NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring

Nipun Batra[1], Jack Kelly[2], Oliver Parson[3], Haimonti Dutta[4], William Knottenbelt[2], Alex Rogers[3],Amarjeet Singh[1], Mani Srivastava[5]

[1]Indraprastha Institute of Information Technology Delhi, India {nipunb, amarjeet}@iiitd.ac.in
[2] Imperial College London {jack.kelly, wjk}@imperial.ac.uk
[3] University of Southampton {op106, acr}@ecs.soton.ac.uk
[4] CCLS Columbia {haimonti@ccls.columbia.edu}
[5] UCLA {mbs@ucla.edu}

ABSTRACT

Non-intrusive load monitoring, or energy disaggregation, aims to separate household energy consumption data collected from a single point of measurement into appliance-level consumption data. In recent years, the field has rapidly expanded due to increased interest as national deployments of smart meters have begun in many countries. However, empirically comparing disaggregation algorithms is currently virtually impossible. This is due to the different data sets used, the lack of reference implementations of these algorithms and the variety of accuracy metrics employed. To address this challenge, we present the Non-intrusive Load Monitoring Toolkit (NILMTK); an open source toolkit designed specifically to enable the comparison of energy disaggregation algorithms in a reproducible manner. This work is the first research to compare multiple disaggregation approaches across multiple publicly available data sets. Our toolkit includes parsers for a range of existing data sets, a standard output format for disaggregation algorithms, a set of statistics for describing data sets, two reference benchmark disaggregation algorithms and a suite of accuracy metrics. We demonstrate that our toolkit makes it easier to enter into energy disaggregation research by simplifying the use of multiple data sets, while supporting the addition of new disaggregation algorithms, and also encouraging direct comparisons to be made between algorithms through common output formats and accuracy metrics.

1. INTRODUCTION

Non-intrusive load monitoring (NILM), or energy disaggregation, aims to break down a household's aggregate electricity consumption into individual appliances [16]. The motivations for such a process are threefold. First, informing a household's occupants of how much energy each appliance consumes empowers them to take steps towards reducing their energy consumption [11]. Second, personalised feedback can be provided which quantifies the savings of certain appliance-specific advice, such as the financial savings when an old inefficient appliance is replaced by a new efficient appliance. Third, if the NILM system is able to determine the time of use of each appliance, a recommender system would be able to inform the household's occupants of the potential savings through deferring appliance use to a time of day when electricity is either cheaper or has a lower carbon footprint.

Such benefits have drawn significant interest in the field since its inception 25 years ago. In recent years, the combination of smart meter meter deployments [10, 13] and reduced hardware costs of household electricity sensors has led to a rapid expansion of the field. Such rapid growth over the past 5 years has been evidenced by the wealth of academic papers published, international meetings held (e.g. NILM 2012[1], EPRI NILM 2013[2]), startup companies founded (e.g. Bidgely, Neurio) and data sets released, (e.g. REDD [23], BLUED [2], Smart* [5]).

However, three core obstacles currently prevent the direct comparison of state-of-the-art approaches, and as a result may be impeding progress within the field. First, each contribution to date has only been evaluated on a single data set and consequently it is hard to assess whether such approaches generalise to new households. Furthermore, many researchers sub-sample data sets to select specific households, appliances and time periods, making experimental results more difficult to reproduce. Second, newly proposed approaches are rarely compared against the same benchmark algorithms, further increasing the difficulty in empirical comparisons of performance between different publications. Moreover, the lack of reference implementations of these state-of-the-art algorithms often leads to the reimplementation of such approaches. Third, each pa-

[1] `http://www.ices.cmu.edu/psii/nilm/`
[2] `http://goo.gl/dr4tpq`

per targets a different use case for NILM and therefore evaluates the accuracy of their proposed approach using a different set of performance metrics. As a result the numerical performance calculated by such metrics cannot be compared between any two papers. These three obstacles have led to the proposal of successive extensions to state-of-the-art algorithms, while a direct comparison between new and existing approaches has remained impossible.

Similar obstacles have arisen in other research fields and prompted the development of toolkits specifically designed to support research in that area. For example, PhysioToolkit offers access to over 50 databases of physiological data and provides software to support the processing and analysis of such data for the biomedical research community [15]. Similarly, CRAWDAD collects 89 data sets of wireless network data in addition to software to aid the analysis of such data for the wireless network community [24]. However, no such toolkit is available to the NILM community.

Against this background, we propose NILMTK[3]; an open source toolkit designed specifically to enable easy access to and comparative analysis of energy disaggregation algorithms across diverse data sets. NILMTK provides a complete pipeline from data sets to accuracy metrics, thereby lowering the entry barrier for researchers to plug in a new algorithm and compare its performance against the current state of the art. NILMTK has been:

- released as open source software (with documentation[4]) in an effort to encourage researchers to contribute data sets, benchmark algorithms and accuracy metrics as they are proposed, with the goal of enabling a greater level of collaboration within the community.
- designed using a modular structure, therefore allowing researchers to reuse or replace individual components as required.
- written in Python with flat file input and output formats, in addition to high performance binary formats, ensuring compatibility with existing algorithms written in any language and designed for any platform.

The contributions of NILMTK are summarised as follows:

- We propose NILMTK-DF (data format), the standard energy disaggregation data structure used by our toolkit. NILMTK-DF is modelled loosely on the REDD data set format [23] to allow easy adoption with the community. Furthermore, we provide parsers from multiple existing data sets into our proposed NILMTK-DF format. In addition, we propose a single output format for the disaggregated data produced by NILM algorithms.
- We provide a set of statistical functions for detailed understanding of each data set. We also provide a set of preprocessing functions for mitigating common challenges with NILM data sets.
- We provide implementations of two benchmark disaggregation algorithms: first an approach based on combinatorial optimisation [16], and second an approach based on the factorial hidden Markov model [23, 21]. We demonstrate the ease by which NILMTK allows the comparison of these algorithms across a range of existing data sets, and present results of their performance.
- We present a suite of accuracy metrics which are able to evaluate the performance of any disaggregation algorithm which produces an output compatible with NILMTK. This allows the performance of a disaggregation algorithm to be evaluated for a range of use cases.

The remainder of this paper is organised as follows. In Section 2 we provide an overview of related work from the field of NILM. In Section 3 we present NILMTK and give a detailed description of its components. In Section 4 we demonstrate the empirical evaluations which are enabled by NILMTK, and provide analyses of existing data sets and disaggregation algorithms. Finally, in Section 5 we conclude the paper and propose directions for future work.

2. BACKGROUND

The field of non-intrusive load monitoring was founded over 20 years ago when Hart proposed the first algorithm for disaggregation of household energy usage [16, 3]. However, the majority of research has been evaluated using either lab-based or simulated data and hence the performance of disaggregation algorithms in real households has remained unknown. More recently, national deployments of smart meters have prompted a renewed interest in energy disaggregation. We now discuss recent research which have contributed new data sets (Section 2.1), disaggregation algorithms (Section 2.2) and evaluation metrics (Section 2.3) to the field. In Section 2.4 we discuss general purpose toolkits, and finally in Section 2.5 we formalise the NILM problem drawing upon notation used in prior literature.

