Applied Energy 275 (2020) 115391

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Near-real-time plug load identification using low-frequency power data in office spaces: Experiments and applications

Zeynep Duygu Tekler[a,*], Raymond Low[b], Yuren Zhou[a], Chau Yuen[a], Lucienne Blessing[a], Costas Spanos[c]

[a] Engineering Product Development, Singapore University of Technology and Design, Singapore
[b] Engineering Systems and Design, Singapore University of Technology and Design, Singapore
[c] Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA

HIGHLIGHTS

- A real-time plug load identification approach was proposed using low-frequency data.
- The robustness of the approach was evaluated under different experimental settings.
- The best online model reported accuracies up to 93% with response time of 5 min.
- Energy dashboards and control systems were highlighted as future application areas.
- The plug load dataset collected in this study was made publicly available online.

ARTICLE INFO

Keywords:
Plug load identification
Intrusive load monitoring
Low sampling frequency
Smart plugs
Energy management
Office spaces

ABSTRACT

Plug loads account for up to one-third of the overall energy use in commercial buildings. There is thus a growing research interest in utilising load monitoring systems to track plug load usage by installing smart plugs to capture high-resolution consumption data. The availability of such data has also enabled the development of automatic plug load identification models that enhance the capabilities of existing load monitoring systems. Through our literature review, we highlighted several limitations that impede real-world implementation, such as the limited number of publicly available datasets for commercial buildings, models trained on data with high sampling frequencies while using an extended time window, and data leakage issues during model training. In this study, we proposed a near-real-time plug load identification approach that uses low-frequency power data (1/60 Hz) to identify plug loads in office spaces. The dataset used in this study is processed by first identifying the active periods of the plug loads before applying a novel dynamic time window strategy during feature extraction. These extracted features are subsequently passed through several classification algorithms and evaluated using different accuracy metrics. The proposed approach is also assessed through multiple experiments, including (1) identifying the best online and offline models, (2) comparing between different time window strategies, and (3) evaluating model performances under different sampling frequencies. As a result, the best online model achieved accuracies up to 93% using the Bagging algorithm with a minimum dynamic time window of 5 minutes. Finally, we highlighted two application areas of automatic plug load identification in energy dashboards and personalised control systems as part of future works.

1. Introduction

The energy contribution of plug loads has been steadily rising over recent years, accounting for up to 33% of the overall energy use in commercial buildings [1]. This trend is particularly true in high-efficiency buildings where the development of energy-efficient designs for HVAC and lighting systems has resulted in a gradual decline in their energy contributions, driven by code mandates and industry standards. On the other hand, the development of energy-efficient plug loads remains relatively unregulated, leading to an overall increase in their

* Corresponding author.
E-mail address: duygutekler_zeynep@mymail.sutd.edu.sg (Z.D. Tekler).
URL: https://github.com/zeynepduygutekler/plug-load-dataset (Z.D. Tekler).

https://doi.org/10.1016/j.apenergy.2020.115391
Received 27 April 2020; Received in revised form 27 May 2020; Accepted 9 June 2020
Available online 30 June 2020

relative energy contributions when compared to other energy systems. Based on these energy trends, a recent projection published by the U.S. Energy Information Administration predicts an average 2.6% increase in energy use per year for office equipment between 2019 and 2050 [2]. Due to the lack of a standardised definition in the literature, we define plug loads as electrical devices that draw power from a building's electrical sockets, excluding conventional heating, cooling, and lighting loads in the building.

1.1. Load monitoring approaches and applications

Given the rise in plug load energy contribution in commercial buildings, there is a growing research and commercial interest in utilising load monitoring systems to efficiently track and manage plug load energy use through various applications. The two main approaches used in existing load monitoring systems can be categorised under non-intrusive load monitoring (NILM) and intrusive load monitoring (ILM) [3].

NILM involves installing a power meter at the building's main electrical panels to obtain the aggregated power consumption data at the floor or building level. This aggregated data is subsequently passed through a load disaggregation algorithm to identify the operational states of individual plug loads in the building based on their unique power consumption patterns [4]. NILM has been a popular and low-cost option among researchers as it does not intrude upon the occupants during data collection and only requires the installation of a single power meter for the entire floor or building [5]. However, the performance of load disaggregation algorithms is significantly affected by the number of plug loads that are operating simultaneously due to the increased combination of consumption patterns to be identified, as well as signal distortion due to overlapping use [6]. Moreover, it was highlighted by Antonio et al. [7] that the NILM approach tends to perform poorly in detecting plug loads that do not consume a significant amount of power. In the context of office buildings where a large number of similar plug loads can be found operating simultaneously (i.e., laptops and monitors), the lack of a differentiable power consumption pattern between these plug loads poses significant challenges when attempting to perform load disaggregation.

With the recent advancements in sensing and communication technologies, the reduced cost of sensors and improved device integration with smart home systems have led to the rising popularity of the ILM approach [8]. ILM involves installing smart power plugs directly at each electrical socket, which allows high-resolution and accurate energy consumption information of individual plug loads to be collected, due to the elimination of the load disaggregation step. The availability of this data has also opened the doors for exciting applications [3] such as:

- **Activity Recognition:** Inferring the occupants' activity patterns based on the usage of specific plug loads.
- **Occupancy Detection:** Inferring the occupants' presence based on the operation of specific plug loads located in different parts of the building [6].
- **User-Appliance Interaction:** Analysing the occupants' interaction with different plug loads such as their usage frequency, duration of use, and switch-off behaviours [9].
- **Automatic Plug Load Identification:** Automatically identifying the plug load's operational state (ON/OFF) and type (desktop, monitor, laptop) based on its power consumption patterns.

1.2. Automatic plug load identification

Among the above applications listed, automatic plug load identification has useful implications on effectively managing plug loads to promote energy savings in buildings. A detailed breakdown of the energy contributions of each plug load on an energy dashboard provides occupants with real-time feedback on their energy consumption levels to encourage them to reduce their energy footprint [8,10,11]. This approach is commonly known as eco-feedback. Furthermore, the availability of this information would facilitate the design of conservation initiatives that can be targeted towards reducing energy wastage from high-energy loads. The fine granularity of the consumption data at the temporal level also allows building managers to deploy load scheduling strategies for optimal energy utilisation within the building [12], as well as perform load profiling and benchmarking between different buildings [13,14].

Many studies have proposed various automatic plug load identification approaches based on the data collected using an ILM approach. Some of the features used during plug load identification include instantaneous voltage, root-mean-square voltage (V_{RMS}), root-mean-square current (I_{RMS}), active power (P), reactive power (Q), apparent power (S), frequency, peak voltage (V_{peak}), peak current (I_{peak}), as well as the current, voltage, and power harmonics. However, our review of past plug load identification studies conducted in commercial buildings (refer to Section 2) reveal several limitations that impede the practical implementation of these models in real-world settings. The first limitation is related to the fact that most publicly available datasets were collected in the context of residential buildings, which narrows down the number of public datasets that can be used for model development in commercial buildings. Furthermore, most of the plug load identification models proposed in past studies were trained on datasets that were collected using sensors with relatively high sampling frequencies (e.g., 1/2 Hz). During a building-wide implementation, the use of such high-frequency sensors will result in a substantial cost due to data storage. Some models also reportedly use an extended time window (e.g., 1 hour) when performing plug load identification, which is considered too long and impractical for online systems that often require the identification process to occur in real-time or near-real-time settings. Moreover, this limitation results in inaccurate identifications for plug loads that are only connected for a short period. The term "time window" refers to the duration of time where historical consumption data is collected and processed to perform plug load identification. Therefore, data must be collected for the duration of the time window before the first identification is made when the plug load is initially connected to the smart plug. Finally, due to the time-series nature of the plug load dataset, some studies have also adopted unsuitable approaches when dividing the dataset during model development and evaluation, leading to data leakage and the reporting of overly optimistic model performance.

1.3. Objective and contributions of the study

The objective of this study is to perform automatic identification of different plug loads typically found in office spaces, based on their power consumption patterns under low sampling frequencies and within a short time interval.

