Electricity Demand Activation Extraction: From Known to Unknown Signatures, Using Similarity Search

Pauline Laviron
EDF
Paris, France
pauline.laviron-petit@edf.fr

Xueqi Dai
Huazhong University of Science and Technology
Wuhan, China
daixueqi@hust.edu.cn

Bérénice Huquet
EDF
Paris, France
berenice.huquet@edf.fr

Themis Palpanas
University of Paris, French University Institute (IUF)
Paris, France
themis@mi.parisdescartes.fr

ABSTRACT

A powerful tool for reducing energy consumption is energy disaggregation (also called NILM Non-Intrusive Load Monitoring), where the goal is to disaggregate the smart meter readings of a household's total electricity consumption to the consumption of that household's individual appliances. State-of-the-art machine learning methods are widely used to solve the NILM problem, but in order to generalize well they require a large amount of data, which are not readily available. We thus need labeled electricity consumption readings from individual appliance activations. Though, manually annotating the start and end of single-appliance activations is extremely laborious and time consuming. Therefore, automated activation extraction methods are needed. Earlier approaches to solve this problem suffer from limitations, such as incomplete signatures, double signatures, and outliers. In this work, we introduce three scalable methods based on techniques that use time series similarity search. The first method is Cartesio that (improves on earlier work that relies on known features of the appliance) and separately detects the start and end times of an appliance activation. The second method is ValmA, a method for identifying previously unknown candidate signatures of variable length, which is essentially parameter-free. The third method is SimBA, a similarity search based method for efficient detection of known signatures in large datasets. These signatures can be computed from the activations extracted using the previous methods. Our experimental results with real 6 and 10 seconds-sampling data demonstrate that, compared to a state-of-the-art solution, our methods improve the accuracy and robustness of appliance activation extraction in very large time series collections. To compare these methods, we also describe a new accuracy measure that takes into account the special characteristics of subsequences, leading to more precise performance evaluation results.

e-Energy '21, June 28-July 2, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8333-2/21/06…$15.00
https://doi.org/10.1145/3447555.3464865

CCS CONCEPTS

• **Computing methodologies** → **Motif discovery**.

KEYWORDS

Electricity Demand Disaggregation, Similarity Search, Matrix Profile, NILM, 10 seconds power data

ACM Reference Format:
Pauline Laviron, Xueqi Dai, Bérénice Huquet, and Themis Palpanas. 2021. Electricity Demand Activation Extraction: From Known to Unknown Signatures, Using Similarity Search. In *The Twelfth ACM International Conference on Future Energy Systems (e-Energy '21), June 28-July 2, 2021,Virtual Event, Italy*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3447555.3464865

1 INTRODUCTION

Energy management is the center of attention in current environmental and economical debates. Energy efficiency, one of the key of a successful ecological transition, requires precise and robust information about energy consumption. Davis and all [4] advocates that aggregate energy feedback can reduce energy consumption by about 3%. Non-Intrusive Load Monitoring (NILM) or energy disaggregation is one way to give this feedback. NILM is the process of estimating information (consumption, time of use...) on each individual appliance (e.g., heating, water heating, washing machine, refrigerator) from the global consumption of a household.

The NILM problem was formalized in the mid-1980s by George Hart [11]. More recently deep learning for NILM was introduced by Jack Kelly [12] with major progress on state of the art models, it made a major breakthrough against old methodology.

In this paper, we focus on automated activation extraction from single appliance load curve methods. This is an intermediate step in order to work on NILM techniques. It can be useful in two cases:

(1) To tag start and end of single-appliance activations in order to train NILM algorithms which aims at detecting the operating time of an appliance.
(2) To create a signature collection. Indeed, lack of supervised data (with both the sub-meter and aggregated load curves) in the NILM field is one of the main problems stalling the improvement of disaggregation algorithms, especially deep learning ones. One way to overcome this limitation is to

perform data augmentation. It consists in combining operating period of different appliances according to a certain scenario of usage to create a synthetic aggregated curve as in [23]. These scenarios are generated from realistic schedule of appliance usage and need in input examples of appliance signatures.

Though, manually annotating the start and end of single-appliance activations is extremely laborious and time consuming. Therefore, automated activation extraction methods are needed.

[Contributions] In this work, we develop three new methods, i.e., Cartesio, ValmA and SimBA, for automated activation extraction from single appliance load curves. Cartesio is inspired by previous work [12] and identifies activations based on thresholds on their values and durations. Though, it separates the detection of the start and end parts of the activations, which in some cases (e.g., for washing machines that have complex activation signatures) leads to superior performance. On the other hand, ValmA and SimBA are novel techniques based on time series methods. They operate directly on the time series signal, and exploit similarities in the shapes of subsequences in order to detect the activation signatures. SimBA is able to detect activations given an example set of signatures as input, while ValmA is able to detect activations with no prior input, or domain knowledge, which makes it suitable for identifying new, previously unknown activations. In this paper, we also describe new measures suitable for evaluating the different aspects of the performance behavior of the different methods. Finally, we contribute to the evaluation of the algorithms with new, manually-annotated seconds-sampling datasets that we use in our experimental comparison.

[Outline] The rest of this paper is structured as follows. In Section 2, we present existing related works on the subject. Then, we introduce in Section 3 similarity search and matrix profile methods, on which ValmA and SimBA are based. Section 4 describes our proposed approaches. In Section 5 we show the experimental evaluation. We finally conclude in Section 6 and promising research directions for future work.

2 RELATED WORK

The subject of single load activation extraction has not been discussed a lot in the NILM community since the question of data augmentation is relatively new to the field. A method for activation extraction has been proposed by Kelly and Knottenbelt [12] (refer to Section 3.2 in their study). Their method, named get_activations, is a solution based on thresholds for the various appliances. Figure 1 shows an illustration of the get_activations method implemented in NILMTK package [2].

The get_activations method uses several parameters. The definition of these parameters are the same for all appliances, but each type of appliance has its own values. These parameters were tuned specifically on UK-DALE dataset [13]. This method is easily understandable but encounters four main limitations. First, it can cut the beginning or the end of an activation. Secondly, it sometimes struggles to separate one activation following another, as for instance in a washing machine load curve. Thirdly, it fails to exclude outliers or other appliance (if two appliances are mixed on the same sub-meter) that have similar power and activation duration. Finally, it can generalize badly on another dataset.

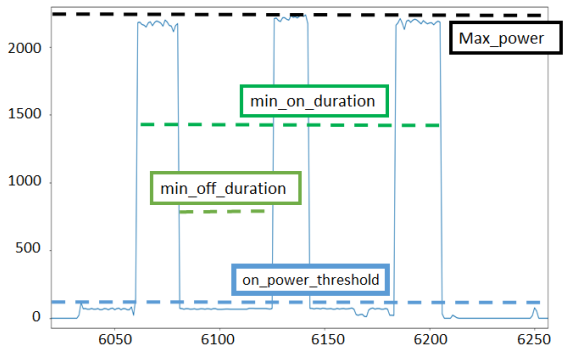

Figure 1: Illustration of Kelly get_activations method. We keep activations whose power are more than on_power_threshold and less than max_power. Activation also needs to last more than min_on_duration and with interwindow duration shorter than min_off_duration.

