM. The following assumptions are made:

1. The model assumes that an appliance signature to be classified (i.e. query rule) is already present inside BoR previously presented in Table 3.
2. Such query rules are either complete (e.g. [eebb-debb]) or partial (e.g. [xxxx-edbb], [fcbb]). A query rule of the pattern [xxxx-edbb] suggests that the first part of the query rule can be anything but the second part should exactly match “edbb”. Similarly, query rule of the pattern [eebb-xxxx] suggests that the first part of the query rule should match “eebb” while the second part can be anything. Another query type could possibly be of the pattern [bfbf] which suggests that this substring rule could anywhere within a complete grammar rule irrespective of its position, unlike the previous two query types.

A simplified conceptual schematic of such a Naïve classification approach to disambiguating appliance signatures using BoR is presented in Fig. 6. The approach is intuitive and self-explanatory. The classification model outputs an appliance type and its operational state based on a search from BoR repository given a query energy signature in the form of SAX rule. At first, the classification approach works by searching for the query rule within the BoR. This is followed by scoring the confidence level of the queried rule to aid in the classification. The example scores of confidence levels range from 0 to 1.

In this example, a confidence score greater than 0.75 strongly favors the (predicted) classification output, whereas scores between 0.5 and 0.75 suggest a relatively weak classification, and scores less than 0.5 suggest that the output classification is weak. A simple scheme for scoring the confidence levels in classifying the query rules is described by Eq. (4).

\[
\text{Confidence Level Score} = \frac{\text{Frequency of the rule in classified output type}}{\text{Overall frequency of the rule in entire BoR}}
\]

5.3. Results and discussion

The results of classifying some example query rules into underlying hidden appliance and their associated events using the proposed subsequence classification approach are presented in Table 4.

Table 4 depicts the summary of observations drawn from the analysis of classification results.

The main observations are the following:

- The first three example query rules labelled [eebb-debb], [bfbf-cfbb] and [debb-edbb] are of complete query type. The classification outputs based on BoR strongly suggest that these query rules correspond to MFD-COPY, MFD-SCAN and MFD-PRINT respectively. This is because they all represent respective characteristic rules within BoR and confidence level in classifying them scores highest i.e. equal to 1.0.
- The query rule labelled [xxxx-edbb] is a partial query type. The classification approach strongly suggests that such a query rule be classified as MFD-PRINT. This is because such rules are relatively strongly prevalent among PRINT than COPY. Besides, such a subsequence pattern is found within the characteristic rule for MFD-PRINT.
- The query rule labelled [eebb-xxxx] is also a partial query type. The classification approach strongly suggests that such a query rule would be classified as MFD-COPY. This is because such a query is found only within MFD-COPY class and also part of its characteristic rule.
- The query rule labelled [fcbb] is also a partial query type. The classification model weakly suggests that such a subsequence pattern would be classified as MFD-PRINT. This is because such a query rule is equally likely to be found in both MFD-PRINT and MFD-SCAN indicated by their confidence score = 0.5.
- The query rule labelled [bfbf] is also a partial query type. The classification model however strongly suggests that such a subsequence pattern would be classified as MFD-SCAN. This is simple because it is part of the characteristic scan rule also indicated by confidence score = 1.
- This partial query rule [debb] is relatively weakly classified as MFD-COPY. This is because it is equally likely to be found in both MFD-PRINT and MFD-COPY indicated by their confidence score = 0.51.

The results obtained by the proposed Naïve classification model using BoR for appliance/events in aggregate signature are promising. The classification model adopts a supervised approach. Subsequence signature classification is assisted through a newer concept referred to as “Bag of Rules” in the context of office NILM. The extension of the existing concept of BoR for characterizing appliance events is a unique contribution of this study.