The main contributions of this paper are as follows:

- A near-real-time plug load identification approach was proposed with an accuracy of up to 93% and was capable of making its first identification within 5 min while using low-frequency power consumption data (1/60 Hz).
- A comprehensive evaluation was conducted to assess the robustness of our proposed approach through multiple experiments. These experiments include (1) identifying the best models for online and offline settings, (2) comparing between dynamic and fixed time window strategies, and (3) evaluating model performance under different sampling frequencies.
- The dataset used in this study was collected by retaining the real-world usage patterns of different plug loads typically found in an office environment and has been made publicly available to contribute to the existing repository of plug load datasets.

2. Literature review

This section provides a critical review of several studies found in the literature that used smart plugs to obtain high-resolution consumption data for performing automatic plug load identification. The review also covers a brief description of the datasets that were used during model development.

A study conducted by Reddy et al. [15] proposed a plug load identification model capable of performing three levels of identifications, including (1) identifying the model of the individual plug load, (2) identifying different plug load types, and (3) identifying the device state (ON/OFF). The dataset used in this study contains the power consumption patterns of 90 plug loads categorised into nine different plug load types commonly found in educational buildings and has a sampling frequency of 1/2 Hz. Several classification algorithms were used to develop the identification model, including K-Nearest Neighbours (K-NN), Support Vector Machine (SVM), Naive Bayes (NB), Logistic Regression (LR), and Random Forest (RF), where the best performing model reported an accuracy close to 100%. This work was extended when Reddy et al. [16] proposed another plug load identification model that was able to identify the plug load type and state (ON/OFF) under different operating conditions, such as different voltage levels and operational states. The dataset used in this study contains the power consumption patterns of 70 plug loads grouped into seven classes, including desktop, kettle, laptop, monitor, projector, printer, and network switch. The data was collected at a sampling frequency of 1/5 Hz (equivalent to 1 sample every 5 seconds) using a custom smart power strip. Using a hybrid combination of regression analysis and weighted K-NN, the proposed model outperformed other baseline algorithms by reporting accuracies up to 100% and 79% during random cross-validation and device-wise cross-validation, respectively.

Another related work was conducted by Abeykoon et al. [17], who proposed a prototype data acquisition platform that collected the power consumption information of individual plug loads found in a residential apartment and performed real-time plug load identification based on their power consumption patterns. In the end, the Silhouette and K-means algorithms reported accuracies up to 98%, outperforming other supervised and unsupervised classification algorithms considered.

Ridi et al. [18] also conducted several plug load identification studies using the Appliance Consumption Signature-Fribourg 2 (ACS-F2) database. The ACS-F2 database contains the power signatures of 225 plug loads from 15 different categories and is an extension of the ACS-F1 database, which was released one year earlier in 2013 [19]. The power consumption information of each plug load is recorded at a sampling frequency of 1/10 Hz (equivalent to 1 sample every 10 s) using smart power plugs for one hour per session, over two separate sessions. By using classification algorithms such as K-NN, Gaussian Mixture Models (GMM), and Hidden Markov Models (HMM), the author was able to report classification accuracies of up to 94% [18,20]. A time window of one hour was also used during the feature extraction step to capture the dynamic characteristics of the plug load.

Finally, one of the most recent plug load identification studies was conducted by Tundis et al. [21], who proposed the use of a set of 19 input features, including the power consumption data, temporal usage, and location information to perform plug load identification. The model was trained on 33 different plug load types and was evaluated between different classification algorithms including RF, Bagging, LogitBoost (LB), Decision Tree (DT), NB, and SVM before concluding that the RF algorithm was able to report the highest classification accuracy of 96.51%.

Despite the excellent model performances reported by these studies, several limitations impede the implementation of these proposed models into real-world settings.

The first limitation is the fact that many of the proposed models are trained on datasets that are collected using sensors with relatively high sampling frequencies of up to 1/2 Hz (equivalent to 1 sample every 2 seconds). While this sampling frequency is useful in capturing the minute transition behaviours of the plug load leading to better identification performance, the large amount of data that has to be stored as a result of the high sampling frequency translates to high storage costs, in the case of a building-wide implementation. Therefore, load monitoring systems tend to be set at a lower sampling frequency to reduce the cost of data storage, and the performance of the identification model should also be robust under such low sampling conditions.

The second limitation is the use of an extended time window (e.g., 1 hour) when extracting input features that require historical data (i.e., maximum power, average power, minimum power) [18,20]. In the case of a real-time system, the use of an extended time window will result in significant delays before the model can identify the plug load and also leads to inaccurate identification if the plug load is only connected for a short period.

The last limitation is related to the issue of data leakage. In the field of machine learning, data leakage occurs when data from the test set is unknowingly revealed to the model during the training phase. Since the model is subsequently evaluated on previously encountered data, this will cause the model to report overly optimistic test results that are not reflective of the model's actual performance and prevent a fair comparison between different plug load identification approaches [15–17]. In this case, where time-series data is considered, the input features are calculated based on historical plug load consumption data that was recorded in the previous timesteps, thereby creating input features that are highly correlated and, in some instances, are identical within the same period. By performing a random split to obtain the training and test datasets, these highly correlated sets of input features will be randomly assigned between both datasets, leading to data leakage.

3. Methodology

This section provides a detailed description of the proposed plug load identification approach, starting with a data collection exercise to obtain the power consumption data of several plug loads typically found in an office space, to the data pre-processing steps taken for feature extraction and model development. Several performance metrics are also proposed to evaluate the resulting model performance.

3.1. Data collection

The first part of our proposed approach begins with a data collection exercise in a typical office environment to obtain the power consumption signatures of several plug loads commonly found on the occupants' desks. The power consumption data was captured in a naturalistic setting and comprised of different plug loads, including laptops, desktops, monitors, fans, and task lamps. A total of 36 occupants participated in this study consisting of a mixture of researchers and administrative staff. Within the study area, every two connecting desks are equipped with fifteen electrical sockets, where each electrical socket is connected to a ZigBee-based smart power plug from Plugwise [22] to monitor the power consumption of any connected plug load (refer to Fig. 1). Each smart plug is also equipped with a ZigBee module that transmits data to a nearby gateway device within the same ZigBee network and contains a relay that allows the plug load to be switched ON/OFF remotely. The gateway device is a Raspberry Pi, equipped with a Plugwise Zigbee dongle that allows it to query each smart power plug within the network for the plug load's instantaneous power at a sampling frequency of 1/60 Hz (equivalent to 1 sample per minute). The data received is then transmitted to a central server through TCP/IP via Ethernet connection [23]. The central server is implemented based on the OSIsoft PI database [24], which stores the power consumption data and allows users to remotely extract the data for subsequent processing using various RESTful API services [25]. Fig. 2 provides a graphical representation of the ZigBee network and the OSIsoft PI database.

Based on the setup described above, three weeks of data collection

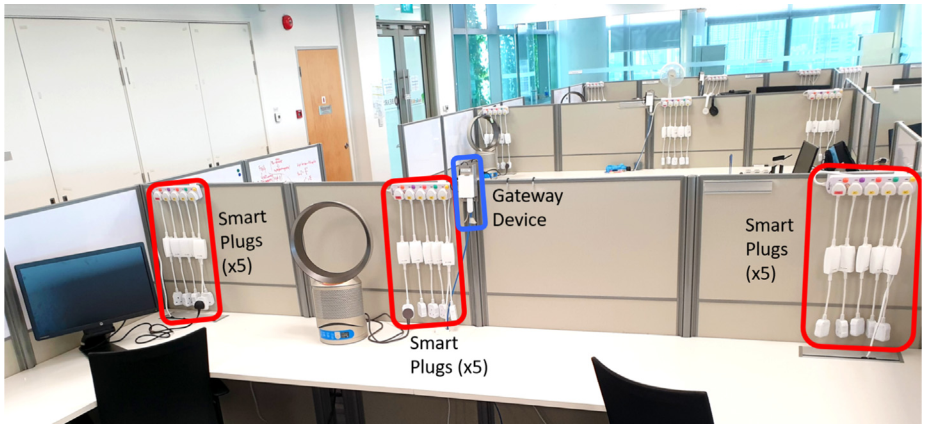

Fig. 1. Layout of the study area. Every two connecting desks are equipped with fifteen electrical sockets, with each electrical socket connected to a Zigbee-based smart power plug to monitor the power consumption of any connected plug load.

