Andreas Reinhardt and Christoph Klemenjak. "How does Load Disaggregation Performance Depend on Data Characteristics? Insights from a Benchmarking Study." In: *Proceedings of the 11th ACM International Conference on Future Energy Systems (e-Energy)*. 2020, pp. 167–177.

How does Load Disaggregation Performance Depend on Data Characteristics? Insights from a Benchmarking Study

Andreas Reinhardt
Department of Informatics
TU Clausthal, Germany
reinhardt@ieee.org

Christoph Klemenjak
Institute of Networked and Embedded Systems
University of Klagenfurt, Austria
klemenjak@ieee.org

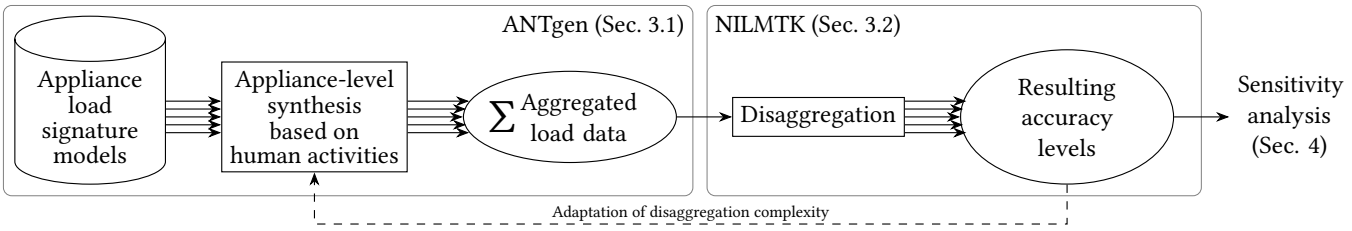


Figure 1: Evaluation cycle used to quantify the impact of aggregate trace characteristics on the achievable NILM performance.

ABSTRACT

Electrical consumption data contain a wealth of information, and their collection at scale is facilitated by the deployment of smart meters. Data collected this way is an aggregation of the power demands of all appliances within a building, hence inferences on the operation of individual devices cannot be drawn directly. By using methods to *disaggregate* data collected from a single measurement location, however, appliance-level detail can often be reconstructed. A major impediment to the improvement of such disaggregation algorithms lies in the way they are evaluated so far: Their performance is generally assessed using a small number of publicly available electricity consumption data sets recorded from actual buildings. As a result, algorithm parameters are often tuned to produce optimal results for the used data sets, but do not necessarily generalize to different input data well. We propose to break this tradition by presenting a toolchain to create synthetic benchmarking data sets for the evaluation of disaggregation performance in this work. Generated synthetic data with a configurable amount of concurrent appliance activity is subsequently used to comparatively evaluate eight existing disaggregation algorithms. This way, we not only create a baseline for the comparison of newly developed disaggregation methods, but also point out the data characteristics that pose challenges for the state-of-the-art.

CCS CONCEPTS

• **Hardware** → **Smart grid**; • **Computing methodologies** → **Machine learning**.

e-Energy'20, June 22–26, 2020, Virtual Event, Australia

ACM ISBN 978-1-4503-8009-6/20/06…$15.00
https://doi.org/10.1145/3396851.3397691

KEYWORDS

non-intrusive load monitoring, load disaggregation, performance evaluation, data set characteristics, synthetic load signatures, NILM benchmarking

ACM Reference Format:
Andreas Reinhardt and Christoph Klemenjak. 2020. How does Load Disaggregation Performance Depend on Data Characteristics? Insights from a Benchmarking Study. In *The Eleventh ACM International Conference on Future Energy Systems (e-Energy'20), June 22–26, 2020, Virtual Event, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3396851.3397691

1 INTRODUCTION

Non-Intrusive Load Monitoring (NILM), or "load disaggregation" for short, is a technique to identify individual contributing appliances from the total electrical power consumption of a building. Since the initial presentation of this concept in [11], numerous authors have made algorithmic contributions to disaggregate data from private homes as well as commercial and industrial facilities (e.g., [15, 20, 28, 38]). An important aspect during the development of disaggregation algorithms is to ensure their applicability in practical settings. Algorithms are hence often trained and tested on data collected in real-world campaigns because only such data inherently feature the full spectrum of characteristics present in reality. This step is facilitated by the availability of data sets that contain time series data of a building's total consumption, as well as data on the electricity demand of the individual devices present within. Thus, these data sets provide the possibility to verify the correct attribution of disaggregated electrical power to each appliance.

Besides the fact that only a small number of such data sets have been publicly released, the evaluations of most proposed NILM algorithms are confined to the use of a single data set only. Thus, it generally cannot be ascertained that the achieved disaggregation results can be generalized. In fact, as soon as an algorithm's parameter settings are optimized for one specific building, it is not unlikely to observe degraded accuracy levels for other nearby houses (cf. Sec. 2.1). This is particularly aggravated when the data sets originate from different geographical areas or different levels of

Table 1: Disaggregation performance results when running eight disaggregation methods (cf. Sec. 4.1 for details) on data from five data sets for load disaggregation. Results for the best-performing disaggregation algorithms are highlighted in bold font.

	MAE by disaggregation algorithm								F1 score by disaggregation algorithm							
Data set (house)	CO	DSC	Hart85	FHMM	DAE	RNN	S2S	S2P	CO	DSC	Hart85	FHMM	DAE	RNN	S2S	S2P
DRED (1)	52.86	41.55	34.36	44.44	**32.65**	33.12	42.05	33.35	0.50	0.49	0.40	0.49	0.48	0.47	**0.51**	0.47
ECO (5)	134.12	94.16	43.88	81.75	37.63	32.11	30.47	**28.43**	0.47	0.46	0.50	0.47	0.55	0.61	0.56	**0.64**
ECO (6)	86.67	40.28	30.20	42.86	17.26	**6.09**	17.33	17.30	0.32	0.36	0.21	0.32	0.32	**0.74**	0.31	0.32
REDD (1)	230.88	163.34	30.93	138.66	34.56	34.30	**25.75**	31.72	0.41	0.44	**0.77**	0.42	0.66	0.63	0.67	0.65
REDD (6)	248.70	257.67	32.54	70.72	**22.11**	45.83	46.11	45.53	0.71	0.71	0.80	0.71	0.71	**0.72**	0.71	0.71
REFIT (2)	72.25	104.04	31.55	72.25	36.08	**31.50**	52.80	51.28	0.52	0.51	0.53	0.52	0.59	**0.61**	0.52	0.52
REFIT (5)	311.48	139.40	86.50	66.04	34.63	52.05	**28.19**	35.44	0.67	0.66	0.50	0.67	0.75	0.68	0.73	**0.76**
UK-DALE (1)	140.78	44.99	21.98	51.96	16.10	16.52	15.67	**13.80**	0.65	0.55	**0.91**	0.65	0.89	0.90	0.87	0.90
UK-DALE (5)	321.08	68.80	33.53	64.61	25.08	29.67	**24.16**	24.21	0.59	0.52	0.70	0.59	0.80	0.77	0.80	**0.81**

building occupancy [6]. An intuitive way to overcome this obstacle, and a method widely used in other disciplines, would be to release a benchmarking data set that features sufficient training data from diverse locations in a data format that can be easily interpreted. The compilation of a representative data set, however, is complicated by the licensing requirements of existing data sets and the unavailability of data from many regions around the globe.

We thus follow an alternative approach in this work. Instead of attempting to compile a benchmarking corpus from existing data sets, we present a methodological way to synthetically create data sets of definable disaggregation complexity. A high degree of realism can be accomplished by using accurate models of existing appliances and user activities. By forwarding synthetically generated data of gradually increasing levels of concurrent appliance activity to state-of-the-art disaggregation algorithms, we determine their sensitivity to specific data characteristics in a much more fine-grained way. The key contributions of this work are:

(1) We present a toolchain, *ANTgen*, to synthetically generate load signature data (both the household aggregate and individual data for all appliances) to synthesize data with a highly realistic appearance while permitting fine-grained control over the contained appliances and activities.
(2) We *benchmark* the performance of eight disaggregation algorithms when applied to generated synthetic data. By iteratively increasing the complexity of the used input data, we methodologically determine the characteristic features that complicate the disaggregation step.
(3) We discuss and interpret the results and *highlight the limitations* of current disaggregation algorithms that necessitate more consideration in future work.