2.1 Data Sets

In 2011, the Reference Energy Disaggregation Dataset (REDD) [23] was introduced as the first publicly available data set collected specifically to aid NILM research. The data set contains both aggregate and sub-metered

[3] Code: `http://github.com/nilmtk/nilmtk`

[4] Documentation: `http://nilmtk.github.io/nilmtk`

Data set	Institution	Location	Duration per house	Number of houses	Appliance sample frequency	Aggregate sample frequency
REDD (2011)	MIT	MA, USA	3-19 days	6	3 sec	1 sec & 15 kHz
BLUED (2012)	CMU	PA, USA	8 days	1	N/A*	12 kHz
Smart* (2012)	UMass	MA, USA	3 months	1	1 sec	1 sec
Tracebase (2012)	Darmstadt	Germany	N/A	N/A	1-10 sec	N/A
Sample (2013)	Pecan Street	TX, USA	7 days	10	1 min	1 min
HES (2013)	DECC, DEFRA	UK	1 or 12 months	251	2 or 10 min	2 or 10 min
AMPds (2013)	Simon Fraser U.	BC, Canada	1 year	1	1 min	1 min
iAWE (2013)	IIIT Delhi	Delhi, India	73 days	1	1 sec	1 sec
UKPD (2014)	Imperial College	London, UK	3-14 months	4	6 sec	1-6 sec & 16 kHz

Table 1: Comparison of household energy data sets. *BLUED labels every state transition for each appliance.

power data from 6 households, and has since become the most popular data set for evaluating energy disaggregation algorithms. In 2012, the Building-Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) [2] was released containing data from a single household. However, the data set does not include sub-metered power data, and instead events triggered by appliance state changes were recorded. As a result, it is only possible to evaluate whether changes in appliance states have been detected (e.g. washing machine turns on), rather than the assignment of aggregate power demand to individual appliances (e.g. washing machine draws 2 kW power). More recently, the Smart* [5] data set was released, which contains household aggregate power data from 3 households, while sub-metered appliance power data was only collected from a single household.

In 2013 the Pecan Street sample data set was released [17], which contains both aggregate and sub-metered power data from 10 households. Later, the Household Electricity Survey data set was released [35], which contains data from 251 households although aggregate data was only collected for 14 households. The Almanac of Minutely Power dataset (AMPds) [27] was also released that year containing both aggregate and sub-metered power data from a single household. Subsequently, the Indian data for Ambient Water and Electricity Sensing (iAWE) [8] was released, which contains both aggregate and sub-metered power data from a single house. Most recently, the UK Power Dataset (UKPD) [20] was released which contains data from four households using both aggregate meters and individual appliance sub-meters. Unfortunately, subtle differences in the aims of each data set have led to completely different data formats being used. As a result, a time-consuming engineering barrier exists when using the data sets, each of which are in different formats. This has resulted in publications using only a single data set to evaluate a given approach, and consequently the generality of results over large numbers of households are rarely investigated. We summarise these data sets in Table 1.

2.2 Disaggregation Algorithms & Benchmarks

The REDD data set was proposed along with a performance result of a benchmark disaggregation algorithm using 10 second data across 5 of the 6 households [23]. Kolter and Jaakkola later proposed an extension to the benchmark algorithm [22], however the extension was only evaluated using features extracted from 14 kHz data from a single house from the data set, and therefore the performance results are not directly comparable. Later, Zeifman [34] and Johnson and Willsky [18] evaluated various approaches using the same data set, although both selected a different subset of appliances and calculated an artificial household aggregate from these appliances, therefore simplifying the disaggregation problem and preventing a numerical comparison with other publications. Subsequently, Parson et al. [28] and Rahayu et al. [30] both proposed new approaches, although each were evaluated using a different set of 4 houses from the REDD data set, again preventing a numerical comparison between publications. Last, Batra et al. [7] evaluated their approach on the REDD data set using a different household to Kolter and Jaakkola. As a result, it has not been possible to deduce whether one approach is preferable to another from the literature.

Similarly, BLUED was introduced along with a benchmark algorithm [2], but has since only been used by one other publication [1]. Similarly, AMPds and iAWE have only been used to evaluate disaggregation algorithms proposed by the data set authors [27, 8]. Clearly, the variety of different formats is slowing the uptake of new data sets, and also preventing algorithms from being tested across multiple data sets.

It is essential to compare newly proposed disaggregation algorithms to the state of the art in order to assess the increase in an algorithm's performance. However, the lack of available reference implementations of state-of-the-art disaggregation algorithms has led to authors often comparing against more basic benchmark algorithms. This problem is further compounded since there is no single consensus on which benchmarks to use, and as a result most publications use a different benchmark algorithm. For example, Kolter and Jaakkola compared their approach to a set of decoupled HMMs [22], Parson et al. and Batra et al. both evaluated their approaches against variants of their own approaches [28, 7], Zeifman compared their approach to a Bayesian classifier, while Rahayu et al. and Johnson and Willsky both compared against a Factorial Hidden Markov Model (FHMM) [30, 18]. Clearly, further publications would benefit from openly available common benchmark algorithms against which newly proposed algorithms could be easily compared.

2.3 Evaluation metrics

The range of different application areas of energy disaggregation has prompted a number of evaluation metrics to be proposed. For example, four disaggregation metrics labelled *energy correctly assigned* have recently been used to evaluate the performance of disaggregation algorithms using the REDD data set. First, Kolter and Johnson [23] proposed an accuracy metric which captures the error in assigned energy normalised by the actual energy consumption in each time slice averaged over all appliances, which was also later used by Rahayu et al. [30] and Johnson and Willsky [18]. However, large errors in the assigned energy in some time slices will result in a negative accuracy, making this an ill-posed metric. Second, Kolter and Jaakkola [22] proposed an equivalent metric wherein the error is presented individually for each appliance rather than an average across all appliances. Third, Parson et al. [28] proposed a metric which captures the error in assigned energy consumed over the complete duration of the data set rather than per time slice. This metric allows overestimates and underestimates in the assigned energy in different time slices to cancel out, and therefore does not represent all disaggregation errors. Fourth, Batra et al. [7] proposed a subtly different metric to Kolter and Johnson [23], in which error is reported instead of accuracy, and also energy assigned to an incorrect appliance is double counted as both an overestimate of one appliance's energy consumption and an underestimate of another. The differences between these four metrics prevent numerical comparisons between publications, and motivate the use of common metrics.

2.4 General Purpose Toolkits

Although no toolkit currently exists specifically for energy disaggregation, various toolkits are available for more general machine learning tasks. For example, scikit-learn is a general purpose machine learning toolkit implemented in Python [29] and GraphLab is a machine learning and data mining toolkit written in C++ [26]. While such toolkits provide generic implementations of machine learning algorithms, they lack functionality specific to the energy disaggregation domain, such as data set parsers, benchmark disaggregation algorithms, and energy disaggregation metrics. Therefore, an energy disaggregation toolkit should extend such general toolkits rather than replace them, in a similar way that scikit-learn adds machine learning functionality to the numpy numerical library for Python.

2.5 Energy Disaggregation Definition

The aim of energy disaggregation is to provide estimates, $\hat{y}_t^{(n)}$, of the actual power demand, $y_t^{(n)}$, of each appliance n at time t, from household aggregate power readings, $\bar{y}_t$. Most NILM algorithms model appliances using a set of discrete states such as off, on, intermediate, etc. We use $x_t^{(n)} \in \{1, \ldots, K\}$ to represent the ground truth state, and $\hat{x}_t^{(n)}$ to represent the appliance state estimated by a disaggregation algorithm.

