Title

Building plug load mode detection, forecasting and scheduling

Authors

Botman, Lola

Lago, Jesus

Fu, Xiaohan

et al.

Publication Date

2024-06-01

DOI

10.1016/j.apenergy.2024.123098

Building plug load mode detection, forecasting and scheduling

Lola Botman[a,*], Jesus Lago[b,1], Xiaohan Fu[c], Keaton Chia[c], Jesse Wolf[c], Jan Kleissl[c], Bart De Moor[a,2]

[a] *KU Leuven, ESAT, Stadius, Belgium*
[b] *Amazon, Netherlands*
[c] *University of California, San Diego, Mechanical and Aerospace Engineering Department, United States*

Abstract

In an era of increasing energy demands and environmental concerns, optimizing energy consumption within buildings is crucial. Despite the vast improvements in HVAC and lighting systems, plug loads remain an under-studied area for enhancing building energy efficiency. This paper studies smart plug active operating mode detection, plug-level load forecasting, and plug scheduling methodologies. This research leverages a unique dataset from the University of California, San Diego, consisting of readings from over 150 smart plugs in several office buildings for more than a year, notably during the post-Covid era. This dataset is made publicly available. A comprehensive literature review on plug, i.e., appliances, operating mode detection is presented. Novel unsupervised learning approaches are applied to identify plug operating modes. A pipeline integrating the detected modes with forecasting and scheduling is developed, aiming at building energy consumption reduction. Our findings offer valuable insights and promising results into smart plug management for energy-efficient buildings.

Keywords: Smart Plug, Operating Mode, Plug Scheduling, Building Consumption

1. Introduction

1.1. Motivation

Non-residential buildings account for 18% of the total electricity consumption of the United States [1, 2]. This combined with increasing local solar power generation, has led recent research to focus on building energy efficiency and demand flexibility. Plug loads are defined as loads not associated with lighting; heating, ventilation and air conditioning (HVAC) and water heating. Note that in the literature, plug loads are sometimes called plug and process loads (PPLs) [2], Miscellaneous Electric Loads (MELs) [3, 4, 5], small power equipment [6], or plug-in equipment loads [7]. Plug loads in commercial buildings accounted for 47% of the total commercial buildings energy consumption in the United States in 2020 [8, 1]. Despite advancements in the efficiency of lighting and HVAC systems, consumption from unregulated plug loads has increased [9, 5]. Plug loads must be targeted as well to further reduce building energy use.

The smart plug research field mostly started by focusing on home energy management systems [10, 11, 12, 13, 14, 15]. The installation of smart plugs in homes is easier, requires fewer devices and thus less investment, and is easier to maintain. The integration of smart plugs into commercial buildings however, is more challenging. Chia et al. [8] discuss these challenges comprehensively, shedding light on data access, control, and integration within commercial buildings. The availability of smart plug data remains limited, and the infrastructure for seamless integration is still evolving.

The utilization of smart plugs to completely dis-

[*]Corresponding author

Email address: lola.botman@esat.kuleuven.be (Lola Botman)

[1]Jesus Lago contributed to this work as an outside activity and not as part of his role at Amazon.

[2]Fellow IEEE, IFAC, SIAM

connect appliances, thus reducing standby and/or idle energy consumption, presents a promising avenue for reducing energy use [16]. This underscores the importance of comprehensive study around plug loads as significant contributors to energy consumption. Yet, plug loads are only minimally explored for potential energy-saving interventions, primarily due to data scarcity and the absence of economical plug load controllers [17]. Existing studies emphasize the potential benefits of an open-source dataset comprising plug load measurements [5].

This paper addresses plug load operating mode detection, i.e., active vs. unused states (i.e. idle, standby, off), plug load forecasting, and smart plug scheduling. Plug load operating mode detection is critical to estimate potential energy savings and to design plug load control schemes. Plug load forecasting provides the predictions for plug load usage patterns to inform plug load scheduling. Plug load scheduling leverages plug load forecast to implement decision on when to turn off plug loads without disrupting plug load users while maximizing energy savings. The dataset originates from the University of California, San Diego and includes over 150 smart plugs distributed across several office buildings over more than a year [8]. The objective of this research is to contribute to the understanding of energy-efficient building management through smart plug time series analysis and to explore the potential for energy savings.

1.2. Contributions

This paper presents several advancements in the study of smart plugs. The key contributions are as follows:

1. A novel open dataset is introduced, encompassing readings from over 150 smart plugs situated in various office buildings, spanning a duration exceeding a year, with specific emphasis on the recent post-Covid era.
2. The first literature review on methodologies for plug load operating mode detection is presented coupled with a compilation of terminologies that have been employed in prior research.
3. The paper proposes an unsupervised learning approach aimed at discerning the operating modes of a plug load, addressing the challenges documented in the existing literature.

Finally, a methodological pipeline is outlined, combining the detected modes with individual plug load forecasting techniques, designed to schedule smart plugs (on/off) a day in advance, with the overarching aim of decreasing building energy consumption.

2. Literature review

2.1. Smart plug hardware

The advent of smart plugs has led to significant advancements in both hardware and software research related to energy consumption and management. Smart plugs enable fine-grained monitoring and control of plug loads and can be configured through Wi-Fi or cellular networks [16]. Plug-level data is becoming more accessible, mainly thanks to these advances in hardware. Many commercially available plug load controllers exist that include plug and telecommunication solutions originally proposed by researchers. For instance, a hardware plug that turns off if the consumption is below a user-defined threshold was showcased in [18]. Another example is a smart plug equipped with a wireless Zigbee sensor, primarily designed to measure power consumption of electrical appliances in home energy management systems [19]. A prototype for home units that can recognize device types based on statistics and labeled data, was described in [20].

2.2. Non-intrusive and intrusive load monitoring

There are two main different approaches[3] to study building energy efficiency. **Non-Intrusive Load Monitoring** (NILM) consists of disaggregation strategies, seeking to break down the total building energy consumption into individual device consumption without the need for extensive hardware installations [21], requiring only one smart meter at the building level. A notable disaggregated model of plug loads in an office building was introduced by Doherty et al. [2]. This method uses power data from a few monitored devices combined with a detailed

[3]Note that in some papers, the term "invasive" is used instead of "intrusive" [21, 22]

device inventory. The derived model provides non-measured loads based on logical assumptions and technical specifications. Crucially, these insights, when coupled with the disaggregated model, help identify devices that can be optimized for energy savings. Though NILM offers the advantage of reduced hardware requirements, making it more cost-effective and easier to maintain, it does have its challenges. Primarily, while NILM has been popular in the residential sector since the 1990s, there are concerns about its accuracy, especially in the commercial sector. Larger spaces tend to have a diverse range of devices, often with multiple instances of the same device, making them hard to distinguish. Skepticism persists regarding NILM's accuracy in such settings [2]. However, it is worth noting that Lanzisera et al. demonstrated that by taking an inventory of just 40% of their building floor space, they achieved less than a 10% error in their projection estimates of number of devices in the entire building (compared to the true inventory), especially for computers and displays [3].

On the other hand, **Intrusive Load Monitoring** (ILM), also termed component-based load monitoring or distributed sensing, operates by monitoring and recording the electrical consumption data –or voltage and current– of each individual device or appliance separately. ILM provides accurate disaggregation, particularly in commercial settings. The capability of smart plug technology in ILM is well-acknowledged. Yet, ILM's major challenge remains its cost, given the hardware and labor demands to monitor each device in a building individually [2, 21]. This study presents an ILM dataset.

2.3. Smart plug data

Comprehensive ILM datasets are crucial for advancing research in the dynamic field of smart plug loads. Making such datasets openly accessible not only fosters progress but also facilitates benchmarking. A recent overview of public datasets for home management systems and typical household appliance recordings is provided in [21]. While some papers presented results without sharing their datasets, others commendably made them available. In 2013, Lanzisera et al. [3] offered insights from a year-long study that meticulously inventoried over 4,000 devices. 455 plug load meters monitored the plug loads in an office building. However, this extensive data is not publicly accessible. Likewise, in 2015, [23] defined the characteristics that a useful open dataset should possess. While the authors stated their intention to release the data post a comprehensive analysis, this dataset has not been made available. Similarly, Das et al. [24] stated their intention to publish their dataset in the future, but it has not been made available yet. Das et al.'s dataset consists of six different appliances recorded during six months. It includes a docking station, a laptop, two monitors, a lamp and a "miscellaneous" plug load, where the user was allowed to plug in any device. They also recorded environmental data such as the lighting, occupancy, temperature, and humidity of the room. To the best of the authors' knowledge, the first office smart plug open data set 'BLOND' was made available in 2018. The data, from 2017, spans over 213 days with voltage and current recordings from 53 distinct appliances at 50 kSps (kilo Sample per second). The dataset occupies 38.7 TB with nearly a million files [25]. In 2019, Doherty et al. published a dataset detailing 14 different plug loads. The data collection covered October to December 2017, with power consumption recordings made every 5 minutes. Among the items tracked were common office appliances like coffee makers, laptops, microwaves, and video conference cameras [2]. An open three-week-long dataset from February and March 2020, recording data every minute incorporates information from 88 smart plugs, which include a diverse range of devices: 31 laptops, 9 desktops, 35 monitors, 13 fans, and 11 task lamps [26]. An open Matlab formatted dataset was also made available in 2020 incorporating six days of readings in May 2018, every second, from 19 sensors in an office space. These sensors range from SmartSense motion sensors to temperature-humidity sensors, ultrasonic sensors, and Belkin Wemo switches [27]. The ancillary sensor data allow studies beyond just smart plug data; for example, occupancy information can be added to plug load forecasting.

2.4. Plug load operating modes detection

2.4.1. Plug load operating mode terminology

The task of discerning the operating mode of an electrical device, specifically determining if it is active, idle, in standby or off, is an essential aspect of all forecasting and scheduling applications. Surprisingly, research addressing this topic specifically remains scarce in the literature. A notable challenge faced during the literature review is the lack of standardization in terminology. This absence of a consistent terminology not only complicates the review process but also makes comparisons between different research difficult. To palliate this lack of standardization this section provides an exhaustive list of the different terminology used in 19 papers which address implicitly the problem of operating mode detection.

In the context of smart plugs and related studies, what this paper refers to as the **procedure of operating mode detection**, is referred to as threshold detection, mode detection, status monitoring, load state estimation, recognition of appliances states, or activity recognition.

Operating modes themselves, as they are named in this paper, are referred to as mode of operation, operating patterns, operational characteristics, operation mode, operating state, operational energy use, power state, power status, power management modes, power levels, appliance status, or appliance activity.

Lastly, when naming the **categories of operating modes**, there's a plethora of terms used both in academic research and industry practices. Some papers provide explicit definitions [28, 29] while most studies assume a general understanding even though this understanding often differs between them. Expanding on [29] definitions, as a generalization, and common ground between existing studies, four modes can be defined. For standardization, for each mode, we suggest one term, that is already prevalent in most literature, and recommend its continued adoption in future work:

(1) The mode off or energy zero represents a device that is completely turned off, consuming no power. We recommend to designate this mode as off.

(2) The next mode is typically defined as a state the device automatically enters after prolonged inactivity [6]. It is known by terms like low-power, passive, suspend, sleep, asleep, standby, or even the descriptive "turned off but still consuming power". This is also sometimes called the vampire current. We recommend to designate this mode as standby.

(3) The following mode is denoted as idle or ready, it signifies a device that is not in use, but turned on and ready for activation, consuming more power than the previous mode. We recommend to designate this mode as idle.

(4) The terms on, high-power, active, in use, and running are used interchangeably to denote this mode, representing the highest power consumption level of a device. We recommend to designate this mode as active.

It should again be noted that there are varied conceptualizations across studies, mostly depending on the application. Some consider only two states, off and a conglomerate of (2), (3) and (4). Others might recognize three distinct states off, a combination of (2) and (3) and on. In this paper, the end goal is to turn a smart plug off when it is not in use. We thus aim to find out when the connected appliance is used/active and when it is not used. The "unused" mode collectively refers to the modes (1),(2),(3), as opposed to mode (4), which is defined as "active".

2.4.2. Plug load operating modes detection methodologies

Approaches to discern plug load operating modes are applicable in both residential and commercial settings and cover a range of methods. Operating modes detection has never been the main research topic of a research paper; instead it is usually a byproduct of a broader study, or a step in a smart plug application such as appliance scheduling. Four approaches for plug load operating modes detection have been identified when reviewing the existing literature. Most of the studies focused on empirical techniques, such as visualizing measurements and determining a standby threshold, or choosing a threshold value based on the average load measurement. Some researchers used literature benchmarks and referred to appliances specification sheets. Device user feedback is sometimes used to qualitatively estimate which percentage of the time a device is on or off, necessitating questionnaires or interview of users. Finally, some papers relied on labeling a dataset manually and then applying supervised learning techniques. This section highlights each of these approaches.

2.4.2.1. Empirical techniques.
Leonardi et al. [11] proposed an innovative solution using machine learning to streamline metadata management for smart homes. While the study's core focus was on identifying swapped devices and newly connected devices on the network, it acknowledged the necessity of threshold identification for operating mode detection. They decided to empirically set the threshold at 15 W. Filtering out standby values improved device identification accuracy. Ambati et al. [30] proposed "AutoPlug", an automated system for tracking and identifying devices plugged into smart outlets in real-time. AutoPlug presumed a smart building environment with interconnected smart outlets transmitting power consumption data in real-time. To segregate active from unused periods, the researchers empirically established a 5 W threshold. Tekler et al. [26] observed that many plug loads occasionally utilize a slight amount of standby power, approximately 2.3 W, when not in active use. Hence, the researchers designated a power threshold of 2.5 W to accurately detect active periods. Wang et al. [31] focused on home energy management and differentiated plugs load as either continuous, e.g. a fridge, or intermittent, e.g. a toaster, power consumers. This categorization seemed to have been done manually. They then determined the active versus idle states based on the effective state ratio (ESR) that depends on the type of power consumption of the device (continuous or intermittent) and its measured mean consumption value. Chia et al. [8] adopted a visual method by examining power recordings of plug loads. They found that the majority of devices have distinct active use peaks during business hours, with standby powers ranging from 2 W to 60 W.

Siebert et al. [32] proposed both centralized and decentralized methodologies for scheduling algorithms. While the focus of the paper was not on determining standby values, the determination of such values was crucial, given the study objective to mitigate standby consumption, also termed "vampire current". In the centralized approach, a home automation controller aggregated data from smart plugs, the smart meter, and utility information, like tariff tiers. The centralized algorithm initially determined the standby energy of devices by computing the daily average consumption for each smart plug. A plug load was deemed in standby mode when its power lies between a lower and an upper threshold, defined respectively as a given percentage of the mean consumption and an upper limit. However, the case study section suggested that these parameters, were respectively set to 75% and 50 W empirically without detailed justification. The decentralized method proposed an in-built scheduling algorithm within the smart plug itself. To deduce standby consumption, this simpler algorithm evaluated if the peak demand during a given assessment period was at least double the minimum. If not, the load was considered constant, implying no standby state. However, if this criterion was not met, the algorithm delineated the range between minimum and maximum demand into ten segments. It then examined, starting from the minimum, whether the cumulative number of measurements within a segment exceeded a reference value (RV) which is a fraction of the total count. If this condition held true, the standby threshold was recognized as the value of that segment. Yet, similar to the centralized method, the case study indicated an empirical selection of parameters, with RV set ad hoc at 25%.