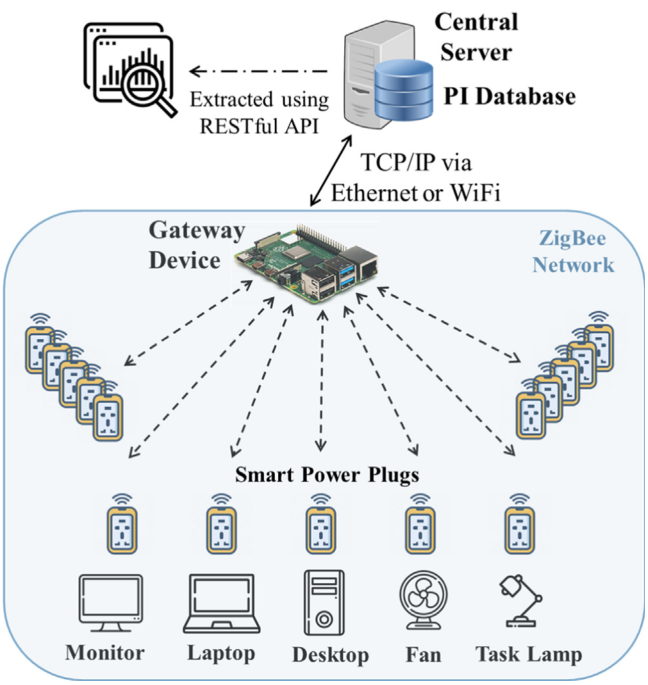

Fig. 2. System architecture of the Zigbee network connected to the PI database.

effort was conducted in the study area between February 2020 and March 2020. In total, 99 smart power plugs were deployed to collect the power consumption data of 9 desktops, 31 laptops, 35 monitors, 13 fans, and 11 task lamps at a sampling frequency of 1/60 Hz (equivalent to 1 sample every minute). The number of devices considered in this study is representative of a medium-sized office space where the data collection exercise is conducted. Based on the number of devices collected for each plug load type, our dataset was found to be comparable and in some instances (e.g., laptops and monitors) more comprehensive when compared to other datasets used in past studies [15,18], albeit considering only five different plug load types. Furthermore, to ensure that our dataset is scalable to other study areas, a particular emphasis is placed on ensuring that the dataset comprises of devices from different manufacturers and models to capture a wide range of power signatures for the same plug load type.

A close inspection of the dataset shows that the power consumption patterns of different plug loads tend to fluctuate throughout the day, even during business hours, where occupants are expected to be present in the office. Fig. 3 shows the power consumption patterns of a laptop and a task lamp during an average workday between 8 AM and 6 PM. It can be observed from Fig. 3 that each device was used intermittently throughout the day as it alternates between the active state, where the device consumes a significant amount of power, and the inactive state, where it has a power consumption close to 0 W. The usage patterns of each device could be influenced by different factors, including the occupants' presence at their desk areas, the flexibility of their work schedules, as well as thermal and lighting conditions. It can also be observed from Fig. 3 that each device was demonstrating sharp power fluctuations when it is both in use and when it is not in use, with the power fluctuations occurring in the latter case caused by standby power. These fluctuations can lead to significant energy wastage if it continues for a prolonged period. Each entry in the dataset also contains four fields, including the timestamp information, the instantaneous power value of the connected plug load recorded up to two decimal places, a unique ID indicating the smart power plug that recorded the information, and the label of the corresponding plug load type manually provided post-data collection. This dataset is made publicly available online [26] to contribute to the existing repository of plug load datasets.

3.2. Data pre-processing

Based on the power consumption data that was collected during the data collection period, a data pre-processing step was performed to prepare the data for model development. This step includes identifying the active periods of the individual plug load and performing feature extraction during the active periods.

3.2.1. Active period identification

As a plug load is used intermittently throughout the day, it alternates between the active state and the inactive state. Since the power consumption patterns are very similar when the plug loads are in the inactive state, identification of the plug load type is only performed when the plug load is in the active state for a prolonged period. This event is also defined as an active period.

By adopting a similar approach taken by [27], the active period is identified by first differentiating when the plug load is in the active state and the inactive state. This differentiation is achieved by setting a power threshold of 2.5 W as it was observed from the power consumption data collected that several plug loads occasionally consume a small amount of standby power when they are not in use (~2.3 W). Based on this power threshold, the active period is identified when the plug load remains in the active state for a prolonged period. Furthermore, as some plug loads frequently fluctuate between the active state and the inactive state while they are in use, a threshold duration of 10 minutes (equivalent to 10 samples) is set such that the plug load must be in the inactive state for this minimum duration of time

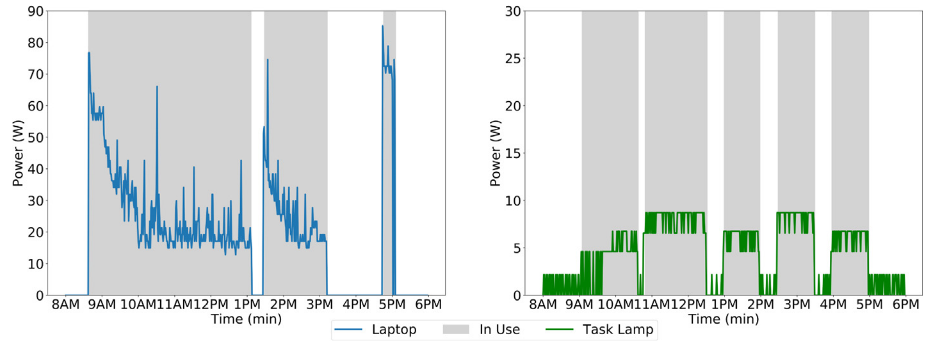

Fig. 3. Power consumption and usage patterns of a laptop (left) and a task lamp (right) during an average workday between 8 AM and 6 PM.

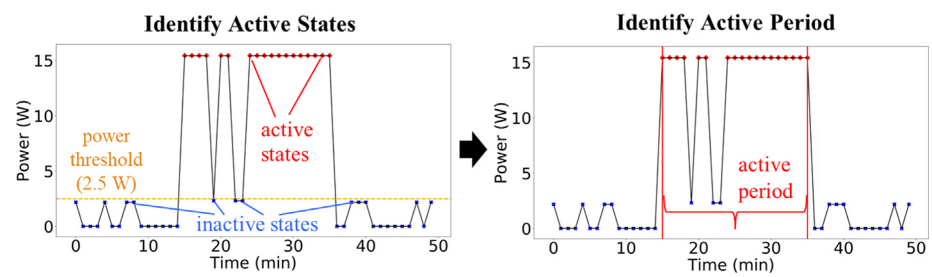

Fig. 4. Step-by-step representation of the active period identification process. The process begins with identifying the active states of the plug load by setting a power threshold value of 2.5 W before identifying the periods where the plug load remains in the active state for an extended duration of time.

consecutively before the active period is considered to have ended. The last 10 minutes when the plug load is in the inactive state is also excluded from the active period. Otherwise, they are considered as random power fluctuations within a single active period. Fig. 4 provides a step-by-step graphical representation of the process described in this subsection. While other studies have proposed the use of a changepoint detection algorithm [28] to determine the active periods, we found this heuristic to be computationally efficient and produced a similar result.

3.2.2. Feature extraction

Based on the active periods identified, feature extraction is performed on the power consumption data recorded within each active period based on a dynamic time window strategy with a minimum length of T_{min} minutes. In other words, feature extraction will be first performed at the T_{min}th minute mark based on the power consumption data that has been recorded so far, and the length of the time window used during feature extraction continues to increase at an incremental step of 1 minute as the active period continues to extend. This approach of using a dynamic time window not only allows us to utilise the maximum amount of historical data available when making each identification, but it also allows the model to make near-real-time identifications with reasonable accuracy if T_{min} is chosen to be a small value. Furthermore, if the minimum time window was set at T_{min} minutes, it should be noted that active periods with a duration shorter than T_{min} minutes will be dropped. Fig. 5 demonstrates the dynamic time window strategy adopted in this study during feature extraction.

