

# SMELs: A Data-driven Middleware for Smart Miscellaneous Electrical Load Management in Buildings

Balaji Kalluri  
Future Cities Laboratory  
Singapore-ETH Centre  
Singapore  
kalluri@arch.ethz.ch

Sekhar Kondepudi, Tham Kwok  
Wai, Kua Harn Wei  
National University of Singapore  
Singapore  
bdgsnk@nus.edu.sg,  
bdgtkw@nus.edu.sg,  
bdgkuahw@nus.edu.sg

Andreas Kamilaris  
Department of Computer Science  
University of Twente  
The Netherlands  
a.kamilaris@utwente.nl

**Abstract**—Growth in Information and Communication Technology (ICT) has triggered an unprecedented proliferation of appliances a.k.a. Miscellaneous Electrical Loads (MELs) in buildings. Till now, managing MELs energy consumption in an optimum, cost-effective and intelligent manner in buildings remain an open-challenge. This article introduces a new supervised, data-driven middleware towards Smart Miscellaneous Electrical Load management in buildings (SMELs). It can perform automatic extraction, modeling and classification of the semantics of office appliances by analyzing aggregated electrical load signatures from several electrical outlets in the workplace environment. The results of analyzing more than 2,000 electrical load signatures from office workstations present classification performance ranging from 79.4% upto 95.8%. The preliminary findings from the study demonstrate the potential of SMELs as a middleware technology in Internet-of-Things (IoT) enabled smart buildings. The novelty of the proposed approach lies in combining the use of optimum sensors and existing data-driven techniques to extract detailed insights about appliances operation in real buildings.

**Keywords**—Miscellaneous electrical loads; middleware; building technology; temporal features; classification; office buildings

## I. INTRODUCTION

The growth in ICT over the years has transformed the nature of work across workplace environments. The proliferation of MELs such as personal computers, monitors, Multi-Functional Devices (MFDs) and VoIP phones in buildings on one hand improved productivity, but on the other hand affected the overall building energy trends. Evidences show MELs as one of the largest loads in commercial buildings [1]. Another research estimated more than 130,000 kWh energy equivalent to 100 tons of carbon-di-oxide savings, when good practices in the usage of MELs are implemented across a large university campus [2]. Although researchers apply several emerging techniques to intelligently measure and manage energy usage of MELs in buildings, several challenges

are still open in practice [3]. Most often building occupants lack the knowledge about operating appliances rationally [2]. Today, IoT and pervasive computing paradigm has multiplexed the role of MELs as not only building loads, but also as catalysts for energy conservation in buildings. Embedding appropriate middleware technology into existing Building Management System (BMS) enable managing buildings even more efficiently. There are no evidences for middleware that supports smarter management of MELs in buildings. A review on relevant pervasive computing middleware approaches from literatures such as SensorAct [4], Cassowary [5] presents a void in this regard.

In the current context, the middleware is a software layer that bridges end-use applications (e.g. email to remind an employee about rational appliance usage tips, goals and competition) with information about MELs operating modes (cf. Fig. 1). The interaction between occupants and MELs in office buildings consume energy. The information pertaining to pattern of usage of MELs are collected as temporal load signatures by multiple IoT-enabled sensors (e.g. smart plugs, occupancy detectors) deployed in the buildings for this purpose. Such information is passed onto the proposed middleware (SMELs) which processes the data from multiple workstations. The goal of SMELs is to facilitate end-use applications in future IoT deployments in smart buildings with information about (real-time) usage patterns of multiple office appliances by occupants. Thus, enable identifying opportunities and developing intelligent interventions to conserve energy wastes, especially at the human-machine interface using limited hardware sensors.

The purpose of this article is to introduce a new supervised, data-driven approach to develop a middleware for smart miscellaneous electrical load management in buildings. The proposed middleware helps automatically extract, model and classify the language or semantics of MELs obtained by monitoring the aggregated load signatures at several electrical outlets in workplace environment. To the best of our knowledge, this article is one of the first to introduce a data-driven middleware for this purpose.