2.4.2.2. Benchmarks and technical specification sheets.
Kawamoto et al. [28] explored three distinct operating modes: active, low-power, and off. The power consumption data for these modes was derived from the authors' unpublished measurements (empirically) or literature benchmarks. For servers specifically, measurements reflected power demands ranging between 50 W and 270 W, with an estimated average of 75 W. Interestingly, copier and laser printer power levels varied considerably with the speed (measured in images per minute) of the device. Doherty et al. [2] developed a disaggregated model to evaluate plug loads in office buildings. The novelty was the utilization of power data from a lim-

ited number of monitored devices combined with a device inventory. The methodology to ascertain standby load was twofold: direct measurements of specific devices in standby, for instance, observing the power consumption of an elliptical trainer for 30 minutes, or extracting the required data from specification sheets. Menezes et al. [6] created two models to estimate plug load power consumption in office buildings, complemented by typical power demand profiles. The paper leaned heavily on data obtained from published benchmarks, especially for Energy Star-rated devices, for which data sheet specification can be sourced from their online database. For refurbished devices or specific appliances that were not in online databases, short-term monitoring was used to provide better input data. They presented a table on published energy use for both desktop and laptop computers in low-power and active modes. Another table captured the power levels of other installed equipment, i.e., computers, printers, copiers and monitors, based on short-term monitoring and specification sheets.

2.4.2.3. User feedback.

Ghatikar et al. [17] evaluated the Infosys Plug Load Manager (PLM), a commercial solution that includes smart plugs, sensors, a server and a PLM application, which includes a user interface. This interface allows users to view the instantaneous power and the mode (on or off) for all connected smart plugs. Note that the definition of the on/off mode is not explicitly stated. One of the features of the PLM is that it allows building managers to identify plug load patterns and evaluate wasted consumption. The wasted consumption appeared to be computed based on the energy consumption outside office hours defined per the users' manual input.

Hafer et al. [29] extracted assumptions about the duration spent in distinct operating modes from user feedback gathered during preliminary phases of the research. Hafer et al. mentioned that given the largely qualitative nature of these assumptions, there is room for improvement. Potential future studies could aim to refine operating mode estimations by collecting more rigorous data about the periods spent in each power state with an emphasis on the most used devices.

2.4.2.4. Labeling data.

Webber et al. [33] collected data on turn-off rates for an array of office equipment, including but not limited to computers, monitors, and printers. They visited office spaces and physically observed each device, and noted down its power status, essentially a form of manual labeling. For instance, a laptop's power status was determined by visual indicators, like a screen being on. Similarly, monitors usually manifested their low-power mode via a distinct light, and printers, particularly ink jets, were simply categorized as either "on" or "off." Identifying the power status for copiers proved relatively straightforward. Ruzzelli et al. [34] introduced RECognition of electrical Appliances and Profiling in real-time (RECAP) with a focus on residential systems. RECAP consisted of three main parts: directing users to catalog electrical devices and establish a unique database of their distinct features, utilizing this database to train a model to differentiate and understand appliance operating modes, and implementing appliance signatures to facilitate comparisons between appliance. The authors meticulously crafted a labeled database of these distinct appliance signatures based on six parameters: real power, power factor, peak current, RMS current, peak voltage, RMS voltage. The challenge of operating mode detection (called device activity recognition in [34]), especially with appliances possessing multiple settings, was highlighted. Lin et al. [35] identified appliance states using circuit-level energy consumption. The study leaned heavily on manually labeled data. Kalluri et al. [23] emphasized the significance of visual inspection for data labeling. A case in point was the BLUED dataset, wherein appliance activity transitions are manually labeled based on the visual assessment of energy signatures. Specifically, any power level fluctuation exceeding 30 W and persisting for a minimum of 5 seconds was documented. Masoso et al.[36] examined energy consumption during unoccupied hours in Botswana and South Africa. Surprisingly, their research revealed that a substantial portion of energy (56%) is consumed during non-working hours, exceeding the consumption during working hours (44%). These results are based on a series of detailed energy audits, with data acquired via on-site assessments and subsequent manual labeling.

2.4.2.5. Other studies.
Several other studies acknowledged the significance of identifying plug load operating modes, though their primary focus was not the detection of these modes. For instance, in the work of Ahmed et al. [37], there is mention of on/off states, but the emphasis was primarily on shifting peak consumption. Kamilaris et al. [5] underscored the importance of discerning operating modes in their conclusions, asserting that it is crucial for enhancing the analysis and understanding of total consumption. However, tangible solutions for this identification were not provided.

2.4.2.6. Summary.
Determining plug load operating modes, both in residential and office settings, has been approached through a diverse set of methods, as discussed in the preceding sections. Four predominant methodologies are identified: empirical methods, benchmarks and specification sheet information, user feedback, and supervised learning when a manually labeled dataset is available. However, each of these approaches presents its challenges. Empirical methods face challenges in their generalizability. Given the differences in power consumption levels across devices, from computers, to copiers, to coffee machines, crafting a one-size-fits-all model threshold value can be problematic. Moreover, addressing each device individually visually is labor-intensive. Relying on averages or percentiles for decisions can be an issue due to potential imbalances in device usage patterns. Turning to benchmarks and specification sheets, challenges are: (i) procuring information for a large number of devices; (ii) The information may sometimes be absent from specification sheets; (iii) Over their lifecycle, devices might deviate from their initial specifications; (iv) The swift advancements in technology necessitate constant updates to benchmarks; (v) The dynamic nature of a workplace, where devices are continuously added or replaced; (vi) Device active states may be shorter than the averaging interval such that the specification sheet value would not manifest in the data. User feedback, while valuable, is inherently qualitative. Determining plug load operating modes demands consistent updates, which is both labor-intensive and challenging to scale. The approach of manual labeling followed by supervised learning, though promising, is not without pitfalls. It demands meticulous experimentation and manual state annotations. The scalability issue resurfaces, coupled with potential inaccuracies due to human errors in labeling.

A recurring observation is that all of these methodologies for plug load operating mode detection play a secondary role in broader research projects. The significance of plug load operating mode detection is frequently understated, even though it can profoundly impact plug load control applications. An additional layer of complexity arises from the lack of standard terminology, leading to potential misinterpretations and ambiguities in research reporting. This literature review underscores the knowledge gap and challenges and accentuates the need for focused research in this domain. We believe that the most promising approach is unsupervised learning and automatically updating plug load operating modes, removing the dependence on labor-intensive manual processes. We adopt an unsupervised learning approach in this paper.

2.5. Forecasting plug load energy use

There are numerous forecasting techniques and papers about electricity consumption time series, designed to predict energy use at varying scales —from national to household levels. However, these broader approaches fall outside the scope of this paper and its literature review. We are specifically interested in forecasting at the individual plug load level.

The literature appears sparse when it comes to forecasting at the finer granularity of individual plug or appliance levels, a niche that poses unique challenges. These challenges arise from the inherent stochasticity of appliance use, largely driven by unpredictable human behavior. Additionally, the data is frequently unbalanced, characterized by prolonged periods of low consumption values (indicating idle, standby, or off modes) with occasional higher consumption when the appliance is in use. In the literature, intermittency is defined as prolonged sequences of consecutive zero values with brief periods of higher values. The domain of intermittent forecasting is predominantly explored within economic contexts, such as sales predictions for inventory systems [38, 39]. In this study, the prolonged sequences do not always represent zero power levels; rather, they reflect prolonged lower power levels when the device is idle yet powered on or in standby versus higher power levels during shorter active usage periods.

Furthermore, any model designed for such forecasting must be inherently scalable, due to the potential need to predict usage across a multitude of plug loads. The current literature landscape offers little insight into individual plug load forecasting, also underscored by Wang et al. [40]. However, some research has been identified that addresses aggregate plug load forecasting at the room, zone, or building level, and these works will be the focus of this section.

Gunay et al. [7] constructed a model that splits plug load patterns across five distinct time frames: occupancy periods, intermediate breaks, weekday evenings, weekends, and vacations. By integrating the predictions of an occupancy model, the research demonstrated accurate plug load forecasts at the room level. Wang et al. [40] focused on HVAC predictive control, underscoring the importance of plug load forecasts to estimate internal heat gains within office environments. Their research findings highlighted the significant role of occupant count in predicting building level plug loads. They developed a forecast model for an 8-hour horizon, utilizing features such as the last 24-hour consumption, the day of the week, hour of the day, and hourly occupancy data. They found that a Long Short Term Memory (LSTM) network outperforms the ARIMA model by 4%. Including occupancy data improved the accuracy of the predictions. Srinivasan et al. [41] discussed plug load forecasting, leaving certain specifics such as the forecasting horizon and the exact consumption entity (whether individual device or aggregate) undefined. Abdullah et al. [14] also discussed forecasting, however they used an artificial neural network controller for appliance management, predicting on or off states based on customer preferences and appliance priorities rather than forecasting individual plug loads based on historical data. Maltais et al. [42] compared various data-driven forecasting models to predict aggregated residential lighting and plug load electricity demands in individual dwellings. While the gradient boosted regression trees model performed best, the distinction between the best models remained nuanced and dependent upon specific residential units and time horizons. Das et al. [24] tackled individual plug load forecasting. They compared the performance of an LSTM network, Bi-LSTM network and a GRU model on a dataset composed of six different office appliances. Their analysis concluded that Bi-LSTM and GRU models outperformed LSTM.

2.6. Scheduling smart plugs

Different smart plug control strategies can be used to optimize energy consumption in commercial spaces. Tekler et al.[26] divided strategies into time-based (also called schedule-based), occupancy-based, and system-based controls. Time-based controls are designed to deactivate plug loads according to a set schedule, typically during periods like non-office hours. If occupants have variable schedules, occupancy-based controls can be more dynamic, by determining presence within a designated area and deactivating plug loads after prolonged absence. Occupancy-based controls are necessary if occupancy patterns are unpredictable. System-based controls are components of a comprehensive building management mechanism. Beyond these established control strategies, innovative hybrid automated strategies have been developed.

Teckler at al. [26] proposed high-fidelity occupancy detection methods to trace user movements, combined with a plug load identification model. Alternatively, users can define their preferred settings per plug type through an interface. Hafer et al.[29] combined energy monitoring gamification on a mobile platform with plug load control schedules, resulting in a 21% reduction in plug loads across three buildings. Choi et al.[43] introduced a location-aware energy-saving service. This service utilized smartphone Bluetooth connections to local beacons to determine the occupancy and turn devices on or off accordingly, yielding energy savings of approximately 32% for PCs and monitors, and about 15% for lights. Ghatikar et al. [17] proposed three approaches to schedule small appliances in office buildings, both drawing upon preset user inputs, using an interface. The first approach is time-based and requires that the user pre-programs each monitored plug load with his personal daily schedule. The second approach is based on a user-specified maximum energy usage (in kW h). Once this limit has been reached, the smart plug turns itself off. The researchers also suggested a third control framework based on a master/slave model. In this setup, the power states of principal devices, such as personal computers, had an influence on the power states of their logically associated devices, like speakers, monitors, and printers. These proposed plug load control system depend on persistent user engagement, along with the requisite infrastructure in terms of reliable communication technologies and hardware. [1] offered a different perspective, recommending that devices switch to standby mode instead of being completely shut off. Standby could be triggered manually or embedded in the device hardware settings. For computers, triggers could be the power button, designated keyboard buttons, or even an icon on the desktop. Multifunctional devices, like printers and scanners, should ideally shift to standby after 15 minutes of inactivity.

3. Methodology

We propose a pipeline for office plug load scheduling composed of four main steps, as illustrated in Figure 1: (i) Data preparation, (ii) operating mode detection, (iii) plug load forecasting, and (iv) plug load on/off scheduling. This pipeline assumes static loads, i.e., the plugged in appliances remain plugged in and do not change.

3.1. Data preparation

A large data preparation process was necessary to obtain a usable dataset. This is a one-time procedure, and the specific details are presented in Section 4.2. In this section, some general guidelines to apply the pipeline in Figure 1 on any plug load dataset are provided.

The dataset is expected to have the same frequency through the entire recording period, in our case it is a quarter hourly resolution. NaNs indicate missing recordings, that can occur due to system updates or failure, or employees unplugging appliances. These can either be filled using data imputation techniques, or left as such and removed during the training and testing. We opted for the latter.

Data exploration and analysis using visualization has been done iteratively, using time series plot, heatmaps, and stacked diagrams with a weekly or yearly evolution. For example, heatmaps allowed identifying wrongly assigned device types and profiles with only zero. The correct device type was reassigned in the metadata and fully zero profiles were dropped. This visual data exploration is highly recommended to obtain insights into the dataset, and identify additional pre-processing steps but is out of scope for this paper.

Another recommended pre-processing step is to select only plugs load that have small amounts of NaNs (missing values) and zero recordings, since the aim is to schedule appliances that are used frequently. Underutilized appliances could instead be turned off semi-permanently or removed. We set the threshold to at most 30% of NaNs and 30% of zeros.

3.2. Operating mode detection

This section covers the proposed plug load operating mode detection procedure. As concluded from the literature review, a recurring observation is that all the existing methodologies for plug load operating mode detection play a secondary role in broader research projects. This section addresses the knowledge gap and challenges in plug load operating mode detection.

As mentioned in Section 2.4.1, in this study, we focus on two operating modes. The first mode corresponds to when the electrical appliance is active, as denoted by (4) in Section 2.4.1. The second mode encompasses states (1), (2), and (3) from the same section, and is designated as unused. With the aim of minimizing energy consumption in buildings, the final objective of the proposed pipeline is to ensure plug loads are tuned on only when active, i.e., in use, and turned off otherwise. This includes situations when appliances are off but drawing a minimal current, in standby, or idle.

The proposed solution is based on an ensemble of statistical and unsupervised learning methods. The method automatically detects plug load

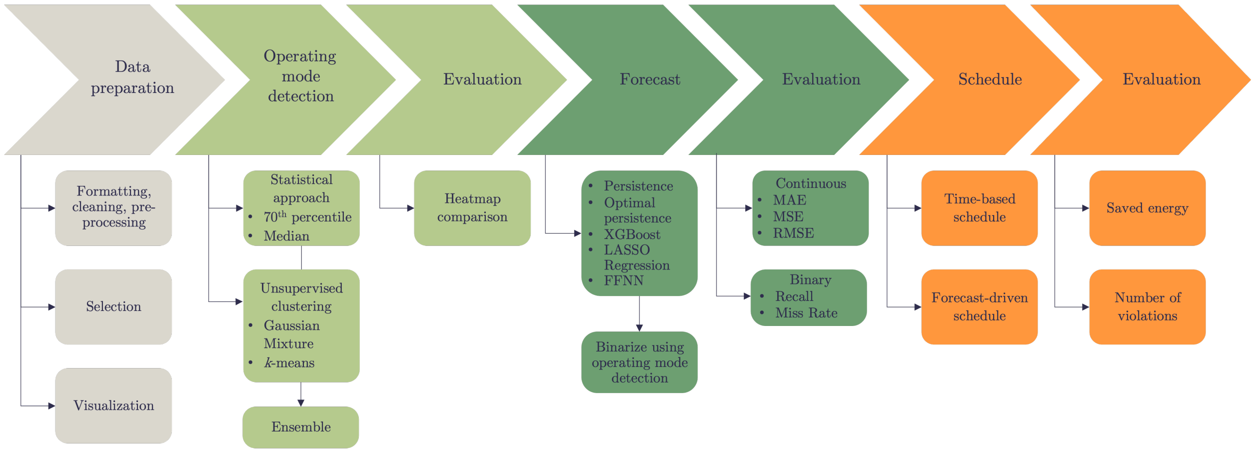


Figure 1. The proposed pipeline for plug load scheduling, designed to enhance energy building efficiency encompasses four principal phases: (i) Data preparation, a one-time procedure that consolidates disparate data recordings into a single, clean dataset ready for analysis; (ii) Operating mode detection, employing automated methods to distinguish between active and unused power levels for each plug load; (iii) Forecasting, which predicts plug load power a day in advance; and (iv) Scheduling, which controls plug power states to reduce energy usage, turning off appliances that are not in use.

operating modes, and removes the dependence on labor-intensive manual processes discussed in Section 2.4.2. Section 3.2.1 details the two statistical approaches, Section 3.2.2 explains the three unsupervised approaches, and finally Section 3.2.3 proposes an ensemble scheme to robustly combine these. Each method, is applied individually on each plug load, and only uses the first 80% of the plug load time series. A visual evaluation is performed against each of the method applied on the test set, i.e., the remaining 20%.