3. NILMTK

NILMTK is designed keeping in mind three different use case scenarios:

1. Analysis of existing data sets and algorithms.
2. Ease of adding new algorithms proposed or new data sets released for broad comparison across existing benchmarks.
3. Ease of deploying learnt models for processing online as well as offline data.

We implemented NILMTK in Python due to the availability of a vast set of libraries supporting both machine learning research (e.g. Pandas, scikit-learn) and the deployment of such research as web applications (e.g. Django, Flask). Furthermore, Python allows easy deployment in diverse environments including academic settings and is increasingly being used for data science. Our API design is heavily based on scikit-learn [29, 9].

Figure 1 presents the NILMTK pipeline from the import of data sets to the evaluation of various disaggregation algorithms over NILM specific metrics. We discuss each module of the NILMTK pipeline in the remainder of this section.

3.1 Data Format

Motivated by our discussion of the wide differences between multiple data sets released in public domain in Section 2.1, we propose NILMTK-DF; a common data set format inspired by the REDD format [23], into which existing data sets can be converted. NILMTK currently includes importers for the following data sets: REDD, Smart*, Pecan Street, HES, iAWE, AMPds and UKPD. BLUED was excluded due to the lack of sub-metered power power data. Tracebase was not included due to the lack of information regarding which appliances were present in the same household.

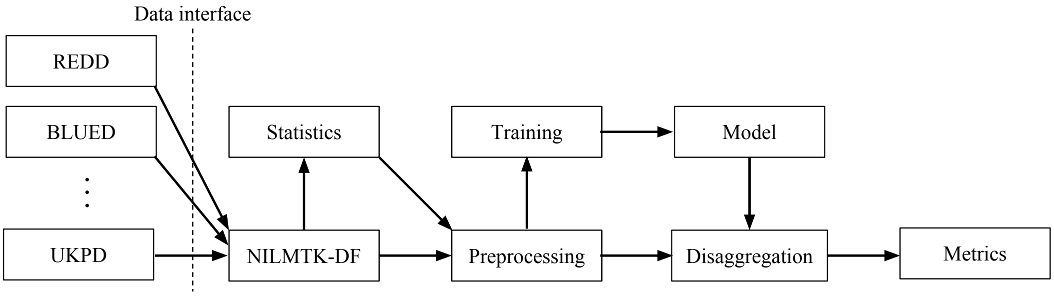

Figure 1: NILMTK pipeline. At each stage of the pipeline, results and data can be stored to or loaded from disk.

After import, the data resides in our NILMTK-DF in-memory data structure, which is used throughout the NILMTK pipeline. Data can be saved or loaded from disk at multiple stages in the NILMTK processing pipeline to allow other tools to interact with NILMTK. We provide two CSV flat file formats: a rich NILMTK-DF CSV format and a "strict REDD" format which allows researchers to use their existing tools designed to process REDD data. We also provide a more efficient binary format using the Hierarchical Data Format (HDF5). In addition to storing electricity data, NILMTK-DF can also store relevant metadata and other sensor modalities such as gas, water, temperature, etc. It has been shown in the past that such additional sensor and metadata information may help enhance NILM prediction [32].

Another important feature of our format is the standardisation of nomenclature. Different data sets use different labels for the same class of appliance (e.g. REDD uses 'refrigerator' whilst AMPds uses 'FGE') and different names for the measured parameters. When data is first imported into NILMTK, these diverse labels are converted to a standard vocabulary[5].

In addition, NILMTK allows rich metadata to be associated with a household, appliance or meter. For example, NILMTK can store the parameters measured by each meter (e.g. reactive power, real power), the geographical coordinates of each house (useful for exploring the relationship between appliance usage and weather), the mains wiring defining the meter hierarchy (useful if a single appliance is measured at the appliance, circuit and aggregate levels), whether a single meter measures multiple appliances and whether a specific lamp is dimmable. A full description of NILMTK-DF is provided in Appendix B.

Through such a combination of metadata and standard nomenclature, NILMTK allows for analysis of appliance data across multiple data sets. For example, users can perform queries such as: 'what is the average energy consumption of refrigerators in the USA compared to the UK?'. Further examples are given in the Appendix F.

We have defined a common interface for data set importers which, combined with the definition of our in-memory data structures, should make it relatively easy for developers to add new data set importers to NILMTK.

3.2 Data Set Diagnostics

No data set is perfect so researchers often spend a long time exploring the detailed characteristics of each data set before proceeding with their research. To help diagnose these issues, NILMTK provides diagnostic functions including:

Detect gaps: Many NILM algorithms assume that each channel is contiguous but this assumption is false when sensors are off or malfunctioning. A 'gap' exists between any pair of consecutive samples if the time elapsed between them is larger than some threshold.

Dropout rate: One common problem is missing data. The dropout rate is the total number of recorded samples, divided by the number of expected samples (which is the length of the time window under consideration multiplied by the sample rate).

Dropout rate (ignoring gaps): To quantify the rate at which a wireless sensor drops samples due to radio issues we first remove large gaps where the sensor is off and then calculate the dropout rate for the remaining contiguous sections.

Up-time: The up-time is the total time for which a sensor was recording. It is the last timestamp, minus the first timestamp, minus the duration of any gaps.

Diagnose: NILMTK provides a single `diagnose` function which checks for all the issues we have encountered.

3.3 Data Set Statistics

Distinct from *diagnostic* statistics, NILMTK also provides functions for exploring appliance usage, e.g.:

Proportion of energy sub-metered: data sets rarely sub-meter every single appliance or circuit so it is useful to quantify the total energy measured by all sub-metered channels, divided by the total energy measured by the mains channels. Prior to calculating this statistic, all gaps present in the mains recordings are

[5] `nilmtk/docs/standard_names/appliances.txt`

masked out of all appliance recordings; and any missing sub-meter data is assumed to be due to the meter being switched off (along with its load).

This section has only described a subset of the diagnostic and statistical functions in NILMTK. Further functions are listed in Appendix E and in the statistics section of the online documentation[6]. Section 4 provides results of the statistical functions described above applied to multiple data sets.

3.4 Preprocessing of Data Sets

The data sets described in Table 1 were collected with a range of hardware and under different settings. For example, the sampling rate of appliance monitors varies from 0.008 Hz to 15 kHz across the data sets. NILMTK provides filters to down-sample such data sets at a specified frequency. Further, these data sets have been collected from different countries, where voltage fluctuations vary widely. Batra et al. showed voltage fluctuates from 180-250 V in the iAWE data set collected in India [8], while the voltage in the Smart* data set varies across the range 118-123 V. Hart et al. showed the need to take into account these voltage fluctuations as they can significantly impact power draw [16]. Therefore, NILMTK has a voltage normalisation function which implements Hart's normalisation equation:

$$Power_{normalised} = \left(\frac{Voltage_{nominal}}{Voltage_{observed}}\right)^2 \times Power_{observed} \tag{1}$$

As explained in Section 3.5, the memory required by some NILM algorithms is exponential in the number of appliances. Thus, when a large number of appliances are considered, a single centralised model may not fit into memory, thus motivating the need for distributed algorithms. This necessitates the need to include top-k appliances by their contribution which can be in terms of either energy or power. We also created preprocessing functions for fixing other common issues with these data sets, such as (i) interpolating small periods of missing data when appliance sensors did not report readings; (ii) filtering out implausible values (such as readings where observed voltage is more than twice the rated voltage) and (iii) filtering out appliance data when mains data is missing.

