Clustering appliance operation modes with unsupervised deep learning techniques

Original
Clustering appliance operation modes with unsupervised deep learning techniques / Castangia, Marco; Barletta, Nicola; Camarda, Christian; Quer, Stefano; Macii, Enrico; Patti, Edoardo. - In: IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. - ISSN 1941-0050. - 19:7(2023), pp. 8196-8204. [10.1109/TII.2022.3217495]

Availability:
This version is available at: 11583/2972716 since: 2023-06-22T14:32:32Z

Publisher:
IEEE

Published
DOI:10.1109/TII.2022.3217495

(Article begins on next page)

Clustering appliance operation modes with unsupervised deep learning techniques

Marco Castangia, Nicola Barletta, Christian Camarda, Stefano Quer, Enrico Macii and Edoardo Patti

***Abstract*—In smart grids, consumers can be involved in demand response programs to reduce the total power consumption of their households during the peak hours of the day. Unfortunately, nowadays, utility companies are facing important challenges in the implementation of demand response programs because of their negative impact on the comfort of end-users. In this paper, we cluster the different operation modes of household appliances based on the analysis of their power signatures. For this purpose, we implement an autoencoder neural network to create a better data representation of the power signatures. Then, we cluster the different operational programs by using a K-means algorithm fitted to the new data representation. To test our methodology, we study the operation modes of some washing machines and dishwashers whose power signatures were derived from both submeters and non-intrusive load monitoring techniques. Our clustering analysis reveals the existence of multiple working programs showing well-defined features in terms of both average energy consumption and duration. Our results can then be used to improve demand response programs by reducing their impact on the comfort of end users. Furthermore, end users can rely on our framework to favor lighter operation modes and reduce their overall energy consumption.**

***Index Terms*—smart grids, appliance operation modes, appliance program, clustering, autoencoder, deep learning.**

I. INTRODUCTION

ELECTRICITY systems need to constantly keep a perfect balance between supply and demand to operate properly [1]. Unfortunately, this equilibrium is not always easy to achieve for several reasons. On the one hand, energy generators and utility operators have to face many unpredictable events that may compromise their energy supply. These situations have been exacerbated by the higher penetration of renewable energy sources [2]. On the other hand, consumers can show significant variations in their energy demand during the day, which makes more difficult to operate an optimal energy distribution from utility operators. Since grid interventions are highly capital intensive and energy shortages are almost unpredictable, it is more convenient to take action on the consumer side in order to reduce fluctuations in demand and increase the overall system efficiency.

Smart Grids allow bidirectional communications between consumers and providers. In this way, they can cooperate to improve the overall reliability of the network and lessen the risks of energy outages [3]. In this new paradigm, utilities started to adopt a set of measures aimed at reducing the overall energy demand during the peak hours of the day, following the so called Demand Response (DR) programs [4], [5]. In summary, DR indicates any change in the electricity usage of end-users in response to variations in the energy prices over time. Changes may include *load shifting* from periods of high demand to periods of lower demand or *load reduction* during peak hours. In either cases, the end-users have to relinquish part of their comfort for the sake of energy efficiency, and this may represent a serious obstacle when consumers are not completely convinced to change their habits.

Thanks to the development of advanced metering infrastructures, end-users are now able to monitor the total power consumption of their houses with various degrees of resolution [6], [7]. Previous studies demonstrate that end-users are spontaneously encouraged to reduce their energy requirements once provided with detailed information on their power usage [8]. In particular, several investigations report that complete information can save up to 12% of the total residential energy [9], [10]. In addition, power usage profiles can be used to reschedule some activity with the goal of better exploiting the production of on-site PV energy [11].

In this work, we cluster the different operational modes of household appliances based on the analysis of their power signatures. To achieve this goal, we leverage the capabilities of autoencoders to automatically extract a better data representation of the collected power signatures, thus completely avoiding the cumbersome task of creating handcrafted features. Then, in order to cluster together operation cycles with similar power signatures, we fit a K-means algorithm to the latent representation extracted by the autoencoder. To test our methodology, we analyze the power profiles generated by some residential washing machines and dishwashers belonging to the public (including submetered power signatures) and a private dataset (storing disaggregated energy profiles). The clustering results are evaluated quantitatively by means of the silhouette score, and qualitatively by analyzing the consistency of relevant attributes such as the average energy consumption and the mean duration. The results obtained in this work can enable novel services for both end-users and utilities. On the one hand, utility companies may ask to the end-user to just select a lighter program instead of shifting it, thus causing only a minor impact on the user's comfort. On the other hand, end-users can easily reduce their energy bills by just favoring lighter operational modes over more energy-intensive programs, once properly informed about the energy consumption of their appliances.

M. Castangia, N. Barletta, S. Quer, E. Macii and E. Patti are with Politecnico di Torino, Turin, IT. E-mails: name.surname@polito.it.
C. Camarda is with Midori S.R.L., Torino, Italy. E-mail: christian@midorisrl.eu
This work has been partially funded by Arrowhead Tools, which is an EU H2020 - ECSEL Joint Undertaking research project, Grant Agreement no. 826452.

The remaining parts of this paper are organized as follows. Section II reports the most relevant works aimed at clustering the appliance's programs; Section III provides a thorough description of the datasets; Section IV describes in detail the different processing stages of our clustering methodology; Section V discusses the results of our clustering analysis for the washing machines and dishwashers separately; finally, Section VI reports the most relevant findings of this work, proposing possible future research directions.