3.2.1. *Statistical approaches*

Two statistical approaches are proposed. They each generate a threshold, which is then used to binarize the power values. The power values higher than the computed thresholds are considered as active (1) and values lower than the thresholds are considered as unused (0).

The first statistical approach considers the median of all recordings as the threshold. The median is favored over the mean, since it is less sensitive to outliers. A limitation of this approach is that this threshold assumes that an appliance is on average used more often than not.

The second statistical approach tries to address this limitation by selecting a threshold value equal to the 70$^{\text{th}}$ percentile of the recording values. The underlying assumption is that an appliance is typically active during weekdays from 8 am to 6 pm. Mathematically, this implies the appliance is active approximately 29.76% of the time. Therefore, the top 30% of values are considered as active, establishing a threshold to differentiate between active and unused states based on the 70$^{\text{th}}$ percentile. Percentile also have the advantage that no mathematical assumption about the underlying distribution of the data has to be made.

3.2.2. *Unsupervised clustering approaches*

Three unsupervised clustering approaches are proposed to classify the power values as the appliance being active (1) or unused (0), i.e., either in idle, standby, or off operating mode. For each approach, the number of clusters is an input parameter and is set equal to two. The same pre-processing is applied in each method: the 0 W recordings and NaN values are dropped, as well as extreme positive outliers. Outliers are identified as values surpassing a cut-off, determined by the mean plus 30 standard deviations. It is improbable for a device to exhibit such drastic consumption variations. This detection process is applied only when there are fewer than five such outliers. This approach seeks to mitigate potential data recording errors. Observing up to four values significantly deviating from the distribution suggests potential recording anomalies. We observe that this outlier removal occurs for less than 5% of the plug loads. Another purpose of this outlier detection is to address the sensitivity of both k-means and Gaussian mixture models to outliers.

The first unsupervised clustering approach uses the k-means algorithm. The missing values (NaN) are dropped and k-means clustering is applied with default parameters, i.e., k-means++ initialization and squared Euclidean distances metrics. This method aims to partition the data into distinct, non-overlapping subgroups, or "clusters". k-means assumes clusters to be spherical and of comparable size. The squared Euclidean distance renders it sensitive to outliers. Its scalability, especially with large datasets, makes k-means a popular choice across diverse research areas. For more details on k-means, the reader is directed to foundational literature [44, 45].

The second and third unsupervised clustering approaches are based on Gaussian Mixture Models (GMM). One of them uses the entire training set while the other only uses the most recent three months of data, and can be updated over time. Fundamentally, a GMM is a probabilistic model that assumes data points are derived from a mixture of several Gaussian distributions with unknown parameters. In this use case, the number of underlying Gaussians is set to two, since we are aiming to determine active versus unused modes. The initialization used in both GMM approaches is the same as for the k-means approach. The model achieves convergence through the expectation-maximization (EM) algorithm. In terms of scalability, GMMs are well-suited for datasets of moderate size but may demonstrate slower performance with larger datasets. While k-means strives for clusters of similar spatial shape, algorithms like GMM allow for clusters with various shapes and sizes. The reader is referred to the existing literature on GMM and for theoretical and implementation details [46, 45]

In all methods, we consistently apply three post-processing steps. Initially, we ensure the cluster labeled as 0 corresponds to the values with the lowest average, given that our intent is for the 0 label to represent unused mode (including off, standby and idle); and 1 to indicate active mode. Subsequently, power readings of 0 W are always labeled as 0, knowing the device is off (thus unused). Lastly, and mainly relevant for the GMM clustering, values under the median of the 0 cluster but tagged as 1 are reclassified as 0. Conversely, values above the median of the 1 cluster but marked as 0 are re-labeled 1. This avoids the outlier effect, which GMM can sometimes produce.

A limitation of both the k-means and GMM algorithms is that they depend on the initialization step. Further experiments with varying initialization could be done in future work [47].

3.2.3. Ensemble

An ensemble technique [48] is introduced to combine insights from all the aforementioned methods. The ensemble works on a majority principle, with a safety margin, specifically designed for appliance scheduling. If none of the five operating mode detection approaches assigned a power value as active, this power value is labeled as unused (0). Similarly, if only one method considered it active, it was still labeled as unused (0). When at least two out of the five methods agree upon active mode, a power value is considered active (1). The ensemble adopts a sort of weighted average, while prioritizing user convenience, keeping the end goal of appliance scheduling in mind. It is preferable to label a power recording as active, ensuring the appliance remains scheduled on, than to mistakenly label it as unused and potentially schedule the appliance off, while a user might utilize the appliance. Note that in the clustering literature, ensembles are sometimes also called consensus clustering.

3.2.4. Operating mode detection evaluation

There is no ground truth (or "true threshold") to evaluate the proposed clustering ensemble approach. We suggest assessing the mode detection accuracy using heatmap comparisons. One can visualize the time series of plug power levels as a heatmap, as depicted in Figure 2. A heatmap displays the plug power level using color intensity, with days on the x-axis and hours on the y-axis. This representation captures the weekly plug load usage pattern. Two heatmaps of the same plug load can be constructed: the original based on the original continuous power levels, and the heatmap after applying

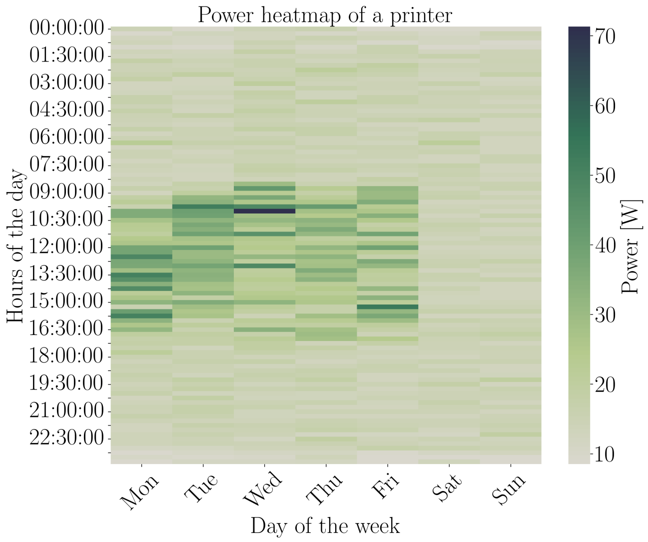


Figure 2. Representation of a three-months long time series of plug power values, visualized as a heatmap. The power level is indicated using color intensity, the days of the week are on the x-axis and hours of the day on the y-axis. Each cell represents the average power value for a combination of a day of the week and a time of the day, e.g., the average power level on a Wednesday at 1000 h is around 70 W. A weekly appliance usage pattern is clearly visible between Monday and Friday during office hours from 0830 h to 1700 h.

mode detection with binary levels. The binarized heatmap should mirror the original's weekly usage pattern. This pattern remains crucial as the pipeline end goal is to schedule appliances in line with their individual usage.

Quantifying the similarity between two heatmaps is not straightforward. Out of the many existing image similarity metrics in the literature, we selected three simple metrics: the MSE [49], the RMSE [50], and the Pearson correlation coefficient [51]. We also considered two more advanced and established metrics: the structural similarity index (SSIM) [52, 53] and the Kullback-Leibler (KL) Divergence [49]. However, none reflected the similarity as the human eye perceives it, nor are they suited for this particular application. Patterns that align perfectly should receive high similarity scores, and minor deviations in timing and values be acceptable. A significant disparity, such as continuous high power on a Monday in the original data that is not present (marked as active) in the binarized heatmap, should not score highly. The literature agrees that evaluating image similarity as perceived by human eye is challenging [52, 49]. Consequently, we decided to exclude these metrics and results from the paper, as they were found to be unsuitable for numerical mode detection evaluation.

Our proposed evaluation method involves visual qualitative heatmap comparisons. We classify the results into three distinct categories based on how well the algorithm captures the underlying usage pattern:

- High-fidelity detection: the binarized heatmap closely mirrors the original heatmap, capturing the usage pattern of the plug power levels. This suggests that the algorithm has successfully detected and represented the true underlying usage pattern.
- Partial-fidelity detection: while the binarized heatmap generally aligns with the original, there are discrepancies. For example, the power value is always marked as active correctly when it is indeed active, but power levels are also sometimes incorrectly labeled as active during some of the unused periods. This indicates that while the algorithm detects most active patterns, it does not provide a high-fidelity match.
- Low-fidelity detection: the binarized heatmap significantly deviates from the original. This could manifest as the plug load being marked as continuously active or continuously unused despite contrary evidence in the original data, or the algorithm failing to capture the usage pattern at all.

This approach has limitations, such as the inability to quickly assess multiple plug loads and subjectivity. Further research in that direction, to find, or design, an appropriate similarity measure with the right properties for this use case is left for future work.

3.3. Forecasting

The forecasting problem, highlighted in dark green in Figure 1, is characterized by two dimensions: spatial aggregation and horizon. Spatial aggregation is the level of load being predicted. In this context, we focus on forecasting the average power of individual plug loads in 15 min intervals. Regarding the horizon, the model produces forecasts for the upcoming day, with data recorded in quarter-hourly intervals, this yields 96 predictions. This section elaborates on several implemented forecasting models based on diverse approaches: a naive baseline, an enhanced naive approach, a linear regression, and finally a more advanced machine learning approach. It is good practice to clarify all forecast time parameters [54]. In this paper the forecast update rate is 24 h (daily at midnight), the forecast resolution is 15 min, the forecast horizon is 24 hours, and the forecast lead time is 0 h.

3.3.1. Persistence

A persistence model serves as a naive baseline, where no modeling is involved. In the absence of complex training, the most straightforward assumption is that today's consumption will be similar to yesterday's consumption.

Another possible assumption is that the load pattern of an electrical appliance on a Monday is more likely to look like the load pattern of the previous Monday than the previous day, i.e., a Sunday. Another persistence model using the same day of the previous week is proposed.

3.3.2. Optimal Persistence

An enhanced version of the persistence model, incorporating calendar information, holiday, and using averaging, is developed. Similar to the persistence model, this optimal persistence model is based on historical data without training. For Tuesday through Friday and Sunday, the forecast is generated by averaging the consumption values of the previous day and the corresponding weekday from the previous week. When forecasting Mondays or Saturdays, the prediction is generated by averaging the values from the same weekday of the prior two weeks. Additionally, when the forecast day is designated as a holiday in the UCSD calendar, the model selects the two nearest holidays or Sundays in the past and averages their consumption data.

3.3.3. Autoregressive model with exogenous input and Lasso regression

An Auto Regressive Model with eXogenous input (ARX) is developed, utilizing Lasso regression. This model has two days of historical data (one day ago and one week ago) along with a holiday binary indicator as inputs. Historical data inputs are scaled between 0 and 1. The output of the model is a one-day-ahead point prediction, i.e., 96 values are predicted since the dataset is in quarter-hourly resolution. The least absolute shrinkage and selection operator, commonly referred to as Lasso regression, is a technique in statistics and machine learning that combines variable selection with regularization. By constraining the sum of the absolute values of the regression coefficients to be below a set threshold, i.e., the L1-norm, Lasso regression effectively drives certain coefficients to zero [55, 56]. We set the parameter α, the penalty term that denotes the amount of shrinkage, equal to 0.001.

3.3.4. XGBoost regression

A more advanced machine learning approach is based on gradient boosting regression using the XGBoost library [57]. It has often proven successful in load forecasting research [58]. XGBoost models construct ensembles from decision tree models. Trees are incrementally added to the ensemble, each tailored to correct the errors of its predecessors. The fitting process involves minimizing a differentiable loss function using gradient descent. More details about XGBoost can be found in [57].

We implemented a local and a global approach. In the local approach, one XGBoost model is trained per plug load time series. In both approaches, seven days of historical data, scaled between 0 and 1, are used as inputs, and one day ahead is predicted, i.e., 96 values. In the global approach, one model is trained over all the plug load time series. In the global approach, early stopping and L1-regularization (alpha = 0.1) are implemented. In the local approach, more regularization is necessary to avoid overfitting. The parameters used are the following: the minimum sum of instance weight needed in a child is set to 8 (min_child_weight = 8), the maximum depth of the trees is set to three (max_depth = 3), the L1 regularization parameter alpha is set to 0.2 (reg_alpha = 0.2), the number of early stopping rounds is set to 20 (early_stopping_rounds = 20), the data is 80% subsampled (subsample = 0.8), and 80% of the features (col_sample = 0.8) are retained. The other parameters are kept to their default values.

3.3.5. Feedforward neural network regression

The proposed feedforward neural network (FFNN) takes two days of historical data as inputs, i.e., one day and one week prior, along with a binary holiday indicator. Inputs are scaled between 0 and 1 to normalize the range of data values. The network generates a one-day-ahead point prediction for the plug load, i.e., 96 values.

The architecture of the FFNN is adapted for two distinct approaches: a local and global approach. In the local approach, one FFNN is trained per plug load. The structure includes an input layer with 128 neurons using the Rectified Linear Unit (ReLU) activation function, a hidden layer with 64 neurons also employing ReLU activation, and an output layer comprising 96 neurons with linear activation to correspond with the 96 outputs for the day-ahead forecast.

In the global approach, one model is trained for all plug loads. The architecture incorporates an input layer with 526 neurons followed by a dropout layer with a rate of 0.2 to prevent overfitting. This is succeeded by two hidden layers with 256 and 128 neurons, respectively, both utilizing ReLU activation. The architecture ends with an output layer with linear activation and 96 neurons, similar to the local approach.

In both scenarios, the training process is governed by two callbacks: early stopping to prevent overfitting and a reduction in the learning rate on a plateau to fine-tune the learning process for better performance. The Adam optimizer is utilized for its efficiency in both cases, with a maximum of 100 epochs for the local model and 200 epochs for the global model, ensuring adequate learning while avoiding excessive computational demands.

3.4. Scheduling

In the pipeline depicted in Figure 1, the scheduling step is emphasized in orange. The proposed approach integrates the outputs from both the ensemble clustering and forecasting models. Here, one-day-ahead forecasting is applied on the test set. Subsequently, the continuous forecast is binarized through the ensemble clustering model. This binary forecast then guides the individual appliance scheduling, determining the activation and deactivation commands and their respective timings. We decided to call this schedule “forecast-driven”. A baseline scheduling approaches is also proposed as a comparison. This baseline schedule, referred to as ’time-based’ control in the literature, relies solely on predefined office hours. We also compare the baseline and the forecast-driven schedules with no schedule.

3.4.1. Time-based schedule

A fix time-based schedule is developed. Monitored appliances are turned off on weekends and every weekday between midnight and 5 am.