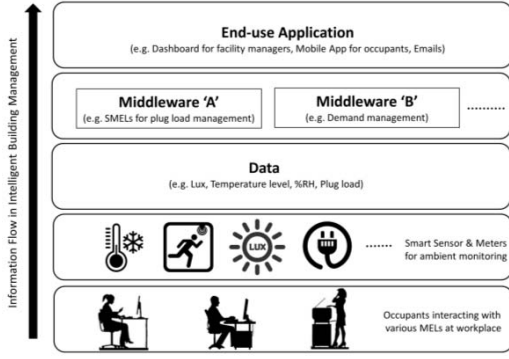


Fig. 1. Application of SMELs middleware in Intelligent Building Management

This paper is organized as follows: section II presents the background to the proposed approach to implement the middleware, section III describes the actual implementation of the middleware, section IV presents key findings from case study of SMELs applied to Office Plug Load Dataset (OPLD) [6] and section V presents the conclusion, limitation and possible next steps for future research in this direction.

## II. BACKGROUND AND PRIOR WORKS

### A. Challenges in energy management of MELs in office buildings

Monitoring every appliance on workstations (MELs) in an operational office building using individual IoT enabled meters, continuously for long duration is not only challenging, but also expensive and impractical [3]. Additional challenge lies in identifying the state-of-operation of MELs beyond simple on-off mode recognition using limited meters. To alleviate this problem, research efforts aimed at identifying individual appliances and their energy consumption breakdown from single point measurements are evolving since 1990s [7]. This classical approach is termed as *Non-Intrusive Load Monitoring* or otherwise called *load disaggregation*. The survey in literature shows several approaches to non-intrusive appliance mode identification for residences [8], but limited evidences in workplace/office environments [6, 9].

### B. Temporal Subsequence Mining applied to MELs

The continuous monitoring of MELs result in load signatures that are fundamentally temporal in nature. Moreover, characteristic signatures of appliance operations are relatively shorter in duration and hence lie hidden within subsequences. Time Series Subsequence (TSS) refers to short temporal patterns within time-series data. Mining such TSS patterns within temporal datasets across diverse fields can be found in literature [10]. However, there are limited evidences of its application to energy and built-environment in particular. The applicability of TSS mining to characterize and classify load signatures of a common office MELs (i.e. MFD) is first introduced in our previous work [11]. It describes several steps towards manual feature extraction, characterization and

supervised classification of temporal load signatures using GrammarViz2.0 [12]. The findings indicate the potential of Symbolic Aggregate approxImation (SAX) [13] to model temporal appliance features and further classify ground-truth operations of MELs from aggregated load signatures. However, the approach presented in [11] is not scalable across large MELs datasets due to manually supervised feature extraction. Our subsequent research work successfully extended the application of SAX-VSM [14] framework to classify MELs operations from a large dataset [15]. This framework intelligently combined unsupervised feature extraction, parameter optimization, SAX discretization and supervised classification. The results showed good accuracy (~92%) in identifying ground-truth operations of MFD from a large collection (>1700) of aggregated load signatures. The approach to implementing middleware presented in Section III shall help understand how SAX enables extracting, modeling characteristics subsequences and using them intelligently towards benchmark classification. Other variants of TSS mining such as PowerSAX [16] and shapelets [17] have also reported success in residential smart metering applications such as customer segmentation and energy consumption forecasting.

## III. PROPOSED MIDDLEWARE

The approach to implement the proposed middleware (SMELs) for appliance mode identification in workplace environment is discussed in this section. The schematic in Fig. 2 describes the data-driven process through various functional blocks across multiple stages.

### A. Intelligent temporal feature engineering

The semantics of MELs are obtained by collectively measuring the aggregated load at several electrical outlets in workplace environment. The OPLD is a large repository of labelled temporal load signatures of multiple office MELs from several individual workstations. Such a dataset forms the input (stage 1) to the middleware presented in Fig. 2. Specific characteristics of the dataset analyzed in this article is presented in Section IV.