II. Related works

To the best of our knowledge, the number of publications that focus on clustering the operational modes of household devices is very limited. Wang et al. [12] used a K-means clustering algorithm to classify the different operational modes of a fridge based on its maximum power usage and total energy demand. The clustering algorithm identified three different modes of operation for that device: The normal mode, which presents the largest number of instances and the lower power consumption, and the defrost and post-defrost modes, which occur less frequently and present a higher energy demand. Overall, the authors demonstrated that it is possible to cluster the different operational cycles of the fridge based on their peak power and energy consumption. Jaradat et al. [13] attempted to classify the different operational modes of three commonly used deferrable appliances, i.e., a washing machine, a clothes dryer, and a dishwasher. The authors implemented a supervised model based on the Dynamic Time Warping distance. Their approach classifies new operational modes by comparing their power profile with a predefined set of operational modes previously collected from the monitored device. In particular, the authors assumed three modes of operations for each piece of equipment, which they broadly categorized into light, medium and heavy, based on their total energy demand. In a later work, the same authors [14] conducted a new study to improve the previous results. This time, the authors used a K-nearest neighbors algorithm to classify the different apparatus operational modes. This new strategy slightly overcame the previous approach by exploiting a set of handcrafted features describing the power states of each appliance's operational mode. Marcu et al. [15] attempted to categorize the various programs of two washing machines based on the differences in their power signatures. According to their results, it is particularly difficult to cluster different washing machine programs as these devices, in order to maximize the program's efficiency in terms of water and energy, adapt their operational stages to the specific run time conditions.

Previous works present clustering methodologies either based on raw data or manual features. Jaradat et al. [13] compare the raw power signatures to a set of reference power loads representing the various operational modes of the monitored appliances. However, methods based on raw data are susceptible to noisy points and outliers, thus they often lead to poor clustering results. In addition, these approaches can be highly inefficient in the presence of long time series. The authors of [12], [14], [15] adopt a feature-based approach relying on the design of handcrafted features to recognize the different operational modes. On the one hand, the extraction of manual features is certainly faster and more robust to noise than the methodologies based on raw data. On the other one, their application is confined to the specific power signatures of the monitored device. Indeed, manually extracted features are hardly reusable with devices from other brands, which may present very heterogeneous power profiles. Furthermore, the design of effective features often requires advanced domain knowledge and can be a very time-consuming task.

In the past decade, researchers discovered that deep learning can efficiently solve various clustering tasks [16]. Among the various possible applications, researchers use deep autoencoders to learn effective data representations of time series and use them to help classical clustering algorithms in their categorization task [17], [18]. Richard et al. [19] investigate the combination of a convolutional autoencoder and a K-medoids algorithm to cluster time series data. Interestingly, the authors verify their clustering methodology on a dataset consisting of daily power consumption. The results show that convolutional autoencoders outperform both raw-data-based methods and feature-based methods. In particular, the embeddings extracted by the autoencoder achieved better results than the clustering methods applied directly to raw data, including the techniques adopting shift and scale-invariant distance metrics (such as dynamic time warping, i.e., DTW). Furthermore, deep autoencoders produce better data representations than linear techniques (such as principal component analysis, i.e., PCA), thanks to the high non-linearity of the encoder's layers.

In the last five years, researchers started to heavily apply deep learning techniques to solve the problem of Non-Intrusive Load Monitoring (NILM) [20]. NILM aims at estimating the energy consumption of the various household appliances through the analysis of the total power consumption of the house [21]. Kelly et al. [22] pioneered the application of deep neural networks to NILM, demonstrating that the deep learning approach easily outperforms all previous methods in terms of disaggregation accuracy. Zhang et al. [23] slightly improved the results of the previous work by employing a better neural network architecture. Most importantly, the authors proved that deep neural networks can distinguish the different power signatures of the device. D'Incecco et al. [24] demonstrated that the features learned for certain appliances can be transferred to estimate the energy consumption of other devices, providing further insights on the electrical features learned by deep neural networks.

Motivated by the exceptional capabilities of deep learning in the extraction of electrical features for energy disaggregation and inspired by the recent success demonstrated by autoencoders in the context of time series clustering, we decided to apply them to efficiently cluster the different operational modes of household instruments. To overcome the limitations of previous works, we decided to adopt a solution based on unsupervised learning techniques. Indeed, the unsupervised approach does not require any class label to work, which guarantees its applicability in a real-world scenario. Furthermore, we refused to assume a fixed number of appliance programs, thus adapting our solution to the

TABLE I: UK-DALE dataset.

		Num. of cycles	Monitoring period
UK-house-1	Washing machine	1433	2012/11/09 - 2017/04/26
	Dishwasher	679	2012/11/09 - 2017/04/26
UK-house-2	Washing machine	59	2013/05/20 - 2013/10/10
	Dishwasher	106	2013/05/20 - 2013/10/10

specific preferences of end-users. The deep learning approach can be easily generalized to multiple devices and brands, thus preventing the cumbersome task of creating handcrafted features for each device. Most importantly, autoencoders can efficiently handle complex patterns in the power consumption of the appliance, where simple statistics such as the mean energy absorbed may fail to distinguish between different operational modes.

III. Dataset

In this section, we introduce the two datasets used to evaluate our methodology. The first one is public and it includes submetered power measurements adopted to verify the performance of our clustering algorithm on a set of clean power signatures. The second dataset has been generated by adopting NILM techniques to the aggregated power consumption of five different residential houses and it is used to test the performance of our algorithm in the presence of noisy power signatures.

A. UK-DALE

The UK Domestic Appliance-Level Electricity (UK-DALE) dataset contains the power consumption of five houses in the UK both at the aggregate-level and at the appliance-level collected with a sampling frequency of six seconds [25]. In this work, we focused on the submetered power consumption of two appliances, i.e., the washing machine and the dishwasher, mainly because they usually comprehend a multitude of operational modes and are present in almost every household. To demonstrate the effectiveness of our methodology, we studied the power consumption of House 1 and House 2 of the UK-DALE dataset, since they both include two dedicated submetered channels for the washing machine and the dishwasher. Table I reports the number of operation cycles and the monitoring periods of the appliances under study. The length of the monitoring period between the two households explains the large difference in the total number of cycles. Indeed, House 1 provides more than four years of submetered data from 9/11/2012 to 26/04/2017, whereas House 2 covers almost five months of measurements from 20/05/2013 to 10/10/2013. Notice that the power signatures of the UK-DALE dataset are collected directly at the appliance level adopting smart plugs. Working on submetered data can be useful to verify the performance of our algorithm on clean power signatures, before moving to other kinds of data sources such as NILM algorithms.