3.4.2. Forecast-driven schedule

In the proposed forecast-driven schedule, logic rules determine the on/off commands of the appliances based on the binary predictions. These binary predictions are the outputs of the subsequent application of the forecast model, in a one-day-ahead rolling window fashion, and the operating mode detection clustering. In the clustering output, label 0 denotes that the appliance is unused (encompassing operating modes off, standby and idle). Conversely, label 1 signifies that the device connected to the smart plug is actively used. The rule for on/off scheduling is straightforward: if a device is consistently labeled as 0 (unused) for a minimum of n consecutive time steps, it is directed to switch off. In any other scenario, which includes instances where it is marked as 0 (unused) for fewer than n time steps or when it is labeled as 1 (active), the device is turned on, or remains on.

3.4.3. Scheduling evaluation

The end schedule is evaluated using four different metrics:

- Saved energy: The absolute saved energy is the difference in consumption between the true test set and the test set with the implemented schedule, i.e., time steps marked as turned off in the schedule have a zero consumption. The percentage of saved energy over the test set is computed by dividing the absolute value by the total energy. These relative and absolute saved energy can be computed per day, per week, or per appliance to obtain different insights. The schedule aims to save as much energy as possible.
- Number of violations: The number of violations is the number of times that the appliance is scheduled as off while it is actually marked active in the mode detection (on the test set). The

schedule of off periods is informed by the mode detection algorithm based on the forecast load values, followed by the scheduling step. The ground truth for assessing violations is based on the mode detected on the actual test set, rather than on the forecast values. The percentage of violations is also computed. These relative and absolute number of violations can be computed per day, per week, or per appliance to obtain different insights. The schedule should not prevent employees from working or using their appliances as they need. A minimum number of violations is desired for user convenience.

- Number of missed chances: The number of missed chances is the number of times that the appliance is scheduled on while it is not used. It is the number of time we could have saved energy but did not. The percentage of missed chances is also computed. These relative and absolute number of missed chances can be computed per day, per week, or per appliance to obtain different insights. When designing the schedule, a greater safety margin or expanded time-based schedules can result in increased missed chances for energy savings. It is crucial to strike a balance between energy conservation, the number of violations, and missed chances.
- Number of turn on/off commands: The number of times that an appliance is turned on and off in a day or in a week can also be computed. This can help assess if the schedule might adversely impact appliances, as some appliances experience increased wear-and-tear when turned on or off.
- Energy efficiency: We also employ a metric termed energy efficiency. This metric is the ratio of useful energy to total energy consumed. Useful energy refers to the energy expended while the appliance is active. If the appliance operates solely when in use, the total energy consumed matches the useful energy, resulting in an efficiency of 100%.

4. Case Study

4.1. Problem statement

The pipeline proposed in this paper tackles the challenge of office plug load scheduling. In a first step, the data is pre-processed. Secondly, the operating mode of each plug load is detected, two modes are targeted: active or unused (encompassing idle, standby and off. Third, the plug load state (active or unused) is predicted one day ahead. Finally, the forecast and the mode detection results are used in the scheduling step, along with logic rules to determine whether an appliance should be turned on or off one day ahead.

4.2. Dataset

The raw dataset is collected through smart plug sockets between the wall plugs and the electric appliance as detailed in [8]. About 650 smart plugs from Best Energy Reduction Technologies (BERT) are deployed in fifteen buildings of the campus of the University of California, San Diego (UCSD). The power level of each individual appliance is recorded in mW at fifteen or at five minutes interval in Jason files and accessible in a data lake. The smart plugs were installed in 2020 and–with a few exceptions such as office moves–are still active to date.

The process of transforming the individual Jason file recordings into a usable dataset necessitated intensive data cleaning and formatting. The specific details lie outside the scope of this paper and will be elaborated upon in a subsequent publication centered on the dataset. This section provides a brief overview of critical pre-processing steps.

The dataset contained a blend of 5-minute (6% of data) and 15-minute (94% of data) recordings. To maintain consistency, these were resampled to a quarter-hourly resolution. Plug load profiles that were only composed of zero recordings or lacking metadata were excluded. While there were no missing values in terms of empty recordings or NaNs in the raw data, instances were identified where the system was inoperative, such as during system-wide updates. It was observed that in such cases, the recording system substituted the gaps with zeros. We elaborated several scenarios to discern these erroneous zeros from actual zero power recordings and replace them with NaNs. One such scenario involved instances where all plug loads in a building simultaneously displayed zero readings over three or more time steps. In such cases, since it is unlikely that all appliances are unused for a consecutive 45 minutes, these readings were replaced with NaNs. Another scenario considered the start of a plug load time series with prolonged time of zero values, succeeded by consistent power level fluctuations for the rest of the recording period. This suggested non-usage during the COVID lockdown, followed by reactivation of the devices. The initial zeros did not truly represent genuine usage trends. Lastly, prolonged zero sequences towards the end of a plug load time series were substituted with NaNs, assuming the appliance's disconnection from the smart plug, preventing recordings. After these steps, 625 plug load recordings spanning over at least a year remained.

In the subsequent phase, a subset of plug loads was chosen for this particular study. Plug loads with over 30% of NaNs or 30% of zeros in total were excluded. Additionally, it was ensured that the test set contained less than 3% NaNs, as the aim was not to schedule underutilized appliances.

The final selection was based on the plug device type. Although a large number of computers were monitored, they were excluded from scheduling to avoid potential disruptions for employees, especially those who may be running models overnight or who might not have backed up recent work. This rigorous selection yielded a dataset of 169 high-quality smart plug time series spanning 498 days, from November 18^{th}, 2021, to March 31^{st}, 2023. The 169 plug loads consist of 146 printers, 16 copiers, 4 TVs, and 3 fax machines.

This final pre-processed dataset is openly accessible [59]. In the upcoming publication centered on the dataset, further analysis and visualizations that were a central part of the data exploration, will be detailed.

4.3. Results & discussion

4.3.1. Mode detection

The mode detection approach based on the clustering ensemble is applied on the given dataset. This method converts the power level values into binary values based on the detected mode. A label of 1 indicates an active appliance, while a label of 0 indicates an unused appliance, encompassing off, standby, or idle states. As explained in Section 3.2.4, a visual evaluation of mode detection accuracy is undertaken through a heatmap comparison. The accuracy of the mode detection is classified into three categories: high-fidelity, partial-fidelity, and low-fidelity. The results are detailed in Table 1. Close to 70% of the plug loads fall under the high-fidelity category, indicating that the algorithm correctly discerns the appliance usage patterns. Another 17% of the plug loads demonstrate partial-fidelity detection. In these instances, while the algorithm identifies the core active usage patterns, it also mistakenly labels some unused power values as active. Lastly, the algorithm exhibits low fidelity for 12% of the plug loads.

	Number of plugs	Percentage of plugs
High-fidelity	119	70.41 %
Partial-fidelity	29	17.16 %
Low-fidelity	21	12.43 %
Total	169	100 %

Table 1. Mode detection results are presented based on a qualitative comparison between original and binarized heatmaps. High fidelity denotes accurate usage pattern detection. Partial fidelity suggests the detected pattern includes the original, but some values labeled as active may be inactive. Low fidelity means the algorithm does not capture the underlying usage pattern accurately.

Given the qualitative nature of the evaluation, we believe it is essential to provide a detailed illustration of these results. All figures discussed in this section are located in Appendix B to enhance textual continuity and readability. The first heatmap is built using the original power data from the test dataset. The second heatmap is generated with the binarized test data through our detection algorithm. The comparison of these two heatmaps is used to categorize our algorithm performance. The last heatmap is formed with the binarized test data, based on power thresholds specified in product specification sheets. Appliance metadata, including aspects like type, brand, and model, facilitated the acquisition of the pertinent specification sheet and power threshold details. A dotted line on the initial time series plot also depicts this specification sheet threshold. The discussion around these specification sheets thresholds concludes this section. Note that the units and scales of the heatmaps are different, as the original data is in Watts, while the data after mode detection is in binary format.

Figure B.3 presents instances of high-fidelity detection results. In every example, our mode detection algorithm effectively captures the continuous power usage pattern in the binarized format. Devices showcased include a printer, a TV, and a copier.

Figure B.4 displays several plug loads classified as partial-fidelity. In each instance, the detected modes encompass the active usage pattern, but the algorithm occasionally labels unused power values as active. However, this does not pose an inconvenience to the user; the appliance will not be inadvertently turn off when it should remain on. For some plug loads, there is no distinct original usage pattern, which could be a reason for the lower performance of the algorithm.

In our focused examination of the **low-fidelity** detection results, we aim to understand potential sources of failure and avenues for the improvement of the mode detection algorithm. We illustrate, discuss and propose solutions to each challenge. *The primary set of challenges relates to the inherent attributes of the plug load data.* There are appliances for which the absolute difference in Watts between the active and standby mode is very small, as illustrated in Figure B.5a. Given the fact that GMM clustering allows overlapping clusters, if the small absolute difference between two modes makes them hard to detect, there exists a risk of unintentionally turning the device off during its active mode. It might be prudent to consider whether these devices should be scheduled at all.

Additionally, situations arise where three distinct modes are perceptibly present. However, the current design of the mode detection discerns only two modes, given the set input parameter in the k-means as well as the GMM clustering. This might potentially affect the efficacy of the proposed algorithm on these specific profiles. An example is illustrated in Figure B.5b. To mitigate this, a solution would be to apply mode detection with parameters adjusted for both two and three modes. Subsequent performance evaluation on a validation set would then determine the most optimal parameterization. Another observed phenomenon is the occurrence of concept drift [39]: over time, devices can exhibit changing behavior. An example is shown in Figure B.5c. This might be due to wear and tear, plug load operating software updates, users changing the plugged appliances, or users employing power strips. This challenge can be addressed by detecting concept drift and subsequently re-training the mode detection algorithm. Raising awareness among users about the scheduling system and the repercussions of unplanned alterations is also an option.

The second type of challenges is inherent to the intrinsic attributes of the implemented algorithm. It is well known that algorithms like k-means and GMM are sensitive to their initialization. To improve the robustness of the algorithm, one approach would involve running the mode detection task multiple times, and then averaging the outcomes. Alternatively, another strategy would entail focusing on adopting a more robust initialization method. Furthermore, in our algorithm, each power value is currently evaluated and labeled individually, not accounting for its sequential or temporal context. This can lead to overlooking specific transitional phases. Including additional features, like the time of day, could potentially increase the accuracy of the algorithm. Another potential improvement is linked to the resolution of measurements. Recordings with a higher frequency might improve detection. Indeed, devices like televisions or printers typically transition between off and active modes within a 15 min interval; therefore 15 min averages may contain power values from multiple modes.

Lastly, we compare our mode detection algorithm with thresholds from appliance specification sheets. For each example highlighted in this section, we determined the power for both active and unused modes, as provided in the appliance specification sheets. Despite the extensive manual effort required, it is worth noting that such information is not always accessible. Out of the 13 appliances presented in Figures B.3, B.4, B.5 only one specification sheet thresholds yields high-fidelity results: Figure B.4(a). The printer in Figure B.4(b) receives a partial-fidelity classification; it aptly labels the active mode but also mislabels certain power values that could be marked as unused. The remaining mode detection results using specification sheets fall into the low-fidelity category. It is worth observing that the copier in Figure B.3(b) seems to capture the general usage pattern but mislabels several active power values as unused, posing a risk of appliance deactivation during actual use. One reason for apparently overly large specification sheet thresholds could be the 15 min averaging period. The specification sheet threshold assumes that the device is active during the entire 15 minute period. For example, for a copier or printer, this would mean continuous printing or copying for 15 minutes. Such usage is unlikely in practice. Most often, a 15 min interval will contain periods of activity together with inactivity or even standby modes. The inactive (unused) periods would then cause the 15 min average power to decrease below the specifications threshold.

4.3.2. Forecasting

The different forecasting model detailed in Section 3.3 were implemented in python and evaluated on an unseen test set, i.e., the last 20% of the dataset. Traditional distance metrics, the Mean Absolute Error, Mean Squared Error and Root Mean Squared Error were used. Smaller values indicate better performance. Table 2a presents the average metrics over all plug loads. Additionally, the percentage improvement in terms of RMSE is computed compared to the naive persistence model based on the consumption of the same day a week ago.

Another evaluation of these models is proposed, oriented towards the end goal of the forecasting: the appliance scheduling. The ground truth consists of the unused / active labels of the operating mode detection algorithm applied directly on the true test set power values, as described in Section 3.2. The continuous power point predictions from the forecasting are also binarized using the operating mode detection algorithm and compared with the ground truth. The evaluation consists of two traditional and complementary classification metrics: the recall and the miss rate. The recall is also called true positive rate, sensitivity, or probability of detection. In our specific case, it is an indication on how many times the appliance has been labeled as active out of all the times it was actually active. We aim to have a recall as high as possible. The miss rate, also termed false negative rate, is computed as 1 – recall. It highlights the number of times that the appliance was labeled as unused while it was active. We want the miss rate to be as low as possible, as this can cause inconvenience to the users of the appliances. Additionally, the percentage improvement in terms of recall is computed as a function of the naive persistence model based on the consumption of the same day a week ago. These results are presented in Table 2b.

In both tables, the models are ordered from the best performing to the least performing based on the respective metric. This ordering varies significantly depending on the chosen evaluation metric. This variation underscores the importance of selecting appropriate metrics that align with the end goals of the forecasting process. In both evaluation methods, the global feedforward neural (FFNN) network model ranks as the top performer. These findings align with Spiliotis et al.'s [60] results on intermittent time series forecasting one-step-ahead. The global FFNN model achieves a 26% improvement in RMSE compared to the baseline and a 42% improvement in recall. Interestingly, the addition of holiday data does not reduce RMSE for both local and global feedforward neural networks, nor for the LASSO autoregressive model. However, adding holiday data does seem to reduce the miss rate by nearly 5% for the local FFNN.

It is important to note that the ground truth used in these evaluations is derived from the preceding operating mode detection algorithm and is not an absolute measure of the ground truth. Consequently, while this ranking offer useful insights into model performance, the results should be interpreted with some caution.

4.3.3. Scheduling

The scheduling results are presented in Table 3. The "true test set schedule" is the schedule obtained using the true values of the test set, as if the predictions were perfect. The true values of the test set are binarized using the operating mode detection algorithm, and the schedule is generated using the forecast-driven approach described in Section 3.4.2. This can be considered as the perfect schedule in terms of forecast, still keeping in mind that there is no real ground truth for the operating mode detection

	MAE [W]	MSE	RMSE	Improvement (%)
Global FFNN	3.85	458.26	10.53	26.00
Individual FFNN	3.53	482.49	10.79	24.17
Global FFNN with holidays	3.65	467.21	11.07	22.21
Individual FFNN with holidays	4.00	489.28	11.15	21.64
Global XGBoost	2.86	529.23	11.51	19.11
Individual XGBoost	4.67	522.19	12.18	14.41
Optimal Persistence	2.79	642.54	12.28	13.70
ARX	4.49	606.44	12.33	13.35
ARX with holidays	4.51	597.54	12.38	13.00
Persistence 1 day	2.61	809.95	13.91	2.25
Persistence 1 week	3.03	817.65	14.23	/

(a) Forecasting results based on the evaluation of Mean Absolute Error, Mean Squared Error and Root Mean Squared Error on the continuous predictions.