By referencing some of the useful features proposed in past studies [21,27], we extracted the following list of input features based on the historical power consumption data recorded during time window T.

Power p_t : This feature captures the plug load's instantaneous power consumption at time t where t refers to the latest time stamp in time window T.

Maximum power p_{max}: This feature calculates the plug load's maximum power consumption during time window T.

$$p_{max} = max\{p_i\}, \quad where\ i \in T \tag{1}$$

Minimum power p_{min}: This feature calculates the plug load's minimum power consumption during time window T.

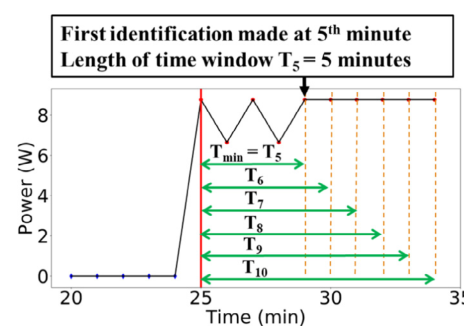

Fig. 5. Feature extraction based on the proposed dynamic time window strategy. The minimum length of the dynamic time window T_{min} is set at 5 minutes in this example, which increases at 1-minute increments.

$$p_{min} = min\{p_i\}, \quad where\ i \in T \tag{2}$$

Average power p_{ave}: This feature calculates the plug load's average power consumption during time window T.

$$p_{ave} = \frac{\sum_{i \in T} p_i}{|T|} \tag{3}$$

Power variance p_{var}: This feature calculates the variation in the plug load's power consumption during time window T.

$$p_{var} = \frac{\sum_{i \in T} (p_i - p_{ave})^2}{|T|} \tag{4}$$

Average active power $p_{aveactive}$: This feature calculates the plug load's average power consumption during time window T when the plug load is in the active state.

$$p_{aveactive} = \frac{\sum_{i \in T,\ p_i > 2.5W} p_i}{\sum_{i \in T,\ p_i > 2.5W} 1} \tag{5}$$

Average peak power $p_{avepeak}$: Given that the time window is made up of a series of disjointed subintervals, s_j, where $j \in S$, separated by power values that fall below the power threshold (2.5 W), this feature calculates the average of the peak power values, from each of these subintervals.

$$p_{avepeak} = \frac{\sum_{j \in S} max\{p_k\}, \ \ where \ \ k \in s_j}{\sum_{j \in S} 1} \tag{6}$$

Power delta d_t**:** This feature captures the absolute power difference, from the previous power value, p_{t-1}, at time t.

$$d_t = |p_t - p_{t-1}| \tag{7}$$

Duty cycle: This feature represents the fraction of time where the plug load is in the active state, given time window T. This information is useful in distinguishing between plug loads that consistently consume a stable amount of power when in use and plug loads that report fluctuating power values during operation.

Histogram features: As some plug loads may exhibit specific patterns of power consumption that cannot be easily captured using aggregated statistical metrics, we attempt to capture any remaining patterns by allocating the reported power delta values (within time window T) into eight overlapping bins of a histogram. Each bin is adjusted to have a width spanning from X to $5X$, where X represents the smallest power delta value that should be placed in that bin. Therefore, the eight bin widths that have been chosen for this study are 1–5 W, 2–10 W, 5–25 W, 8–40 W, 10–50 W, 12–60 W, 15–75 W, and 20–100 W. By calculating the corresponding bin sizes and average time interval between the energy delta values in each bin, we obtained a total of 16 input features. In the end, a combination of all of the above input features resulted in a vector containing 25 dimensions.

3.3. Model development

After the feature extraction step, the processed data is split into a training set and a test set for model training and evaluation, based on a 70/30 ratio, respectively. Due to the time-series nature of the dataset, the processed data is first grouped based on their original devices before randomly splitting each data group either into the training set or the test set. In other words, if the dataset contains a total of ten monitors, seven of the monitors will be randomly assigned to train the model, and the remaining three monitors are used to evaluate the model performance. This simple approach limits the occurrence of data leakage as the highly correlated input features from the same device will always be found in the same dataset (i.e., training or test) while maintaining a low correlation between different datasets since they are based on data obtained from different devices. After defining the training and test datasets, several classifiers were developed using a variety of supervised classification algorithms such as K-NN [29], Gradient Boosting (GB), RF, and Bagging [30] due to their reported performance in past studies.

In a classification problem, K-NN works simply by assigning a class to a data point based on the most common class found among its k nearest neighbours. Therefore, the algorithm assumes that data points from the same class tend to lie in close proximity to each other in geometric space. GB is an iterative functional gradient descent algorithm that minimises its loss function by iteratively introducing a weak learner (i.e., a decision tree) in a forward stage-wise fashion to improve the performance of the current model [31]. Due to its robust performance, it is also a popular algorithm used in many other application areas [32]. RF is an ensemble algorithm that aggregates the predictions made by a large number of relatively uncorrelated decision trees to produce a more accurate model than any individual tree. The correlation between each tree is reduced by training each classifier on different subsets of the dataset with replacement and randomising the set of features considered when splitting each node of the decision tree [33]. Finally, Bagging, or Bootstrap Aggregation, is an ensemble algorithm similar to RF that aggregates the predictions made by a large number of relatively uncorrelated decision trees to produce a more accurate model than any individual tree [34]. However, unlike RF, Bagging reduces the correlation between each tree by using different subsets of the dataset with replacement during model training.

The parameters of each classifier were optimised by performing hyperparameter tuning using a 3-fold cross-validation approach where the training set is randomly split into three equal-sized portions using the stratified (or device-wise) sampling approach to obtain the training and validation datasets to avoid data leakage. While it is more common to adopt a 5-fold or 10-fold cross-validation approach, the desktops in the training dataset might be under-represented in each fold due to its small numbers (i.e., 6), leading to models with high variance.

3.4. Model evaluation

The performance of the plug load identification model is evaluated based on four performance metrics, including overall accuracy, balanced accuracy, weighted accuracy, and majority-voted accuracy.

Overall accuracy, $accuracy_{overall}$, is a standard performance metric used by many studies to evaluate the performance of their machine learning models. The metric is calculated by counting the number of correct classifications made, $classifications_{correct}$, over the total number of classifications made, $classifications_{total}$, as shown in Eq. (8).

$$accuracy_{overall} = \frac{classifications_{correct}}{classifications_{total}} \tag{8}$$

Due to the infrequent use of plug loads such as fans and task lamps, the amount of data collected for these plug load types are comparatively lesser when compared to other commonly used plug loads such as laptops and monitors. Therefore, balanced accuracy, $accuracy_{balanced}$, was proposed as a second performance metric to evaluate the model performance by assigning equal weights to each plug load type regardless of its frequency in the test set. Balanced accuracy is calculated by first applying the overall accuracy equation (refer to Eq. (8)) to each plug load type l, where $l \in L$, and finding the average of these accuracy scores among all classes, as shown in Eq. (9).

$$accuracy_{balanced} = \frac{\sum_{l \in L} \frac{classifications_{correct,l}}{classifications_{total,l}}}{|L|} \tag{9}$$

The weighted accuracy, $accuracy_{weighted}$, was proposed as a third performance metric, which places different weights on each classification that was made during the active period a_m, where $m \in M$. More specifically, a heavier weight was placed on classifications that were made towards the end of the active period while a smaller weight is placed on classifications made at the beginning. Intuitively, since a dynamic time window strategy was used during feature extraction, the input features obtained towards the end of the active period will be based on more historical data and is hence more likely to capture the characteristics of the plug load, leading to more accurate identifications. Exponential smoothing was applied to the probability vector, $P_m(t)$ containing the probability of each plug load type at time t, with $\alpha = 0.1$, before summing all of the probability vectors within the active period a_m to obtain the final weighted probability vector, P_m. The final classification was made for the entire active period by identifying the plug load type with the highest probability in P_m. Since a single class is assigned to each active period, the final accuracy value is calculated by counting the number of correctly labelled active periods $classifications_{correct,m}$, over the total number of active periods, $|M|$, as shown in Eq. (11).