B. Non Intrusive Load Monitoring

"Omitted for Double Blind Review" is a startup applying Non-Intrusive Load Monitoring techniques to provide detailed reports on the power consumption of their users. The firm kindly provided us a large number of power signatures derived from the disaggregation of some washing machines and dishwashers operating in their monitored lodgings. Hereinafter, we refer to this information as the NILM dataset and we use it to prove that our methodology can work with disaggregated data. As reported in Table II, this dataset embraces data collected in five different residences, covers a period of 16 months (from 01/09/2020 to 31/12/2021), and includes a varying number of operational cycles. NILM data are essentil to demonstrate the potentiality of our algorithm in a real-world scenario. Disaggregation algorithms are usually preferred to submeters for large commercial applications, given their low costs and high scalability. Moreover, recent NILM algorithms demonstrated high fidelity in the disaggregation of dishwashers and washing machines, motivating their use as an alternative to submeters [26], [27].

TABLE II: NILM dataset.

		Num. of cycles	Monitoring period
NILM-house-1	Washing machine	267	2020/09/01 - 2021/12/31
	Dishwasher	270	
NILM-house-2	Washing machine	170	2020/09/01 - 2021/12/31
	Dishwasher	196	
NILM-house-3	Washing machine	162	2020/09/01 - 2021/12/31
	Dishwasher	479	
NILM-house-4	Washing machine	463	2020/09/01 - 2021/12/31
	Dishwasher	302	
NILM-house-5	Washing machine	479	2020/09/01 - 2021/12/31
	Dishwasher	372	

IV. Methodology

In this section, we describe the processing stages that constitute our clustering methodology. The main steps are reported in the pipeline of Figure 1. The input consists of a set of power signatures that can be collected either from submeters or NILM algorithms. The submeters continuously monitor the power consumption of the device both during its active and idle states. Therefore, a first segmentation procedure is applied to submeters to recognize the active states containing the actual power signatures of the monitored device. Conversely, NILM algorithms already provide the disaggregated power signatures of the appliance with their start and stop times, which are inferred during the disaggregation process. Both the power signatures extracted from submeters and those generated by NILM are passed to the pre-processing step in the same way. In the pre-processing stage, all instances are normalized to be fed into a deep autoencoder. This autoencoder is trained to reconstruct the operation cycles extracted in the previous step and to learn the best latent representation of the input data. Finally, the latent representation learned by the neural network is used by the K-means clustering algorithm, which is in charge of recognizing the different programs of the device. The main phases of our methodology are described in the following sub-sections.

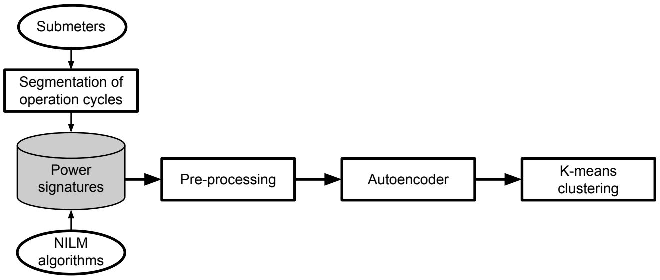

Fig. 1: Pipeline structure of our proposed clustering methodology.

A. Segmentation of operation cycles

The purpose of the segmentation procedure is to extract the individual operations from the continuous active power measurements received from the submeter. With reference to Figure 1, note that this phase is exclusively applied to the submetered data, since the operation cycles coming from NILM algorithms have been already extracted during the disaggregation process. The input of the procedure is a window of active power measurements of arbitrary length received from the submeter, whereas the output is a list of operation cycles extracted from the corresponding input window. The segmentation algorithm used in this work is very similar to the one already proposed by Massidda et al. [28]. The authors used a rule-based procedure to extract the operations by using a limited set of parameters. In summary, an appliance is considered active when its power consumption exceeds a predefined power threshold, that we called *min power*. In order to cope with the temporary idle times of washing machine and dishwasher, we used an additional parameter called *min off* to decide what is the minimum number of samples under the power threshold such that the appliance operation can be considered truly completed. Finally, we used a parameter called *min on* to filter out false positives that present too short operations. Table III reports the value of these parameters used during the segmentation phase for the washing machine and the dishwasher. We use the same values for House 1 and House 2. Notice that this set of optimal values is found in an empirical way, considering the expected duration of a normal operation and trying to separate individual cycles as well as possible.

TABLE III: Selected values for the main parameters of the segmentation procedure.

	min power [W]	*min on* [s]	*min off* [s]
Washing machine	25	300	180
Dishwasher	25	300	900

B. Pre-processing

All training instances have been zero padded up to 1600 samples, which correspond exactly to two hours and forty minutes of activity. Training instances longer than 1600 samples have been truncated down to 1600 samples to conform with shorter instances. Neural networks are known to converge faster when the input values are scaled within a small interval, whereas normalization generally improves the overall training process. Therefore, we decided to scale the input data by using the classical standardization formula reported in Equation 1, in which the mean μ_x and standard deviation σ_x are computed across all the power measurements x.

$$x_{scaled} = \frac{x - \mu_x}{\sigma_x} \tag{1}$$

C. Autoencoder

The autoencoder is a neural network architecture capable of learning *latent representations* of the inputs in a totally unsupervised way [29]. The general architecture of the autoencoder presents an encoder network followed by a decoder network. The encoder network is responsible for mapping the inputs into a lower dimensional space such that the amount of information preserved from the inputs is maximized. The decoder network is in charge of mapping back the latent representations to the original inputs. The autoencoder architecture is constrained by the amount of information that can be retained by the network during the training process. In this way, the autoencoder is forced to capture only the most significant features of the inputs in order to minimize the overall reconstruction error across the entire training set.