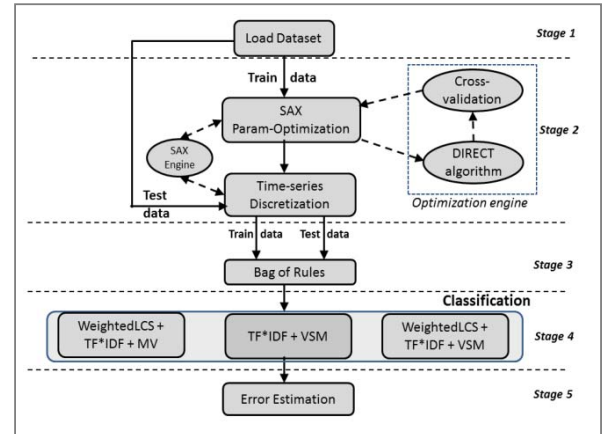


Fig. 2. Data-driven implementation of SMELs

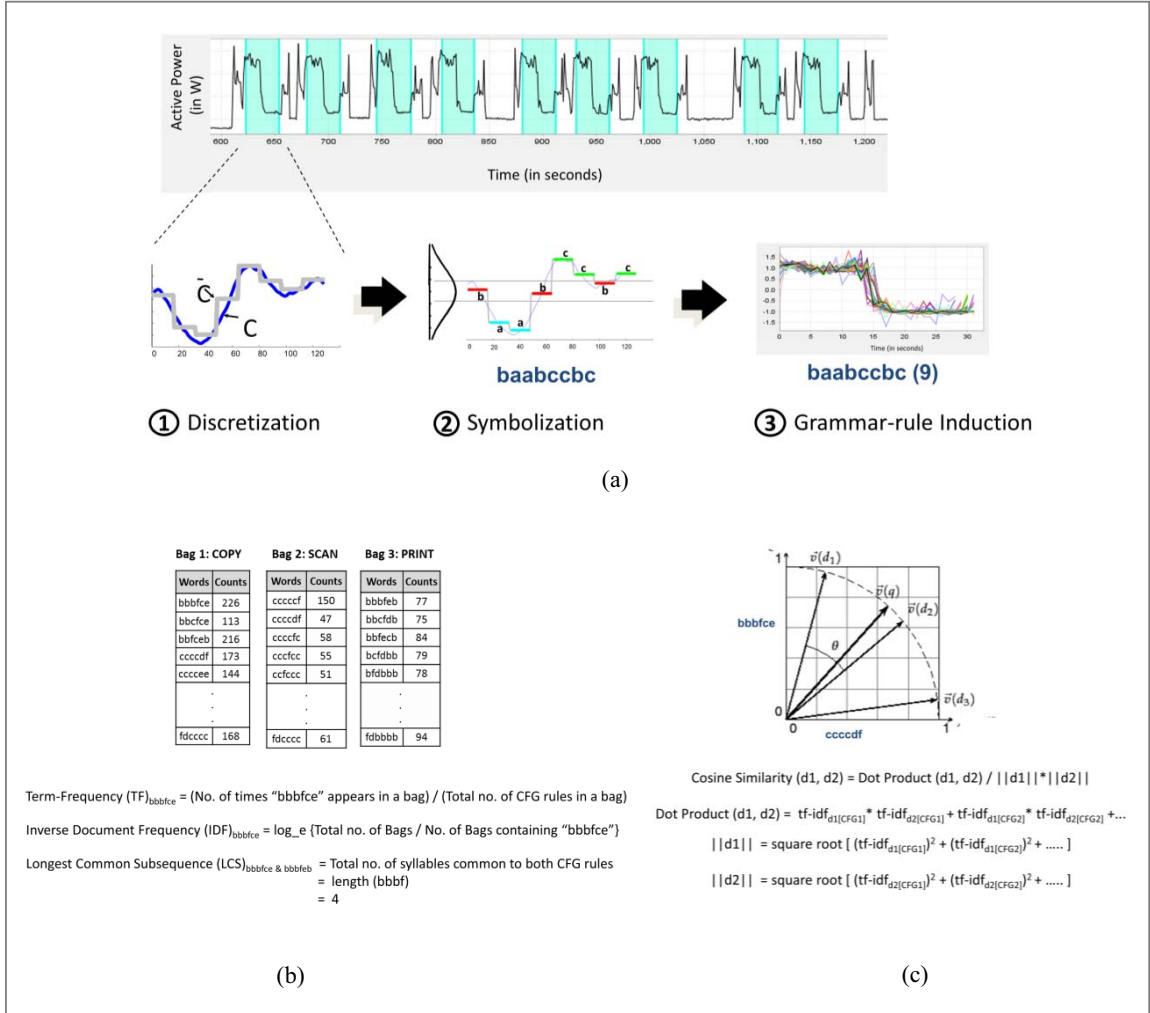


Fig. 3. (a) Illustration of feature engineering in SMELs using SAX is presented in 3 steps using an aggregate load signature as input from OPLD, with highlighted episodes representing MFD in *Copy* operation to be extracted; (b) An example of term-frequency, inverse document frequency and longest common subsequence computation using Bag-of-Words; (c) A vector space representation of temporal load signatures and cosine similarity measure to classify them using Vector Space Model

This is followed by the process of transforming temporal load signatures of appliances into intelligent features for improved appliance mode identification. The intelligent feature engineering process in SMELs is based on an existing TSS mining technique called SAX [13]. It follows three main steps as intuitively described in Fig. 3(a) namely: temporal discretization, symbolization and grammar-rule induction. A temporally aggregated load signature (600sec) when a desktop PC, a monitor and a MFD operate concurrently across different modes of operation is presented as an example. The highlighted subsequences within the input episode present similarity in the ground-truth appliance operations (e.g. MFD in *Copy* operation), exhibited as approximate (temporal) signature similarity. The purpose of feature engineering step is to discover and model such similarities from a large, labelled appliance load signature repository. This is done by sampling temporal load signatures into relatively smaller subsequences