2 DATA SET DEPENDENCIES OF EXISTING LOAD DISAGGREGATION ALGORITHMS

Recent surveys [35, 40, 41] indicate that close to one hundred approaches to tackle the load disaggregation challenge have been proposed in literature to date. An objective way to evaluate their accuracy is strongly needed to find the best-performing candidates and identify promising avenues for future research. Many of the publications rely on real-world data sets (e.g., REDD [21] or UK-DALE [16]) for this purpose, as they inherently reflect the consumption characteristics of actual users in real buildings. Only by providing NILM mechanisms with the same input data set (or excerpts thereof), their disaggregation performance can be evaluated in a *comparative* manner.

2.1 Disaggregating Existing Data Sets

We demonstrate the variations when running NILM algorithms on existing data sets through a practical experiment. For this purpose, we run eight disaggregation methods that are part of the NILMTK [3] on data from five widely used data sets. Our choice of data sets is governed by the requirement of them containing both aggregate consumption data (i.e., the household total) as well as values for the individual electrical appliances, in order to verify the correct attribution of power usage. As a result, we use two monitored houses each from ECO [4], REDD [21], REFIT [26], and UK-DALE [16], as well as the DRED data set [37], which only covers one house. Our choice of disaggregation algorithms is motivated by their availability in NILMTK as well as the diversity in their fundamental mode of operation; more detailed explanations of the algorithms are given in Sec. 4.1. The default parameter settings were used for all employed algorithms, and methods based on neural networks were trained with a batch size of 128 and a sampling interval of 10 s for a duration of 10 epochs. For data sets that contain more than 150 days worth of data, 150 consecutive days were randomly selected for analysis; smaller data sets were used completely. In each case, 70 % of the input data were used for training, and the disaggregation performance was evaluated on the remaining 30 %.

As our objective is not to determine the single best disaggregation scheme, but rather to highlight the variations in the results they accomplish, we confine our evaluation results to two evaluation metrics, namely the MAE and the F1 score (cf. Sec. 4.1.2). While an F1 score of 1.0 indicates a perfect disaggregation result, lower values indicate that not all activities were detected correctly. In contrast to this, a low MAE value indicates a small deviation between the disaggregated data and is thus an indicator for a good disaggregation accuracy. Results for the disaggregation of a refrigerator appliance are given in Table 1. The refrigerator has been

specifically selected to allow for a comparability across the data sets because it is present in all of the monitored houses. The tabulated results clearly indicate that none of the compared NILM algorithms universally achieves the best disaggregation result. Quite to the contrary, algorithms that excel in one case (e.g., S2P applied to house 1 of UK-DALE) exhibit poor results when used to disaggregate data from another source (e.g., house 2 of REFIT). Furthermore, even different houses within the same data set partially exhibit significant differences with respect to the extent to which the refrigerator's power demand can be correctly disaggregated. In some cases, we even observe that the CO algorithm (which dates back to 1985) shows comparable disaggregation performance to RNNs, even though the latter approach has been published 30 years later.

2.2 Implications on Disaggregation Algorithm Development

The insights gained in our brief preliminary study already show the variations in disaggregation performance, even though an appliance type was used that is often considered "easy to estimate" [10, 24]. Furthermore, the data sets we have used were collected in geographic areas with comparable climate conditions and economic situations (DRED: The Netherlands, ECO: Switzerland, REDD: United States, REFIT: Scotland, UK-DALE: England). Consequently, even more heterogeneous results can be anticipated when trying to disaggregate data from different regions [2], e.g., Africa or Asia. The unavailability of data from many geographic areas, however, represents a major obstacle to the comprehensive evaluation of newly proposed disaggregation algorithms. In fact, using the current approach, i.e., confining performance evaluations to limited input data sets, it is impossible to deduce that attained results will also *generalize* to other buildings, cities, countries, or even continents.

For an objective evaluation of an algorithm's disaggregation performance, we thus argue that it is necessary to supply it with input traces of variable disaggregation complexity, featuring the specific features of the targeted geographic area. Moreover, in order to allow the creators of novel disaggregation techniques to compare the performance of their approaches to the state of the art, a compilation of such input traces should be available, ideally in the form of a benchmark data set. By applying different algorithms to the same set of data, their performance differences (and possibly even the underlying reasons) can be determined in much greater detail, and the design of new solutions and/or the improvement of existing algorithms be fostered.

3 TOWARDS *COMPARABLE* PERFORMANCE EVALUATIONS OF NILM ALGORITHMS

To accomplish an automated and comparable evaluation of NILM disaggregation performance, we propose to use the data processing toolchain depicted in Fig. 1 in this paper. On its left-hand side, the part to create synthetic load signatures at configurable disaggregation complexity is shown. More details about its design and selected implementation details are provided in Sec. 3.1. The right-hand side part of the figure shows our evaluation setup. To accommodate the flexible extensibility of our toolchain, we rely on NILMTK [3]. This open-source framework combines implementations of several NILM algorithms in conjunction with a range of evaluation metrics, and can be easily extended by new algorithms. More details on its usage within the scope of our evaluation setup are given in Sec. 3.2. The presented toolchain is used to synthetically generate data for the benchmarking of NILM algorithms, which we describe in Sec. 4.

3.1 Synthetic Load Signature Generation

In order to assess the sensitivity of disaggregation algorithms to the characteristics of load signatures, a way to provide them with input data of varying complexity is required. Publicly available data sets, collected by means of real-world sensor deployments, thus take an important role in the evaluation of NILM algorithms and to support their general operability (cf. Sec. 2.1). The limitations of real-world data sets (i.e., their confinement to specific geographic areas and particular demographics) [18], however, limit the extent to which they can be used to draw generalizable conclusions. This is a severe impediment, given that intentional variations of the data to disaggregate are required in order to judge an algorithm's sensitivity to specific data characteristics.

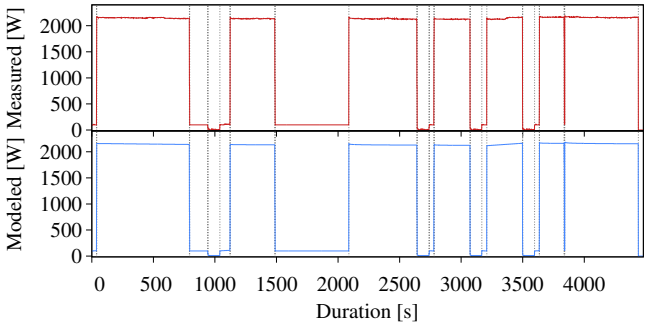
To overcome this limitation of existing data sets, we have developed ANTgen (the "**A**MBAL-based **N**ILMTK **T**race **gen**erator")[1], a tool to facilitate the generation of electrical consumption data at definable levels of disaggregation complexity. The realistic appearance of ANTgen's output data is vital to allow for the generalization of observations to data collected in real-world smart meter deployments, and specifically catered for by three mechanisms.

(1) ANTgen relies on appliance load signatures (i.e., time series of an appliance's power intake over time) from existing data sets. By transforming them into synthesizable models, generated load data implicitly reflects the consumption characteristics observed from the operation of actual appliances.
(2) ANTgen schedules the operation of appliances according to models of daily activities, which define the operation of a set of appliances in a given order. This way, the order of appliance operation in synthetic data resembles reality (i.e., a meal is first cooked, then eaten).
(3) User models are employed that determine what activities will be present in the data, during which times they can be taking place, as well as the frequency of their occurrence. Data complexity can thus be easily increased by adding more (virtual) members to a synthetic household.