The 1D convolutional layers (Conv 1D) are composed by multiple filters which have the purpose of selecting the most salient features from the input sequence received from the previous layer. The greater the number of filters, the larger the amount of features that can be potentially learned. The filter's size defines the receptive field of a single filter. To capture space-invariant features, the filter slides across the entire input sequence with a certain step size (or stride). Each filter performs the following computation:

$$\hat{y} = f(\sum_{i=1}^{n} w_i x_i + b) \tag{2}$$

where n is the filter's size, x_i are the inputs, w_i are the filter's weights, b is a bias term, and f is the activation function.

Recurrent layers are specifically designed to capture temporal relations in the input thanks to the capabilities of long short-term memory (LSTM) cells [30]. At each timestamp the LSTM cell receives the current input x_t plus a digest of all previous inputs, which is summarized by the short-term state h_t and the long-term state c_t. The following set of equations describes the different operations performed by the LSTM cell:

$$\begin{aligned} i_t &= \sigma(W^{xi} x_t + W^{hi} h_{t-1} + b_i) \\ f_t &= \sigma(W^{xf} x_t + W^{hf} h_{t-1} + b_f) \\ o_t &= \sigma(W^{xo} x_t + W^{ho} h_{t-1} + b_o) \\ g_t &= tanh(W^{xg} x_t + W^{hg} h_{t-1} + b_g) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes g_t \\ y_t &= h_t = o_t \otimes tanh(c_t) \end{aligned} \tag{3}$$

where W^{xi}, W^{xf}, W^{xo}, and W^{xg} are the weighted connections of the input vector x_t; W^{hi}, W^{hf}, W^{ho}, and W^{hg} are the weights of the previous short-term state vector h_{t-1}; b_i, b_f, b_o, and b_g are the bias terms.

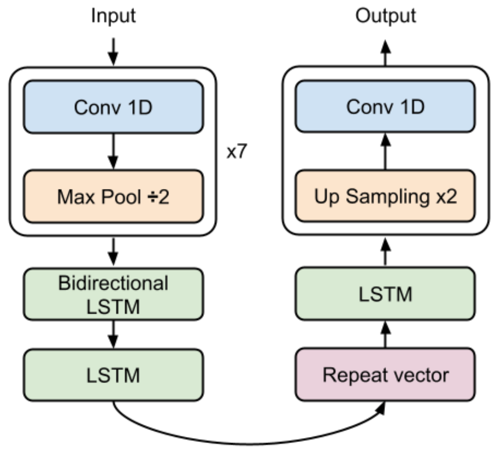

Fig. 2: Autoencoder architecture.

The autoencoder architecture used in this work is described in Figure 2. The encoder (on the left-hand side) is composed by a set of convolutional layers alternated with max pooling layers, which have the twofold function of reducing the input dimensions and selecting the most relevant features in the input [31]. The encoder terminates with two LSTM layers with the purpose of capturing temporal relations among input values. Notice that the first recurrent layer implements a bidirectional LSTM in order to take into consideration temporal relations in both directions [32]. The decoder (on the right-hand size) presents an architecture almost symmetrical to the encoder. It starts with a recurrent layer followed by a set of convolutions and up-sampling layers. All convolutional layers contain 32 filters of size 3 which use a rectified linear unit (Relu) as activation function. To maintain the same sequence length after the convolution, we used a stride equal to 1 and zero padded the input sequence on both sides. The LSTM cells work with 128 units and use a hyperbolic tangent (tanh) activation function. The max pooling layers in the encoder network progressively downsample the input sequences in order to constraint the amount of information retained in the latent representation. The up-sampling layers in the decoder network increment the the previous layer's dimension back to the original input shape.

The loss function of the network is the *mean square error* computed between the reconstructed signal and the original input. The network parameters are optimized by using the adaptive moment estimation (Adam) algorithm with a default learning rate of 0.001 and a batch size equal to 32. The duration of the training process is capped at 500 epochs, but it stops earlier if the reconstruction error does not show improvements for more than 25 epochs (early stopping criteria). Table IV reports the training hyper-parameters used for the autoencoder.

TABLE IV: Training hyper-parameters of the autoencoder.

Hyper-parameter	Value
Optimizer	Adam
Loss	mean squared error
Learning rate	0.001
Epochs	1500
Batch size	32
Stopping criteria	early stopping with patience equal to 50

D. K-means algorithm

K-means is a partitional clustering algorithm that, given a number of clusters K, assigns each instance to the cluster with the closest centroid, such that the mean squared distance between the instances and their closest centroid is minimized through an iterative process [33]. Centroids are defined as the mean of all instances belonging to a cluster. At the beginning, the algorithm select K random instances in the training set to be designated as initial centroids. Then, the algorithm iteratively assigns each instance to the cluster with the closest centroid. Finally, it updates the centroids of each cluster by computing the mean value of all instances belonging to it. The procedure continues until convergence, i.e., until centroids do not change anymore. The hyper-parameters selected for the K-means algorithm are reported in Table V. Notice that, as we are going to discuss in Section V, the number of clusters K has been omitted from Table V as this value depends on the specific dataset under study.

TABLE V: K-means hyper-parameters.

hyper-parameter	Value
K	-
Initialization	random
Num. of random initializations	10
Max iterations	300
Tolerance	0.0001

V. RESULTS

In this section, we discuss the results of the clustering analysis for the analyzed devices, i.e., washing machines and dishwashers. Firstly, we introduce the silhouette score, which is the metric adopted to evaluate the clustering results and to select the optimal number of clusters for each piece of equipment. Then, we report the most relevant features of the extracted operational modes, including the average energy consumption, the mean duration, and the number of cycles found during the monitoring period.