using sliding window technique with window size,  $w=120$ . Subsequently, these continuous-time subsequences are discretized using Piecewise Aggregate Approximation (PAA) technique with PAA size,  $p=8$ . This is described as discretization step in Fig. 3(a). The resulting discretized subsequences in the next step are symbolized with alphabet size,  $a=3$  and thus transformed into Context-Free-Grammar (CFG) rules e.g. *baabccbc* through symbolization. This is followed by discovering, extracting and modeling several approximately similar subsequences from input dataset based on matching CFG rules. In the example, nine similar CFG rules are discovered from the example input aggregate load signature. This step is termed as grammar-rule induction. Together all these three steps represent SAX feature engineering. It is important to note that SAX is a parameterized technique and determining the optimum parameters ( $w$ ,  $p$ ,  $a$ ) follows  $n$ -fold cross validation and DIRECT optimization

algorithm as implemented in [14]. Collectively these steps represent stage 2 in SMELs and it results in intelligent appliance features called CFG rules.

A large repository of input temporal load signatures is intelligently processed using SAX<sup>1</sup>, resulting in a huge collection of labelled CFG rules with appliance modes as labels. Collectively they form Bag-of-Rules (BoR) dictionary for OPLD as shown in Fig. 3(b). This process is labelled stage 3 in Fig. 2. Typically, BoR consist of multiple bags composed of several instances of CFG rules together with their frequency count. Each bag corresponds to individual ground-truth appliance operation modes. These CFG rules are employed as “intelligent” features in the subsequent stages. Specific details on employing these features towards improved appliance (modes) recognition is discussed in the following section.