Each data set importer in NILMTK defines a `preprocess` function which runs the necessary preprocessing functions to clean the specific data set.

3.5 Training and Disaggregation Algorithms

NILMTK provides implementations of two common benchmark NILM algorithms: combinatorial optimisation (CO) and factorial hidden Markov model (FHMM). CO was proposed by Hart in his seminal work [16], while techniques based on extensions of the FHMM have been proposed more recently [23, 21]. The aim of the inclusion of these algorithms is not to present state-of-the-art disaggregation results, but instead to enable new approaches to be compared to well-studied benchmark algorithms without requiring the reimplementation of such algorithms. We now briefly describe these two algorithms.

Combinatorial Optimisation: CO finds the *optimal* combination of appliance states, which minimises the difference between the sum of the predicted appliance power and the observed aggregate power, subject to a set of appliance models.

$$\hat{x}_t^{(n)} = \underset{\hat{x}_t^{(n)}}{\operatorname{argmin}} \left| \bar{y}_t - \sum_{n=1}^{N} \hat{y}_t^{(n)} \right| \tag{2}$$

Since each time slice is considered as a separate optimisation problem, each time slice is assumed to be independent. CO resembles the subset sum problem and thus is NP-complete. The complexity of disaggregation for T time slices is $O(TK^N)$, where N is the number of appliances and K is the number of states for each appliance. With the complexity of CO being exponential in the number of appliances, the approach is only computationally tractable when a small number of appliances are modelled. However, it should be noted that more efficient pseudo-time algorithms can often be applied in practice.

Factorial Hidden Markov Model: The power demand of each appliance can be modelled as the observed value of a hidden Markov model (HMM). The hidden component of these HMMs are the states of the appliances. Energy disaggregation involves jointly decoding the power draw of n appliances and hence a factorial HMM [14] is well suited. A FHMM can be represented by an equivalent HMM in which each state corresponds to a different combination of states of each appliance. Such a FHMM model has three parameters: (i) prior probability (π) containing K^N entries, (ii) transition matrix (A) containing $K^N \ * K^N$ or K^{2N} entries, and (iii) emission matrix (B) containing $2K^N$ entries. The complexity of exact disaggregation for such a model is $O(TK^{2N})$, and as a result FHMMs scale even worse than CO. From an implementation perspective, even storing (or computing) A for 14 appliances with 2 states each needs 8 GB of RAM. Hence, we propose to validate FHMMs on preprocessed data where either top-k appliances are modelled, and appliances contributing less than a given threshold are discarded.

For certain algorithms such as FHMMs, modeling relationships amongst consecutive samples is necessary. Thus, NILMTK provides facilities for dividing data into train and test while still maintaining the notion of time.

3.6 Appliance Model Import and Export

[6] `http://nilmtk.github.io/nilmtk/stats.html`

Recently, a lot of interest has arisen in deploying live NILM systems, as evidenced by the many startup companies currently aiming to provide disaggregated consumption feedback to end consumers. However, the application of recent NILM algorithms proposed in academic research for live disaggregation is hindered by their primary suitability for offline statistical analysis of fixed data sets. Motivated by reducing the barrier from offline algorithms to those suitable for live deployments, we developed the Model module in NILMTK which encapsulates the results of the training module required by the disaggregation module. It is the responsibility of the model developer to create export and import functions to interface with a JSON file. NILMTK currently includes importers and exporters for both the FHMM and CO approaches. Further, in Section A, we show how if the rated power is known, we can avoid learning on sub-metered power data. At the time of deployment, this learnt JSON model can be easily imported and used for disaggregation.

3.7 Accuracy Metrics

As discussed in Section 2, a range of accuracy metrics are required due to the diversity of application areas of energy disaggregation research. To satisfy this requirement, NILMTK provides a set of metrics which combines both general detection metrics and those specific to energy disaggregation. We now give a brief description of each metric implemented in NILMTK along with its mathematical definition.

Error in total energy assigned: Represents the difference between the total energy assigned to appliance n and the actual energy consumed by appliance n over the entire data set.

$$\left| \sum_t y_t^{(n)} - \sum_t \hat{y}_t^{(n)} \right| \tag{3}$$

Fraction of total energy assigned correctly: Represents the overlap between the fraction of total energy assigned to each appliance and the actual fraction of total energy consumed by each appliance over the entire data set.

$$\sum_n \min \left(\frac{\sum_n y_t^{(n)}}{\sum_{n,t} y_t^{(n)}}, \frac{\sum_n \hat{y}_t^{(n)}}{\sum_{n,t} \hat{y}_t^{(n)}} \right) \tag{4}$$

Normalised error in assigned power: Represents the sum of the differences between the power assigned to appliance n and the actual power of appliance n in each time slice t, normalised by the total energy consumption of appliance n.

$$\frac{\sum_t \left| y_t^{(n)} - \hat{y}_t^{(n)} \right|}{\sum_t y_t^{(n)}} \tag{5}$$

RMS error in assigned power: Represents the root mean square error between the power assigned to appliance n and the actual power of appliance n in each time slice t.

$$\sqrt{\frac{1}{T} \sum_t \left(y_t^{(n)} - \hat{y}_t^{(n)} \right)^2} \tag{6}$$

Confusion matrix: Represents the number of time slices in which each of an appliance's states were either confused with every other state or correctly classified.

True positives, False positives, False negatives, True negatives: Represents the number of time slices in which appliance n was either correctly classified as being on (TP), classified as being on while it was actually off (FP), classified as off while is was actually on (FN) and correctly classified as being off (TN).

$$TP^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = on, \hat{x}_t^{(n)} = on \right) \tag{7}$$

$$FP^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = off, \hat{x}_t^{(n)} = on \right) \tag{8}$$

$$FN^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = on, \hat{x}_t^{(n)} = off \right) \tag{9}$$

$$TN^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = off, \hat{x}_t^{(n)} = off \right) \tag{10}$$

True/False positive rate: Represents the fraction of time slices in which an appliance was correctly predicted to be on that it was actually on (TPR), and the fraction of time slices in which the appliance was incorrectly predicted to be on that it was actually off (FPR). We omit appliance indices n throughout the following metrics for conciseness.

$$TPR = \frac{TP}{(TP + FN)} \tag{11}$$

$$FPR = \frac{FP}{(FP + TN)} \tag{12}$$

Precision, Recall: Represents the fraction of time slices in which an appliance was correctly predicted to be on that it was actually off (Precision), and the fraction of time slices in which the appliance was correctly predicted to be on that it was actually on (Recall)

$$Precision = \frac{TP}{(TP + FP)} \tag{13}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{14}$$

F-score: Represents the harmonic mean between precision and recall.

$$F\text{-}score = \frac{2.Precision.Recall}{Precision + Recall} \tag{15}$$

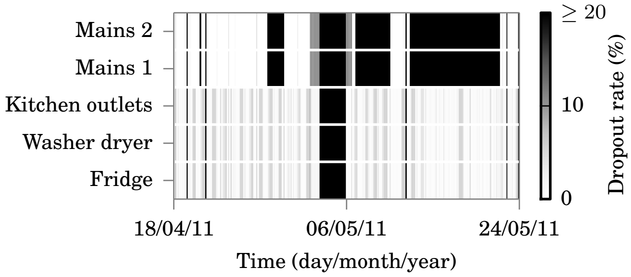

Figure 2: Lost samples per hour from a representative subset of channels in REDD house 1.