3 BACKGROUND

3.1 Similarity Search

Time series similarity search is a key operation relevant to several time series analysis tasks, such as classification, anomaly detection, and frequent pattern identification. Similarity search can also be used for searching and detecting appliance activation signatures. However, executing the similarity search operation on very large time series collections is notoriously challenging, due to the high dimensionality of the time series (here we use the term *dimensionality* to refer to the length, or number of points in a time series).

In response to the need for fast and scalable similarity search [5, 7, 19, 21], several indexing approaches have been developed [6, 8–10, 20]. The goal of these indexes is to group similar subsequences together, and when a query arrives, to guide the search to a subset of the dataset (i.e., groups of subsequences) that will contain the answer. Therefore, by pruning the rest of the search space, these approaches lead to fast execution times for similarity search queries.

However, almost all proposed indexing techniques share a common restriction: they only support similarity search with queries of a fixed length (i.e., dimensionality). The length of the query has to be the same as the length of the subsequences in the index, and its value is determined at index construction time. We note that this requirement leads to limitations during the analysis phase, since it restricts all the results to be of that length. In our case, this means that we can only identify appliance signatures of the exact same length, even though in practice signatures have slight variations in their duration (such as the time duration of a washing machine cycle).

In order to overcome this problem, we use ULISSE (ULtra compact Index for variable-length Similarity SEarch in data series) [14,

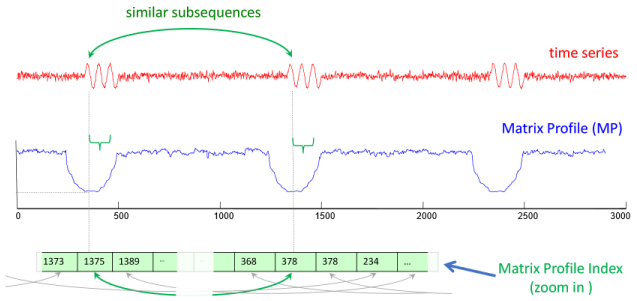

Figure 2: Matrix Profile (MP) example [1] ©Eamonn Keogh

15], which is the first single-index solution that supports fast answering of variable-length similarity search queries for both non Z-normalized and Z-normalized time series collections.

ULISSE produces exact results, and is based on the following key idea: a data structure that indexes subsequences of a given length extracted from a long time series using a sliding window already contains all the information necessary for reasoning about subsequences of any length of the original time series. Therefore, the problem of enabling a time series index to answer queries of variable-length becomes a problem of how to reorganize the information that already exists in the index. To this effect, ULISSE proposes a new summarization technique that is able to represent contiguous and overlapping subsequences, leading to succinct summaries. It combines the representation of several subsequences within a single summary, and enables fast similarity search for variable-length queries. For a detailed description of the ULISSE approach, we refer the reader to the original study [14, 15].

3.2 Motif Discovery

The problem of efficiently extracting unknown signatures from smart meter data is very relevant to the *motif discovery* problem. In the time series literature, a motif is an unknown pattern that repeats (approximately the same) unusually often in the data. Over the last decade, motif discovery has emerged as an important primitive operation for time series analysis, and has been studied a lot, leading to substantial progress on its scalability. The Matrix Profile (MP) [1] is a data structure that annotates a time series, and efficiently solves the motif discovery problem.

The concept of MP is illustrated in Figure 2. This example shows a long time series (top/red), with a subsequence pattern (sinus wave) that repeats approximately the same three times. The MP is itself a sequence (middle/blue), where at every point it records the distance of the subsequence of the original time series starting at that point to its nearest neighbor in the same series. Therefore, the valleys (lowest points) in the MP correspond to subsequences that are very similar to one another. The matrix profile index (bottom) records the position of these nearest neighbors, and allows us to access them in constant time.

Nevertheless, almost all existing solutions require the user to choose as a parameter the desired length of the motifs. Once again, this is a limitation for the subsequent analysis steps, which would benefit from the discovery of motifs of different lengths. The only available solution in this case would be to run the algorithm over all lengths in the desired range and, rank the various motifs discovered, picking eventually the subsequences that contain the desired pattern. Clearly, this solution is not optimal for at least two reasons: scalability, since finding motifs is an expensive operation, and effectiveness, since it does not provide a way to compare motifs of different lengths.

In our study, we use VALMOD (Variable Length Matrix Profile) [16, 18], which is the first approach for discovering motifs of variable length. VALMOD is up to orders of magnitude faster than the baseline solution. This efficiency derives from the use of a lower bounding technique that enables the algorithm to quickly estimate distances of neighboring subsequences without spending time to perform the corresponding exact calculations. Therefore, VALMOD prunes a considerable part of the search space, while still providing the correct final results. We refer the reader to the original studies for a detailed description of the VALMOD algorithm [16, 18] and corresponding system [17].

Table 1: Parameters used in get_activations

Appliances	WM	DW	FR	FZ	FR-FZ	MW	KE
Max power (watts)	2700	2600	3000	1500	1600	1500	3000
On power threshold (watts)	20	30	20	20	20	10	10
Min. on duration (secs)	1600	1400	600	550	350	60	60
Min. off duration (secs)	800	1600	100	100	50	40	50
border (points)	1	1	1	1	1	1	1

4 PROPOSED APPROACH

In the following sections, we use the following abbreviations, WM for Washing Machine, DW for Dishwasher, FR for Fridge, FZ for freezer, FR-FZ for Fridge-Freezer, MW for Microwave and KE for Kettle .

4.1 Threshold Method

In order to compare with the state-of-the art method, we compute get_activations method. We tune the parameters for REFIT dataset [3]. The new value of the parameters are listed in Table 1. The former parameters tuned for UK-DALE [13] are listed in Table 2

4.2 Cartesio

Threshold method (get_activations from the NILMTK [2]) meets the needs of most scenarios, but the Figure 3 reports some critical cases such as incomplete signatures, double signatures and outliers.