	Recall	Miss Rate	Improvement (%)
Global FFNN	74.09	25.91	42.55
Individual XGBoost	71.27	28.73	40.27
ARX	69.55	30.45	38.80
ARX with holidays	68.59	31.41	37.94
Global XGBoost	59.72	40.28	28.72
Global FFNN with holidays	59.17	40.83	28.06
Individual FFNN with holidays	58.68	41.32	27.45
Individual FFNN	54.14	45.86	21.38
Persistence 1 day	48.73	51.27	12.65
Optimal Persistence	45.52	54.48	6.48
Persistence 1 week	42.57	57.43	/

(b) Forecasting results based on the evaluation of recall and miss rate on the binarized predictions. The continuous predictions are binarized using the operating mode detection algorithm.

Table 2. Forecasting models evaluated on the test set, i.e., the last 20% of each plug load time series and averaged over all plugs. (a) Forecasts of mW of individual plug load average power consumption over 15 min intervals are compared to plug load control measurements. (b) The binarized predictions derived from the forecasts are compared to the binarized output of the ensemble plug load operating mode detection model presented in Section 3.2. The binarized predictions from the forecasts are obtained by applying the plug load operating mode detection model to the forecast data. Therefore active versus inactive thresholds may differ between forecasts and ground truth. In both tables, the models are ordered from the best performing to the least performing based on the respective metric. This ordering varies significantly depending on the chosen evaluation metric. This variance underscores the importance of selecting appropriate metrics that align with the end goals of the forecasting process. In both evaluation methods, the global feedforward neural network model ranks as the top performer.

step. The "no schedule" approach considers the appliances as always on. The baseline schedule is the simplistic time-based approach, as described in Section 3.4.1. These three specific schedules are interesting to compare to the forecast-driven methods, in the sense they each give us insights into the realistic expectation of the pipeline. The true test set schedule indicates the maximum amount of energy that could be saved, i.e., 42.07% along with the best energy efficiency that could be obtained, i.e., 87.32%. This perfect schedule also shows that there are at least 6.28% of missed chances, these are occurrences when the appliance is marked as unused for such a short period that the schedule does not deem it long enough to actually turn the appliance off. This perfect schedule has one of the highest number of turn on/off commands. When no schedule is applied, the percentage of missed chances is 70%. This confirms our assumption in Section 3.2.1 that the appliances are, on average, approximately used 30% of the time. In the no schedule case, we also have an indication of the worst energy efficiency, i.e., 39.79%. Finally, the baseline schedule provides insights on the metrics that can be attained without complex computing, training, nor data collection and the proposed approaches should at least be better than the baseline.

Selecting the right schedule depends on the specific application. For instance, when scheduling critical appliances, minimizing the miss rate is important to ensure reliability and continuity for the users. Conversely, in scenarios where non-critical appliances are scheduled, the objective may shift towards maximizing energy savings. Formulating an optimization problem that integrates these different considerations, would allow selecting a schedule that aligns with the specific needs.

The results table clearly indicates that the global FFNN leads to the most effective forecast-driven scheduling. The proposed pipeline, while innovative and promising, also highlights that enhancements in forecasting, i.e., moving closer to the "True Test Set Schedule" could substantially increase energy savings and reduce the miss rate.

5. Future research directions

In this section, based on the reviewed literature, the presented research, and the result discussion, we outline potential research directions in smart plug utilization and building energy efficiency. These suggestions provide potential extensions and improvements to the existing study.

- *Mode detection*: The mode detection algorithm is the first to provide an automated way to discern between plug load modes, yet it has limitations that could be addressed through several improvements. Firstly, a suitable metric for evaluating mode detection algorithms would enable more effective comparisons. Secondly, exploring alternative algorithms, such as kernel spectral clustering, could yield promising results [**?**]. Lastly, determining the optimal number of clusters, i.e., modes, using methods like the elbow method, and then applying a post-processing step to select only the relevant modes, could enhance the performance of the mode detection.
- *Occupancy data*: The integration of occupancy data into plug load forecasting models is recommended to increase accuracy. Several studies [43, 61, 40] have underscored the significant role of occupancy data in plug load predictions.
- *Categorizing plug loads*: Wang [31] suggests categorizing plug loads based on their operational modes: continuous versus intermittent. For instance, while water dispensers generally run continuously, devices like TVs operate intermittently. Adopting distinct mode detection and forecasting strategies for each category could yield more accurate predictions. Note that the plug loads in the proposed dataset are all intermittent. Another approach could be to cluster the time series and then developing a forecasting model per cluster or device type, leveraging more historical data. Cluster-based forecasting is consistent with the new trend of global forecasting models [39].
- *Addressing plug load appliance movement*: this research paper assumes static loads, i.e., the plugged in appliances remain plugged in and do not change. Detecting novel devices or changes in plugged in device, and adapt the scheduling accordingly would provide more flexibility and robustness.
- *Transfer learning*: The idea of leveraging hand labeling from existing datasets for transfer learning on new datasets, given device type consistency, is promising. The dataset by Kalluri et al. [23], which was visually inspected and manually labeled, could provide insights for similar applications.
- *Impact of each pipeline step*: It would be interesting to assess numerically the performance at each stage of the proposed pipeline. Specifically, the impact of accurate operating mode detection and the impact of improvements in forecasting on the subsequent scheduling tasks could be assessed.
- *Return on investment*: The economic implications of plug load monitoring cannot be overlooked. Comprehensive financial analyses, as

	Number of violations (%)	Missed chances (%)	Energy saved (%)	Number of turn on/off commands per plug per day	Energy Efficiency (%)
True Test Set Schedule	0.0	6.28	42.07	4.33	87.32
No Schedule	0.0	70.26	0.0	0.0	39.79
Global FFNN Schedule	2.21	33.11	21.92	1.62	45.89
Individual XGBoost Schedule	2.75	35.82	16.51	1.82	44.77
Global XGBoost Schedule	3.27	27.67	25.84	3.33	47.92
Individual FFNN Schedule	4.61	23.77	29.38	3.71	47.85
ARX Schedule	5.09	28.63	20.38	2.18	47.35
Optimal Persistence Schedule	6.05	10.03	40.15	3.4	55.77
Persistence 1 day Schedule	6.05	12.70	46.17	4.26	54.51
Persistence 1 week Schedule	7.95	14.30	47.70	4.32	50.31
Baseline Schedule	11.4	38.09	41.88	1.45	43.28

Table 3. The different forecast-driven schedules are compared. The “true test set schedule” can be considered as the perfect schedule, it indicates the maximum amount of energy that could be saved if the forecast were perfect. The “no schedule” approach considers the plugs as always on. In the no schedule case, we have an indication of the worst energy efficiency and the worse percentage of missed chances. The baseline schedule is the simplistic time-based approach, it provides insights on the metrics that can be attained without complex computing, training nor data collection and the proposed approaches should at least be better than this. The results table clearly indicates that the global feedforward neural network leads to the most effective forecast-driven scheduling.

demonstrated in Trenbath et al.[1] and Wang et al.’s[62] return on investment analysis, could shed light on the actual monetary benefits and potential payback periods of such initiatives.

- *Frame a global optimization problem*: Optimizing the scheduling of energy consumption should be approached from a multifaceted perspective, encompassing occupants’ convenience, financial gains, and the overarching goal of energy reduction. This multi-objective optimization problem can pave the way for sustainable energy management solutions in the future [12].
- *Randomized control trial*: In the context of smart plug scheduling, implementing a randomized controlled trial could offer valuable insights into the practical effectiveness of the proposed scheduling strategies. Specifically, we suggest applying plug load scheduling for an 24-hour duration, refraining from scheduling in the subsequent 24 hours, and then reintroducing the scheduling for another 24 hours. This cycle would be maintained over several weeks. By comparing the scheduled hours against the non-scheduled hours, performance metrics such as energy conservation and user satisfaction could be assessed.
- *Eco-feedback*: Promotion of occupants’ awareness, also called eco-feedback, emerges as a promising element in energy reduction. Kamilaris et al.[5] conducted an illuminating study over 22 weeks, targeting the understanding of office workers’ energy consumption behaviors. Results revealed the compelling impact of continuous personal feedback, with notable reductions in energy use observed in the third and fourth months. Moreover, the study highlighted the efficacy of group-level feedback and peer-education interventions in fostering energy-saving behavior. Doherty et al.[2] and Gandhi et al. [63] further emphasize the potential of informational campaigns, rewards programs, gamification, and regular energy usage feedback in enhancing occupants’ energy con-

servation habits.

6. Conclusion

This paper introduces a pipeline for scheduling plug loads in office buildings with the aim of energy reduction. The pipeline integrates three components: plug load operating mode detection, plug load forecasting, and plug load scheduling based on the two prior components. The objective is to reduce the energy consumed when appliances are not actively used.

Significant contributions of this research include the development of an extensive dataset, capturing over a year's worth of data from 169 plug loads in multiple office buildings. This dataset is made publicly available to foster further research. The first literature review on plug load operating mode detection methodologies and terminology is presented, clarifying terminology and providing a foundation for future work in this domain. An unsupervised approach for determining plug load operating modes is proposed, which has been a challenge in past literature. Additionally, the research opens the domain of individual plug load forecasting, an under-explored area. The resulting pipeline demonstrates potential plug load energy savings of up to 50%, marking a significant step towards building energy efficiency. The developed code is made publicly available for reproducibility and benchmarking in further research work.

The research also acknowledges certain limitations, such as the absence of a numerical metric for evaluating the plug load mode detection algorithm accuracy and opportunities in improving forecasting models. A specific section is dedicated to future research in the smart plug field.

In conclusion, the research presented proposes a novel and promising framework for office plug load scheduling and underscores the importance of sharing data and methodologies. It lays a foundation for collaborative progress and sets a benchmark for plug load mode detection, forecasting, and appliance scheduling towards energy efficient buildings.

Acknowledgment

The work of this paper has been carried out as part of a research stay at the University of California, San Diego, in collaboration with the research group STADIUS from KU Leuven. This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program, from the KU Leuven Research Fund (projects C16/15/059, C3/19/053, C24/18/022, C3/20/117, C3I-21-00316), from the European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Adv. Grant, grant agreement N^{o}885682) and a FWO travel grant for long stay abroad (V408523N). The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government. During the preparation of the paper, the authors used ChatGPT in order to improve some language formulations. After using this tool, the authors carefully reviewed and edited the content as needed and take full responsibility for the content of the publication. The UC San Diego team acknowledges funding from the California Energy Commission contract EPC-20-009 on "Smart Plug Load Controls Integrated with Building Energy Management Systems".

References

[1] K. Trenbath, R. Tuttle, K. Maisha, A. LeBar, Assessing and Reducing Plug and Process Loads in Office Buildings, Tech. rep., National Renewable Energy Laboratory (2020).
URL `https://www.osti.gov/biblio/1530714/`

[2] B. Doherty, K. Trenbath, Device-level plug load disaggregation in a zero energy office building and opportunities for energy savings, Energy and Buildings 204 (2019). `doi:10.1016/j.enbuild.2019.109480`.

[3] S. Lanzisera, S. Dawson-Haggerty, H. Y. Cheung, J. Taneja, D. Culler, R. Brown, Methods for detailed energy data collection of miscellaneous and electronic loads in a commercial office building, Building and Environment 65 (2013) 170–177. `doi:10.1016/j.buildenv.2013.03.025`.
URL `http://dx.doi.org/10.1016/j.buildenv.2013.03.025`

[4] B. Kalluri, A. Kamilaris, S. Kondepudi, K. H. Wei, Classifying Office Plug Load Appliance Events in the context

of NILM using Time-series Data Mining, Proceedings of the 12th REHVA World Congress and CLImA, Aalborg, DenmarN 6 (2016) 1–11.

[5] A. Kamilaris, B. Kalluri, S. Kondepudi, T. Kwok Wai, A literature survey on measuring energy usage for miscellaneous electric loads in offices and commercial buildings, Renewable and Sustainable Energy Reviews 34 (2014) 536–550. doi:10.1016/j.rser.2014.03.037.

[6] A. C. Menezes, A. Cripps, R. A. Buswell, J. Wright, D. Bouchlaghem, Estimating the energy consumption and power demand of small power equipment in office buildings, Energy and Buildings 75 (2014) 199–209. doi:10.1016/j.enbuild.2014.02.011.
URL http://dx.doi.org/10.1016/j.enbuild.2014.02.011

[7] H. B. Gunay, W. O'Brien, I. Beausoleil-Morrison, S. Gilani, Modeling plug-in equipment load patterns in private office spaces, Energy and Buildings 121 (2016) 234–249. doi:10.1016/j.enbuild.2016.03.001.
URL http://dx.doi.org/10.1016/j.enbuild.2016.03.001

[8] K. Chia, A. Lebar, V. Agarwal, M. Lee, J. Ikedo, J. Wolf, K. Trenbath, J. Kleissl, Integration of a Smart Outlet-Based Plug Load Management System with a Building Automation System, 2023 IEEE PES Grid Edge Technologies Conference and Exposition, Grid Edge 2023 (2023). doi:10.1109/GridEdge54130.2023.10102749.

[9] R. Langner, K. Trenbath, Integrating Smart Plug and Process Load Controls into Energy Management Information System Platforms: A Landscaping Study, Tech. rep., National Renewable Energy Laboratory NREL (2019).
URL www.nrel.gov/publications.

[10] M. S. Ahmed, A. Mohamed, H. Shareef, R. Z. Homod, J. A. Ali, Artificial neural network based controller for home energy management considering demand response events, 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering, ICAEES 2016 (2017) 506–509doi:10.1109/ICAEES.2016.7888097.

[11] A. Leonardi, H. Ziekow, D. Konchalenkov, A. Rogozina, Detecting Smart Plug Configuration Changes in Smart Homes, Smart SysTech 2014; European Conference on Smart Objects, Systems and Technologies (2014) 1–7doi:10.1109/SmartSysTech.2014.7156016.

[12] H. Shareef, M. S. Ahmed, A. Mohamed, E. Al Hassan, Review on Home Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers, IEEE Access 6 (2018) 24498–24509. doi:10.1109/ACCESS.2018.2831917.

[13] S. Sharda, K. Sharma, M. Singh, A real-time automated scheduling algorithm with PV integration for smart home prosumers, Journal of Building Engineering 44 (June) (2021) 102828. doi:10.1016/j.jobe.2021.102828.
URL https://doi.org/10.1016/j.jobe.2021.102828

[14] A. N. Abdullah, A. H. Mutlag, M. S. Ahmed, Neural Network Based Home Energy Management for Modelling and Controlling Home Appliances under Demand Response, Journal of Physics: Conference Series 1963 (1) (2021). doi:10.1088/1742-6596/1963/1/012097.

[15] R. H. Hussain, M. I. Shujaa, IoT residential appliance monitoring and controlling system using MQTT protocol, the Fourth Scientific Conference for Electrical Engineering Techniques Research (Eetr2022) 2804 (February) (2023) 020027. doi:10.1063/5.0154764.

[16] A. Santos, G. P. Duggan, J. Davis, D. Zimmerle, A Cautionary Note on Using Smart Plugs for Research Data Acquisition, e-Prime - Advances in Electrical Engineering, Electronics and Energy 4 (December 2022) (2023) 100137. doi:10.1016/j.prime.2023.100137.
URL https://doi.org/10.1016/j.prime.2023.100137

[17] G. Ghatikar, I. Cheung, S. Lanzisera, Miscellaneous and electronic loads energy efficiency opportunities for commercial buildings : a collaborative study by the United States and India, Tech. rep., Lawrence Berkeley National Laboratory (2013).
URL https://escholarship.org/uc/item/80b1c401

[18] J. Han, H. Lee, K. R. Park, Remote-Controllable and Energy-Saving Room Architecture based on ZigBee Communication, IEEE Transactions on Consumer Electronics 55 (1) (2009) 264–268. doi:10.1109/TCE.2009.4814444.