$$P_m = \alpha\left[P_m(t) + (1-\alpha)P_m(t-1) + \ldots + (1-\alpha)^{t-1}P_m(1)\right] + (1-\alpha)^t P_m(0) \tag{10}$$

$$accuracy_{weighted} = \frac{classifications_{correct,m}}{|M|} \tag{11}$$

Finally, the majority-voted accuracy, $accuracy_{majority-voted}$, was proposed as a fourth performance metric, which aggregates the classifications that were made during the active period and selects the most frequently chosen plug load type as the final class. This metric differs from weighted accuracy as it places equal weight on each classification that was made regardless of the amount of historical data that was used to make the classification. Since a single class is assigned to each active period, the final accuracy value is calculated by counting the number of correctly labelled active periods over the total number of active periods, as shown in Eq. (13).

$$classification_m = mode\{classification_{t\in a_m}\} \tag{12}$$

$$accuracy_{majority-voted} = \frac{classifications_{correct,m}}{|M|} \tag{13}$$

4. Experimental results

In this section, we provided a comprehensive evaluation of the proposed plug load identification approach through three separate experiments, including (1) identifying the best performing models for online and offline settings; (2) comparing between dynamic and fixed time window strategies; and (3) evaluating model performance under different sampling frequencies. In each of these experiments, multiple models were developed under the same conditions (i.e., sampling frequency, time window strategy, input features) while varying only one condition depending on the experiment conducted.

4.1. Experiment 1: Identifying the best models for online and offline settings

In this experiment, different machine learning models were developed using the proposed plug load identification approach by varying the minimum length of the dynamic time window and plotting their corresponding model performance in Fig. 6. In other words, each data point in Fig. 6 represents the model performance of a unique model with a minimum dynamic time window of T_{min} minutes. K-NN, RF, GB, and Bagging algorithms were used to develop different models to enable a comparison between these classification algorithms and identify the best performing model for an online and offline setting.

In the case where the model is deployed in an online setting, the instantaneous power consumption of any connected plug load is used to perform real-time identification of its plug load type. Since the data follows a time-series, each identification can only be made based on the historical data that has been collected up to this point. Furthermore, the length of the model's time window relates to the amount of historical data that will be used to calculate the input features for plug load identification. Since the model only performs identification during an active period and uses a dynamic time window strategy, the minimum length of the dynamic time window corresponds to the response time of the model when making its first identification during the start of each active period. In other words, if the model is defined to have a minimum dynamic time window of 10 minutes, the model is only able to make its first identification at the 10-minute mark from the start of the active period. Based on this definition, an ideal online model should (1) identify the plug load type accurately and (2) within the shortest time possible during the start of an active period. It can be observed from Fig. 6 that there is a gradual rise in the model performance for all four performance metrics as the length of the minimum dynamic time window increases. Given that both the model performance and minimum length of the dynamic time window possess a positive correlation, there is a need to strike the right balance between both factors when identifying the best online model. This balance is achieved by calculating the accuracy delta values for each minimum dynamic time window (refer to Fig. 7) and identifying the critical point where any further increase in window length only results in a minimal gain in model performance. It can be observed from Fig. 7 that K-NN reaches its critical point at around the 3-minute mark, RF and Bagging both reach their respective critical points at around the 5-minute mark, and GB reaches its critical point beyond the 20-minute mark, thus eliminating it from consideration due to its long time window. By referring back to Fig. 6, K-NN is also eliminated from consideration since it consistently underperforms in all four performance metrics when compared to RF and Bagging by reporting 0.760, 0.720, 0.749, and 0.773 for overall accuracy, balanced accuracy, weighted accuracy, and majority-voted accuracy, respectively. Finally, the Bagging algorithm with a minimum dynamic time window of 5 minutes is determined to be the best performing model for an online setting by outperforming the RF algorithm with the same time window, reporting 0.935, 0.890, 0.807, and 0.785 for overall accuracy, balanced accuracy, weighted accuracy, and majority-voted accuracy, respectively.

On the other hand, in the case where the model is deployed in an offline setting, all of the power consumption information recorded during the study period is assumed to be accessible as historical data and can be used for feature extraction. Since the output of the model is not time-critical, the minimum length of the dynamic time window is not crucial, unlike the case of a real-time system. Therefore, the best performing model for an offline setting is identified solely based on the four performance metrics. Based on the results reflected in Fig. 6, it can be observed that the Bagging algorithm with a minimum dynamic time window of 50 minutes was able to outperform all classification

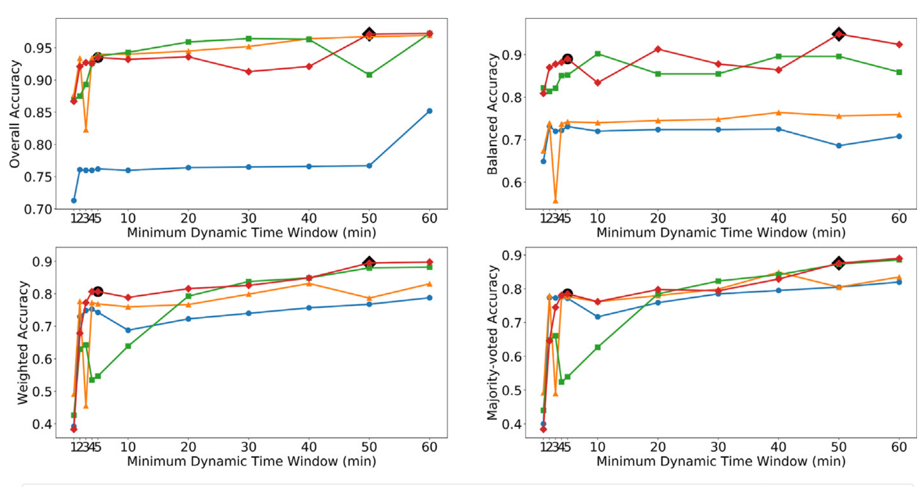

Fig. 6. Model performance based on different minimum dynamic time windows T_{min}. In this experiment, different models are developed under the same conditions while only varying the minimum length of the dynamic time window T_{min}.

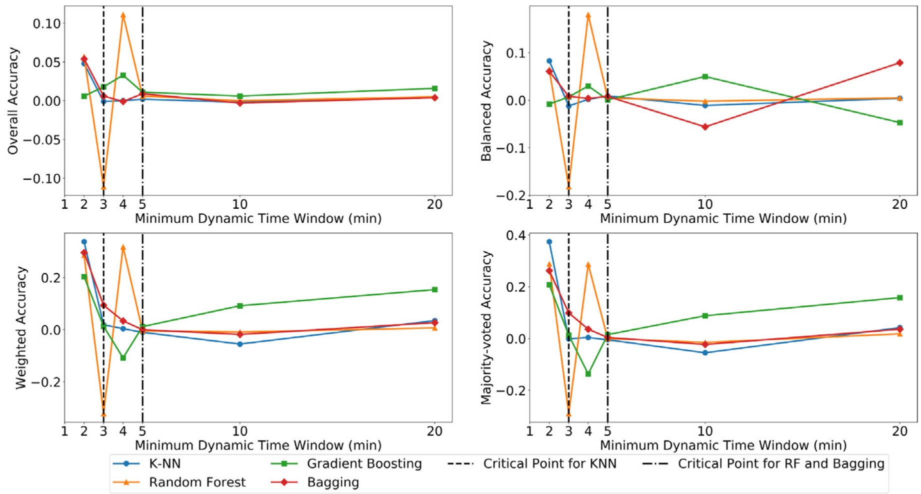

Fig. 7. Accuracy delta values based on different minimum dynamic time windows T_{min}. In this experiment, different models are developed under the same conditions while only varying the minimum length of the dynamic time window T_{min}.

algorithms under other time windows by reporting 0.971, 0.948, 0.895 and 0.876 for overall accuracy, balanced accuracy, weighted accuracy, and majority-voted accuracy, respectively.