A. Silhouette score

To validate the results of our clustering, we cannot rely on typical classification metrics, as we do not have a detailed annotation of the actual appliance programs selected by end-users. As a consequence, the most reliable metric to validate the clustering results is the *silhouette score* [34]. This represents a unique measure describing how well a single event fits the occurrences of its cluster with respect to the instances of other clusters. The silhouette score $s(i)$ of a single sample is computed by the following formula:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \quad (4)$$

where $a(i)$ is the mean squared distance between the selected instance and all other occurrences within its cluster, and $b(i)$ is the smallest mean squared distance computed between the selected instance and the samples belonging to all other clusters. The silhouette score takes values between -1 and 1, where 1 indicates very good and -1 very bad clustering results. To validate the quality of our clustering phase across the entire dataset, we compute the average silhouette score obtained by all the instances in the dataset.

B. Selecting the optimal number of clusters

As described in the previous section, we used the silhouette score to find the optimal number of clusters K. More in detail, we run the K-means algorithm with different values of K and we keep only the solution with the highest silhouette score. For our purposes, we used a number of clusters varying from 2 up to 10, considering a realistic scenario in which the end-user does not use more that 10 different operational modes for the same appliance. The silhouette scores obtained for these values of K are reported in Table VI. The highest scores for each device have been highlighted using boldface. In general, the optimal number of clusters varies between 2 and 4, which is far less than the maximum number considered in our procedure (i.e., 10 clusters). Moreover, the silhouette score assumes a minimum of two clusters to be computed, thus ignoring the case in which the end-user utilizes only a single program. We are perfectly conscious that the end-user may easily stick with a single operational mode for the entire monitoring period. However, we must consider that two operational modes with almost the same energy consumption and duration are practically identical for our applications. Therefore, we can assume that two clusters with very similar energy consumption and duration belong to the same program, without compromising the effectiveness of our clustering methodology in detecting operational modes with very distinct characteristics.

C. Analysis of washing machine cycles

The clustering analysis performed on the washing machine cycles shows the presence of power signatures recurring several times during the monitoring period. These signatures can be associated with the most common operational modes. Table VII reports the average energy consumption, the mean duration time, and the number of cycles for each washing machine programs traced in the UK-DALE and the NILM dataset. Overall, we can notice that end-users prefer less energy-intensive programs that appear with the highest frequency in our analysis. We can also notice that the duration time is not indicative of the total energy consumption of a washing machine program. Indeed, the power used by the washing machine is generally dominated by the water heating stage occurring in the first phases of the working cycle. Figure 3 shows the typical power signature of the four washing machine programs used in the first house of the UK-DALE dataset. Figure 4 depicts the three signatures found in the fourth house of the NILM dataset. These signatures clearly show that longer heating stages correspond to a greater power absorption, whereas the spin cycles represent only a minor contribution to the total energy demand.

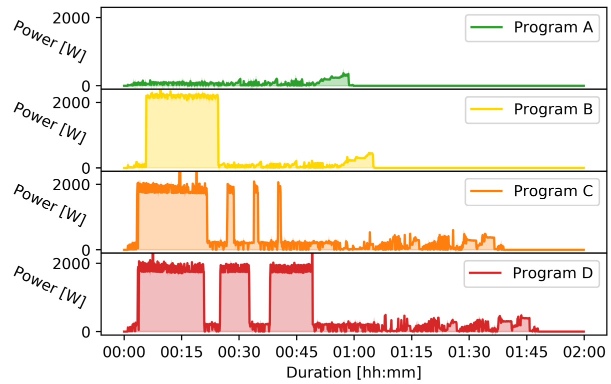

Fig. 3: Operational modes of the washing machine in the first house of the UK-DALE dataset.

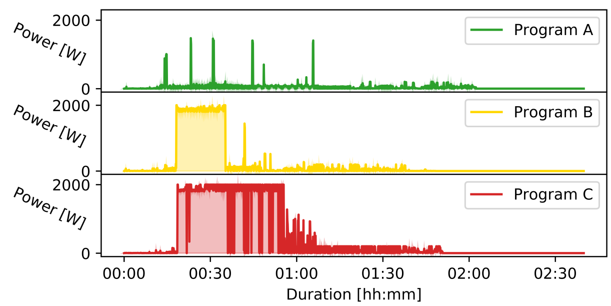

Fig. 4: Operational modes of the washing machine in the fourth house of the NILM dataset.

D. Analysis of dishwasher cycles

The clustering results of the dishwasher identify several well-defined power signatures that can be assigned to as many working modes for that device. Table VIII reports the average energy consumption, the mean duration, and the number of cycles identified for each dishwasher program in the UK-DALE and in the NILM dataset. The results show that end-users tend to favor lighter dishwasher programs in most cases, except for the houses in the UK-DALE dataset and the third houses in the NILM dataset. Similarly to the results for the washing machine, the duration of the operation cycle is not correlated with its total energy consumption. Figure 5 plots two power signatures of the dishwasher from the first house of UK-DALE. Figure 6 illustrates four different programs from the first house of the NILM dataset. Overall, the dishwasher presents very distinctive patterns characterized by the alternation of multiple heating stages with different durations and distances.

E. Discussion

Our methodology is particularly effective in clustering the different operational modes of both washing machines and

TABLE VI: Silhouette scores obtained with a different number of clusters (K) for the washing machine and the dishwasher in the houses of the UK-DALE's dataset and the "Omitted for Double Blind Review"'s dataset.