A washing machine whole process follows these sequential steps:

- a pre-washing phase (usual but not systematic)
- a water heating phase with a very high power and long duration,

Table 2: Parameters used in get_activations by Kelly

Appliances	WM	DW	FR	FZ	FR-FZ	MW	KE
Max power (watts)	2500	2500	300	300	300	3000	3100
On power threshold (watts)	1800	1800	12	12	12		
Min. on duration (secs)	1800	1800	60	55	35	60	12
Min. off duration (secs)	160	1800	12	12	12	30	0
border (points)	1	1	1	1	1	1	1

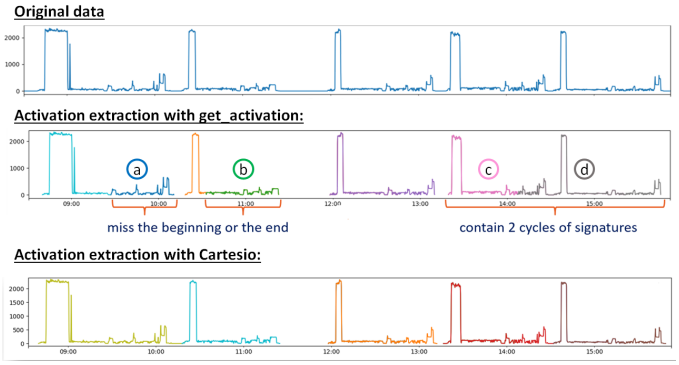

Figure 3: Example of activations extraction for the get_activations method and Cartesio. At the top the original data. In the middle results from get_activations: a&b) omission of the beginning of the activation sequence; c) omission of the end of the activation sequence; d) get_activations considers two neighboring activations as a single activation. At the bottom, results of Cartesio: Cartesio correctly identifies all activations.

- a first washing phase,
- an optional heating phase between the two washing phases
- a second washing phase,
- a rinsing phase (spin cycles)
- a drain and drying phase (1000 rotations/minute) which power will be a little higher than the rinsing phase.

Therefore, we proposed a new activation extraction method, called Cartesio which identities the start phase and end phase. It's suitable for appliances which have an obvious start and end characteristic like washing machine, or the cases where the activations tend to be close to each other (when a washing machine ends, another one starts). It also can be applied on appliances which work continuously (as television) by setting the end-phase related parameters to zero.

Cartesio's main steps are the following:

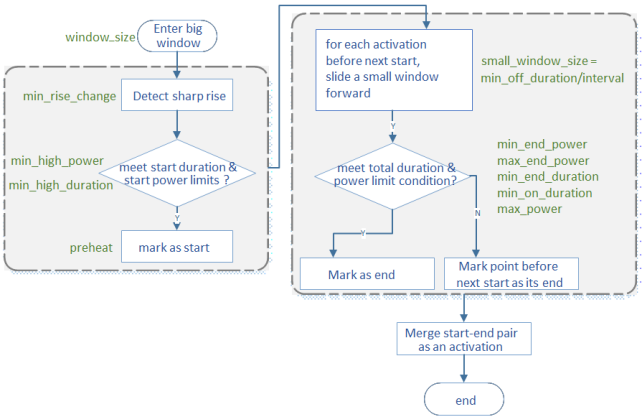

Figure 4: Flowchart of Cartesio approach

(1) Detect a sharp rise greater than **(min_rise_change)**: each of sharp rise has the potential to be a start phase,
(2) Check if this potential peak reaches a minimal duration **(min_high_duration)** and minimal power **(min_high_power)**,
(3) If it does, subtract **preheat** from this time point to obtain a start point and wait for the next step. If it does not, this point is not a true start and we skip it,
(4) Slide a small window of length of **min_off_duration** seconds between this start and the next sharp rise point. Check if any of these windows meets an "end of activation" requirement: power between **[min_end_power, max_end_power]**. If it does, then keep this end of this window as an end. If no small windows contains an end and in order to ensure there is always an end to a start, we set the next high rise (minus border) as an end.
(5) Finally, when the start and the end of an activation are set, check the whole activation integrity regarding the minimum of duration **(min_on_duration)** and the max power **(max_power)**.

The flowchart of the method is summarised in Figure 4. The values of the threshold parameters tuned for Cartesio are listed in Table 3, and those introduced in Cartesio are listed in Table 4.

4.3 ValmA

Activation extraction procedure using ValmA is as follows:

(1) Preprocessing (very important step): Add some white noises to the baseline. Indeed, single-appliance load curve have sometimes some long segments of low constant power. The Euclidean distance between these flat areas is 0, consequently ValmA may classify all these segments as motifs while missing the correct activations. We then add **Noise Gain** to the points below the percentile and we also add a small noise to all the data series.
(2) Set the minimal and maximal lengths **Length range**, then run the VALMOD algorithm. VALMOD returns several result files including matrix profile index (VALMAP) and length

Table 3: Old parameters used in Cartesio

Appliances	WM	DW	FR	FZ	FR-FZ	MW	KE
Max power (watts)	2700	2700	1500	1500	3000	1500	4000
On power threshold (watts)	20	20	5	5	5	2	2
Min. on duration (secs)	800	1600	600	600	600	60	60
Min. off duration (secs)	600	1600	100	100	50	40	50
border (points)	0	10	10	10	10	0	0

Table 4: New parameters used in Cartesio

Appliances	WM	DW	FR	FZ	FR-FZ	MW	KE
Min rise change (watts)	700	900	50	50	50	500	500
Min high power (watts)	1500	2000	80	50	50	1000	1000
Min. high duration (secs)	180	600	800	800	800	40	40
Preheat (secs)	300	10	10	10	10	10	10
Min end power size (secs)	150	2000	0	0	0	0	0
Max end power (watts)	600	2600	0	0	0	0	0

profile which records the length of the best match among $[min\_length, max\_length]$.

(3) Post processing: Detect peaks in VALMAP time series to get the start time of the activations. Then, find the corresponding length in length profile and add it to the start time to obtain the complete activation. We perform peaks detection on the opposite signal cleared form baseline, this procedure requires two parameters: the value of the peak that should be greater than 30% of the maximum and the minimal interval between peak inferior to 1000 seconds. We also suppress redundant activations and zeros at the beginning and end of each signature.

The diagram of ValmA process is given Figure 5.

Figure 6 shows the result of ValmA for washing machine. Above, we draw the original time serie with several washing machine (WM) activations. In the middle, we see that the VALMAP matrix profile

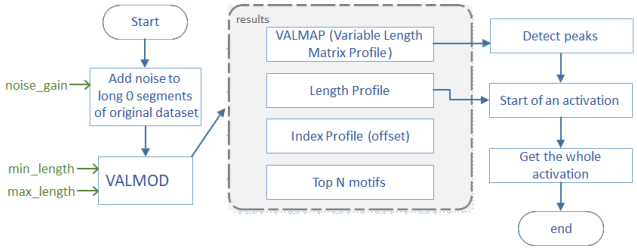

Figure 5: Flowchart of ValmA approach

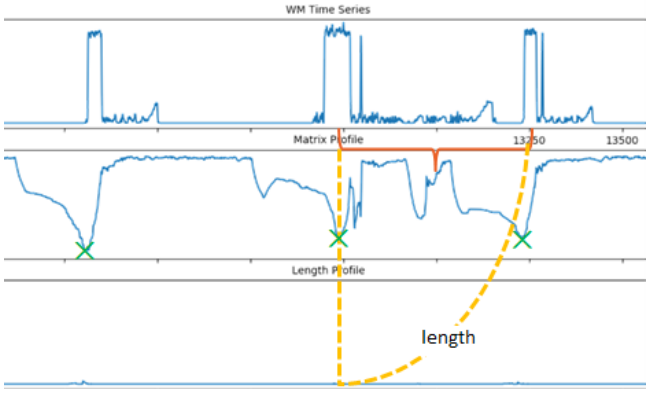

Figure 6: ValmA: How to extract activations from the original results. The yellow line delimits the signature length. The green crosses indicate the beginning of a new activation.