Since the plug load identification model is initially trained on a group of devices and subsequently tested on a completely different set of devices that it has not encountered before, the excellent test results (i.e., 93% for online model and 97% for offline model) indicate that the proposed approach is highly scalable and can be generalised to other study areas containing a different and larger set of devices.

4.2. Experiment 2: Comparing between dynamic and fixed time window strategies

In this second experiment, we are interested in performing a comparison between the proposed dynamic time window strategy against the fixed time window strategy commonly used in other studies. By developing different models using the fixed time window strategy and varying the length of the time window, their corresponding model performance is plotted in Fig. 8, together with the best performing online model identified in the previous experiment (Bagging algorithm with a minimum dynamic time window of 5 minutes).

It can be observed from Fig. 8 that the model using a dynamic time window strategy significantly outperforms any models using a fixed time window strategy for all four performance metrics when the time window is similarly set at 5 minutes. While the performance of the models using a fixed time window strategy gradually rises when the length of the time window increases, they were only able to match the performance of the best performing online model when the time window is set at 20 minutes, making it four times slower when making its first identification at the start of an active period. Therefore, based on these results, the effectiveness of the fixed time window strategy is limited as it requires a choice between choosing a model with excellent classification performance but has a long response time or a model that is capable of making real-time identifications at significantly reduced accuracy. Both of these cases are not ideal in an online setting that requires the model to exhibit robust performance while having a short time window.

4.3. Experiment 3: Evaluating model performances under different sampling rates

In our final experiment, we are interested in evaluating the robustness of the proposed plug load identification approach under other sampling frequencies that are lower than 1/60 Hz (equivalent to 1 sample every minute) and observe how the model's performance is affected. Therefore, different models are developed based on the proposed plug load identification approach while using the same power consumption dataset, which has been down sampled to lower sampling frequencies. Their model performances are evaluated and compared in Fig. 9.

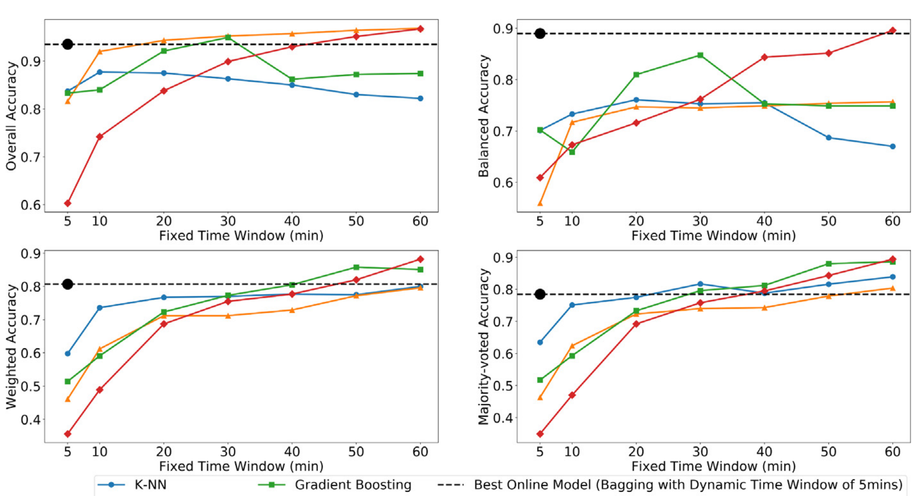

Fig. 8. Model comparison between dynamic time window strategy and fixed time window strategy. In this experiment, different models are developed under the same conditions while using a fixed time window strategy with varying time windows. The results are compared with the best performing online model identified in Experiment 1, which uses a dynamic time window strategy of 5 minutes.

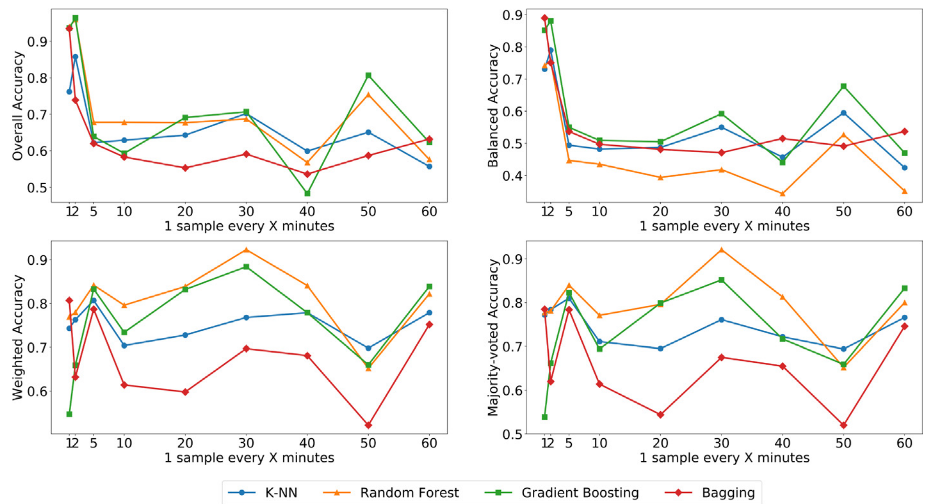

Fig. 9. Model performance under different sampling frequencies. In this experiment, different models are developed under the same conditions while only varying the sampling frequency of the input power consumption data.

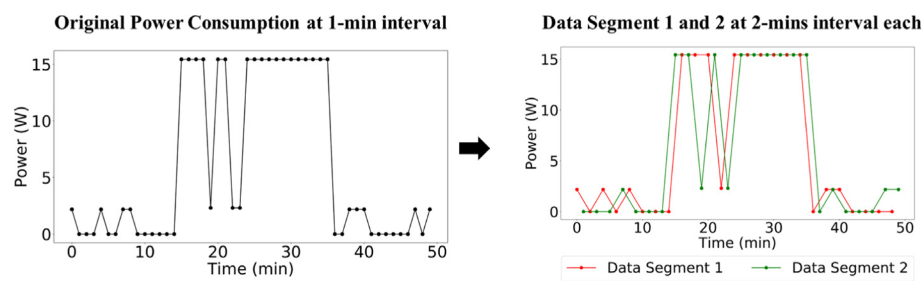

Fig. 10. Step-by-step representation of the down sampling process to obtain a final sampling frequency of 1/120 Hz (equivalent to 1 sample every 2 minutes).

The down sampling process is conducted by sampling the dataset at regular intervals based on a lower sampling frequency starting from the beginning until the end of the dataset to form a data segment. This down sampling process is repeated multiple times, starting once again from the beginning of the dataset for the remaining data points to result in multiple data segments when all data points have been sampled. For instance, if we are interested in obtaining a dataset with a sampling frequency of 1/120 Hz (equivalent to 1 sample every 2 minutes), we start by sampling the original dataset for every other data point to form the first data segment. Following this, the remaining data points are sampled to form the second and last data segment, resulting in a total of two data segments obtained. In the case where the sampling frequency is chosen to be 1/300 Hz (equivalent to 1 sample every 5 minutes), we will obtain five data segments where the time interval between each data point is 5 minutes apart. This down sampling process is adopted to ensure that all data points are used for model development and is best represented in Fig. 10, where the sampling frequency of interest is 1/120 Hz.