Number of clusters	UK-house-1		UK-house-2		NILM-house-1		NILM-house-2		NILM-house-3		NILM-house-4		NILM-house-5	
	WM	DW	WM	DW	WM	DW	WM	DW	WM	DW	WM	DW	WM	DW
K = 2	0.26	**0.73**	**0.51**	0.84	0.064	0.293	0.132	**0.304**	**0.143**	**0.229**	0.061	**0.286**	**0.173**	**0.244**
K = 3	0.31	0.52	0.13	**0.86**	0.057	0.325	**0.139**	0.152	0.119	0.198	**0.110**	0.183	0.159	0.148
K = 4	**0.32**	0.39	0.12	0.61	**0.068**	**0.351**	0.113	0.119	0.126	0.168	0.065	0.158	0.140	0.158
K = 5	0.27	0.38	0.11	0.61	0.058	0.275	0.116	0.128	0.141	0.147	0.063	0.155	0.134	0.133
K = 6	0.27	0.40	0.12	0.56	0.063	0.286	0.114	0.099	0.138	0.145	0.066	0.149	0.131	0.132
K = 7	0.26	0.42	0.12	0.54	0.060	0.288	0.112	0.114	0.134	0.142	0.060	0.145	0.105	0.122
K = 8	0.20	0.44	0.11	0.49	0.036	0.304	0.115	0.102	0.114	0.145	0.057	0.129	0.100	0.129
K = 9	0.20	0.45	0.12	0.50	0.038	0.302	0.106	0.118	0.116	0.112	0.059	0.131	0.100	0.124
K = 10	0.20	0.44	0.11	0.48	0.042	0.311	0.120	0.109	0.105	0.097	0.044	0.129	0.114	0.121

TABLE VII: Average energy consumption, average duration, and number of cycles for the washing machine use cases.

		Energy consumption [kWh]	Duration [hh:mm]	Number of cycles
UK-house-1	Program A	0.08 ± 0.03	0:49 ± 0:16	133 (9%)
	Program B	0.67 ± 0.22	1:06 ± 0:12	708 (50%)
	Program C	0.75 ± 0.17	1:34 ± 0:15	474 (33%)
	Program D	1.40 ± 0.30	1:56 ± 0:18	118 (8%)
UK-house-2	Program A	0.23 ± 0.01	0:17 ± 0:02	2 (3%)
	Program B	0.41 ± 0.09	0:40 ± 0:05	57 (97%)
NILM-house-1	Program A	0.87 ± 0.49	1:52 ± 0:39	66 (24.72%)
	Program B	0.95 ± 0.09	2:08 ± 0:09	79 (29.59%)
	Program C	1.21 ± 0.13	2:37 ± 0:18	72 (26.97%)
	Program D	1.29 ± 0.12	2:17 ± 0:09	50 (18.73%)
NILM-house-2	Program A	0.34 ± 0.22	2:07 ± 0:29	60 (35.29%)
	Program B	0.66 ± 0.11	1:38 ± 0:20	80 (47.06%)
	Program C	1.10 ± 0.20	1:50 ± 0:24	30 (17.65%)
NILM-house-3	Program A	0.34 ± 0.18	2:04 ± 0:35	94 (58.02%)
	Program B	0.57 ± 0.21	1:47 ± 0:17	68 (41.98%)
NILM-house-4	Program A	0.69 ± 0.20	1:55 ± 0:25	225 (48.60%)
	Program B	0.86 ± 0.18	2:00 ± 0:16	71 (15.33%)
	Program C	1.32 ± 0.23	1:58 ± 0:18	167 (36.07%)
NILM-house-5	Program A	0.24 ± 0.07	1:12 ± 0:23	142 (58.68%)
	Program B	0.55 ± 0.14	1:09 ± 0:16	100 (41.32%)

TABLE VIII: Average energy consumption, average duration, and number of cycles for the dishwasher use cases.

		Energy consumption [kWh]	Duration [hh:mm]	Number of cycles
UK-house-1	Program A	1.16 ± 0.10	1:39 ± 0:03	257 (35%)
	Program B	1.61 ± 0.10	1:25 ± 0:03	422 (65%)
UK-house-2	Program A	0.97 ± 0.04	1:17 ± 0:01	7 (7%)
	Program B	1.11 ± 0.07	0:47 ± 0:02	90 (85%)
	Program C	1.33 ± 0.09	1:13 ± 0:02	9 (8%)
NILM-house-1	Program A	0.93 ± 0.13	2:06 ± 0:04	129 (47.78%)
	Program B	1.08 ± 0.35	1:22 ± 0:24	59 (21.85%)
	Program C	1.16 ± 0.15	2:01 ± 0:07	48 (17.78%)
	Program D	1.48 ± 0.14	2:11 ± 0:06	34 (12.59%)
NILM-house-2	Program A	0.68 ± 0.11	2:08 ± 0:06	181 (92.35%)
	Program B	1.03 ± 0.13	1:30 ± 0:08	15 (7.65%)
NILM-house-3	Program A	1.02 ± 0.19	1:29 ± 0:10	74 (15.45%)
	Program B	1.28 ± 0.20	1:56 ± 0:07	405 (84.55%)
NILM-house-4	Program A	0.97 ± 0.11	1:33 ± 0:03	246 (81.46%)
	Program B	1.23 ± 0.14	1:53 ± 0:15	56 (18.54%)
NILM-house-5	Program A	0.86 ± 0.31	1:46 ± 0:20	109 (29.30%)
	Program B	0.96 ± 0.10	1:19 ± 0:02	263 (70.70%)

dishwashers. However, we must highlight that the clustering analysis is more effective for the dishwasher as the washing machine operations present a greater variance within the same cluster both in terms of energy consumption and duration. On the other hand, dishwasher operations are well-defined and do not present significant deviations within the same cluster, neither in energy consumption nor in duration. These considerations are proved by the standard deviation values, which are smaller in Table VIII for the dishwasher (i.e., 0.14 kWh) than for the washing machine in Table VII (i.e., 0.17 kWh). Moreover, the average standard deviation in the duration of the washing machine programs (i.e., 14 minutes) is about two times the one for the dishwasher (i.e., 7 minutes). The reason for this discrepancy in the cohesion is probably due to the different water temperatures selected by the user of the washing machine, which may cause a lot of variability within the same operational mode. Nevertheless, the clustering algorithm managed to extract significant groups of operations showing a strong affinity in their power signatures.