Table 5: Parameters used in ValmA

Appliance	Noise Gain	Length Range
WM	2000	200-500
DW	800	200-900
FR	100	150-250
FZ	100	150-250
FR-FZ	100	150-350
MW	400	30-200
KE	50	100-200

presents peaks (drawn in green) at the beginning of the activations. Below, the length profile indicates the activation duration.

Table 5 lists the ValmA parameters we used in this study for each appliance.

Figure 7 shows that even though VALMOD examines several subsequence lengths at once, it exhibits very good scalability behavior. This is because VALMOD uses an effective pruning strategy, which allows the algorithm to prune a large percentage of the computations.

4.4 SimBA

In this section, we assume that a signature library has been built from a dataset (from the Cartesio method for example). The idea is to define a template for each type of signatures of each appliance through clustering and barycenter computation. Indeed, a same

Figure 7: ValmA: Relationship between running time and data/query length

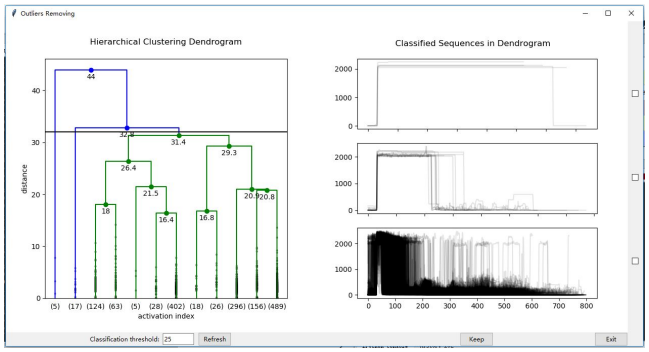

Figure 8: GUI that allows the user to select barycenters (washing machine example)

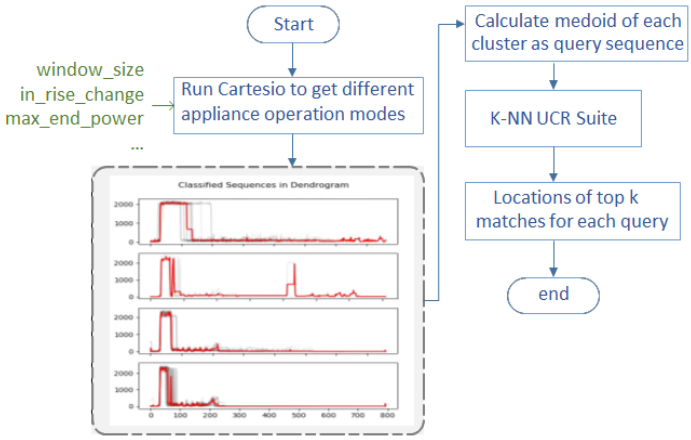

Figure 9: Flowchart of SimBA method

appliance can have several templates (for example, a washing machine can have one corresponding to short cycle and an other to long cycle). These templates are then used to search for similar signatures in new single appliances load curves.

(1) Use Cartesio or threshold method to extract activations from single appliance load curves from appliances of several house,
(2) Classify the activations using hierarchical clustering using DTW-distance between standardized time series. Thanks to a graphical user interface (GUI cf Figure 8), we can change the black line which represents classification threshold in the dendogram by updating the value on the left. Therefore, we will see different clustering results and choose a satisfactory classification result. For simplicity and easy reproduction, we choose a cut of the dendogram at 10 for all appliances except kettle and some microwave at 5.
(3) Draw the barycenter by calculating the DTW medoid of each clusters (in red in Figure 9). We then select reasonable barycenters as the input query to similarity search on the right. At the end of this step, one or more templates represent each appliance.
(4) Run k-nn similarity search program 'ULISSE' [14] for each template of the appliance you are searching for. It will return the k most similar motifs to the template in the input single appliance load curve. k is thus a parameter to tune: if k is too small, we can't recover all the activations; if k is too big, we will increase false alarm and thereby obtain a bad accuracy either. The optimal k depends on the numbers of real activations in the input single appliance load curve. In general, the higher the occurrence frequency, the bigger the k. We can use basic prior knowledge about appliance occurrences in time. For instance, a washing machine has few activations per week and a fridge several per day.
(5) Post treatment: find the start time of each real activation. The k most similar motifs were obtained with ULISSE. Several of them can represent the same real activation. We thus use a similar technic as ValmA. We perform peaks detection on the opposite signal cleared form baseline and fill with zeros on the remaining points, this procedure requires two parameters: the value of the peak that should be greater than 30% of the maximum and the minimal interval between peak inferior to **next_act** seconds.
(6) Post treatment: If the highest percentile is above **min_high_power**, add a new activation from the start time (defined in the previous step) to the start time plus the length of the template.

Table 6: Parameters of SimBA

	WM	DW	FR	FZ	FR-FZ	MW	KE
next_act (s)	6000	6000	2000	2000	2000	50	50
k of k-nn	5000	5000	5000	5000	5000	5000	5000
min_high_power (watts)	1000	1000	20	20	20	20	20

	Institution	Location	Duration per House	Number of houses	Sample frequency
UK-Dale (2015)	Imperial College	London, UK	39-786 days	5	6 s
REFIT (2015)	U. Of Strathclyde	Loughborough, UK	2 years	20	6-8 s

Figure 10: REFIT and UK-DALE Datasets

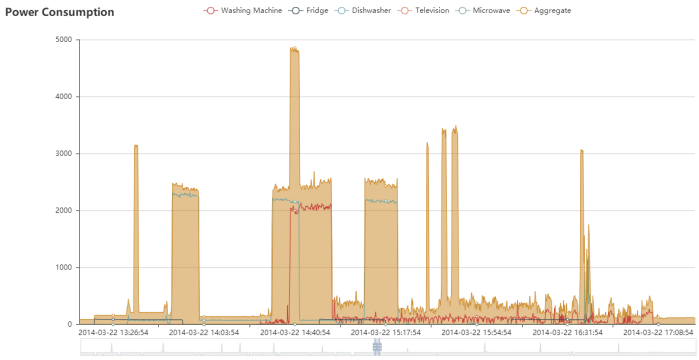

Figure 11: Part of the REFIT dataset (from house 2)

5 EXPERIMENTAL SETUP

5.1 Setup

All the results are tested on laptop, Intel Core i5-8265U. Cartesio was written completely in Python 3.6. ValmA uses Python 3.6 for the preprocessing and post processing part but calls C code for the VALMOD algorithm. SimBa was written in Python 3.6 but calls ULISSE C code for the similarity search step.