All of the aforementioned concepts are presented in more detail in the following subsections.

3.1.1 Appliance models. In order to generate aggregated load signatures with realistic appearances, the electrical power consumptions of all contributing appliances must closely resemble reality. We hence build on existing concepts for synthetic appliance modeling [5, 7] in order to extract of appliance models from appliance-level load signature data. In preparation for their modeling, the operational cycles of individual appliances are extracted, and all periods of appliance inactivity removed. The operational cycles identified in the previous step are then divided into segments, each of which is approximated separately by one of four models: (1) A constant value, (2) a linear change of the power from a given start value to an end value, (3) a logarithmic growth or decay function,

[1] Available at https://github.com/areinhardt/antgen



(a) Power consumption during one operation cycle extracted from input data (top) vs. its modeled representation (bottom). The dashed vertical lines depict the change-points, in-between which individual models are being used.

t_{start}	t_{end}	Power consumption model ($\Delta t = t - t_{start}$)
0	38	$P(t) = 82.7 + 2.4778 * log(80.834 * (\Delta t + 1))$
39	39	$P(t) = 1039.0$
40	792	$P(t) = 2158.774 - 0.0243 * \Delta t$
793	794	$P(t) = 859.0$
795	941	$P(t) = 99.606 - 0.002 * \Delta t$
942	943	$P(t) = 64.0$
944	1039	$P(t) = 9.748 + 4.0925 * e^{(0.198 * \Delta t)}$
1040	1122	$P(t) = 100.734 + 0.1237 * \Delta t$
…	…	…

(b) Excerpt of the resulting approximation model (unit of time: seconds).

Figure 2: Example result of the model extraction step for a dishwasher's load signature.

or (4) an exponential growth or decay function. A graphical user interface is provided for the model extraction step, in order to verify and optionally amend the choice of the parameter settings. Given that modeling only needs to be executed once, yet poorly fitting models will render the synthetic data unusable, we believe that it is meaningful to leverage expert knowledge at this stage. An XML-based model representation is stored after completion of the model extraction step for each modeled appliance load signature. This approach allows us to combine the advantages of real-world data (to ensure that synthetically generated traces have a realistic appearance) with those of generative models (being able to schedule them as often as needed).

The measured power consumption of a dishwasher appliance (from Tracebase [32]) as well as its modeled approximation are shown in Fig. 2a, and a tabular representation of the first segments of the output model is given in Fig. 2b. Without loss of generality, we source appliance-level input data from the ECO [4] and Tracebase [32] data sets. The extraction of appliance models from other data sets is easily possible, as long as appliance-level load signatures are available. Our decision to rely on data sets collected in the same region (ECO: Switzerland, Tracebase: Germany), however, ensures a greater degree of realism in the synthetic data because the contained electrical appliances are likely to originate from the European Single Market. More specifically, the appliance models listed in Table 2 are used throughout the tests we conduct in Sec. 4.

Table 2: List of appliances for which ANTgen models have been extracted from the Tracebase and ECO data sets.

Category	Appliances
kitchen	refrigerator, dishwasher, toaster, microwave oven, kettle, stove, coffee maker, bread cutter
housekeeping	washing machine, vacuum cleaner
entertainment	PC, TV, CD player, amplifier

3.1.2 Activity models. The second type of model required to generate load signatures with a realistic appearance is the consideration of user behavior. In this work, we assume that users have specific routines for certain actions; a widely used assumption in research on Activities of Daily Living (ADL), such as [8, 25]. For example, the preparation of a breakfast meal might include the operation of a stove, water kettle, and toaster in a given order and for a given duration. For the sake of simplicity, we model user activities in the form of state machines. By annotating the state transitions with probabilities, our system accommodates variations in the order (e.g., first operate the toaster, then the kettle, or vice versa) as well as to make certain steps optional (e.g., skipping the use of the stove for breakfast altogether). Activities thus govern the use of appliance models over time, as well as the (temporal and logical) dependencies between them. A sample application state machine for the "breakfast" activity is given in Fig. 3, whilst the state machine modeling the cyclic operation of a refrigerator is given in Fig. 4.

Besides the definition of their logical sequence of steps (and the appliances required to execute them), activities also feature annotations whether the user's presence is required during the entire activity (e.g., having breakfast), only during its initial phase (e.g., to start the washing machine), or if appliances run fully unattended (e.g., a refrigerator). Our definition language facilitates the provision of such annotations in order to cater for the correct synthesis of load profiles. Like for the extraction of appliance models in Sec. 3.1.1, the definition of activity models is a manual step and requires expert knowledge to ensure their realistic appearance.

3.1.3 User models. As a third and final component of our synthetic load signature generation tool, we model the behavior of users by creating a mapping between their physical presence in the building, the activities they can perform during this time, as well as the number of times a certain activity is performed per day on average. The independent modeling of users allows for the synthesis of data with a low degree of concurrency (such as exhibited by rental apartments inhabited by single persons) up to multiple parallel appliance operations, like in multigenerational homes. User-independent base loads, such as exhibited by refrigeration equipment and the standby power consumption of appliances, are modeled in the same fashion, yet simply not attributed to a particular user identity.

3.1.4 Trace synthesis. The logical connections between users, activities, and corresponding appliance operations are established by ANTgen. Scheduling user-driven activities in a certain time frame requires three conditions to be met: (1) The user's presence at home, (2) the user's ability to execute a certain activity during that time, and (3) the availability of the appliances required to this end (i.e.,

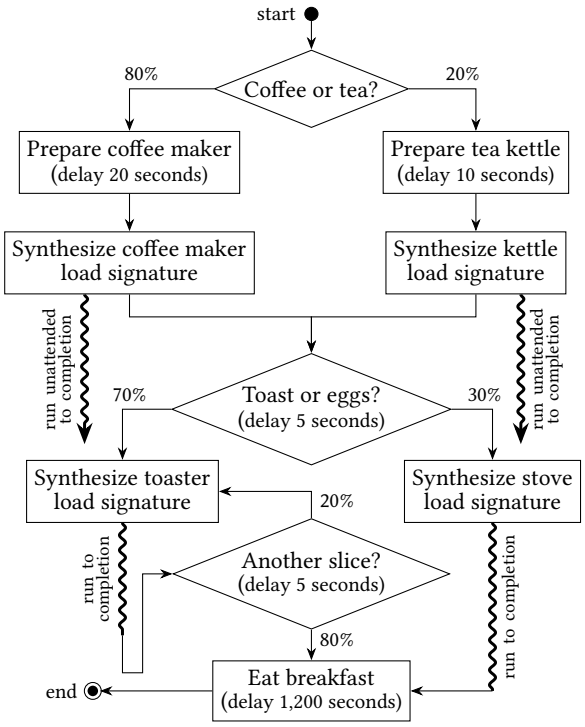


Figure 3: Sample activity model for the "preparing breakfast" activity. Curly lines indicate the synthesis of appliance load signatures into the output data. This activity requires the user to be present in the building during its execution.

they must not occupied by any inhabitant of the building simultaneously). ANTgen stores all of these conditions in the form of bitmaps, and applies a logical conjunction of these bitmaps to determine time frames during which activities can place. Activities (and the ensuing appliance operations) are then scheduled at random within these time frames, permitted that the duration of the activity fits into the available time. For activities that rely on the unattended operation of appliances, the user presence is disregarded when finding suitable operational times.