[19] M. S. Ahmed, A. Mohamed, R. Z. Homod, H. Shareef, A. H. Sabry, K. Bin Khalid, Smart plug prototype for monitoring electrical appliances in Home Energy Management System, 2015 IEEE Student Conference on Research and Development, SCOReD 2015 (2015) 32–36doi:10.1109/SCORED.2015.7449348.

[20] H. Morsali, S. M. Shekarabi, K. Ardekani, H. Khayami, A. Fereidunian, M. Ghassemian, H. Lesani, Smart plugs for building energy management systems, 2012 2nd Iranian Conference on Smart Grids, ICSG 2012 (2012) 1–5.

[21] A. F. Moreno Jaramillo, D. M. Laverty, D. J. Morrow, J. Martinez del Rincon, A. M. Foley, Load modelling and non-intrusive load monitoring to integrate distributed energy resources in low and medium voltage networks, Renewable Energy 179 (2021) 445–466. doi:10.1016/j.renene.2021.07.056.
URL https://doi.org/10.1016/j.renene.2021.07.056

[22] H. Çimen, N. Bazmohammadi, A. Lashab, Y. Terriche, J. C. Vasquez, J. M. Guerrero, An online energy management system for AC/DC residential microgrids supported by non-intrusive load monitoring, Applied Energy 307 (July 2021) (2022) 118136. doi:10.1016/j.apenergy.2021.118136.
URL https://doi.org/10.1016/j.apenergy.2021.118136

[23] B. Kalluri, S. Kondepudi, K. H. Wei, T. K. Wai, A. Kami-

laris, OPLD: Towards improved non-intrusive office plug load disaggregation, 2015 IEEE International Conference on Building Energy Efficiency and Sustainable Technologies, ICBEST 2015 5 (2015) 56–61. doi:10.1109/ICBEST.2015.7435865.

[24] A. Das, M. K. Annaqeeb, E. Azar, V. Novakovic, M. B. Kjærgaard, Occupant-centric miscellaneous electric loads prediction in buildings using state-of-the-art deep learning methods, Applied Energy 269 (February) (2020) 115135. doi:10.1016/j.apenergy.2020.115135. URL https://doi.org/10.1016/j.apenergy.2020.115135

[25] T. Kriechbaumer, H. A. Jacobsen, BLOND, a building-level office environment dataset of typical electrical appliances, Scientific Data 5 (2018) 1–14. doi:10.1038/sdata.2018.48.

[26] Z. D. Tekler, R. Low, Y. Zhou, C. Yuen, L. Blessing, C. Spanos, Near-real-time plug load identification using low-frequency power data in office spaces: Experiments and applications, Applied Energy 275 (May) (2020). doi:10.1016/j.apenergy.2020.115391.

[27] S. Heo, S. Song, B. Kim, H. Kim, Sharing-aware Data Acquisition Scheduling for Multiple Rules in the IoT, Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2020-April (2020) 43–55. doi:10.1109/RTAS48715.2020.00-18.

[28] K. Kawamoto, J. G. Koomey, B. Nordman, R. E. Brown, M. A. Piette, M. Ting, A. K. Meier, Electricity used by office equipment and network equipment in the US, Energy 27 (3) (2002) 255–269. doi:10.1016/S0360-5442(01)00084-6.

[29] M. Hafer, Quantity and electricity consumption of plug load equipment on a university campus, Energy Efficiency 10 (4) (2017) 1013–1039. doi:10.1007/s12053-016-9503-2.

[30] L. P. R. Ambati, D. Irwin, AutoPlug: An automated metadata service for smart outlets, 2016 7th International Green and Sustainable Computing Conference, IGSC 2016 (2017) 1–8doi:10.1109/IGCC.2016.7892604.

[31] S. Wang, X. Deng, H. Chen, Q. Shi, D. Xu, A bottom-up short-term residential load forecasting approach based on appliance characteristic analysis and multi-task learning, Electric Power Systems Research 196 (April) (2021). doi:10.1016/j.epsr.2021.107233.

[32] L. C. Siebert, L. R. Ferreira, E. K. Yamakawa, E. S. Custódio, A. R. Aoki, T. S. Fernandes, K. H. Cardoso, Centralized and decentralized approaches to demand response using smart plugs, Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference (2014) 1–5doi:10.1109/tdc.2014.6863547.

[33] C. a. Webber, J. a. Roberson, R. E. Brown, C. T. Payne, B. Nordman, J. G. Koomey, Field Surveys of Office Equipment Operating Patterns, Tech. rep., Lawrence Berkeley National Laboratory (2001). URL https://escholarship.org/uc/item/3r7480zm

[34] A. G. Ruzzelli, C. Nicolas, A. Schoofs, G. M. O'Hare, Real-time recognition and profiling of appliances through a single electricity sensor, SECON 2010 - 2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (2010) 1–9doi:10.1109/SECON.2010.5508244.

[35] G. Y. Lin, S. C. Lee, J. Y. J. Hsu, W. R. Jih, Applying power meters for appliance recognition on the electric panel, Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, ICIEA 2010 (2010) 2254–2259doi:10.1109/ICIEA.2010.5515385.

[36] O. T. Masoso, L. J. Grobler, The dark side of occupants' behaviour on building energy use, Energy and Buildings 42 (2) (2010) 173–177. doi:10.1016/j.enbuild.2009.08.009.

[37] M. S. Ahmed, A. Mohamed, R. Z. Homod, H. Shareef, Hybrid LSA-ANN based home energy management scheduling controller for residential demand response strategy, Energies 9 (9) (2016). doi:10.3390/en9090716.

[38] J. E. Boylan, A. A. Syntetos, Intermittent Demand Forecasting, John Wiley & Sons, Ltd, 2021. doi:https://doi.org/10.1002/9781119135289.

[39] M. Hamoudia, S. Makridakis, E. Spiliotis (Eds.), Forecasting With Artificial Neural Theory and Applications, 1st Edition, Springer Nature Switzerland, 2023.

[40] Z. Wang, T. Hong, M. A. Piette, Predicting plug loads with occupant count data through a deep learning approach, Energy 181 (2019) 29–42. doi:10.1016/j.energy.2019.05.138. URL https://doi.org/10.1016/j.energy.2019.05.138

[41] A. Srinivasan, K. Baskaran, G. Yann, IoT Based Smart Plug-Load Energy Conservation and Management System, 2019 IEEE 2nd International Conference on Power and Energy Applications, ICPEA 2019 (2019) 155–158doi:10.1109/ICPEA.2019.8818534.

[42] L. G. Maltais, L. Gosselin, Forecasting of short-term lighting and plug load electricity consumption in single residential units: Development and assessment of data-driven models for different horizons, Applied Energy 307 (June 2021) (2022) 118229. doi:10.1016/j.apenergy.2021.118229. URL https://doi.org/10.1016/j.apenergy.2021.118229

[43] M. Choi, W. K. Park, I. Lee, Smart Office Energy-Saving Service Using Bluetooth Low Energy Beacons and Smart Plugs, Proceedings - 2015 IEEE International Conference on Data Science and Data Intensive Systems; 8th IEEE International Conference Cyber, Physical and Social Computing; 11th IEEE International Conference on Green Computing and Communications and 8th IEEE Inte (2015) 247–251doi:10.1109/DSDIS.2015.69.

[44] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, Proceedings of the

fifth Berkeley symposium on mathematical statistics and probability 1 (14) (1967) 281–297.

[45] F. Pedregosa, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[46] G. J. McLachlan, K. E. Basford, Mixture models: Inference and applications to clustering, JSTOR, 1988.

[47] K. P. Sinaga, M. S. Yang, Unsupervised K-means clustering algorithm, IEEE Access 8 (2020) 80716–80727. doi:10.1109/ACCESS.2020.2988796.

[48] M. Bramer, Ensemble Classification, in: Principles of Data Mining, Springer London, London, 2013, pp. 209–220. doi:10.1007/978-1-4471-4884-5{_}14.
URL https://doi.org/10.1007/978-1-4471-4884-5_14

[49] D. Krakov, D. G. Feitelson, Comparing performance heatmaps, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8429 LNCS (2014) 42–61. doi:10.1007/978-3-662-43779-7{_}3.

[50] V. Di Gesù, V. Starovoitov, Distance-based functions for image comparison, Pattern Recognition Letters 20 (2) (1999) 207–214. doi:10.1016/S0167-8655(98)00115-9.

[51] A. A. Goshtasby, Similarity and Dissimilarity Measures, in: Image Registration: Principles, Tools and Methods, Springer London, London, 2012, pp. 7–66. doi:10.1007/978-1-4471-2458-0{_}2.
URL https://doi.org/10.1007/978-1-4471-2458-0_2

[52] G. Liu, F. A. Reda, K. J. Shih, T. C. Wang, A. Tao, B. Catanzaro, Image Inpainting for Irregular Holes Using Partial Convolutions, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11215 LNCS (2018) 89–105. doi:10.1007/978-3-030-01252-6{_}6.

[53] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss Functions for Image Restoration With Neural Networks, IEEE Transactions on Computational Imaging 3 (1) (2017) 47–57. doi:10.1109/TCI.2016.2644865.

[54] D. Yang, E. Wu, J. Kleissl, Operational solar forecasting for the real-time market, International Journal of Forecasting 35 (4) (2019) 1499–1519. doi:https://doi.org/10.1016/j.ijforecast.2019.03.009.
URL https://www.sciencedirect.com/science/article/pii/S0169207019300755

[55] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.
URL https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1996.tb02080.x

[56] B. Uniejewski, J. Nowotarski, R. Weron, Automated variable selection and shrinkage for day-ahead electricity price forecasting, Energies 9 (8) (2016) 621. doi:10.3390/en9080621.
URL http://www.mdpi.com/1996-1073/9/8/621

[57] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13-17-Augu (2016) 785–794. doi:10.1145/2939672.2939785.
URL https://dl.acm.org/doi/10.1145/2939672.2939785

[58] L. Barolli, Short term load forcasting using XGBoost, Vol. 1, Springer International Publishing, 2019. doi:10.1007/978-3-030-15035-8.
URL http://dx.doi.org/10.1007/978-3-030-15035-8_108

[59] L. Botman, J. Lago, X. Fu, K. Chia, J. Wolf, J. Kleissl, B. De Moor, Python implementation: Smart Plug Pipeline.
URL https://gitlab.esat.kuleuven.be/Lola.Botman/smart-plug-pipeline

[60] E. Spiliotis, S. Makridakis, A. A. Semenoglou, V. Assimakopoulos, Comparison of statistical and machine learning methods for daily SKU demand forecasting, Operational Research 22 (3) (2022) 3037–3061. doi:10.1007/s12351-020-00605-2.
URL https://doi.org/10.1007/s12351-020-00605-2

[61] B. Acker, C. Duarte, K. V. D. Wymelenberg, Office Space Plug Load Profiles and Energy Saving Interventions, ACEEE Summer Study on Energy Efficiency in Buildings (2012) 1–13.

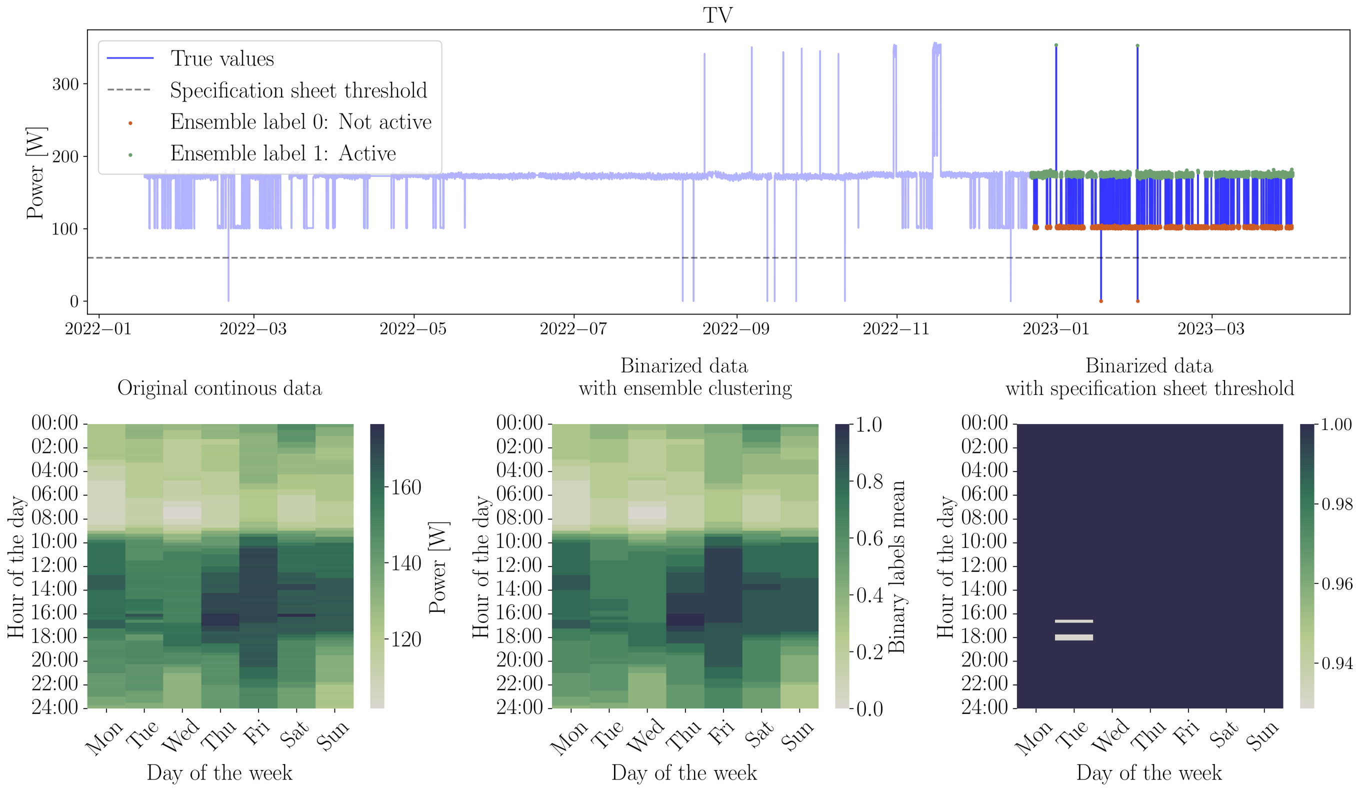
[62] S. Wang, A. A. Kim, E. M. Johnson, Understanding the deterministic and probabilistic business cases for occupant based plug load management strategies in commercial office buildings, Applied Energy 191 (2017) 398–413. doi:10.1016/j.apenergy.2017.01.082.
URL http://dx.doi.org/10.1016/j.apenergy.2017.01.082

[63] P. Gandhi, G. S. Brager, Commercial office plug load energy consumption trends and the role of occupant behavior, Energy and Buildings 125 (2016) 1–8. doi:10.1016/j.enbuild.2016.04.057.
URL http://dx.doi.org/10.1016/j.enbuild.2016.04.057