5.2 Datasets

An energy disaggregation data set is a collection of electrical energy measurements taken from real-world scenarios, without disrupting the everyday routines in the monitored space. These usually contain measurements from the aggregate consumption (taken from the mains) and of the individual loads (i.e., ground-truth data), which is obtained either by measuring each load at the plug-level or measuring the individual circuit to which the load is connected. In a real-world scenario, typically multiple loads are connected to the same circuit. The currently available datasets can be categorized as event-based or event-less datasets. The major difference between the two is that event-less approaches do not require the identification of individual power changes. Consequently, collecting datasets for event-less approaches is more straightforward and less time consuming.

[22] referenced some famous and typical public electrical load measurements datasets. We chose in this work to focus on REFIT and UK-DALE datasets [3, 13]. These two datasets have a relatively large amount of houses over a large period of time with a satisfactory sampling. Having two datasets also enable us to test generalization capabilities of the methods. Furthermore, the former method get_activations has been tuned on UK-DALE dataset.

The visualization of REFIT data set is shown Figure 11. In this work, we resampled REFIT dataset to 10 seconds.

[Dataset overview] We selected seven target appliances in all our experiments: the washing machine (WM), dishwasher (DW), fridge (FR), freezer (FZ), fridge-freezer (FR-FZ), Microwave (MW) and Kettle (KE). Among all these appliances, washing machine, dishwasher, microwave and kettle have high power up to 2-2.5 kW, their frequency and time of occurrence are irregular. The others have lower power around a few hundred watts, they do not have many operative modes so most of their activation signatures are the same, and they follow a certain occurrences frequency. For better parameter setting later, here we analyze through a boxplot the distribution of the length of activations for these seven appliances in our test dataset:

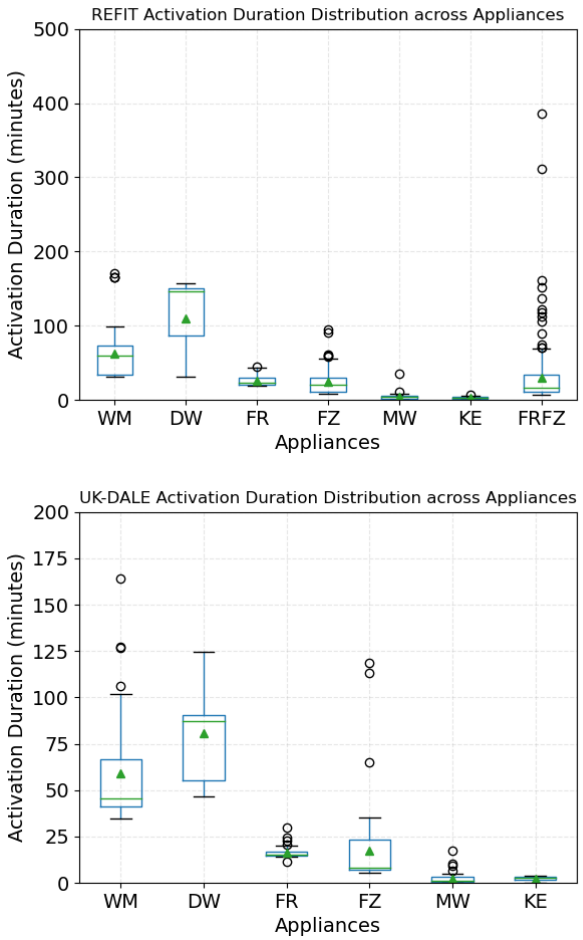

Figure 12: Length distribution of each appliance in the manually annotated dataset for REFIT (above) and UK-DALE (below)

In Figure 12, boxplots of the length of real activations are available per appliance for REFIT and UK-DALE datasets. The 5 horizontal lines for each appliance represent: the maximum, upper quartile,

median, lower quartile and the minimum respectively. The green triangle is the mean value. The black circles are outliers. Distribution of the appliance duration are quite similar between REFIT and UK-DALE datasets. We only select fridge-freezers for REFIT dataset and it shows a large variability of length compare to the others appliances.

[Dataset Partitioning] We manually annotated ground-truth start and end of activations for 3-20 days (depending on the occurrence of the appliance per day) of data from REFIT's houses 1-5 and from UK-DALE's houses 1,2,4 and 5 for some appliances. They serve as test dataset for all the following experiments.

For SimBA, the main goal is to use barycenters computed once on a training dataset because this step is time consuming. We can then compute the similarity search step, which is fast, on a new load curve without recomputing the barycenter extraction step. For REFIT, to avoid an overfitting problem for this method's evaluation, we thus implement a leave-one-out procedure where we train the barycenter on some houses and test it on the remaining house and so on. For UK-DALE, since there were not enough annotated houses to perform a clean leave one out procedure, we use resampled REFIT barycenter computed on all houses. It is thus a strong test of generalization capability for SimBA.

5.3 Evaluation Measures

To attest to the algorithms performances, classical machine learning performances measures as recall or precision are not completely well suited for time series and even less for our specific task. Tatbul and all [24] suggest an extension of precision and recall for time series especially for anomaly detection. We inspired from it but define our own performance measures that allow us to answer our primary questions. These measures complement one another without redundancies.

In the following section, [x1, x2] will refer to the groundtruth interval and [y1, y2] to the one extracted by our method also called result activation. We will denote # as the number element in a set, grt_i (resp. res_i) the ith activation in the groundtruth (resp. in the results), grt (resp. res) all the activations in the groundtruth (resp. in the results). We consider that two time series intersect if and only if there is at least one time point in common:

$$\text{intersect}([\text{x1}, \text{x2}], [\text{y1}, \text{y2}]) = \begin{cases} 1 & \text{if } (\min(x2, y2) - \max(x1, y1)) > 0 \\ 0 & \text{else.} \end{cases} \tag{1}$$

We define the projection of an activation [y1, y2] on an other [x1, x2] as:

$$\text{projection}([\text{y1}, \text{y2}] \rightarrow [\text{x1}, \text{x2}]) = \max\left(\frac{\min(\text{x2}, \text{y2}) - \max(\text{x1}, \text{y1})}{\text{x2} - \text{x1}}, 0\right) \tag{2}$$

The range of the projection measure is between [0,1], 0 indicates that the two activations have no intersection and 1 indicates that the activation ([x1, x2] belongs to the [y1, y2] activation .