A simple case reflecting the following story is visualized in Fig. 5, where the highlighted fields indicate that a given condition is true. User Alice wants to have a cup of tea, for which she needs to operate the water kettle (the *appliance*). Given that she shares an apartment with Bob, the kettle is not available for a part of the day during which Bob does his cooking. Alice only takes tea during certain hours (the late morning and early afternoon), which limits the times during which the *activity* can take place. Lastly, she (the *user*) leaves the apartment for a while, and naturally cannot prepare tea during her absence. Through the logical conjunction of these conditions, only a small number of possible time frames remain during which she can perform this activity. Whether or not ANTgen schedules the activity in one of them finally depends on the probability of having tea on a given day. In the example case, we assume that Alice has tea once every day, thus ANTgen randomly schedules the activity in one of the possible time slots (highlighted in the figure),

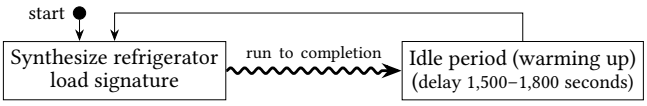


Figure 4: Sample activity model for a refrigerator's operation. This appliance runs unattended and uninfluenced of the user, and does not necessitate the user's presence in the building during its operation.

during which it subsequently marks the user and the appliance busy in the corresponding bitmaps.

In ANTgen, the temporal resolution at which data are being generated is 1 s, and thus each bit in the bitmaps corresponds to one second of synthetic data. The repeatability of experiments is supported in ANTgen through the possibility to seed its random number generator by a known value. Being able to completely replicate experimental runs is a crucial prerequisite for the generation of benchmarking data sets. Alternatively, a pseudo-random seed value can be used to generate synthetic data with the same users, activities, and appliances, yet different schedules, in order to augment the amount of training data available to a disaggregation algorithm.

3.2 Trace Disaggregation using NILMTK

The key contribution of this paper is a comprehensive evaluation and sensitivity analysis of existing NILM algorithms. Two complementary contributions are required in order to this end:

(1) Input data that features both the building aggregate trace as well as the load signatures of all appliances (the *ground truth*). As ANTgen computes the aggregate data by summing up the load signatures of individual appliances, this prerequisite is easily met when using ANTgen.
(2) A disaggregation tool that reconstructs individual load signatures from an aggregate input signal it is provided with. Through the comparison of the disaggregated appliance-level data with the ground truth, the disaggregation accuracy can be evaluated.

We rely on the set on disaggregation algorithm implementations in NILMTK [3], which we discuss in more detail in Sec. 4.1. NILMTK expects input data to be stored in the HDF5 file format, with annotations compliant to the NILM Metadata scheme[2]. We thus convert ANTgen's output to the HDF5 format used in NILMTK, and auto-generate the required metadata annotations to facilitate their import, based on the used appliance models. As ANTgen outputs both individual power consumption traces for each appliance under consideration and an aggregate value that simulates the power demand recorded at the smart meter, all essential data required to run and evaluate NILM algorithms are provided. The entire data generation and processing chain (cf. Fig. 1) allows us to run evaluations in an automated fashion. At the same time, changes to ANTgen's configuration (e.g., adding the model of another user to the simulated building) allow for the deliberate variation of its output data.

[2] Available at https://github.com/nilmtk/nilm_metadata

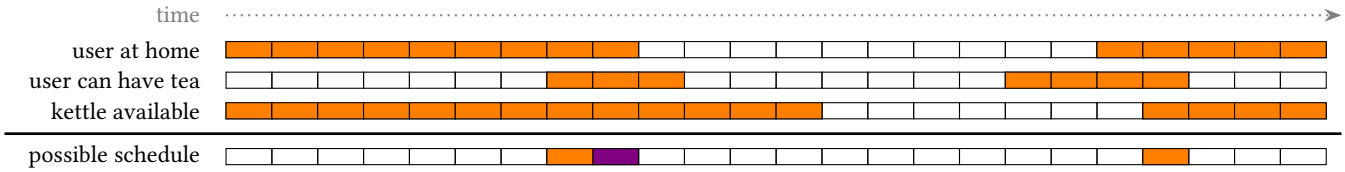


Figure 5: ANTgen internally relies on bitmaps to indicate user presence, activity times, and appliance usage. The logical conjunction of each bit is used to determine possible time slots at which an activity can take place.

4 METHODOLOGICAL NILM EVALUATION

An intuitive expectation to the disaggregation performance is its dependency on the extent of concurrent appliance activity in the underlying data set. While the sole presence of a single appliance is expected to be easily detectable at high accuracy, more variations in the load signatures as well as more simultaneous appliance activities are expected to complicate the disaggregation process significantly. A methodological evaluation of the sensitivity of a set of NILM algorithms to such occurrences is thus needed, in order to get a better understanding of which features have an impact on the disaggregation performance. We conduct an extensive study in this section, and discuss the gained insights in Sec. 4.3.

4.1 NILM Algorithms and Evaluation Metrics

4.1.1 Disaggregation algorithms. Within the scope of our evaluations, we rely on the following eight disaggregation algorithms that are implemented as part of NILMTK. Relying on this many algorithms is essential to meet our objective of providing a comparative performance study.

(1) *Combinatorial Optimization (CO)*, introduced in [11] first tries to determine the power demand of each appliance for each of its modes of operation. Then, the goal of CO is to identify the subset of concurrently active appliances such that the difference between the measured aggregate and the sum of individual appliances' power intakes is minimal.
(2) *Discriminative Sparse Coding (DSC)* [19] first trains models that reflect each considered appliance's hourly energy demand. Subsequently, an "activation schedule" for each of the appliances is determined, which minimizes the error between the modeled consumption and the observed energy demand. This optimization step is accelerated by applying sparsity constraints on the model activations.
(3) *Edge Detection (Hart85)* [11] divides an input time series data into periods of *steady* and *transient* power demand. After each *transient* period, the magnitude differences between its surrounding *steady* states are correlated with appliance state changes, and each appliance's operating state is updated correspondingly.
(4) An *Exact Factorial Hidden Markov Model (FHMM)* represents each appliance by a hidden Markov model, composed of states (with steady power demand) and the transitions between them. Through extracting the mean power demand for each state as well as the probabilities for state transitions, FHMMs attribute observed power data to the appliances whose models allow to fit the shape of the load signature.
(5) Autoencoders are symmetrically designed neural networks trained to closely reproduce their input at their output, while using a sparse internal representation [13]. The *Denoising Autoencoder (DAE)* considers the aggregate power signal as a noisy representation of an appliance's power signal, and uses the autoencoder to eliminate the contributions of other appliances from the signal [15, 22].
(6) The use of *Recurrent Neural Networks* (RNNs) for load disaggregation was proposed in [15]. A neural network with memory cells is trained to recognize the load signatures of individual appliances within aggregate data. After its training, the network outputs an updated estimate of the set of operating appliances for each newly received input sample.
(7) *Sequence-to-Sequence Optimization (S2S)* also uses neural networks, yet without memory elements [39]. Instead, sliding windows across the aggregate input data are mapped to power consumption segments of appliance loads. This way, previously observed patterns in aggregate load data can be matched to the trained characteristics of individual devices.
(8) The *Sequence-to-Point Optimization (S2P)* technique is a variation of S2S and also relies on neural networks, yet instead of outputting a sequence of power consumption values for an appliance's disaggregated power demand, it is specifically trained to only output the value at the midpoint of a time window [39].

Three more algorithms are available in NILMTK, but have been disregarded from our analysis because they either reported consistently poor results (which was the case for the simple mapping of the mean power consumption to the set of contributing appliances) or were too slow to complete our experiments in reasonable time (WindowGRU [22] and AFHMM [20]). Even though they have not been considered in this work, a comparative evaluation can be conducted at a later stage using the same synthetic data.

4.1.2 Evaluation metrics. To measure the classification performance of the considered algorithms, we compute two metrics that describe how well the disaggregated data for given appliances line up with the ground truth. Our first metric, the Mean Average Error (MAE), is computed according to Eq. (1).

$$\mathrm{MAE}^{(i)} = \frac{1}{T} \cdot \sum_{t=0}^{T-1} |\hat{x}_t^{(i)} - x_t^{(i)}| \tag{1}$$

In the equation, x_t and $\hat{x}_t$ are the actual and disaggregated power demands of appliance i, respectively, and T represents the number of samples in the data. The MAE is an absolute measure of the error between actual and disaggregated appliance power demand.