Hamming loss: Represents the total information lost when appliances are incorrectly classified over the data set.

$$HammingLoss = \frac{1}{T}\sum_{t}\frac{1}{N}\sum_{n}\mathrm{XOR}\left(x_t^{(n)}, \hat{x}_t^{(n)}\right) \quad (16)$$

In Appendix C we summarise the NILMTK pipeline with a code snippet to illustrate the ease by which data sets can be imported and preprocessed, algorithms can be trained and used to disaggregate a household's energy usage, and accuracy metrics can be employed to evaluate disaggregation accuracy.

4. EVALUATION

We now demonstrate several examples of the rich analyses supported by NILMTK. First, we diagnose some common (and inevitable) issues in a selection of data sets. Then we show patterns of appliance usage. Third, we give some examples of the effect of voltage normalisation on the power demand of individual appliances, and discuss how this might affect the performance of a disaggregation algorithm. Fourth, we present summary performance results of the two benchmark algorithms included in NILMTK across six data sets using a number of accuracy metrics. Finally, we present detailed results of these algorithms for a single data set, and discuss each algorithm's performance for different appliances.

4.1 Data Set Diagnostics

Table 2 shows a selection of statistics (defined in Section 3.2) computed by NILMTK across multiple public data sets. For example, the table illustrates that AMPds used a robust recording platform because it has a percentage up-time of 100%; a dropout rate of zero and 97% of the energy recorded by the mains channel was captured by the sub-meters. Pecan Street also has a 100% uptime and zero dropout rate. Two homes in Pecan Street registered a proportion of energy submetered of over 100%, despite us removing channels which were clearly measuring circuits (as distinct from appliances), which suggests that other channels must be measuring circuits (and hence some appliances are double-counted). Many data sets do not explicitly describe which sub-meter channels measure circuits and which measure appliances. This illustrates the benefits of data set metadata (proposed as part of NILMTK-DF in Section 3.1) describing the basic mains wiring.

Figure 2 shows the distribution of missing samples for REDD house 1. From this we can see that each mains recording channel has four large gaps (the solid black blocks) where the sensors are off and the sub-metered channels have only one large gap. Ignoring this gap and focusing on the time periods where the sensors are recording, we see numerous periods where the dropout rate is around 10%. Such issues are by no means unique to REDD and are crucial to diagnose before data sets can be used for the evaluation of disaggregation algorithms, or for data set statistics.

4.2 Data Set Statistics

Energy disaggregation systems must model individual appliances. Hence, as well as diagnosing technical issues with each data set, we can also visualise patterns of behaviour recorded in each data set. For example, different appliances draw a different amount of power (e.g. a toaster draws approximately 1.57 kW), are used at different times of day (e.g. the TV is usually on in the evening) and have different correlations with external factors such as weather (e.g. lower outside temperature implies more usage of electric heating). Furthermore, load profiles of different appliances of the same type can vary considerably, especially appliances from different countries (e.g. the two washing machine profiles in Figure 4). Some disaggregation systems benefit by capturing these patterns (for example, the conditional factorial hidden Markov model (CFHMM) [21] can model the influence of time of day on appliance usage). In the following sections, we present examples of how such information can be extracted from existing data sets using NILMTK, covering the distribution of appliance power demands (Section 4.2.1), usage patterns (Section 4.2.2) and external dependencies (Section 4.2.3).

4.2.1 Appliance power histograms

Figure 3 displays histograms of the distribution of powers used by a selection of appliances. Appliances such as toasters and kettles tend to have just two possible power states: on and off. This simplicity makes them amenable to be modelled by, for example, Markov chains with only two states per chain. In contrast, more complex appliances such as washing machines, vacuum cleaners, dimmer lights and computers often have many more states.

The proportion of energy use per appliance varies from country to country. For example, the house recorded in India for the iAWE data set has two air conditioning units, whilst none of the households in UKPD have an

Data set	Number of appliances	Percentage energy sub-metered	Dropout rate (percent) ignoring gaps	Mains up-time per house (days)	Percentage up-time
REDD	9, 16, 23	58, 71, 89	0, 10, 16	4, 18, 19	8, 40, 79
Pecan Street	13, 14, 22	75, 87, 150	0, 0, 0	7, 7, 7	100, 100, 100
AMPds	20	97	0	364	100
iAWE	10	48	8	47	93
UKPD	4, 12, 49	19, 48, 82	0, 7, 22	36, 102, 404	73, 84, 100

Table 2: Summary of data set results calculated by the diagnostics functions in NILMTK. Each cell represents the range of values across all households per data set. The three numbers per cell are the minimum, median and maximum values. AMPds and iAWE each contain just a single house, hence these rows have a single number per cell.

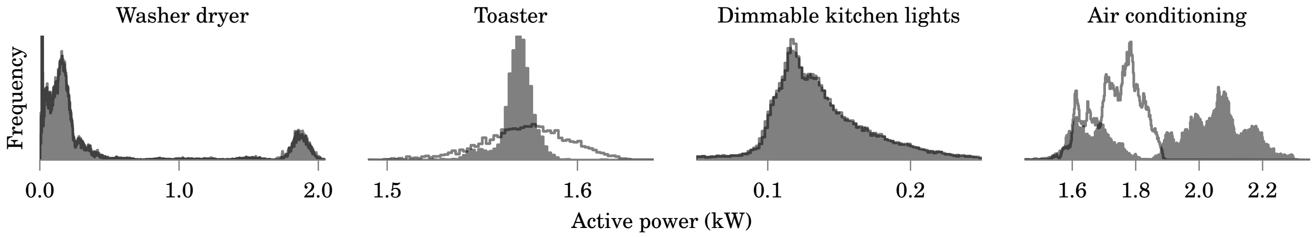

Figure 3: Histograms of power consumption. The filled grey plots show histograms of normalised power. The thin, grey, semi-transparent lines drawn over the filled plots show histograms of un-normalised power.

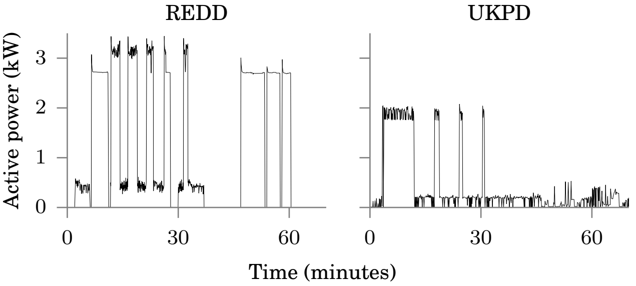

Figure 4: One washing machine from the USA, one from the UK.

air conditioning unit. Examples of the differences in appliance energy use in individual households by country are illustrated in Figure 5.