Does the method recover all activations in the ground-truth? We want to retrieve the most activations possible to use in the simulator, or for the barycenter computation. Thus, we define recovery as the ratio between the number of activations in the groundtruth recovered by the method and the total number of activations in the groundtruth.

$$\text{Recovery} = \frac{\sum_{\text{i}} \text{intersect}(\text{grt}_\text{i}, \text{res})}{\#\text{grt}} \tag{3}$$

If the method recovers a groundtruth activation, is the result complete? To attest the quality of activation extraction, we define a new performance measure that we called completeness. We want to attest how much of the reference activation we recover.

$$\text{Completeness}(\text{grt}_\text{i}) = \arg\max_{\text{j}}(\text{projection}(\text{res}_\text{j} \rightarrow \text{grt}_\text{i})) \tag{4}$$

If the groundtruth is completely recovered by one result, it's completeness will be 1. It will be zero if there is no intersection. In order to have a global measure of completeness, we then average over all the groundtruth activation.

If a result activation has a corresponding groundtruth activation, does the result exceed this groundtruth? To attest the quality of activation extraction, we define a complementary performance measure, called precision, in order to evaluate by how much the results are exceeding the groundtruth.

$$\text{Precision}(\text{res}_\text{j}) = \arg\max_{\text{i}}(\text{projection}(\text{grt}_\text{i} \rightarrow \text{res}_\text{j})) \tag{5}$$

The range of precision is [0, 1]. It is defined for each result that intersects with a groundtruth. If a result doesn't exceed the groundtruth, its precision will be 1. The more the result exceeds the groundtruth, the more its precision will tend to zero. In order to have a global measure of precision, we then average over all the result activations.

If the method recovers a ground truth activation, are there multiple result activations corresponding to the same ground-truth activation? The cardinality measure takes into account that an activation could correspond to multiple activations in the result.

$$\text{Cardinality}(\text{grt}_\text{i}) = \sum_{\text{j}} \text{intersect}(\text{res}_\text{j}, \text{grt}_\text{i}) \tag{6}$$

Cardinality is only defined for groundtruth activation retrieved in the result, thus its range is $[1, \infty]$. In order to have a global measure of cardinality, we then average over all the groundtruth activation.

Does my method find activation whereas there is no activation in the ground truth at this time? False alarm is a classical performance measure to consider activations in the result that do not correspond to any activation in the groundtruth.

$$\text{FalseAlarm} = \#\text{res} - \sum_{\text{res}_\text{j}} \text{intersect}(\text{res}_\text{j}, \text{grt}) \tag{7}$$

Figure 13 depicts the main cases of time series comparison. The case 1 is a perfect case thus all the performance measures are 1 except the false alarm at 0. The case 2 related an incompleteness of 80%, the case 3 an exceeding activation, the case 4, 6 and 7 are different cases of incomplete and exceeding activations. The case 5 shows an example of cardinality higher than 1. Finally, the case 8 describes a case of false alarm.

Cases						
	Groundtruth Results	Recovery	Complete	Exceed	Cardinality	False alarm
①		1	1	1	1	0
②		1	0.8	1	1	0
③		1	1	0.8	1	0
④		1	0.7	0.7	1	0
⑤		1	0.4	0.3 / 0.8	2	0
⑥		1/1	0.4/0.7	0.3	1	0
⑦		1/1	1/1	0.3	1	0
⑧		0	NA	NA	0	2

Figure 13: Possible cases of the results: the case 1 is a perfect case thus all the performance measures are 1 except the false alarm at 0. The case 2 related an incompleteness of 80%, the case 3 an exceeding activation, the case 4, 6 and 7 are different cases of incomplete and exceeding activations. The case 5 shows an example of cardinality higher than 1. Finally, the case 8 describes a case of false alarm.

5.4 Results

Figure 14 shows the performances measures (recover, completeness, precision, false alarm and cardinality) on both datasets of:

(1) get_Kelly: threshold method proposed by Kelly with its own parameters tuned on UK-DALE (in blue)
(2) get_our: threshold method proposed by Kelly with our own parameters tuned on REFIT (in orange)
(3) Cartesio (in green)
(4) ValmA (in red)
(5) SimBA with the cross-validation procedure for REFIT and REFIT trained barycenter for UK-DALE (in purple)

In general, all the methods perform quite well to recover the activations. The threshold method has high performance when tuned on the dataset but can exhibit flows when applied on an other dataset. For instance, our kettle parameters are really good for REFIT but not suited for UK-DALE on the contrary of Kelly's parameters. Appendix B gives more details about performances of threshold method and an extension with outlier removal. Cartesio, which is an extension of threshold method, has a better precision ratio for washing machine since it avoids double activation. However, it behaves poorly regarding the dishwasher, specifically on the completeness ratio and cardinality since it cuts the activation in two parts. For the rest of the appliances, its behaviour is really close to the threshold method. ValmA offers good results on most of the appliances. As expected, ValmA has difficulties to recover correctly the microwave and the kettle since they do not present clear redundant patterns. SimBA is less efficient than other methods. It can be explained by the fact that some appliance have really specific signatures and then the barycenter computed on other houses are not really adequate. A larger training dataset would probably improve SimBA results. Except for the washing machine and fridge, the results on UK-DALE are not really different from those on REFIT. One explanation could be that the two experiments were conducted in UK so the variability across houses from the same dataset is comparable to the one across datasets. Annexe C gives SimBA results with barycenter directly computed on the same house/dataset.

All methods have difficulties recovering microwave activation due to the high variability of the activations.

Regarding results across performances measures, we see that sometimes a tradeoff has to be chosen between them. Indeed an improvement of one measure can lead to an other degradation. This trade-off depends on the application. If we want to create a signature collection, we are interested in clean signatures so we might favour the completeness ratio and precision over recover ratio. If we want to tag start and end activation to train NILM algorithm, recover and false alarm are really important.

Figure 15 shows the running time comparison between the methods in log10. As expected given its simplicity, the threshold method is the fastest one. Cartesio follows closely (1 to ten times longer) behind the threshold method since both the computation complexity are linear. For SimBA running time evaluation, we separate results between the training step when barycenter are extracted and the test step when the similarity search is computed. The two steps have similar computing time and the procedure is approximately 100 time longer than threshold methods. Despite an optimized computation, ValmA is by far the slowest method (10^4,10^5 longer than threshold method). Appendix A discusses the effect of downsampling to accelerate the procedure. The ValmA 's running time for dishwasher and washing machine is significantly longer for UK-DALE compared to REFIT because the sampling is smaller and we extracted around 15/20 days for UK-DALE and 10 for REFIT.