6. Conclusion

Disaggregation of office plug loads is still an open issue, and the time series measurements of office equipment offer many opportunities in discovering multiple hidden patterns and their corresponding operational states. This is one of the first studies addressing this particular challenge by analyzing subsequences of time series data. The SAX-based feature extraction helps in engineering appliance specific features for supervised classification.
The application of existing subsequence feature extraction tools help to create a corpus of grammar rules representing characteristic features in appliance energy signatures. In addition, they also indicate a potential for load disaggregation through a supervised classification approach. This is demonstrated through a Naive classification based on the novel "Bag of Rules" approach, as a proof-of-concept towards classifying unknown appliance signatures. However, the fact that the application of time-series subsequence mining towards office NILM is still at an early stage should be acknowledged. Nevertheless, the study has taught several lessons that are worth highlighting.

- Time series subsequence mining can help to better characterize transient appliance operational states. This is demonstrated by the example of discovering copy, scan and print operations in a MFD that presents characteristic signature patterns. This study is therefore a good step towards characterizing office plug load appliances.
- Discovering the individual appliance states within aggregate signatures using characteristic features or grammar rules is certainly possible.
- The interactive and intuitive design of existing open-source tools e.g. GrammarViz2.0, has enabled quick discovery of characteristic subsequences in a large time series data. Additionally, it is important to note that the characteristics of subsequences discovered are independent of the tool used.
- Classification using the idea of "Bag of Rules" as a proof-of-concept shows promising results in identifying individual appliances and their events within aggregate energy signatures.
- Moving towards true NILM in offices, the findings from this study need to be validated across several aggregate time series data obtained from other appliance types such as desktop PCs, laptops and monitors.
- In the context of OPLD, mining and classifying relevant subsequences automatically from several input time series data still remains a challenge.
- The proposed subsequence mining works on matching similarities in energy signatures exhibited by appliance’s transient events. This poses limitation on characterizing other appliances’ steady-states (e.g. desktop in On state, monitor in 100% brightness etc.) using this technique.

### Table 4

Results of classification of unknown rules within aggregate time series data using Naive approach.

<table>
<thead>
<tr>
<th>Example Query rules</th>
<th>(Predicted) Classification Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Query type (Partial or Complete)</td>
</tr>
<tr>
<td>[eebb-debb]</td>
<td>Complete</td>
</tr>
<tr>
<td>[eebb-cfbb]</td>
<td>Complete</td>
</tr>
<tr>
<td>[debb-edbb]</td>
<td>Complete</td>
</tr>
<tr>
<td>[eebb-xxxx]</td>
<td>Partial</td>
</tr>
<tr>
<td>[bcbbeb]</td>
<td>Partial</td>
</tr>
<tr>
<td>[fcbbeb]</td>
<td>Partial</td>
</tr>
<tr>
<td>[debbeb]</td>
<td>Partial</td>
</tr>
</tbody>
</table>

7. Future work

A potential next step towards office NILM is to mine and discover subsequences automatically from time series data, using OPLD or other relevant datasets that might become available. This will first require making appropriate choices of input SAX parameters for subsequence feature extraction and characterization adaptively. A relevant approach using the DIRECTL optimization algorithm has been found useful in similar scenarios [32]. Combining approaches such as tf*idf weighting from the Information Retrieval domain, together with SAX discretization, can automatically create huge weighted corpus of gram rules. Such large corpus of gramm rules might further improve the accuracy of classification. Another possible next step is to use appropriate classification techniques (e.g. Vector Space Model) for automatically classifying large corpus of SAX words (features). Moreover, the performance of the proposed approach needs to be benchmarked against other relevant approaches.

In addition to transient events discussed in this study, a complete solution towards office NILM should also consider steady appliance states. A simple statistical measure of appliance data using mean and variance from the target episodes can help to approximate the steady appliance operational states. A classification framework combining statistical measures for appliances’ steady states and grammar rules for appliances’ transient states is to be investigated.

### Acknowledgments

The authors would like to thank Singapore’s Ministry of Education for providing the financial support to conduct this research under AcRF grant R-296-000-145-112. A special thanks to Jyothsna Devi in helping collect, clean and prepare the dataset used for analysis in this paper.

### References