### B. Hidden appliance mode recognition via classification

Intelligent energy management of MELs in office workstations requires disambiguating/classifying individual, hidden appliance operation modes from aggregate load signatures. This is the next step in the data-driven process (stage 4 in Fig. 2). Classifying unlabeled, aggregate load signatures containing characteristic episodes of appliances (and their ground-truth operations) require sophisticated techniques. In SMELs, several techniques such as Longest Common Subsequence (LCS), Term-Frequency (TF), Inverse-Document Frequency (IDF), Vector Space Model (VSM) and Maximum Voting (MV) are employed for this purpose. The implementation differs from our previous work [15] by including multiple approaches based on sub-string similarity towards improved classification and performance benchmarking. Three benchmark classification approaches introduced in the classification step (i.e. stage 4) are as follows:

- Approach 1: Feature weighting based on combined LCS and product of TF and IDF but classification based on MV.
- Approach 2: Feature weighting based on product of TF and IDF but classification based on VSM.
- Approach 3: Feature weighting based on combined weighted LCS and product of TF and IDF but classification based on VSM.

The TF represents number of times a specific rule appears in an appliance (mode) class or a bag. On the other hand, IDF represents the specificity of a rule to an appliance (mode) class or a bag. The product of TF and IDF for each rule gives its relative significance to a particular appliance (mode) class. However, LCS differs from the former by considering the substring similarity between two CFG rules. Also, it encapsulates the length of the longest common sub-string. Fig. 3(b) presents the definition of all three feature weighting measures with example of CFG rules from BoR. The next step in SMELs is using the labelled features from BoR towards supervised classification of the unlabeled (test) data. The stages 1, 2 and 3 described above help transform the temporal load signatures into features suitable for representing as vectors.

<sup>1</sup> Implementation inherited from <https://github.com/jMotif/jmotif-R>

This facilitates applying one of the most effective classification techniques from Information Retrieval known as VSM [18]. It helps represent several temporal load signatures in the dataset as vectors in a multi-dimensional vector space and classify them. A simplified two-dimensional representation of a multi-dimensional vector space consisting of both labelled (train) and unlabeled (test) vectors is presented in Fig. 3(c). The vectors labelled  $v(d1)$ ,  $v(d2)$  and  $v(d3)$  denote labelled vectors whose appliance mode class is known apriori, while vector labelled  $v(q)$  denote unlabeled vector whose appliance mode class is to be determined. The cosine similarity which is a measure of distance between vectors in the multi-dimensional vector space is used in classifying the unlabeled vector. The mathematical expression of cosine similarity measure involves the product of TF-IDF weights of individual features constituting a vector as shown in Fig. 3(c). On the other hand, SMELs also integrates simple MV based classification approach in order to facilitate classifying features that match partially between test and train vectors. It computes an aggregate score based on both fully matched features (using TF-IDF) and partially matched features (using LCS) and classifies the test data based on maximum similarity score (votes). Finally, the classification error from multiples approaches are analyzed to benchmark the performance (stage 5 in Fig. 2). One of the key contributions of SMELs is enabling benchmark the performance of classifying ground-truth appliance modes from aggregated load signatures using multiple TSS approaches. The software implementation of the middleware is made in Java and is based on open-source SAX-VSM framework [14]. Typically, running SMELs on a personal computer to train and classify a dataset with 2000 load signatures takes few minutes to complete.

## IV. RESULTS AND DISCUSSION

The aggregate load signatures from OPLD when a desktop or laptop PC, a monitor and an MFD operate concurrently in office workstation is considered for analysis. The analysis of performance of SMELs presented in this section considers disambiguating or classifying multiple, hidden operational modes of an individual office appliance (i.e. MFD) from the aggregated dataset. Table I presents various operational modes of several office appliance categories considered for analysis. It shows three operational modes for desktop or laptop PC (viz. On, Sleep, Off), three operational modes for monitor (viz. 100% Brightness, 50% Brightness, Off) and five operational modes for MFD (viz. Idle, Copy, Scan, Print, Off). For the purpose of illustration, Table I also shows 19 legitimate use-case scenarios when desktop PC, monitor and MFD concurrently operate. However, the analysis presented in this study considers three common appliance combinations (cf. Table II) found in a typical office workstation namely:

- Desktop PC with Monitor and MFD, labelled as D+M+P
- Laptop PC with Monitor and MFD, labelled as L+M+P
- Laptop PC with MFD, labelled as L+P

The analysis of classifying ground-truth operations of three different instances of MFD viz. P1, P2 and P3 (from above concurrent appliance operations) are analyzed in