To compare the UK-DALE and the NILM dataset we can apply the Principal Component Analysis (PCA) to the latent representations extracted by the autoencoder and plot the first two principal components of the clusters. In this way, we discover that the clusters formed in the first case are slightly more cohesive than those created in the second. For the washing machine, Figure 7 shows that the clusters are less sparse for the UK-DALE than for the NILM dataset. This difference is even more visible for the clusters of the dishwasher, reported in Figure 8. This difference is also highlighted by the average standard deviations obtained for the two datasets and reported in Table VII and VIII. The clusters from the submetered data present an average deviation in the power consumption of 0.11 kWh and a variation of only 5 minutes for their duration. In contrast, the clusters obtained from the NILM data set presents higher deviations, i.e., 0.17 kWh in terms of power and 15 minutes in terms of duration. The difference between the two datasets is probably due to inaccuracies in the disaggregation algorithm, which may generate some spurious operation cycles containing traces from other devices in the aggregated power signal. However, the gap in terms of clustering does not interfere with the final goal of our methodology. As a matter of fact, we can still provide a good estimate of the average energy demand for a certain program if we compute the mean of the energy consumption for its cluster. In this sense, learning a good data representation is fundamental to placing spurious operation cycles in the neighborhood of their correct program. Based on the previous observations, we can state that our algorithm can properly manage data from both the submeters and the disaggregation algorithms, once the available power signatures are sufficiently accurate.

F. Applications

The results presented in this study can be beneficial for both end-users and utility companies to optimize their energy

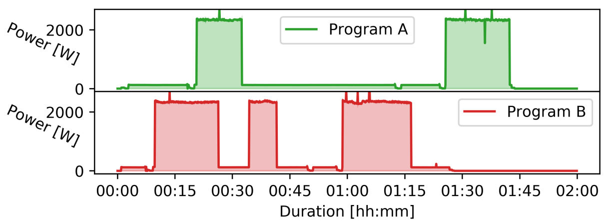

Fig. 5: Operational modes of the dishwasher of the first house of the UK-DALE dataset.

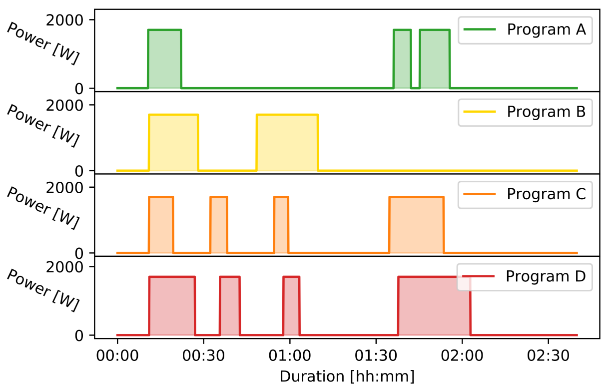

Fig. 6: Operational modes of the dishwasher of the first house of the NILM dataset.

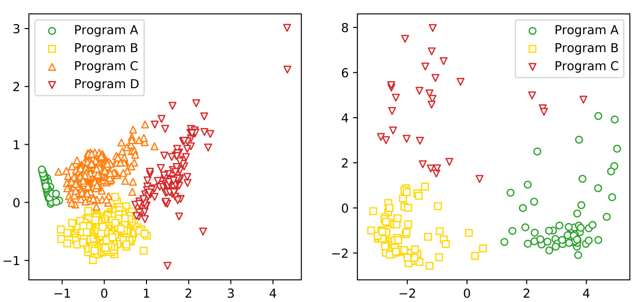

Fig. 7: Two-dimensional PCA projections of the washing machine cycles in the first house of the UK-DALE dataset (left-hand side) and in the fourth house of the NILM dataset (right-hand side).

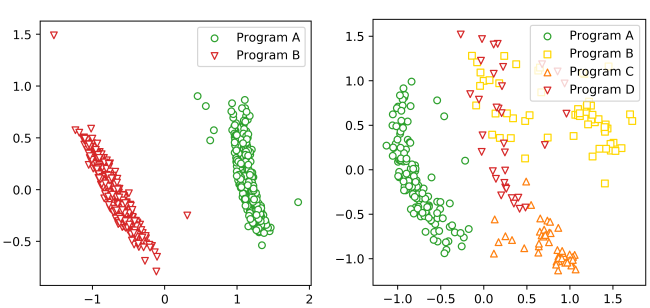

Fig. 8: Two-dimensional PCA projections of the dishwasher cycles in the first house of the UK-DALE dataset (left-hand side) and in the first house of the NILM dataset (right-hand side).

utilization and management. Past publications show that personalized recommendation systems are more effective than traditional real-time feedback in reducing the total energy consumption of end-users [9]. Indeed, people usually prefer to receive precise suggestions to reduce their energy consumption instead of interpreting cryptic energy reports [35].

In this context, the proposed algorithm can be used to implement an advanced recommendation system aimed at suggesting more practical actions to reduce the end-users' energy consumption. These actions can be tailored to the specific habits of consumers, who can be progressively taught to apply better practices in the use of their appliances. Smart challenges and personalized notifications can be used to increase the end-users' engagement and periodical feedback can be provided to check their progress towards the end goals. Thanks to the proposed algorithm, we can track the exact number of working cycles of each appliance's program and suggest new methods to reduce the impact of energy-intensive programs on the energy bill. For example, end-users can choose a lighter operational mode to decrease their energy consumption due to a washing machine, a dryer, or a dishwasher. In those houses equipped with solar panels, the information on the power consumption of each program can be matched with the domestic provisioning of solar power to increase the energy self-sufficiency of the house.

Alternatively, heavy programs can be shifted towards off-peak hours to leverage potential discounts from utility companies, favoring lighter operation modes during peak hours. Indeed, utility companies can improve DR programs by suggesting lighter operational modes instead of shifting the entire program activation, with only a minor impact on the end-user's comfort. Furthermore, the information on the duration of the operational modes can help Demand-Side Management (DSM) programs to provide more precise scheduling of their devices during the day. Finally, the use of lighter programs together with the adoption of green habits can also help to reduce the overall energy demand of entire urban areas during peak hours, which may relax the requirements of DR programs.