4.2.2 *Appliance usage histograms*

Figure 6 shows histograms which represent usage patterns for three appliances over an average day, from which strong similarities between groups of appliances can be seen. For example, the usage patterns of the TV and Home theatre PC are very similar because the Home theatre PC is the only video source for the TV. In contrast, the boiler has a different usage pattern which occurs as a result of the household's occupancy pattern in mornings and evenings, and the hot water timer triggering twice a day.

4.2.3 *Appliance correlations with weather*

Previous studies have demonstrated correlations between temperature and heating/cooling demand in Australia [31]

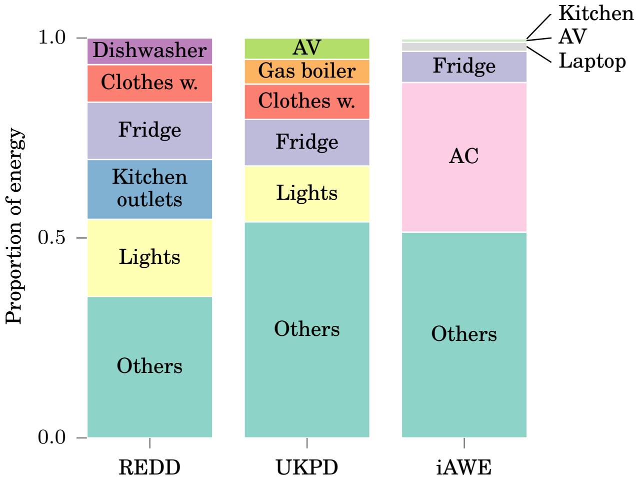

Figure 5: Top 5 appliances in terms of the proportion of the total energy used in a single house (house 1) in each of REDD (USA), iAWE (India) and UKPD (UK).

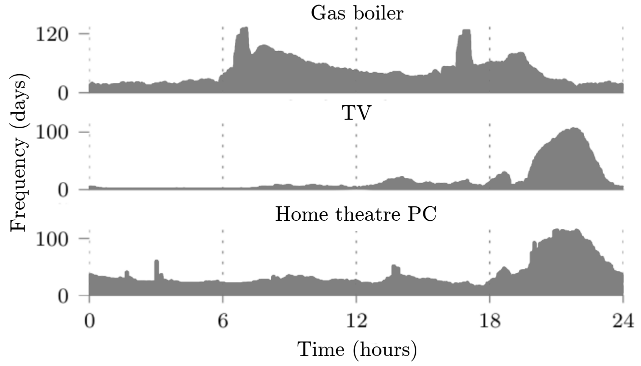

Figure 6: Daily appliance usage histograms of three appliances over 120 days from UKPD house 1.

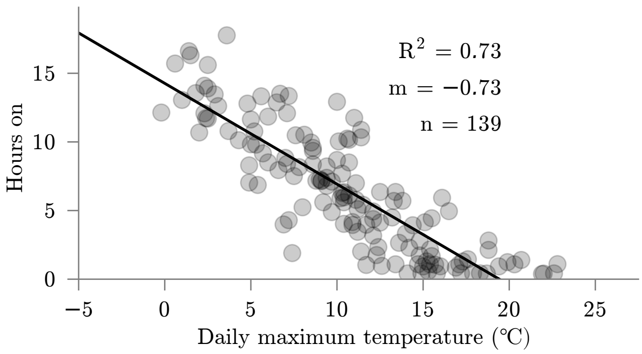

Figure 7: Linear regression showing correlation between gas boiler usage and external temperature. R^2 denotes the coefficient of determination, m is the gradient of the regression line and n is the number of data-points (days) used in the regression.

and between temperature and total household electricity usage in the USA [19]. If such correlations can be demonstrated then a disaggregation system could learn correlations between weather variables and appliance usage in order to refine its appliance usage estimates [33].

Figure 7 shows correlations between boiler usage and maximum temperature (appliance data was from UKPD house 1; temperature data was from the UK Met Office). The correlation between external maximum temperature and boiler usage is strong ($R^2 = 0.73$) and it is noteworthy that the x-axis intercept ($\approx 19\,^{\circ}$C) is approximately the set point for the boiler thermostat, as one might expect.

4.3 Voltage Normalisation

It is inevitable that a household's mains voltage will fluctuate around the nominal value. For example, in the UK the nominal mains voltage is 230 V but is allowed to vary by −6%, +10%. This variation in voltage can produce variations in power demand of 20% in linear loads, such as resistive heaters [16]. Such abrupt changes in reported power can be problematic for disaggregation algorithms. To mitigate this effect, the power consumption of individual appliances can be normalised as outlined in Section 3.4.

Figure 3 shows histograms for both the normalised and un-normalised appliance power consumption. Normalisation produces a noticeably tighter power distribution for linear resistive appliances such as the toaster, although it has little effect on constant power appliances, such as the washer dryer or LED kitchen ceiling lights. Moreover, on non-linear appliances such as the air conditioner, normalisation increases the variance in power draw. This is in conformance to the prior work by Hart et al. [16] which proposed a modified approach to normalisation:

$$Power_{normalised} = \left(\frac{Voltage_{nominal}}{Voltage_{observed}}\right)^{\beta} \times Power_{observed} \tag{17}$$

For linear appliances such as the toaster, β is 2, whereas for appliances such as fridge, Hart found β to be 0.7. Thus, we believe that the benefit of voltage normalisation is dependent on the proportion of resistive (linear) loads in the household.

4.4 Disaggregation Across Data Sets

We now compare the disaggregation results across the first house of each of the following 6 data sets: AMPds, Pecan Street, iAWE, Smart*, UKPD and REDD. Since all the data sets were collected over different durations, we used the first half of the samples for training and the remaining half for testing across all data sets. Further, we preprocessed REDD, UKPD, Smart* and iAWE data set to 1 minute frequency using the down-sampling filter, which we discussed earlier in Section 3.4, to account for different aggregate and mains data sampling frequencies and compensating for intermittent lost data packets. The small gaps in REDD, UKPD, SMART* and iAWE were interpolated, while the time periods where either of mains data or appliance data was missing were ignored. AMPds and Pecan Street data did not require any preprocessing. Since both CO and FHMM have exponential computational complexity in the number of appliances, we consider only those appliances whose total energy contribution was greater than 5%. Across all the data sets, the appliances which contribute more than 5% of the aggregate include HVAC appliances such as the air conditioner and electric heating, and appliances which are used throughout the day such as the refrigerator. We model all appliances using two states (on and off) across our analyses, although it should be noted that any number of states could be used. However, our experiments are intended to demonstrate a fair comparison of the benchmark algorithms, rather than an fully optimised version of either approach. We compare the disaggregation performance of CO and FHMM

Data set	Train time (s)		Disaggregate time (s)		NEP		FTE		F-score	
	CO	FHMM	CO	FHMM	CO	FHMM	CO	FHMM	CO	FHMM
REDD	3.67	22.81	0.14	1.21	1.61	1.35	0.77	0.83	0.31	0.31
Smart*	1.51	1.66	0.03	0.09	0.75	0.70	0.89	0.93	0.80	0.79
Pecan Street	1.72	2.83	0.02	0.12	0.68	0.75	0.99	0.87	0.77	0.77
AMPds	5.92	298.49	3.08	22.58	2.23	0.96	0.44	0.84	0.55	0.71
iAWE	1.68	8.90	0.07	0.38	0.91	0.91	0.89	0.89	0.73	0.73
UKPD	1.06	11.42	0.10	0.52	3.66	3.67	0.81	0.80	0.38	0.38

Table 3: Comparison of CO and FHMM across multiple data sets

across the following three metrics defined in Section 3.7: (i) fraction of total energy assigned correctly (FTE), (ii) normalised error in assigned power (NEP) and (iii) F-score. These metrics were chosen because they have been used most often in prior NILM work. A lower NEP indicates better performance, while a higher F-score and FTE indicate better disaggregation accuracy. The evaluation was performed on a laptop with a 2.3 GHz i7 processor and 8 GB RAM running Linux.