TABLE I. USE-CASE COMBINATIONS WITH DESKTOP PC, MONITOR AND MFD IN OPERATION

Use case #	Desktop or Laptop PC:	Monitor:	MFD:
	• On • Sleep • Off	• 100% Brightness • 50% Brightness • Off	• Idle • Copy • Scan • Print • Off
1	On	100% B	Idle
2	On	100% B	Copy
3	On	100% B	Scan
4	On	100% B	Print
5	On	50% B	Idle
6	On	50% B	Copy
7	On	50% B	Scan
8	On	50% B	Print
9	On	Off	Idle
10	On	Off	Copy
11	Sleep	Off	Idle
12	Sleep	Off	Copy
13	On	100% B	Off
14	On	50% B	Off
15	Sleep	Off	Off
16	On	Off	Off
17	Off	Off	Idle
18	Off	Off	Copy
19	Off	Off	Off

this section. Collectively, more than 2000 aggregate load signatures across all combinations are analyzed using SMELs and the results are summarized in Table II. Each disaggregation use-case in the table have few hundreds of labelled aggregate load signatures (#233, #230, #242 etc.) for analysis. Each signature in them consist of several subsequences, each representing a combination of multiple appliance operations (cf. Table I). The goal is to recognize hidden ground-truth operational modes of MFD from the unlabeled (test) dataset. For example, classifying 1292 operational modes (i.e. 68\*19) of P1 from 68 test signatures based on 3135 trained features

(i.e. 165\*19) from 165 training signatures in use-case 1 in Table II. The analysis presented in the table is categorized based on two approaches to split the entire input dataset: (a) train=70% and test=30%; (b) train=80% and test=20%. The column labelled *classification measures* represent distinct SAX parameters for each use-case (in [w, p, a] format) together with the breakdown of signature count for training and testing. These SAX parameters are obtained using optimization engine as described in the previous section. The percentage of unlabeled (test) signatures classified positively are also presented as results in the table. The following are some observations and findings drawn.

- Depending on the nature of splitting the dataset, SAX parameters vary. For example, use-case 1 presents either [25,6,7] or [27,7,6] corresponding to 70-30 or 80-20 split in the input dataset.
- Deciphering operational modes of MFDs using approach 3 marginally outperforms approach 2, with 70-30 split in the dataset when compared to 80-20 split. This suggests that integrated feature selection based on both complete and partial substring similarity marginally outperforms feature selection based on complete substring similarity, especially when classification is based on VSM. However, the same is not true when classification is based on MV. This suggests the potential of intelligent feature engineering approach that combines LCS and TF-IDF measures.
- Clearly, most use-cases under both approaches 2 and 3 performs better, with performance exceeding 80%. However, in the case of approach 1 only few use-cases present performance exceeding 80%. This indicates that VSM based appliance mode classification is better