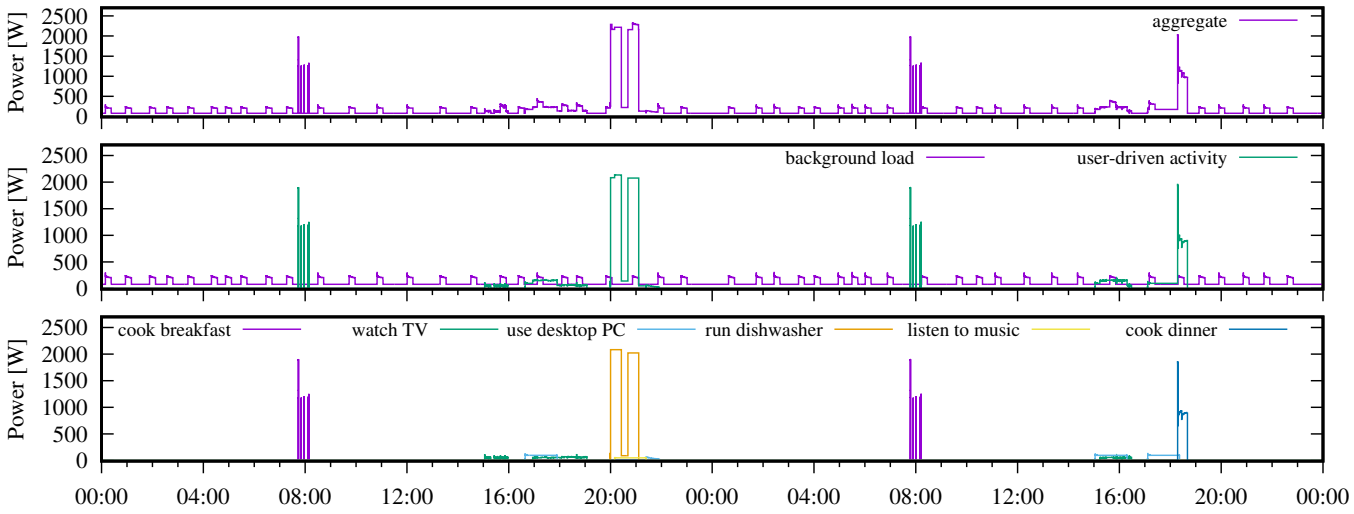


Figure 6: Excerpt of the input data set TC4-N2 (1 simulated user, 80 W additional constant background load) during the course of two days. The top graph shows the total aggregate consumption, the middle one is disaggregated by the nature of power consumption, whereas the bottom diagram provides a separation by each user-driven action.

Thus, it implicitly penalizes the mis-detection of appliances with a greater energy demand. The second metric considered is the F1 score (F-measure), computed as per Eq. (2).

$$\mathrm{F1} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{2}$$

TP represents the number of true positive disaggregation results, FP the false positive detections, and FN the false negatives. In order to compute this metric, appliance operations are discretized to a binary form (active/inactive) for each time interval. The sums of all occurrences are then computed as follows:

$x_t^{(i)}$ = active, $\hat{x}_t^{(i)}$ = active → true positive (TP)
$x_t^{(i)}$ = inactive, $\hat{x}_t^{(i)}$ = active → false positive (FP)
$x_t^{(i)}$ = active, $\hat{x}_t^{(i)}$ = inactive → false negative (FN)

The F1 score has been specifically selected because it excludes true negative results (i.e., the correct detection of an inoperative appliance), given the large number of such occurrences in load signature data. The chosen set of metrics is also well aligned with the proposals presented in other studies on NILM, such as [23, 30], which recommend the use of metrics from at least two categories when evaluating disaggregation performance.

4.1.3 NILMTK settings. We rely on the latest version of NILMTK at the time of writing (`v0.4.0dev1`). Implementations of all disaggregation algorithms were taken the *NILMTK-contrib* repository[3] and used with their default set of options (cf. [3]). Besides resampling ANTgen's output data to a uniform sampling interval of 10 s to cater for a fair evaluation, no further data preprocssing has been applied. All neural networks are trained for 10 epochs to prevent overfitting; for DSC, 100 iterations are executed per training step. These training specifications are motivated by our objective to explore characteristics of existing NILM algorithms in a broad range of cases, rather than optimizing them for their best performance.

[3] Available at https://github.com/nilmtk/nilmtk-contrib

4.2 Input Data

In order to comprehensively evaluate NILM performance, input data of variable complexity are required. For a fair comparison, we consistently created 180 days of synthetic input data, out of which 126 days (70 %) are provided to the algorithms as training data; the remaining 54 days (30 %) are consequently used to test the algorithms. An overview of the generated data sets and their characteristics is given in Table 3, which lists the number of (virtual) users and appliances present in a building, as well as the maximum number of concurrently active appliances. The modeled activities include the different ways to prepare food (for breakfast, lunch, and dinner), entertainment activities (using a PC, watching TV, listening to music) and household chores (running the dishwasher, laundry washing, and vacuuming). We have plotted the first two days of the TC4 data set in Fig. 6 for visual reference. The total aggregate (shown in the top figure) is composed of traces that result from user activities (bottom) figure.

Orthogonal to the addition of appliances and their concurrent operations, we also run five variants of each data set through the disaggregation algorithms:

- N0 The synthetic data that are simply an addition of all individual appliance load signatures.
- N1 A constant load of 20 W is added to the aggregate trace, in order to simulate a small invariable amount of standby loads.
- N2 A constant load of 80 W is added to the aggregate trace, in order to simulate a greater contribution of standby loads.
- N3 Gaussian noise with a mean of 20 W and a standard deviation of 10 W is added to the aggregate trace, in order to simulate a small amount of distortion.
- N4 Gaussian noise with a mean of 80 W and a standard deviation of 20 W is added to the aggregate trace, in order to simulate a greater amount of distortion.

Table 3: Synthetic data sets created in this paper to evaluate the performance of NILM algorithms.

Test case	# users	# appliances	Max. concurrently active
TC1	0	1	1
TC2	1	7	3
TC3	1	8	5
TC4	1	14	5
TC5	2	8	8
TC6	2	14	9

4.3 Insights from our Comparative Study

We have run all variants of the synthetically generated data sets through NILMTK, using 8,400 evaluation runs (6 test cases × 5 noise levels × 8 disaggregation algorithms × 1–7 appliances under consideration). We summarize the key insights as follows:

4.3.1 The refrigerator is (indeed) an easy-to-disaggregate appliance. Aligned with the observations in [10, 24], the refrigerator appliance consistently represents the appliance that can be disaggregated at greatest accuracy. F1 scores above 0.85 can be consistently achieved with many algorithms, as shown in Fig. 7. The best overall results are unsurprisingly accomplished for test case 1 (TC1) with no added noise (N0), in which the refrigerator is the only appliance in the building. Even the presence of added noise (N1–N4) does not lead to drastically degraded disaggregation scores for the refrigerator. A slight tendency to degraded disaggregation F1 scores can be observed for test cases with more concurrent appliance activity (TC2–TC6), yet overall the accuracy is are still remarkably high. The only exception to this general trend is visible for Hart85, which yields a poor F1 score for test case 1 (N0–N2), but reaches much higher values for the rest of the cases. Given that virtually all algorithms fare with the refrigerator well, it should be re-considered whether this appliance is a good measure for disaggregation performance.