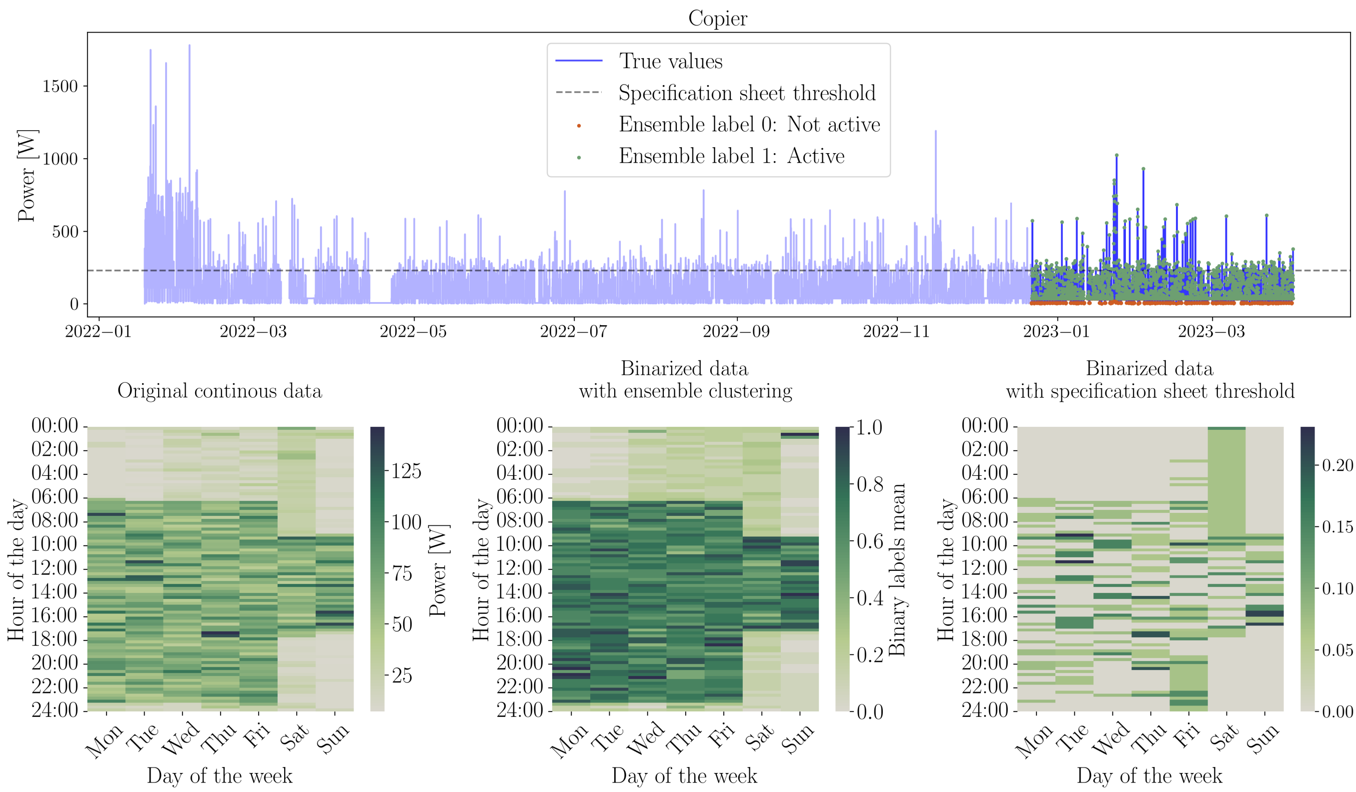
Appendix A. Code and data availability

The code used and implemented in this research study along with the pre-processed datasets are publicly available on GitLab [59].

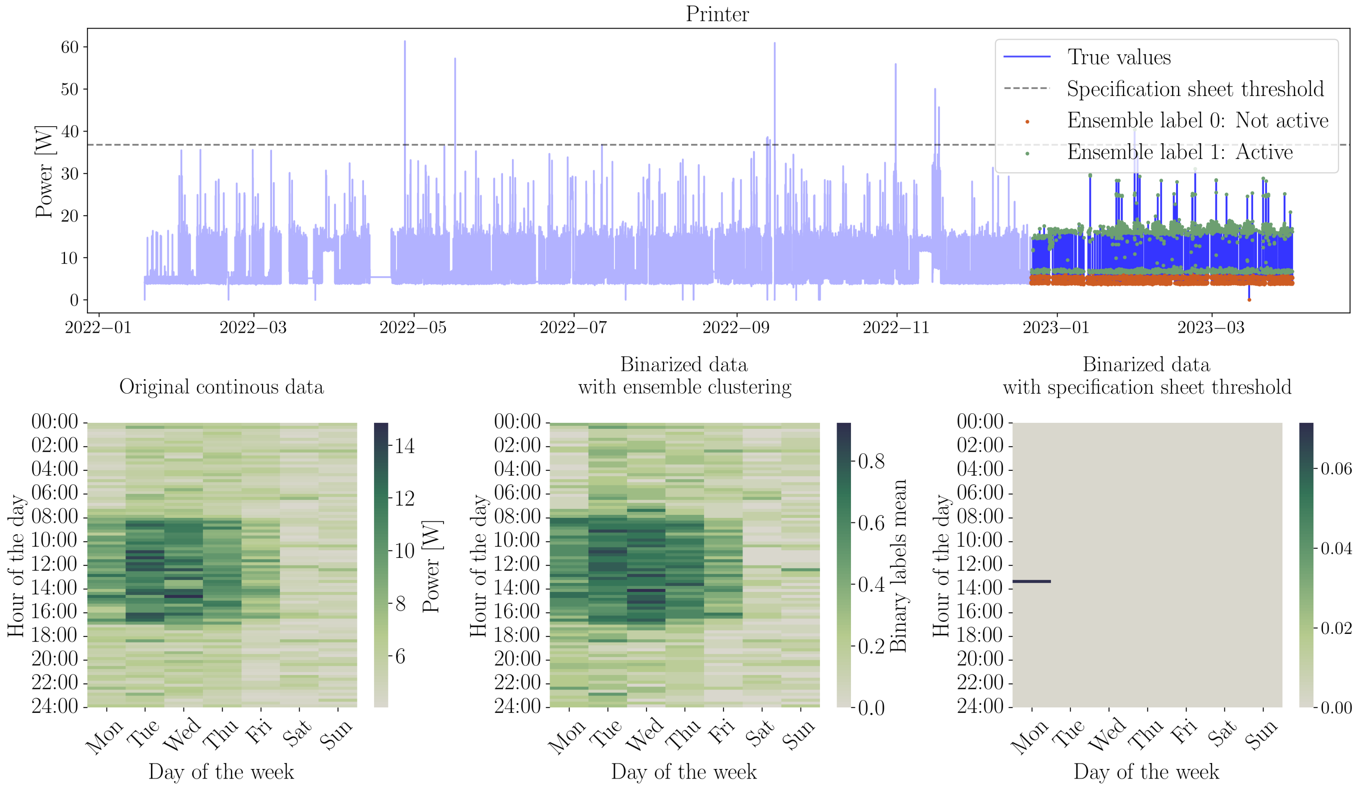
Appendix B. Illustration of the operating mode detection results



(a) High-fidelity mode detection of a TV.

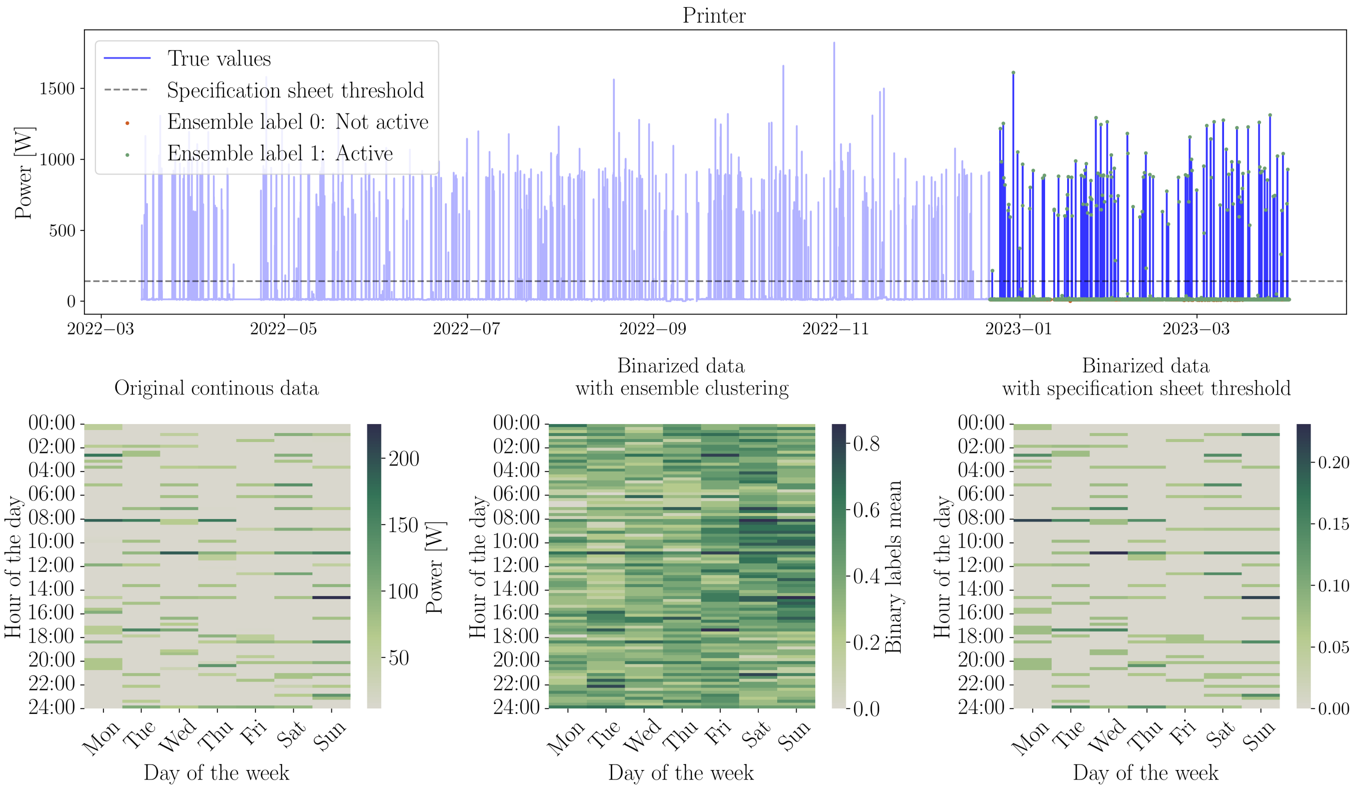


(b) High-fidelity mode detection of a Copier.

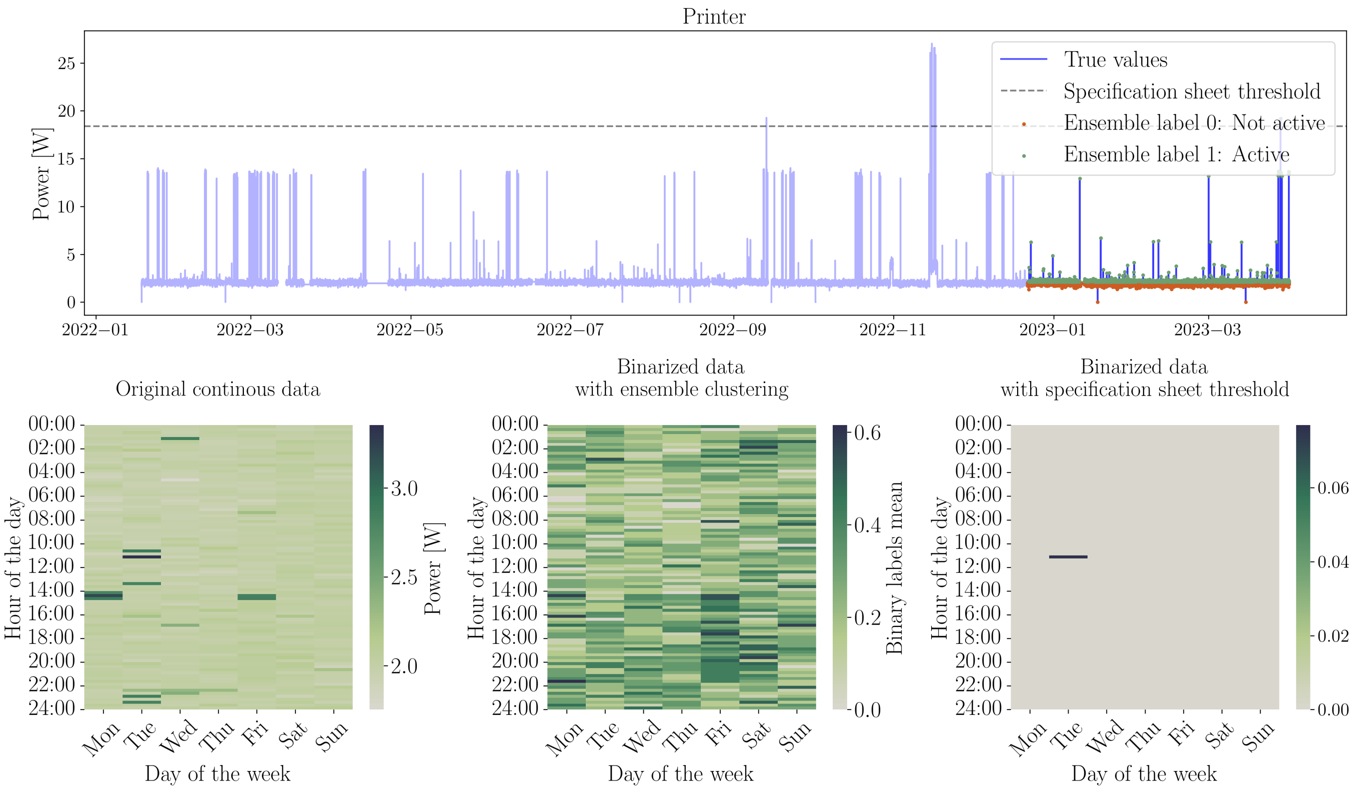


(c) High-fidelity mode detection of a printer.