It can be observed from Fig. 9 that the overall accuracy and balanced accuracy of all models generally decreases as the sampling frequency continues to drop. This result is expected as the low sampling frequency has an effect of masking or smoothening the transient power fluctuations of the plug load during the active period, thereby making plug loads that exhibit such power fluctuations less differentiable from other plug loads that exhibit more stable consumption patterns. Moreover, as the sampling frequency decreases, the length of each active period also decreases, thus resulting in fewer data available for model training. This effect has a more significant impact on plug loads, such as task lamps and fans, that are normally used for shorter durations of time. On the other hand, the weighted accuracy and majority-voted accuracy for most models showed some signs of improvement despite the decrease in sampling frequencies from 1/60 Hz to 1/1800 Hz (equivalent to 1 sample every 30 minutes). However, this improvement in model performance did not continue when sampling frequencies fall below 1/1800 Hz. Therefore, based on these results, we can conclude that a real-time plug load identification model might not be feasible when sampling frequencies fall below 1/120 Hz due to the drastic drop in overall accuracy and balanced accuracy. However, in the case of an offline system, the slight improvement in the models' weighted accuracy and majority-voted accuracy show some potential that an offline system can still be implemented with reasonable accuracy under extremely low sampling frequencies.

Apart from the evaluation results obtained through the above experiments, it should be noted that a truly fair comparison between our proposed approach and past approaches is challenging to achieve due to the different conditions adopted during each study. For instance, some studies included current and voltage information as additional input features during plug load identification, or adopted a higher sampling frequency during data collection (e.g., 1/2 Hz), or used an extended time window during feature extraction (e.g., 1 hour). All of these conditions provide additional information to the model resulting in better model performance but also require additional effort to obtain, which is difficult to quantify. Despite these favourable conditions, our proposed approach was still able to report a comparable model performance of up to 93% while using only the power consumption information, a much lower sampling frequency of 1/60 Hz, and a short time window of 5 minutes to perform near-real-time plug load identification.

5. Practical applications

Given the promising performance of the proposed plug load identification approach, this section provides a detailed description of how the approach can be integrated into existing load monitoring and control systems to improve their performance in reducing plug load energy consumption.

5.1. Automatic plug load identification in energy dashboards

One of the most direct application areas for the plug load identification approach can be found in energy dashboards, which have become increasingly common in residential buildings and smart office spaces. Energy dashboards are designed to provide real-time feedback on the occupants' energy usage using load monitoring approaches such as NILM and ILM, as described in Section 1. This feature is particularly useful in the context of office spaces as occupants typically do not know how much energy they consumed due to their plug load usage. Furthermore, since they do not pay for their energy consumption, there is no incentive for them to adopt environment-friendly habits leading to energy wastage. Therefore, through the use of energy dashboards, different information can be presented to the occupants to inform them of their energy consumption, display educational tips on reducing energy wastage, and even compare their consumption patterns with their peers to encourage more sustainable behaviours. This form of intervention is commonly known as eco-feedback. Based on a study conducted by Yun et al. [35], it was found that the use of an energy dashboard combined with eco-feedback can result in energy savings of up to 9% in an office space.

Despite the potential benefits of utilising energy dashboards to inform the occupants on their energy consumption levels, the highest resolution of information that could be provided is often limited at the desk level or at the electrical socket level, when ILM approaches are adopted. Therefore, the use of the plug load identification approach to automatically identify the plug load type based on its power consumption patterns can further increase the level of data resolution and translate that information into other informative visualisations for the occupants. For instance, a detailed breakdown of the energy contributions of each plug load type can facilitate the design of conservation initiatives that are specifically targeted towards reducing energy wastage from high-energy loads. Some of these initiatives include equipment replacement programs for older and less efficient plug loads, adjusting power settings to reduce energy use during non-working hours, and reminding occupants to turn off their plug loads before leaving the office. Management support in these initiatives is also crucial in ensuring the long-term adoption of these sustainable practices [36]. Furthermore, an automatic plug load identification feature can replace the current method of getting the occupants to provide the same information into the load monitoring system, thereby reducing the chances of mislabels and human effort during a large-scale implementation.

5.2. Automated plug load scheduling and control system

Apart from monitoring the occupants' energy usage and visualising the information via energy dashboards, recent advancements in ILM provides occupants with the flexibility to operate their plug loads remotely using smart power plugs. Different control strategies can also be adopted when automating the controls of these plug loads, which include schedule-based or timer-based controls, occupancy-based controls, and system-based controls [37]. Timer-based controls are used to switch off plug loads based on a predefined schedule, such as during non-office hours when occupants are expected to be away from the office [38]. However, this type of control system is not suitable when occupants exhibit more flexible occupancy schedules that can cause significant disruptions when plug loads are prematurely switched off when the occupant is still present in the office. Occupancy-based controls work by switching off plug loads when the occupant is detected to be away from a predefined space, such as his desk area, for a prolonged period. The advantage of such a control system over time-based controls is that it reacts well to the occupant's stochastic occupancy patterns. Finally, system-based controls are part of a building-wide management system that monitors and controls the energy usage of the entire building, using a web-based software with user-friendly interfaces to provide building managers with a detailed analysis of the building's energy consumption. Based on the same study conducted by Yun et al. [35], it was reported that a combination of automated control strategies and eco-feedback could result in energy savings of up to 35.4%.

Other than the various control strategies highlighted above, our future work on automated plug load control systems will involve the introduction of a new category of control strategies. This novel idea involves a combination of different technologies to create a personalised control system that adapt to the occupant's behaviour over time. Some features of such a control system will include the use of high-resolution occupancy detection approaches [39,40] to capture the occupant's stochastic movement schedules as well as an energy dashboard for energy monitoring. Occupants can also provide their preferred control settings for their plug loads through the same interface. By integrating the plug load identification model proposed in this study into the control system, different control rules can also be automatically assigned to each plug load type to minimise energy wastage and adapt to the occupant's specific usage habits. A prototype of the personalised plug load control system will be developed as part of future work to evaluate the energy-saving potentials of such a system.

6. Conclusion

In this study, we developed a near-real-time plug load identification approach under extremely low sampling frequencies for several plug loads that are typically found in an office space. The resulting model is developed by first conducting a three-week data collection exercise in a study area to obtain the power consumption patterns of different plug loads operating in a real-world office environment. The data is captured through a ZigBee network, which consists of several smart power plugs that record the instantaneous power consumption of the connected plug loads at a sampling frequency of 1/60 Hz. Once the raw power consumption data is obtained, the next step involves identifying the active periods of the plug load and performing feature extraction within each active period using a dynamic time window strategy to capture the unique characteristics of each plug load. Following this, the processed data is randomly split into a training set and test set based on their corresponding devices before training it on several classification algorithms during model development. The performance of each resulting model is evaluated based on four performance metrics, including overall accuracy, balanced accuracy, weighted accuracy, and majority-voted accuracy. A comprehensive evaluation of the proposed plug load identification approach is performed by developing multiple models under different parameter settings and comparing their model performance through three separate experiments. The first experiment attempted to identify the best performing model for online and offline settings by observing the change in model performance when the minimum length of the dynamic time window increases. In the end, the Bagging algorithm with a minimum dynamic time window of 5 minutes was chosen to be the best online model, while the same algorithm with a minimum dynamic time window of 50 minutes was chosen to be the best offline model due to its excellent performance in all four performance metrics. The second experiment was conducted to perform a comparison between the proposed dynamic time window strategy against the fixed time window strategy commonly used in other studies. It was concluded in the experiment that the proposed approach was able to produce models that significantly outperform the other models developed using the fixed time window strategy. Finally, the last experiment was conducted to evaluate the robustness of the proposed approach under other sampling frequencies lower than 1/60 Hz by down sampling the current dataset and developing different models based on those sampling frequencies. It was observed that the model's overall accuracy and balanced accuracy generally decreases as the sampling frequency decreases, while the weighted accuracy and majority-voted accuracy demonstrated some initial signs of improvement when the sampling frequency is slightly decreased. Finally, the study was concluded by highlighting two application areas that would benefit from automatic plug load identification. The first application area is related to energy dashboards where the automatic identification of different plug load types can increase the level of data resolution and result in the computation of other useful visualisations for eco-feedback. The second application area of the plug load identification approach is to support the development of a personalised plug load control system. The novel system uses a combination of different technologies, including (1) high-resolution occupancy detection approaches to capture the occupants' stochastic movement schedules, (2) an user-friendly interface for occupants to provide their preferred control settings, and (3) an automated plug load identification model to assign different control rules to each plug load type. A prototype will also be developed in future works to evaluate the energy-saving potentials of such a system.