4.3.2 Concurrent appliance activity does not influence all algorithms the same way. Let us compare the MAE diagrams of the refrigerator in Fig. 7b with an excerpt of the scores of the coffee maker in Fig. 8. In general, a positive correlation between the extent of noise and appliance concurrency and the resulting MAE can be observed, with occasional outliers for some configurations (most often for the CO, DSC, and FHMM algorithms). A more coherent trend is observed for the coffee maker, but instead the increment in MAE is much more pronounced for these algorithms. This can likely be attributed to the coffeemaker's load signature (which is internally cycling between heating and idle states, with no power drawn during the latter phases), whereas the refrigerator's power demand follows an exponential decay model during its cooling cycle. MAE values of DAE, RNN, S2S, and S2P are largely unchanging or even shrinking with more complex disaggregation scenarios. This hints at the fact that these algorithms have a greater resilience to noise and concurrent appliance operations.

Investigating deeper into this issue, let us look at the F1 score of FHMM when disaggregating difference appliances, as shown in Fig. 9. The values marked "ref" in the figure represent results of a data set in which only the given appliance was present and configured to operate five times per day. Given the strong degradation FHMM experiences beyond the reference case, the figure reveals the algorithm's sensitivity to noise and concurrent activities. However, this limits its usability in practical NILM settings where isolated appliance operation is the exception rather than the norm. Using test data with realistic features (i.e., concurrently operative devices) is thus vital for the comprehensive evaluation of NILM algorithms.

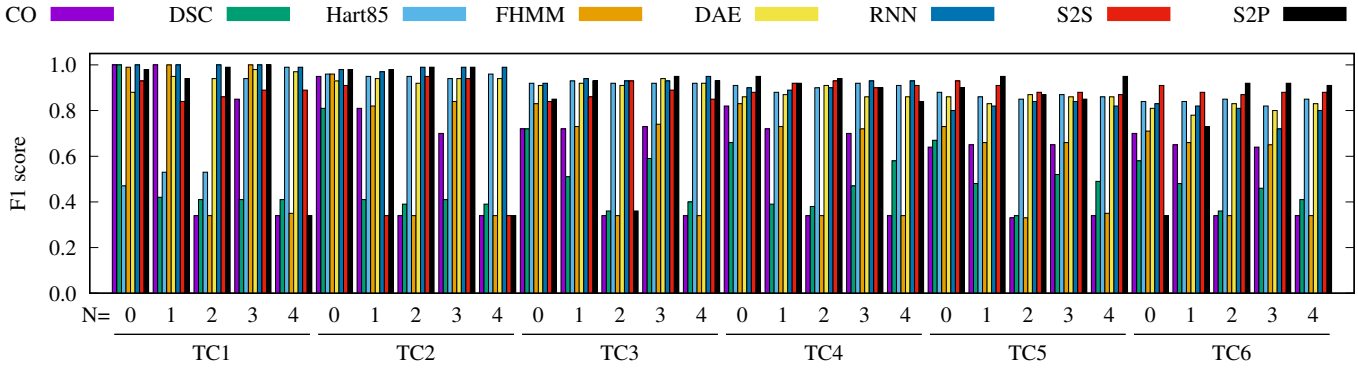
4.3.3 Not all algorithms can disaggregate multi-state appliances. The dishwasher appliance is an exemplary case of a multi-state appliances, i.e., a device that follows a sequence of states (e.g., rinsing, soaking, drying) in a pre-defined order. Disaggregating its power consumption (see Fig. 10), however, shows a rather discernible performance gap between the first four considered algorithms (CO, DSC, Hart85, and FHMM) which exhibit low F1 scores and high MAE values, and the remaining four algorithms under consideration (DAE, RNN, S2S, and S2P). A similar, yet less pronounced, observation can be made for the coffee maker (Fig. 8). Given the prevalence of multi-state appliances (particularly in white goods), we suggest that such devices should be part of all evaluation setups for newly proposed NILM methods.

4.3.4 Current evaluation metrics are not expressive enough. In our evaluations, we have used the correctly attributed power (MAE) and the correct decision if an appliance was operative (F1 score) in conjunction. The underlying reason is that these metric quantify different aspects of disaggregation performance. Other authors (e.g., [30]) have already determined the need for more expressive metrics that combine whether and how accurate appliances could be disaggregated. Rather than using a single metric only, we thus suggest a set of complementary metrics to be computed when comparatively evaluating disaggregation algorithms in order to better highlight the strengths and weaknesses of an algorithm.

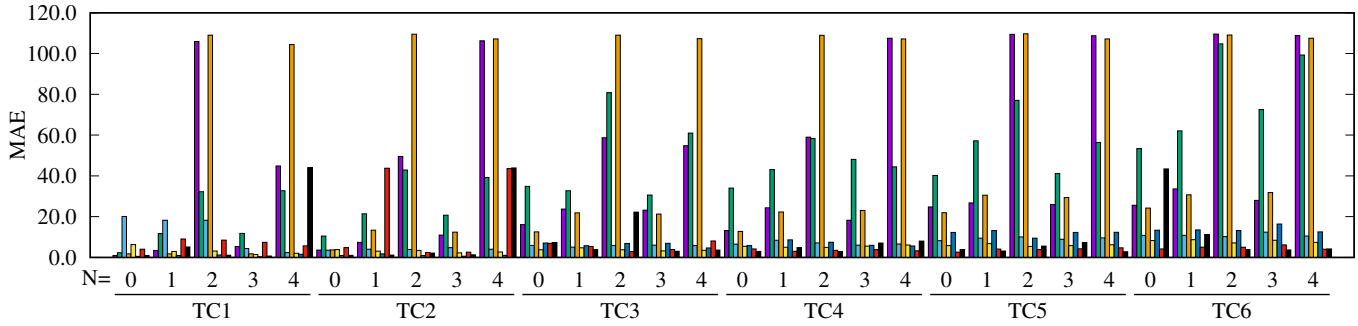
4.3.5 Out of the considered algorithms, S2P is the top-scoring choice. Having observed generally high F1 scores for RNN, S2S, and S2P, and low corresponding disaggregation MAE values, these methods appear as promising avenues for future research. We hence present a more detailed comparison of these three top-scoring algorithms in Fig. 11 when applied to five appliances in the TC6 test case. In general, the F1 score of S2P is almost always superior to the other two mechanisms, and only slightly outperformed by S2S for the case of the refrigerator when no added noise is present. Thus, the approach of using a window of input data to disaggregate just a single value appears as a promising candidate to build upon in new algorithm designs.

5 RELATED WORK

Over the years, an enormous amount of research has been devoted to load disaggregation algorithms. Even though some of their names indicate a generic applicability, e.g., Universal NILM [33], their evaluations are mostly confined to a single data set, such as UK-DALE [16] in [22, 34, 36], REDD [21] in [9], or BLUED [1] in [14]. Confining performance evaluation of NILM algorithms to just one single data set makes it impossible to draw any meaningful conclusions about the performance when applied to data collected



(a) F1 score of the refrigerator appliance when disaggregated from the different test cases.



(b) MAE score of the refrigerator appliance when disaggregated from the different test cases.

Figure 7: Results for the disaggregation of the refrigerator, for each of the metrics defined in Sec. 4.1.2.

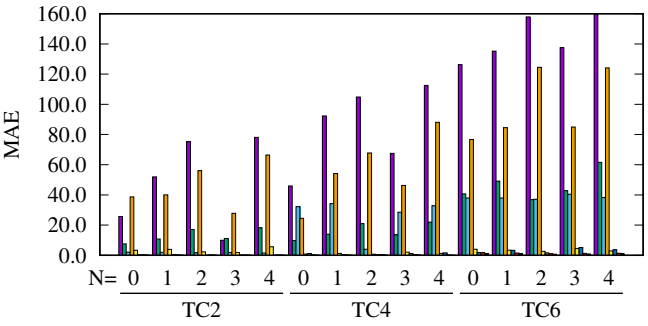


Figure 8: MAE scores for the disaggregation of the coffee maker, on which the extent of concurrent appliance activity has the greatest impact.