TABLE II. SUMMARY OF RESULTS: CLASSIFYING MULTIPLE OPERATIONAL MODES OF MFDs FROM A LARGE AGGREGATE DATASET

Case No.	Disaggregation Use-Cases	Dataset Split : Train (70%) + Test (30%)			Dataset Split : Train (80%) + Test (20%)				
		Classification Measures	Approach 1	Approach 2	Approach 3	Classification Measures	Approach 1	Approach 2	Approach 3
1	P1 in D+M+P (#233)	[25, 6, 7] #Train = 165 #Test = 68	66.2%	100%	98.5%	[27, 7, 6] #Train = 186 #Test = 47	72.3%	100%	97.8%
2	P2 in D+M+P (#230)	[25, 8, 8] #Train = 160 #Test = 70	100%	100%	100%	[23, 8, 6] #Train = 184 #Test = 46	97.8%	100%	100%
3	P3 in D+M+P (#242)	[32, 8, 7] #Train = 170 #Test = 72	80.5%	95.8%	97.2%	[25, 7, 7] #Train = 194 #Test = 48	75%	97.9%	87.5%
4	P1 in L+M+P (#290)	[25, 6, 7] #Train = 200 #Test = 90	74.4%	91.1%	93.3%	[18, 7, 7] #Train = 232 #Test = 58	63.7%	98.2%	98.2%
5	P2 in L+M+P (#284)	[19, 7, 7] #Train = 200 #Test = 84	76.2%	75%	82.1%	[18, 8, 7] #Train = 227 #Test = 57	77.2%	78.9%	84.2%
6	P3 in L+M+P (#275)	[25, 6, 7] #Train = 195 #Test = 80	75%	91.2%	92.5%	[25, 6, 7] #Train = 220 #Test = 55	80%	94.5%	92.7%
7	P1 in L+P (#214)	[32, 7, 6] #Train = 150 #Test = 64	76.5%	93.7%	93.7%	[26, 7, 6] #Train = 171 #Test = 43	69.7%	100%	100%
8	P2 in L+P (#163)	[18, 6, 7] #Train = 115 #Test = 48	89.6%	91.7%	89.6%	[25, 7, 6] #Train = 130 #Test = 33	100%	95.2%	96.9%
9	P3 in L+P (#215)	[26, 7, 6] #Train = 150 #Test = 65	84.6%	84.6%	86.2%	[32, 8, 6] #Train = 172 #Test = 43	79.1%	97.6%	97.6%

than MV when using temporal features.

- Overall, the preliminary benchmark analysis of more than 2,000 aggregate load signatures from multiple office workstations using SMELs present classification performance ranging between 79.4% and 95.8%. This shows the success of the proposed middleware in disambiguating the hidden appliance operations in aggregated load signatures.

## V. CONCLUSION, LIMITATION AND FUTURE WORK

An approach to elucidate the semantics of office appliances hidden in aggregate load signatures from multiple workstations is presented using SMELs. The novelty lies in developing an approach to implement a middleware combining optimum IoT sensors and existing data-driven techniques to facilitate improved appliances (mode) monitoring in office buildings. The preliminary results demonstrate the success of deciphering various operational modes of a common office appliance from a huge collection of aggregate load signatures collected from office workstations (OPLD). In particular, feature engineering approach combining LCS and TF-IDF is found to better model characteristic load signatures for supervised classification. Additionally, classification based on VSM presented better performance over simple MV approach, especially in analyzing large temporal datasets.

In the realm of pervasive computing and IoT, embedding such middleware into the fabric of BMS shall facilitate intelligent applications that require understanding MELs operations in real-time at personalized occupant-level. We envision the future of smart buildings with cooperative and intelligent energy management of MELs in workplaces. This combines use of limited sensors, cloud-based implementation of SMELs middleware, real-time appliance mode identification and improved eco-feedback messaging services.

However, the proposed middleware is still very nascent and pose certain limitations for its real-world application. Firstly, results presented in this article is limited to identifying MFD operational modes alone, which exhibits characteristic signature patterns. Extending the application to other office appliances (e.g. desktops, laptops, monitors) that exhibit relatively steady load signatures is a subsequent step in developing the middleware. A potential approach in this direction is an ensemble of multiple classification approaches based on intelligent features obtained. For example, the operational modes of MFD are relatively short-lived and shall be classified based on similarities between subsequences. On the other hand, the operational modes of other appliances such as desktops, laptops and monitors are relatively steady and long-lived and shall be classified based on similarities between complete temporal sequences in load signatures. Secondly, the proposed approach is supervised using labelled load signatures from handful of appliances and their combinations present in OPLD. Thus, the performance of the proposed middleware beyond the appliance instances in OPLD is not validated. Collaboratively upscaling OPLD through crowd and open sourcing will address this gap. Finally, in the IoT paradigm, future research shall validate SMELs in real world through cloud-based implementation.