5.5 Discussion

For a single appliance extraction activation, threshold methods seem appealing thanks to their fast computation times and relatively high accuracy. However, they are rather sensitive in their parameter tuning that requires a manual calibration. This also implies that they should be fine-tuned independently for each dataset, which may be cumbersome. Nevertheless, we observe that Cartesio achieves better accuracy than the threshold method for the special case of the washing machine appliance, and hence we recommend its use for this case. ValmA is a good alternative to the threshold method since it exhibits very good performance without the need to set (almost) any parameters. Therefore, it requires almost no human intervention. Its major drawback is its relatively high computation time, which could be reduced by using down-sampling, sliding window computations, or modern hardware (e.g., computations in SIMD, multi-core and GPU architectures). Although SimBA results are not very competitive to the other methods, we believe that they will improve when using larger datasets.

In this paper, we focused on the single appliance problem, but in the general case, we may have more than one appliances recorded in the same dataset. In this case, threshold methods will fail to detect and distinguish the various appliances, which will be mixed, especially if they have similar power and duration characteristics. If the two appliances do not activate simultaneously, both ValmA and SimBA would be able to recover their activations. If the two

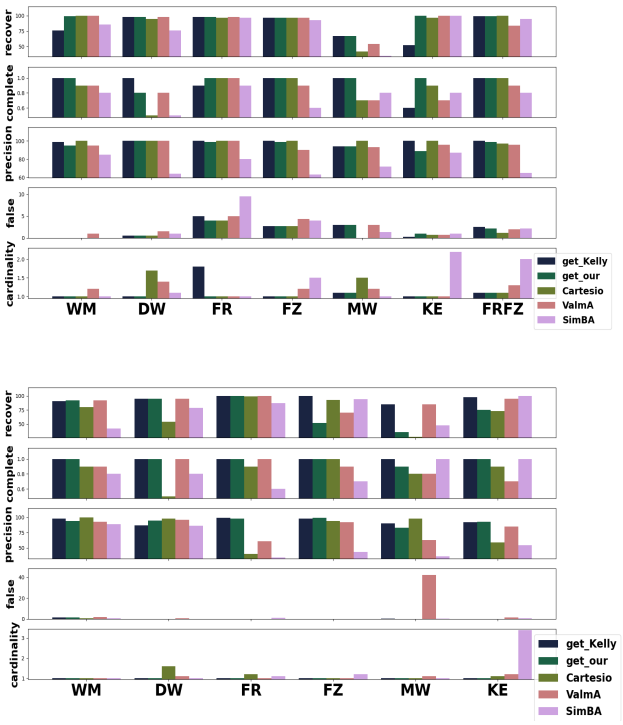

Figure 14: Accuracy comparison of activations extraction for REFIT (above) and UK-DALE (below)

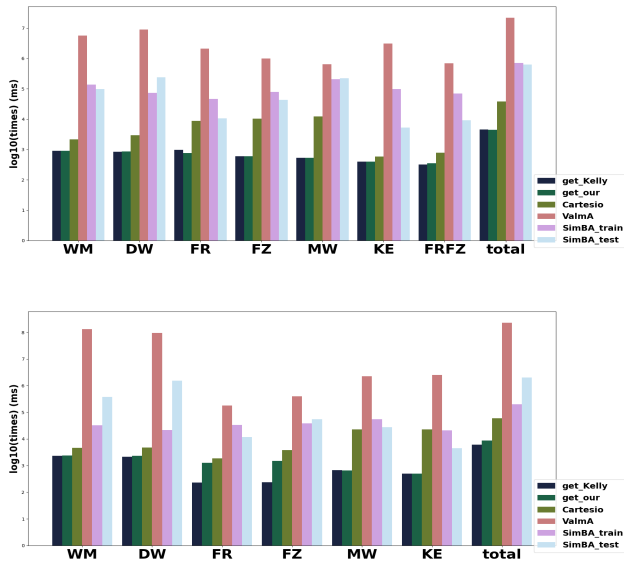

Figure 15: Running time comparison of activations extraction for REFIT (above) and UK-DALE (below)

appliances activate at the same time, only SimBA can potentially recover part of the activations.

In summary, our experimental analysis with the REFIT and UK-DALE demonstrates that:

- The state of the art threshold method exhibits high accuracy and fast computation, but requires a manually fine-tuned calibration on each dataset.
- The Cartesio method provides better results when the activations are close to each other, or for specific activation signatures, such as the washing machine.
- The k-NN similarity search method, SimBA, is suitable for activation extraction, provided we already have a set of example signatures.
- Finally, ValmA is an efficient method for extracting activations, and has very good accuracy and generalization abilities with (almost) no parameters. The biggest advantage of ValmA is that it can detect unknown signatures.

6 CONCLUSIONS AND FUTURE WORK

We developed solutions to the activation extraction problem based on detection of the start and end times, as well as on time series similarity search. To compare these methods, we also describe new accuracy measures that take into account the special characteristics of subsequences, leading to more precise performance evaluation results.

In our future work, we plan to evaluate the algorithms on additional datasets, for which we know the ground-truth, including data that involve more appliances (such as hotplate, oven, and others). We will also test the ValmA and SimBA methods on the problem of activation extraction from multi-appliances load curves. This would be very useful, since in sub-meters data collection, even if the aim is to measure only one appliance, technical problems appear and several appliances are often measured on the same sub-meter. Finally, we would like to use SimBA to extract activations from total load curve as in the NILM disaggregation problem to compare results with NILM literature.

ACKNOWLEDGMENTS

Work partially supported by the FMJH Program PGMO in conjunction with EDF-THALES.

A VALMA AND EFFECT OF DOWNSAMPLING

Since the computing time is the major drawback of the ValmA approach, we looked at the downsampling impact on both computing time and performance measures on REFIT dataset.

Figure 16 shows that downsampling by a factor of 2 (resp. 4) decreases approximately the running time by a factor 4 (resp 16). The total time for the test dataset downsampled by a factor 4 is around 20 minutes. It is an acceptable time but it is still much longer than threshold/Cartesio methods.