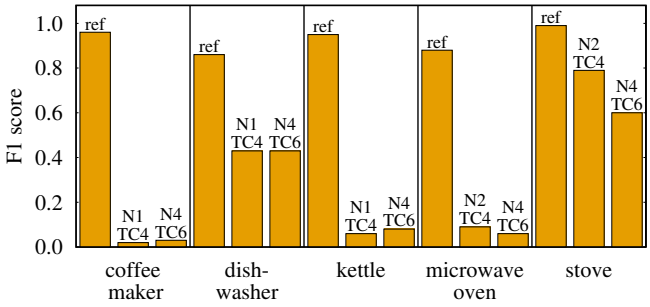
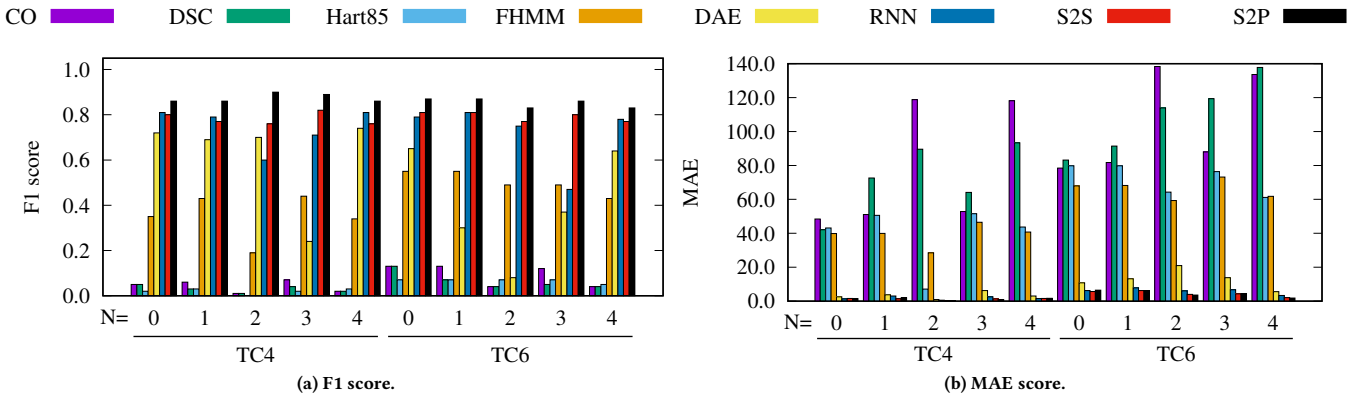


Figure 9: F1 score for the FHMM disaggregation algorithm when disaggregating appliances in a reference case (no other appliance activity) as well as test cases TC4/N1 and TC6/N4.

in different settings. Thus, virtually all of the currently available disaggregation algorithms must be considered to be strongly limited with regard to the possibility to generalize their results.

Only a few studies have considered the comparative assessment of NILM algorithms so far. In [4], five NILM algorithms were comparatively evaluated using the ECO data set, which had been collected specifically for this purpose. Besides the results from the comparison of disaggregation performance, a key insight was the strong heterogeneity of current algorithm implementations with regard to the expected input data format and data granularity, the used programming language, and/or the metrics that have been computed as part of their original evaluations. Consequently, a framework called NILM-EVAL was developed to easily benchmark disaggregation methods using the same input data and the same performance metrics, and even tune their parameter settings. The proposed framework was not widely adopted by developers of NILM algorithms, however, and has not been extended by new methods since its publication. In contrast to this, the Non-Intrusive Load Monitoring Toolkit (NILMTK) [3] is experiencing a greater adoption and greatly contributes to the repeatability of load disaggreagtion research. NILMTK already ships with a set of disaggregation mechanisms (cf. Sec. 4.1) and can be easily extended by new methods.



(a) F1 score. (b) MAE score.

Figure 10: Results for the disaggregation of the dishwasher appliance (which is only part of test cases 4 and 6).

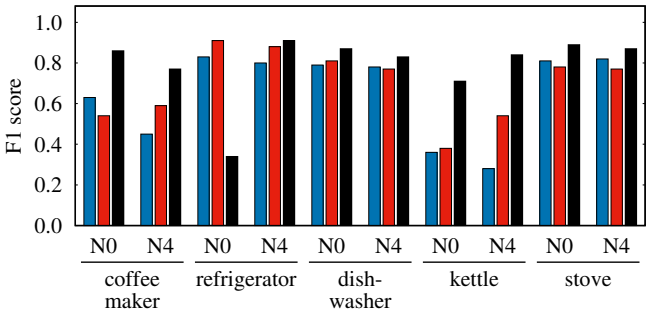


Figure 11: F1 scores of the top-3 disaggregation algorithms across five appliances from test case 6.

Thus, it allows for the same evaluation methodology to be applied through all experiments and algorithms, which effectively results in comparable and reproducible algorithm testing.

An analytical comparison of NILM approaches was presented more recently in [27], in which the authors have compiled the performance evaluation results of ten NILM algorithms, as reported by their original authors. Similar to [4], the plethora of evaluation metrics, the variety and heterogeneity of real-world data sets, as well as the differences in the applied evaluation methodologies were remarked as obstacles to a fair comparative evaluation study. The concern regarding suitable metrics for an objective comparison was furthermore raised in [23], which promotes the choice of metrics to assess both classification and energy estimation accuracy.

The synthetic generation of data represents a viable means to overcome data set limitations by allowing for the generation of a virtually infinite amount of aggregate load signature data with a configurable randomness, noise, and complexity. Correspondingly, tools like AMBAL [5], SmartSIM [7], or the Load Profile Generator [31] have been published. They allow for the generation of appliance-level and building-level load signatures, which can subsequently be exported into file formats ready to be used in disaggregation research. Their adoption in practice is, however, extremely limited; to the best of our knowledge, not a single publication on NILM uses synthetic data for a performance evaluation. Likewise, NILMPEds [29], SHED [12], and SynD [17] are attempts to release a synthetic data set for NILM performance evaluations, yet with neither any means to vary their complexity easily nor access to the tool's source code to create additional data.

Our contribution hence significantly advances the state of the art, in that we not only provide a tool to generate synthetic data, but also to use it as the input for NILMTK as well as evaluating the disaggregation performance of all its included algorithms. By combining our data set generator tool with NILMTK, we construct a toolchain that enables reproducible on-demand performance evaluation of load disaggregation algorithms.

6 CONCLUSION

The development of NILM algorithms is currently seeing a great research interest. Evaluations of such algorithms with regard to their practical usability are, however, often limited to their operation on a small number of available data sets. This represents a strong limitation, as it is not possible to generalize the attained results. In this work, we have made three major contributions to pave the way towards a *comparable* evaluation of NILM algorithms. First, we have introduced ANTgen, a tool to synthetically generate aggregate load signature data with definable degrees of concurrent appliance activity that can be fine-tuned to resemble regular user activities close to reality. Second, we have defined six test cases, composed of up to 9 user activities using 14 devices, and used ANTgen to generate 180 days worth of synthetic data for each of them. These data were then used to comparatively evaluate the disaggregation performance of 8 existing algorithms in more than 8,000 runs of the NILM toolkit. Third and lastly, we have discussed the insights we have gained—some of them counterintuitive—and derived pointers how to steer the future development of NILM algorithms into the right direction.

Benchmarking is a technique widely used in other domains of computer science and engineering, and we have translated this concept to the NILM domain. By making ANTgen publicly available, anyone can contribute activity and appliance models in order to increase the realism of synthetic data even further. Still, NILM algorithms will ultimately be faced with real-world data. We thus primarily see the use of synthetic data as a vehicle to accelerate the development of novel NILM methods rather than to declare data sets collected in real-world scenarios obsolete.

ACKNOWLEDGMENTS

This work was supported by Deutsche Forschungsgemeinschaft grant no. RE 3857/2-1. The authors would like to thank the anonymous reviewers for providing insightful comments and valuable suggestions.