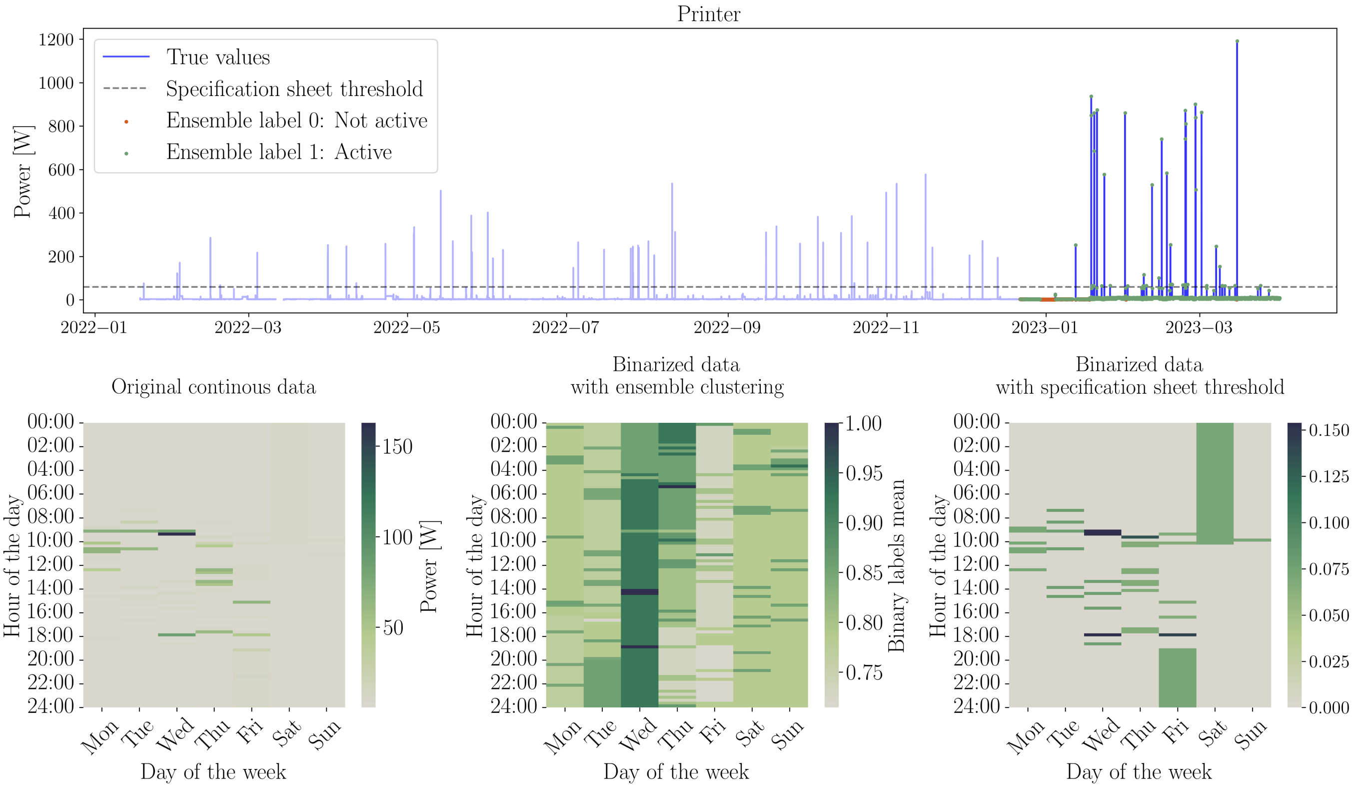
Figure B.3. (Continued) Each figure consists of four plots. First, the complete time series is plotted, with the training set in light blue and the test set in dark blue. In the test set, the labels outputted by our mode detection algorithms are incorporated. Active power values are marked with green dots, while inactive (unused) values are marked in orange dots. The first heatmap is build using the original test data. The second heatmap is generated with the binarized test data through our detection algorithm. The comparison of these two heatmaps is used to categorize our algorithm performance. The last heatmap is formed with the binarized test data, based on power thresholds specified in product specification sheets. The dotted line on the initial time series plot also depicts this data sheet threshold. (a), (b) and (c) are examples of mode detection results classified as high-fidelity. Each binarized dataset captures the underlying usage pattern of the true dataset accurately. Different device types and usage patterns are represented.



(a) Partial-fidelity mode detection of a printer.

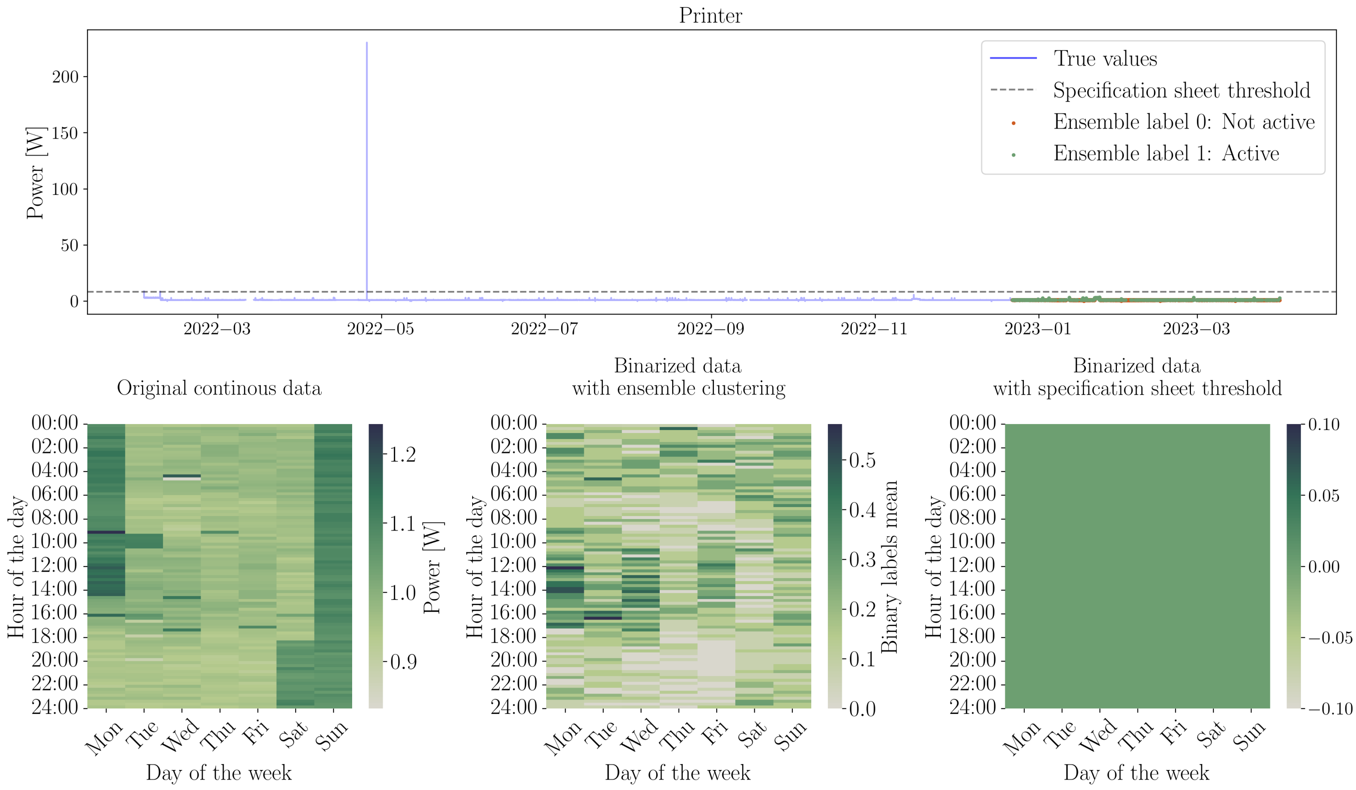


(b) Partial-fidelity mode detection of a printer.

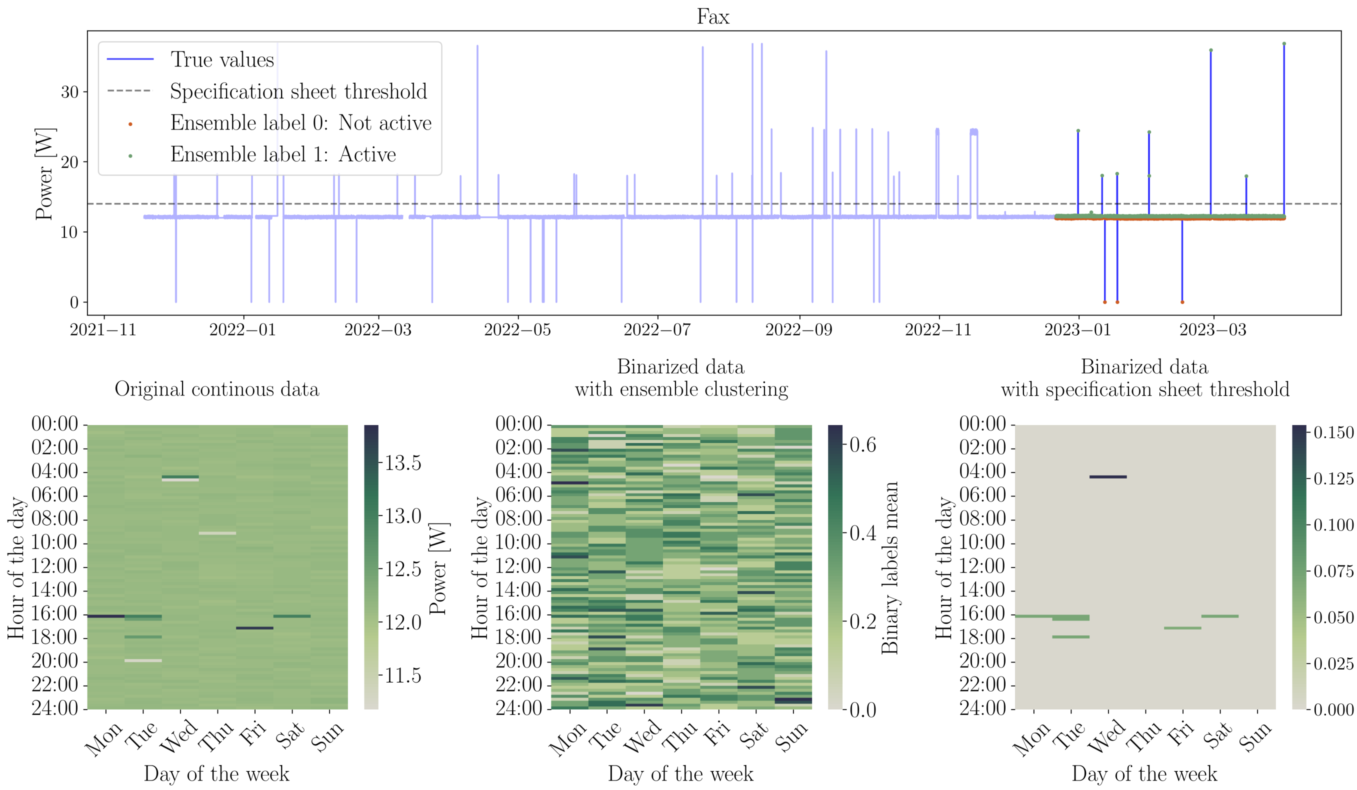


(c) Partial-fidelity mode detection of a printer.

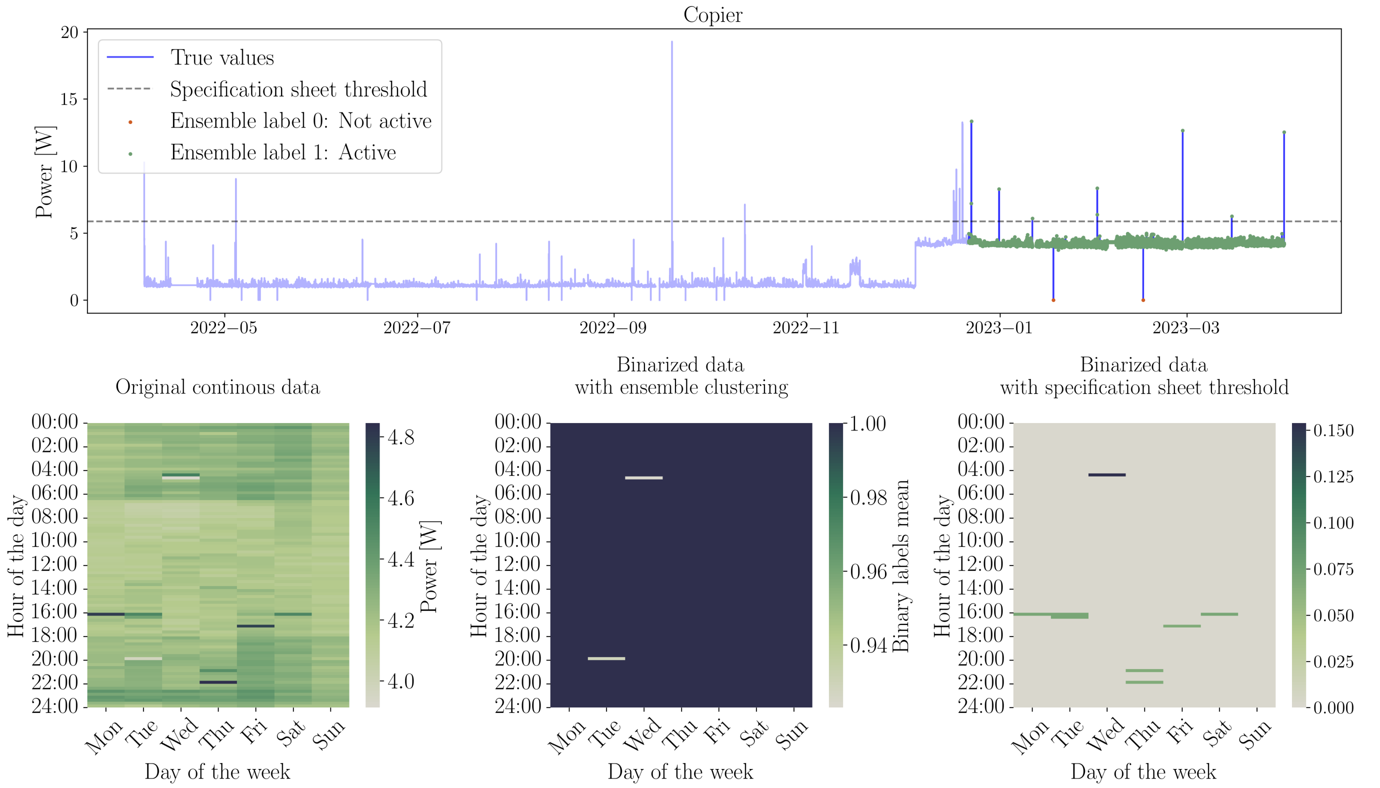
Figure B.4. (Continued) Each figure consists of four plots. First, the complete time series is plotted, with the training set in light blue and the test set in dark blue. On the test set, the labels outputted by our mode detection algorithms are incorporated. Active power values are marked with green dots, while inactive (unused) values in orange dots. The first heatmap is build using the original test data. The second heatmap is generated with the binarized test data through our detection algorithm. The comparison of these two heatmaps is used to categorize our algorithm performance. The last heatmap is formed with the binarized test data, based on power thresholds specified in product sheets. The dotted line on the initial time series plot also depicts this data sheet threshold. (a), (b) and (c) are examples of mode detection results classified as partial-fidelity. Each binarized dataset captures the underlying usage pattern of the true power values, nevertheless it also wrongly labels some unused values as active.



(a) Low-fidelity mode detection, which might be due to the small difference in power absolute values, or the fact that the appliance might never have been used.



(b) Low-fidelity mode detection which might be due to the three existing operating modes.



(c) Low-fidelity mode detection which might be due to the occurrence of concept drift.

Figure B.5. (Continued) Each figure consists of four plots. First, the complete time series is plotted, with the training set in light blue and the test set in dark blue. On the test set, the labels outputted by our mode detection algorithms are incorporated. Active power values are marked with green dots, while inactive (unused) values in orange dots. The first heatmap is build using the original test data. The second heatmap is generated with the binarized test data through our detection algorithm. The comparison of these two heatmaps is used to categorize our algorithm performance. The last heatmap is formed with the binarized test data, based on power thresholds specified in product sheets. The dotted line on the initial time series plot also depicts this data sheet threshold. (a), (b) and (c) are examples of mode detection results classified as low-fidelity. (a) The proposed algorithm has difficulty grasping the usage patterns since the difference between active and unused mode is very small in terms of absolute power values. (b) There seem to be three operating modes for this appliance, while the proposed algorithm is designed to detect two modes. (c) The large change in power dynamics between the training set and the test set, called concept drift, makes it challenging for the proposed algorithm to accurately discern the new usage pattern.