## REFERENCES

- [1] K. McKenney, M. Guernsey, R. Ponoum, and J. Rosenfeld, "Commercial miscellaneous electric loads: Energy consumption characterization and savings potential in 2008 by building type," TIAX LLC, Lexington, MA, Tech. Rep. D, 498, 2010, pp. 224.
- [2] A. Kamilaris, D.T.H. Ngan, A. Pantazaras, B. Kalluri, S. Kondepudi, and T. Kwok Wai, "Good practices in the use of ICT equipment for electricity savings at a university campus," In: Proceedings of Green Computing Conference, pp. 1-11, Dallas, Texas, USA, 2014. IEEE
- [3] A. Kamilaris, B. Kalluri, S. Kondepudi, and T. Kwok Wai, "A literature survey on measuring energy usage for miscellaneous electric loads in offices and commercial buildings," *Journal for Renewable and Sustainable Energy Reviews*, vol. 34, 2014, pp. 536-550.
- [4] P. Arjunan, N. Batra, H. Choi, H. Singh, P. Singh, and M. B. Srivastava, "Sensoract: a privacy and security aware federated middleware for building management," In: Proceedings of the fourth ACM workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, Toronto, Ontario, Canada, 2012, pp. 80-87. ACM
- [5] P. Kathiravelu, L. Sharifi, and L. Veiga, "Cassowary: Middleware platform for context-aware smart buildings with software-defined sensor networks," In: Proceedings of the second workshop on Middleware for Context-Aware Applications in the IoT, Vancouver, BC, Canada, 2015, pp. 1-6. ACM
- [6] B. Kalluri, S. Kondepudi, H. W. Kua, T. Kwok Wai and A. Kamilaris, "OPLD: Towards improved non-intrusive office plug load disaggregation," In: Proceedings of IEEE International Conference on Building Efficiency and Sustainable Technologies, Singapore, pp.56-61. IEEE
- [7] G. W. Hart, "Nonintrusive appliance load monitoring," In: Proceedings of the IEEE, vol. 80, 1992, pp. 1870-1891. IEEE
- [8] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol.12, 2012, pp. 16838-16866.
- [9] A. Rogriguez, S. T. Smith, A. Kiff, and B. Potter, "Small power load disaggregation in office buildings based on electrical signature classification," In: Proceedings of International Energy Conference (ENERGYCON), Leuven, Belgium, 2016, pp.1-6. IEEE
- [10] T. c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, 2011, pp. 164-181.
- [11] B. Kalluri, A. Kamilaris, S. Kondepudi, H. W. Kua, and T. Kwok Wai, "Applicability of using time series subsequences to study office plug load appliances," *Journal for Energy and Buildings*, vol. 127, 2016, pp. 399-410.
- [12] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, S. Frankenstein, and M. Lerner, "Grammarviz 2.0: a tool for grammar-based pattern discovery in time series," In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, vol. 3, 2014, pp. 468-472. ACM
- [13] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Journal for Data Mining and Knowledge Discovery*, vol. 15, 2007, pp. 107-144.
- [14] P. Senin, and S. Malinchik, "SAX-VSM: Interpretable time series classification using sax and vector space model," In: Proceedings of IEEE conference in Data Mining, Dallas, TX, USA, 2013, pp. 1175-1180. IEEE
- [15] B. Kalluri, A. Kamilaris, S. Kondepudi, H. W. Kua, and T. Kwok Wai, "Classifying Office Plug Load Appliance Events in the context of NILM using Time-series Data Mining," In: Proceedings of the 12th REHVA World Congress and CLIMA, Aalborg, Denmark, vol. 6, 2016.
- [16] A. Reinhardt, and S. Koessler, "PowerSAX: Fast motif matching in distributed power meter data using symbolic representations," In: Proceedings of IEEE 39th Conference on Local Computer Networks Workshops, Edmonton, AB, Canada, 2014, pp. 531-538. IEEE
- [17] O. P. Patri, A. V. Panangadan, C. Chelms, and V. K. Prasanna, "Extracting discriminative features for event-based electricity disaggregation," In: Proceedings of IEEE Technologies for Sustainability (SusTech), Portland, OR, USA, 2014, pp. 232-238. IEEE

- [18] C. D. Manning, P. Raghavan, and H. Schütze, "Scoring, term weighting, and the vector space model," In: Introduction to Information Retrieval, 2008, pp. 100-123.