REFERENCES

[1] Kyle Anderson, Adrian Ocneanu, Diego Benitez, Derrick Carlson, Anthony Rowe, and Mario Berges. 2012. BLUED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research. In *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*.

[2] Nipun Batra, Manoj Gulati, Amarjeet Singh, and Mani B. Srivastava. 2013. It's Different: Insights into Home Energy Consumption in India. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys)*.

[3] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. Towards Reproducible State-of-the-Art Energy Disaggregation. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)*.

[4] Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini. 2014. The ECO Data Set and the Performance of Non-Intrusive Load Monitoring Algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys)*.

[5] Nadezda Buneeva and Andreas Reinhardt. 2017. AMBAL: Realistic Load Signature Generation for Load Disaggregation Performance Evaluation. In *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*.

[6] Nadeem A. Burney. 1995. Socioeconomic Development and Electricity Consumption: A Cross-Country Analysis using the Random Coefficient Method. *Energy Economics* 17, 3 (1995).

[7] Dong Chen, David E. Irwin, and Prashant J. Shenoy. 2016. SmartSim: A Device-Accurate Smart Home Simulator for Energy Analytics. In *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*.

[8] Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. 2012. Sensor-Based Activity Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012).

[9] Dominik Egarter, Venkata P. Bhuvana, and Wilfried Elmenreich. 2014. PALDi: Online Load Disaggregation via Particle Filtering. *IEEE Transactions on Instrumentation and Measurement* 64, 2 (2014).

[10] Anthony Faustine, Nerey Henry Mvungi, Shubi Kaijage, and Kisangiri Michael. 2017. A Survey on Non-Intrusive Load Monitoring Methodies and Techniques for Energy Disaggregation Problem. *arXiv preprint arXiv:1703.00785* (2017).

[11] George W. Hart. 1985. *Prototype Nonintrusive Appliance Load Monitor.* Technical Report. MIT Energy Laboratory and Electric Power Research Institute.

[12] Simon Henriet, Umut Şimşekli, Benoit Fuentes, and Gaël Richard. 2018. A Generative Model for Non-Intrusive Load Monitoring in Commercial Buildings. *Energy and Buildings* 177 (2018).

[13] Geoffrey E. Hinton and Ruslan Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006).

[14] Aashish K. Jain, Syed S. Ahmed, Prahalathan Sundaramoorthy, Raghavendran Thiruvengadam, and Vineeth Vijayaraghavan. 2017. Current Peak based Device Classification in NILM on a Low-Cost Embedded Platform using Extra-Trees. In *Proceedings of the IEEE MIT Undergraduate Research Technology Conference (URTC)*.

[15] Jack Kelly and William Knottenbelt. 2015. 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 (BuildSys)*.

[16] Jack Kelly and William Knottenbelt. 2015. The UK-DALE Dataset, Domestic Appliance-level Electricity Demand and Whole-House Demand from Five UK Homes. *Scientific Data* 2, 150007 (2015).

[17] Christoph Klemenjak, Christoph Kovatsch, Manuel Herold, and Wilfried Elmenreich. 2020. A Synthetic Energy Dataset for Non-Intrusive Load Monitoring in Households. *Scientific Data* 7, 1 (2020).

[18] Christoph Klemenjak, Andreas Reinhardt, Lucas Pereira, Mario Berges, Stephen Makonin, and Wilfried Elmenreich. 2019. Electricity Consumption Data Sets: Pitfalls and Opportunities. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)*.

[19] J. Zico Kolter, Siddarth Batra, and Andrew Y. Ng. 2010. Energy Disaggregation via Discriminative Sparse Coding. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems (NIPS)*.

[20] J. Zico Kolter and Tommi Jaakkola. 2012. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[21] J. Zico Kolter and Matthew J. Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. In *Proceedings of the Workshop on Data Mining Applications in Sustainability (SIGKDD)*.

[22] Odysseas Krystalakos, Christoforos Nalmpantis, and Dimitris Vrakas. 2018. Sliding Window Approach for Online Energy Disaggregation Using Artificial Neural Networks. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence (SETN)*.

[23] Stephen Makonin and Fred Popowich. 2015. Nonintrusive Load Monitoring (NILM) Performance Evaluation. *Energy Efficiency* 8, 4 (2015).

[24] Masako Matsumoto, Yu Fujimoto, and Yasuhiro Hayashi. 2016. Energy Disaggregation based on Semi-Binary NMF. In *Proceedings of the 12th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)*.

[25] Judith Michael and Heinrich C. Mayr. 2013. Conceptual Modeling for Ambient Assistance. In *Proceedings of the International Conference on Conceptual Modeling (ER)*.

[26] David Murray, Jing Liao, Lina Stankovic, Vladimir Stankovic, Richard Hauxwell-Baldwin, Charlie Wilson, Michael Coleman, Tom Kane, and Steven Firth. 2015. A Data Management Platform for Personalised Real-Time Energy Feedback. In *Proceedings of the 8th International Conference on Energy Efficiency in Domestic Appliances and Lighting (EEDAL)*.

[27] Christoforos Nalmpantis and Dimitris Vrakas. 2019. Machine Learning Approaches for Non-Intrusive Load Monitoring: From Qualitative to Quantitative Comparation. *Artificial Intelligence Review* 52, 1 (2019).

[28] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. 2012. Non-Intrusive Load Monitoring Using Prior Models of General Appliance Types. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

[29] Lucas Pereira. 2019. NILMPEds: A Performance Evaluation Dataset for Event Detection Algorithms in Non-Intrusive Load Monitoring. *Data* 4, 3 (2019).

[30] 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).

[31] Noah Pflugradt, Jens Teuscher, Bernd Platzer, and Wolfgang Schufft. 2013. Analysing Low-Voltage Grids using a Behaviour based Load Profile Generator. In *Proceedings of the International Conference on Renewable Energies and Power Quality (ICREPQ)*.

[32] Andreas Reinhardt, Paul Baumann, Daniel Burgstahler, Matthias Hollick, Hristo Chonov, Marc Werner, and Ralf Steinmetz. 2012. On the Accuracy of Appliance Identification Based on Distributed Load Metering Data. In *Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT)*.

[33] Alejandro Rodriguez-Silva and Stephen Makonin. 2019. Universal Non-Intrusive Load Monitoring (NILM) Using Filter Pipelines, Probabilistic Knapsack, and Labelled Partition Maps. *arXiv preprint arXiv:1907.06299* (2019).

[34] Cristina Rottondi, Marco Derboni, Dario Piga, and Andrea E. Rizzoli. 2019. An Optimisation-based Energy Disaggregation Algorithm for Low Frequency Smart Meter Data. *Energy Informatics* 2, 1 (2019).

[35] Antonio Ruano, Alvaro Hernandez, Jesus Ureña, Maria Ruano, and Juan Garcia. 2019. NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: A Review. *Energies* 12, 11 (2019).

[36] Valerio Salerno and Graziella Rabbeni. 2018. An Extreme Learning Machine Approach to Effective Energy Disaggregation. *Electronics* 7, 10 (2018).

[37] Akshay S.N. Uttama Nambi, Antonio Reyes Lua, and Venkatesha R. Prasad. 2015. LocED: Location-Aware Energy Disaggregation Framework. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys)*.

[38] Markus Weiss, Adrian Helfenstein, Friedemann Mattern, and Thorsten Staake. 2012. Leveraging Smart Meter Data to Recognize Home Appliances. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*.

[39] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-Point Learning with Neural Networks for Non-Intrusive Load Monitoring. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*.

[40] Mengmeng Zhuang, Mohammad Shahidehpour, and Zuyi Li. 2018. An Overview of Non-Intrusive Load Monitoring: Approaches, Business Applications, and Challenges. In *Proceedings of the International Conference on Power System Technology (POWERCON)*.

[41] Ahmed Zoha, Alexander Gluhak, Muhammad Imran, and Sutharshan Rajasegarar. 2012. Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors* 12, 12 (2012).