Figure 17 presents the impact of downsampling on performance measures. The downsampling does not seem to have an effect on fridge, freezer ad fridgefreezer activation extraction and has a slight effect on dishwasher and washing machine. However, the downsampling affects significantly the performance measures of the

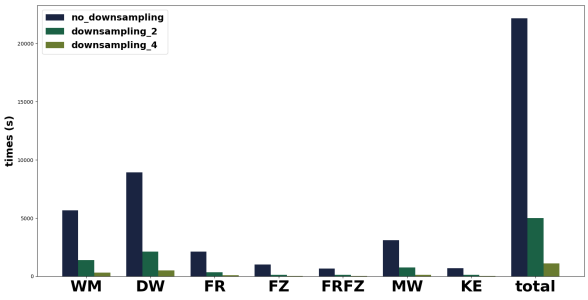

Figure 16: Effect of downsampling on running time

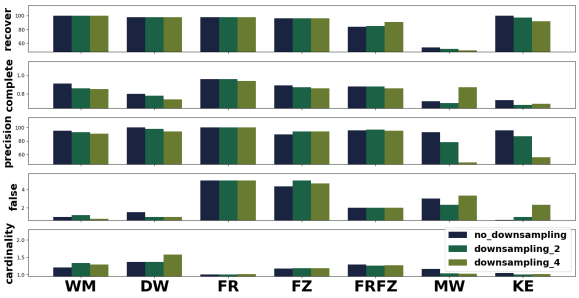

Figure 17: Effect of downsampling on performances measures

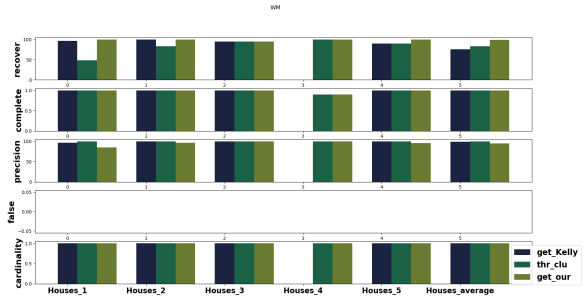

Figure 18: Performances measures on REFIT Washing Machine for get_Kelly, thr_clu and get_our

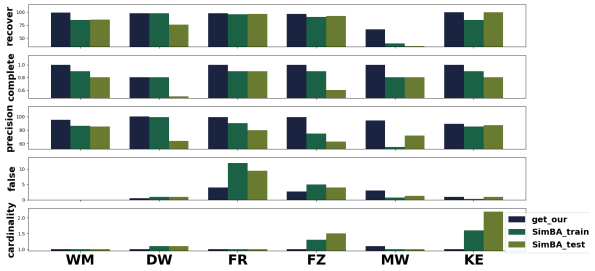

Figure 19: Performances measures on REFIT data for get_our (blue), get_activations_overfit (orange) and get_activations_test (green)

microwave and the kettle. These two appliances can indeed be really in a short time-frame and downsampling can suppress some important points.

B THRESHOLD METHODS AND EXTENSION WITH OUTLIER REMOVAL

We compare here three methods:

(1) get_Kelly: threshold method proposed by Kelly with its own parameters tuned on UK-DALE (in blue)
(2) thr_clus: threshold method with our own parameters followed by the clustering step of SimBA methods. We select only the activations in good clusters and then suppress outliers. (in orange)
(3) get_our: threshold method proposed by Kelly with our own parameters tuned on REFIT (in green)

Figure 18 shows performance results on REFIT washing machine on each houses. We can see that Kelly's parameters are more suited for House 1 than our parameters that lead to more double activation and thus bad precision. However, threshold method with Kelly's parameters fails to recover House 4's washing machine due to too small min_off_duration and too low max_power. This special case is an illustration on the high sensitivity of the parameters regarding the parameters.

In general, the threshold method with clustering has a poorer recovery since it deleted all the multiple activations than get_our but has a way better precision ratio. If you are interested in clean activations, you should favor thr_clus method and get_our if you are more interested in recovering the maximum of activations.

Please note that the thr_clus method can also be really useful to disentangle two appliances with similar parameters as washing machine and dishwasher. Indeed, it will create separate clusters between the two appliances.

C SIMBA: EFFECT OF TRAINING DATASET

In this appendix, we compare the results of SimBA when computing the barycenter on the same houses that we want to extract the signatures. We will call it «SimBA_overfit". The procedure by cross-validation as presented in the section 4.4 will be referred as "SimBA_test".

Figure 19 shows the comparison between the threshold method (blue), the get_activations_overfit (orange) and get_activations_test (green). As expected, the results when performing the similarity search from the barycenter extracted from the same appliance gives better results than when performing with an appliance of an other house. The performances of get_activations_overfit are not far from the one of the threshold method tuned on REFIT dataset.

REFERENCES

[1] Eamonn Keogh Abdullah Mueen. 2017. The First Matrix Profile Tutorial. The UCR Matrix Profile Page. https://www.cs.ucr.edu/~eamonn/MatrixProfile.html.

[2] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring. In *e-Energy*.

[3] Lina Stankovic David Murray. 2015. REFIT: Electrical Load Measurements. https://pureportal.strath.ac.uk/en/datasets/refit-electrical-load-measurements.

[4] Alexander Davis, Tamar Krishnamurti, Baruchm Fischhoff, and Wandi Bruine de Bruin. 2013. Setting a standard for electricity pilot studies. *Energy Policy* 62 (2013), 401–409.

[5] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2020. Big Sequence Management: on Scalability. In *IEEE BigData*.

[6] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2021. Big Sequence Management: Scaling up and Out. In *EDBT*.

[7] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2021. High-Dimensional Similarity Search for Scalable Data Science. In *ICDE*.

[8] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2021. New Trends in High-D Vector Similarity Search: AI-driven, Progressive, and Distributed. In *VLDB*.

[9] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2018. The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. *PVLDB* (2018).

[10] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2019. Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. *PVLDB 13(3)* (2019).

[11] George W. Hart. 1992. Nonintrusive Appliance Load Monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.

[12] Jack Kelly and William Knottenbelt. 2015. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In *BuildSys*.

[13] Jack Kelly and William Knottenbelt. 2015. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. 2, 150007 (2015).

[14] Michele Linardi and Themis Palpanas. 2018. Scalable, variable-length similarity search in data series: The ULISSE approach. *PVLDB* 11, 13 (2018), 2236–2248.

[15] Michele Linardi and Themis Palpanas. 2020. Scalable data series subsequence matching with ULISSE. *VLDB J.* 29, 6 (2020).

[16] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. 2018. Matrix Profile X: VALMOD - Scalable Discovery of Variable-Length Motifs in Data Series. In *SIGMOD*.

[17] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. 2018. VALMOD: A Suite for Easy and Exact Detection of Variable Length Motifs in Data Series. In *SIGMOD*.

[18] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn J. Keogh. 2020. Matrix profile goes MAD: variable-length motif and discord discovery in data series. *Data Min. Knowl. Discov.* 34, 4 (2020), 1022–1071.

[19] Themis Palpanas. 2015. Data Series Management: The Road to Big Sequence Analytics. *SIGMOD Record* (2015).

[20] Themis Palpanas. 2020. Evolution of a Data Series Index. *Communications in Computer and Information Science (CCIS)* (2020).

[21] Themis Palpanas and Volker Beckmann. 2019. Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). *SIGMOD Rec.* 48, 3 (2019).

[22] Lucas Pereira and Nuno Nunes. 2018. Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 6 (2018), 1–17.

[23] Quentin Reynaud, Yvon Haradji, François Sempé, and Nicolas Sabouret. 2017. Using Time Use Surveys in Multi Agent based Simulations of Human Activity. In *ICAART*.

[24] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. 2018. Precision and recall for time series. *NeurIPS* (2018).