VI. Conclusion

We introduce a two-stage clustering approach to identify the different operational modes of household appliances based on the analysis of their power signatures. We first implement an autoencoder neural network to create a better data representation of the power signatures. Then, we fit a K-means algorithm to the latent representation extracted by the autoencoder. In this way, we group operational cycles with similar power signatures. Finally, we test our framework on authentic working cycles. To guarantee the feasibility of the proposed framework, we analyze the power signatures of some washing machines and dishwashers collected using both submeters and non-intrusive load monitoring techniques. Our clustering results reveal the presence of multiple operation modes with well-defined features in terms of both energy consumption and duration. Overall, our analysis demonstrates the effectiveness of our methodology and our study presents the very first framework to automatically classify the different operational modes of deferrable apparatus with a completely unsupervised approach. The adoption of deep learning techniques overcame the limitations of handcrafted features as proposed in previous works, thus allowing a greater flexibility of the methodology, which can now be easily generalized to multiple devices and manufacturers. In addition, the features extracted by the autoencoders speed up the clustering process by reducing the data dimensionality and are more robust to noisy power signatures such as those provided by NILM algorithms. Most importantly, the unsupervised learning approach does not rely on the annotations of our equipment programs, which are rarely available in a real-world scenario. Finally, we provide a set of real-world applications of our work that can help both end-users and utility companies to optimize their energy utilization and management.

REFERENCES

[1] X. Yan, Y. Ozturk, Z. Hu, and Y. Song, "A review on price-driven residential demand response," *Renewable and Sustainable Energy Reviews*, vol. 96, pp. 411–419, 2018.

[2] S. R. Sinsel, R. L. Riemke, and V. H. Hoffmann, "Challenges and solution technologies for the integration of variable renewable energy sources—a review," *renewable energy*, vol. 145, pp. 2271–2285, 2020.

[3] F. Bandeiras, E. Pinheiro, M. Gomes, P. Coelho, and J. Fernandes, "Review of the cooperation and operation of microgrid clusters," *Renewable and Sustainable Energy Reviews*, vol. 133, p. 110311, 2020.

[4] F. Pallonetto, M. De Rosa, F. D'Ettorre, and D. P. Finn, "On the assessment and control optimisation of demand response programs in residential buildings," *Renewable and Sustainable Energy Reviews*, vol. 127, p. 109861, 2020.

[5] A. R. Jordehi, "Optimisation of demand response in electric power systems, a review," *Renewable and sustainable energy reviews*, vol. 103, pp. 308–319, 2019.

[6] L. Wenpeng, "Advanced metering infrastructure," *Southern Power System Technology*, vol. 3, no. 2, pp. 6–10, 2009.

[7] A. Ghosal and M. Conti, "Key management systems for smart grid advanced metering infrastructure: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2831–2848, 2019.

[8] R. Gopinath, M. Kumar, C. P. C. Joshua, and K. Srinivas, "Energy management using non-intrusive load monitoring techniques-state-of-the-art and future research directions," *Sustainable Cities and Society*, p. 102411, 2020.

[9] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

[10] Q. Yan, W. Xudong, and W. Zun, "Applications of nilm in the optimization management of intelligent home energy management system," in *2020 IEEE Sustainable Power and Energy Conference (iSPEC)*. IEEE, 2020, pp. 2300–2305.

[11] S. Gómez Melgar, A. Sánchez Cordero, M. Videras Rodríguez, and J. M. Andújar Márquez, "Matching energy consumption and photovoltaic production in a retrofitted dwelling in subtropical climate without a backup system," *Energies*, vol. 13, no. 22, p. 6026, 2020.

[12] Z. Wang and R. S. Srinivasan, "Classification of household appliance operation cycles: A case-study approach," *Energies*, vol. 8, no. 9, pp. 10 522–10 536, 2015.

[13] A. Jaradat, H. Lutfiyya, and A. Haque, "Demand response for residential uses: A data analytics approach," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–6.

[14] ——, "Appliance operation modes identification using cycles clustering," *arXiv preprint arXiv:2101.10472*, 2021.

[15] M. Marcu, M. Darie, and C. Cernazanu-Glavan, "Comparative analysis of home appliances' functional regimes using power signatures," in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2018, pp. 1–6.

[16] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[17] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *arXiv preprint arXiv:1901.10738*, 2019.

[18] Q. Ma, J. Zheng, S. Li, and G. W. Cottrell, "Learning representations for time series clustering," *Advances in neural information processing systems*, vol. 32, 2019.

[19] G. Richard, B. Grossin, G. Germaine, G. Hébrail, and A. de Moliner, "Autoencoder-based time series clustering with energy applications," *arXiv preprint arXiv:2002.03624*, 2020.

[20] Y. Yang, J. Zhong, W. Li, T. A. Gulliver, and S. Li, "Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 6892–6902, 2019.

[21] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[22] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*, 2015, pp. 55–64.

[23] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[24] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1419–1429, 2019.

[25] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, no. 150007, 2015.

[26] "Omitted for double blind review."

[27] Y. Liu, L. Zhong, J. Qiu, J. Lu, and W. Wang, "Unsupervised domain adaptation for nonintrusive load monitoring via adversarial and joint adaptation network," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 266–277, 2021.

[28] L. Massidda, M. Marrocu, and S. Manca, "Non-intrusive load disaggregation by convolutional neural network and multilabel classification," *Applied Sciences*, vol. 10, no. 4, p. 1454, 2020.

[29] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International conference on artificial neural networks*, 2011, pp. 52–59.

[32] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[33] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[34] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[35] G. T. Gardner and P. C. Stern, "The short list: The most effective actions us households can take to curb climate change," *Environment: science and policy for sustainable development*, vol. 50, no. 5, pp. 12–25, 2008.