Table 3 summarises the results of the two algorithms across the six data sets. It can be observed that FHMM performance is superior to CO performance across the three metrics for both REDD and AMPds. This confirms the theoretical foundations proposed by Hart [16]; that CO is highly sensitive to small variations in the aggregate load. The FHMM approach overcomes these shortcomings by considering an associated transition probability between the different states of an appliance. However, it can be seen that CO performance is similar to FHMM performance in iAWE, Pecan Street, Smart* and UKPD across all metrics. This is likely due to the fact that very few appliances contribute more than 5% of the household aggregate load in the selected households. For instance, space heating contributes very significantly (about 60% for a single air conditioner which has a power draw of 2.7 kW in the Pecan Street house and about 35% across two air conditioners having a power draw of 1.8 kW and 1.6 kW respectively in iAWE). As a result, these appliances are easier to disaggregate by both algorithms, owing to their relatively high power demand in comparison to appliances such as electronics and lighting. However, in the UKPD house the washing machine was one of the appliances contributing more than 5% of the household aggregate load, which brought down overall metrics across both approaches.

Another important aspect to consider is the time required for disaggregation and training, again reported in Table 3. These timings confirm the fact that CO is exponentially quicker than FHMM. This raises an interesting insight: In households such as the ones used from Pecan Street and iAWE in the above analysis, it may be beneficial to use CO over a FHMM owing to the reduced amount of time required for disaggregation and training, even though FHMMs are in general considered to be more powerful. It should be noted that the greater amount of time required to train and disaggregate the AMPds data is a result of the data set containing one year of data, as opposed to the Pecan Street data set which contains one week of data, as shown earlier in Table 1.

Appliance	NEP		F-score	
	CO	FHMM	CO	FHMM
Air conditioner 1	0.3	0.3	0.9	0.9
Air conditioner 2	1.0	1.0	0.7	0.7
Entertainment unit	4.2	4.1	0.3	0.3
Fridge	0.5	0.5	0.8	0.8
Laptop computer	1.7	1.8	0.3	0.2
Washing machine	130.1	125.1	0.0	0.0

Table 4: Comparison of CO and FHMM across different appliances in iAWE data set

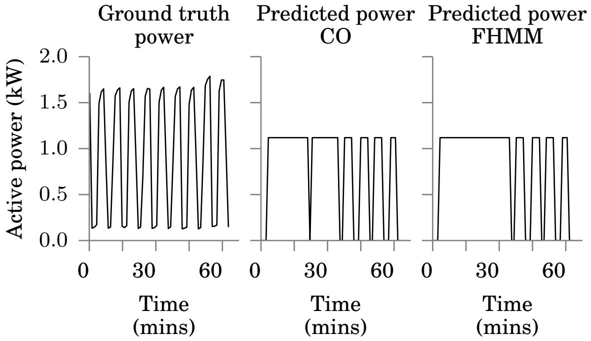

Figure 8: Comparison of predicted power (CO and FHMM) with ground truth for air conditioner 2 in the iAWE data set

4.5 Detailed Disaggregation Results

Having compared disaggregation results across different data sets, we now give a detailed discussion of disaggregation results across different appliances for a single house in the iAWE data set. Table 4 shows the disaggregation performance across the top 6 energy consuming appliances, in which each appliance is modelled using using 2 states as before. It can be seen that CO and FHMM report similar performance across all appliances. We observe that the results for appliances such as the washing machine and switch mode power supply based appliances such as laptop and entertainment unit (television) are much worse when compared to HVAC loads like air conditioners across both metrics. Prior literature shows that complex appliances such as washing machines are hard to model [4].

We observe that the performance accuracy of air conditioner 2 is much worse than air conditioner 1. This is due to the fact that during the instrumentation, air conditioner 2 was operated at a set temperature of 26 °C. With an external temperature roughly $5 - 10\,^{\circ}$C below this set temperature, this air conditioner reached the set temperature quickly and turned off the compressor while still running the fan. However, air conditioner 1 was operated at 16 °C and mostly had the compressor on. Thus, air conditioner 2 spent much more time in this intermediate state (compressor off, fan on) in comparison to air conditioner 1. Figure 8 shows how both FHMM and CO are able to detect on and off events of air conditioner 2. Since air conditioner 2 spent a considerable amount of time in the intermediate state, the learnt two state model is less appropriate in comparison to the two state model used for air conditioner 1. This can be further seen in Figure 8, where we observe that both FHMM and CO learn a much lower power level of around 1.1 kW, in comparison to the rated power of around 1.6 kW. We believe that this could be corrected by learning a three state model for this air conditioner, which comes at a cost of increased training and disaggregation computational and memory requirements. The air conditioner set temperature details quoted above were provided by the iAWE authors as a part of appliance metadata in their data set release.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed NILMTK; an open source toolkit designed to allow empirical comparisons to be made between existing energy disaggregation algorithms. The toolkit defines a common data format, NILMTK-DF, provides parsers from seven publicly available data sets to NILMTK-DF, and provides data set statistics and preprocessing functions to identify and mitigate common problems with NILM data sets. In addition, the toolkit includes implementations of two benchmark disaggregation algorithms based on combinatorial optimisation and the factorial hidden Markov model. Furthermore, NILMTK includes implementations of a set of performance metrics which will enable future research to directly compare disaggregation approaches through a common set of accuracy measures.

We have also demonstrated the analyses which NILMTK can provide across a range of publicly available data sets. We showed how the statistics functions can be used to detect any missing data. We then showed how various appliance models could be learnt from sub-metered power data. We also demonstrated how disaggregation algorithms can be applied to multiple data sets and their performance compared. Last, we provided a breakdown of each algorithm's performance by individual appliances.

Future work will focus upon the addition of recently proposed training and disaggregation algorithms. For instance, both benchmark implementations included in NILMTK are well studied algorithms which require submetered power data from each household for a supervised training phase, while algorithms proposed in more recent research [21, 28] only require aggregate data for an unsupervised training phase.

We have made some initial steps towards creating a metadata schema for appliances. However, we believe the task of maintaining a comprehensive, communal schema may be better suited to a semantic wiki.

An additional direction for future work would be the inclusion of a household simulator (e.g. [25]) within NILMTK. Since all data sets represent a limited number of households, it is currently impossible to test how a disaggregation algorithm might perform in a household other than those present in existing data sets. However, a simulator would overcome this problem by generating data for new households by either combining appliance data from multiple households or simulating appliances using detailed appliance models.