CRediT authorship contribution statement

Zeynep Duygu Tekler: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Raymond Low:** Conceptualization, Investigation, Writing - review & editing. **Yuren Zhou:** Conceptualization, Writing - review & editing. **Chau Yuen:** Supervision, Writing - review & editing. **Lucienne Blessing:** Supervision, Writing - review & editing. **Costas Spanos:** Project administration, Resources, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is supported in part by the Building Efficiency and Sustainability in the Tropics (SinBerBEST) inter-disciplinary research group under the Berkeley Education Alliance for Research in Singapore (BEARS) centre. The authors thank Christopher Soyza, Zuraimi Sultan, and Sneha Ranjit for supporting and facilitating the data collection efforts, and Kenny Choo for his valuable feedback on the manuscript. The opinions, findings, and conclusions expressed in this material are those of the authors only.

References

[1] Gandhi P, Brager GS. Commercial office plug load energy consumption trends and the role of occupant behavior. Energy Build 2016;125:1–8.
[2] U.E.I. Administration, Annual energy outlook 2020: Commercial sector key indicators and consumption; 2020.
[3] Abubakar I, Khalid S, Mustafa M, Shareef H, Mustapha M. Application of load monitoring in appliances' energy management–a review. Renew Sustain Energy Rev 2017;67:235–45.
[4] Welikala S, Thelasingha N, Akram M, Ekanayake PB, Godaliyadda RI, Ekanayake JB. Implementation of a robust real-time non-intrusive load monitoring solution. Appl Energy 2019;238:1519–29.
[5] Aladesanmi E, Folly K. Overview of non-intrusive load monitoring and identification techniques. IFAC-PapersOnLine 2015;48(30):415–20.
[6] Cominola A, Giuliani M, Piga D, Castelletti A, Rizzoli AE. A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring. Appl Energy 2017;185:331–44.
[7] Ridi A, Gisler C, Hennebert J. A survey on intrusive load monitoring for appliance recognition. In: 2014 22nd international conference on pattern recognition. IEEE; 2014. p. 3702–7.
[8] Kleiminger W, Beckel C, Staake T, Santini S. Occupancy detection from electricity consumption data. Proceedings of the 5th ACM workshop on embedded systems for energy-efficient buildings. 2013. p. 1–8.
[9] Tekler Z, Low R, Blessing L. Using smart technologies to identify occupancy and plug-in appliance interaction patterns in an office environment. IOP Conf Ser Mater Sci Eng 2019;609(6):062010.
[10] Wong YF, Şekercioğlu YA, Drummond T, Wong VS. Recent approaches to non-intrusive load monitoring techniques in residential settings. In: 2013 IEEE computational intelligence applications in smart grid (CIASG). IEEE; 2013. p. 73–9.
[11] Tushar W, Yuen C, Chai B, Huang S, Wood KL, Kerk SG, et al. Smart grid testbed for demand focused energy management in end user environments. IEEE Wirel Commun 2016;23(6):70–80.
[12] Zoha A, Gluhak A, Imran MA, Rajasegarar S. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. Sensors 2012;12(12):16838–66.
[13] Park JY, Yang X, Miller C, Arjunan P, Nagy Z. Apples or oranges? identification of fundamental load shape profiles for benchmarking buildings using a large and diverse dataset. Appl Energy 2019;236:1280–95.
[14] Zhou Y, Lork C, Li W-T, Yuen C, Keow YM. Benchmarking air-conditioning energy performance of residential rooms based on regression and clustering techniques. Appl Energy 2019;253:113548.
[15] Reddy RS, Keesara N, Pudi V, Garg V. Plug load identification in educational buildings using machine learning algorithms. In: Proceedings of BS2015: 14th conference of international building performance simulation association, Hyderabad, India; 2015. p. 1940–6.
[16] Reddy R, Keesara N, Garg V, Pudi V. Plug load identification using regression based nearest neighbor classifier. Proceedings of the eighth international conference on future energy systems. 2017. p. 101–10.
[17] Abeykoon V, Kankanamdurage N, Senevirathna A, Ranaweera P, Udawalpola R. Real time identification of electrical devices through power consumption pattern detection. Pervasive Comput 2016;10(1):40–8.
[18] Ridi A, Gisler C, Hennebert J. Acs-f2—a new database of appliance consumption signatures. In: 2014 6th international conference of soft computing and pattern recognition (SoCPaR). IEEE; 2014. p. 145–50.
[19] Gisler C, Ridi A, Zufferey D, Khaled OA, Hennebert J. Appliance consumption signature database and recognition test protocols. In: 2013 8th international workshop on systems, signal processing and their applications (WoSSPA). IEEE; 2013. p. 336–41.
[20] Ridi A, Gisler C, Hennebert J. Appliance and state recognition using hidden markov models. In: 2014 International conference on data science and advanced analytics (DSAA). IEEE; 2014. p. 270–6.
[21] Tundis A, Faizan A, Mühlhäuser M. A feature-based model for the identification of electrical devices in smart environments. Sensors 2019;19(11):2611.
[22] Plugwise smart power plugs. https://www.plugwise.com/en_US/products/plug.
[23] Shetty SS, Chinh HD, Gupta M, Panda SK. User presence estimation in multi-occupancy rooms using plug-load meters and pir sensors. GLOBECOM 2017–2017 IEEE global communications conference. IEEE; 2017. p. 1–6.
[24] Osisoft pi database. https://www.osisoft.com/PI-System/.
[25] Das HP, Konstantakopoulos IC, Manasawala AB, Veeravalli T, Liu H, Spanos CJ. Segmentation analysis in human centric cyber-physical systems using graphical lasso. arXiv preprint arXiv:1810.10533.
[26] Tekler ZD, Low R. Plug load dataset for office spaces. Mendeley https://doi.org/10.17632/dnx6bc59rj.2. https://github.com/zeynepduygutekler/plug-load-dataset.
[27] Ambati LPR, Irwin D. Autoplug: an automated metadata service for smart outlets. In: 2016 seventh international green and sustainable computing conference (IGSC). IEEE; 2016. p. 1–8.
[28] Truong C, Oudre L, Vayatis N. Selective review of offline change point detection methods. Signal Process 2020;167:107299.
[29] Tsai M-S, Lin Y-H. Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation. Appl Energy 2012;96:55–73.
[30] Himeur Y, Alsalemi A, Bensaali F, Amira A. Robust event-based non-intrusive appliance recognition using multi-scale wavelet packet tree and ensemble bagging tree. Appl Energy 2020;267:114877.
[31] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001:1189–232.
[32] Low R, Cheah L, You L. Commercial vehicle activity prediction with imbalanced class distribution using a hybrid sampling and gradient boosting approach. IEEE Trans Intell Transp Syst. 2020. https://doi.org/10.1109/TITS.2020.2970229.
[33] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[34] Breiman L. Bagging predictors. Mach Learn 1996;24(2):123–40.
[35] Yun R, Aziz A, Lasternas B, Zhang C, Loftness V, Scupelli P, et al. The design and evaluation of intelligent energy dashboard for sustainability in the workplace. International conference of design, user experience, and usability. Springer; 2014. p. 605–15.
[36] Tekler ZD, Low R, Chung SY, Low JSC, Blessing L. A waste management behavioural framework of singapore's food manufacturing industry using factor analysis. Procedia CIRP 2019;80:578–83.
[37] A.S. 90.1-2019. Energy standard for buildings except low-rise residential buildings; 2019.
[38] Yanda Zhang NA, Bonneville C, Higa R. Integrated lighting and plug load controls aceee summer study on energy efficiency in buildings. In: ACEE, Summer Study on Energy Efficiency in Buildings.
[39] Tekler ZD, Low R, Gunay B, Andersen RK, Blessing L. A scalable bluetooth low energy approach to identify occupancy patterns and profiles in office spaces. Build Environ 2020;171(1):106681.
[40] Tekler ZD, Low R, Blessing L. An alternative approach to monitor occupancy using bluetooth low energy technology in an office environment. J Phys: Conf Ser 2019;1343(1):012116.