6. REFERENCES

[1] K. Anderson, M. Berges, A. Ocneanu, D. Benitez, and J. Moura. Event detection for non intrusive load monitoring. In *Proceedings of 38th Annual Conference on IEEE Industrial Electronics Society*, pages 3312–3317, 2012.

[2] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Bergés. 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, Beijing, China*, pages 12–16, 2012.

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

[4] S. Barker, S. Kalra, D. Irwin, and P. Shenoy. Empirical characterization and modeling of electrical loads in smart homes. In *International Green Computing Conference*, pages 1–10. IEEE, 2013.

[5] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht. Smart*: An open data set and tools for enabling research in sustainable homes. In *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, Beijing, China, 2011.

[6] N. Batra, P. Arjunan, A. Singh, and P. Singh. Experiences with occupancy based building management systems. In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pages 153–158, 2013.

[7] N. Batra, H. Dutta, and A. Singh. INDiC: Improved Non-Intrusive load monitoring using load Division and Calibration. In *International Conference of Machine Learning and Applications*, Miami, Florida, USA, 2013.

[8] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava. It's different: Insights into home energy consumption in india. In *Proceedings of the Fifth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '13, 2013.

[9] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

[10] California Public Utilities Commission. Final Opinion Authorizing Pacific Gas and Electric Company to Deploy Advanced Metering Infrastructure. Technical report, 2006.

[11] S. Darby. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, 486:2006, 2006.

[12] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 197–210. ACM, 2010.

[13] Department of Energy & Climate Change. Smart Metering Equipment Technical Specifications Version 2. Technical report, UK, 2013.

[14] Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273, 1997.

[15] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[16] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[17] C. Holcomb. Pecan street inc.: A test-bed for nilm. In *International Workshop on Non-Intrusive Load Monitoring*, Pittsburgh, PA, USA, 2012.

[18] M. J. Johnson and A. S. Willsky. Bayesian Nonparametric Hidden Semi-Markov Models. *Journal of Machine Learning Research*, 14:673–701, 2013.

[19] A. Kavousian, R. Rajagopal, and M. Fischer. Determinants of residential electricity consumption: Using smart meter data to examine the effect of climate, building characteristics, appliance stock, and occupants' behavior. *Energy*, 55(0):184 – 194, 2013.

[20] J. Kelly and W. Knottenbelt. Smart meter disaggregation: Data collection & analysis. Poster at UK Energy Research Council Summer School, 2013.

[21] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han. Unsupervised Disaggregation of Low Frequency Power Measurements. In *Proceedings of the 11th SIAM International Conference on Data Mining*, pages 747–758, Mesa, AZ, USA, 2011.

[22] J. Z. Kolter and T. Jaakkola. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1472–1482, La Palma, Canary Islands, 2012.

[23] J. Z. Kolter and M. J. Johnson. Redd: A public data set for energy disaggregation research. In *proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, San Diego, CA, USA, 2011.

[24] D. Kotz and T. Henderson. Crawdad: A community resource for archiving wireless data at dartmouth. *Pervasive Computing, IEEE*, 4(4):12–14, 2005.

[25] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng. Load Signature Study - Part II: Disaggregation Framework, Simulation, and Applications. *IEEE Transactions on Power Delivery*, 25(2):561–569, 2010.

[26] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Graphlab: A new parallel framework for machine learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, CA, USA, 2010.

[27] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajic. AMPds: A Public Dataset for Load Disaggregation and Eco-Feedback Research. In *IEEE Electrical Power and Energy Conference*, Halifax, NS, Canada, 2013.

[28] O. Parson, S. Ghosh, M. Weal, and A. Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 356–362, Toronto, ON, Canada, 2012.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[30] D. Rahayu, B. Narayanaswamy, S. Krishnaswamy, C. Labbe, and D. P. Seetharam. Learning to be energy-wise: Discriminative methods for load disaggregation. In *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on*, pages 1–4, 2012.

[31] Richard de Dear and Melissa Hart. Appliance Electricity End-Use: Weather and Climate Sensitivity. Technical report, Sustainable Energy Group, Australian Greenhouse Office, 2002.

[32] A. Schoofs, A. Guerrieri, D. T. Delaney, G. O'Hare, and A. G. Ruzzelli. ANNOT: Automated Electricity Data Annotation Using Wireless Sensor Networks. In *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks*, Boston, MA, USA, 2010.

[33] M. Wytock and J. Zico Kolter. Contextually Supervised Source Separation with Application to Energy Disaggregation. *ArXiv e-prints:1312.5023*, 2013.

[34] M. Zeifman. Disaggregation of home energy display data using probabilistic approach. *IEEE Transactions on Consumer Electronics*, 58(1):23–31, 2012.

[35] J.-P. Zimmermann, M. Evans, J. Griggs, N. King, L. Harding, P. Roberts, and C. Evans. Household electricity survey. a study of domestic electrical product usage. Technical Report R66141, DEFRA, May 2012.

APPENDIX

A. CASE STUDIES

In this section we consider two deployment scenarios where NILMTK is being used to perform disaggregation. Firstly, we consider the faculty housing deployment at IIIT Delhi [6]. 26 apartments have been instrumented individually with smart meters. Data from these households is being aggregated into a sMAP [12] server instance residing inside IIIT Delhi. Aggregate house electricity data from sMAP is pulled via HTTP request and is fed into NILMTK via a converter. The second case study was a single household deployment in Delhi where the data from a Schneider Electric smart meter was collected using a low power Raspberry Pi. In both these cases, rated power of HVAC based appliances such as air conditioners and always on appliances such as refrigerator were used to specify the model. The model was imported using CO disaggregator and disaggregation performed.

B. NILMTK-DF

We now provide the details of NILMTK-DF. Figure 9 shows NILMTK hierarchical structure for modelling physical hierarchies. Each dataset consists of one of more households. Each house may comprise of sensor broadly divided as utility (eg. power), ambient (eg. temperature) and external sensors (eg. outside weather). Wherever available, we also store meta data for a household such as area, floor, etc. Across, utilities, we focus mainly on electrical data, which is divided into mains (coming from grid), panel (circuits) and appliances. Each of these can store multiple physical measurement quantities such as active power, voltage etc. Further, wherever available, we also store appliance meta data such as rated power, details of instrumenting sensor, etc.

C. SAMPLE CODE FOR NILMTK PIPELINE

We now illustrate the NILMTK pipeline via a minimal code example.

```
dataset = DataSet()

#Load the dataset
dataset.load_hdf5(DATASET_PATH)

#Load first house
building = dataset.buildings[1]

#Remove records where voltage<160
building = filter_out_implausible_values(
    building, Measurement(‘voltage’, ‘’), 160)

#Downsample to 1 minute
building = downsample(building, rule=’1T’)

# Choosing feature for disaggregation
DISAGG_FEATURE = Measurement(‘power’, ‘active’)

# Dividing the data into train and test
train, test = train_test_split(building)

# Train on DISAGG_FEATURES using FHMM
disaggregator = FHMM()
disaggregator.train(train,
            disagg_features=[DISAGG_FEATURE])

# Disaggregate
disaggregator.disaggregate(test)

# F1 score metric
f1_score = f1(disaggregator.predictions,
            test)
```

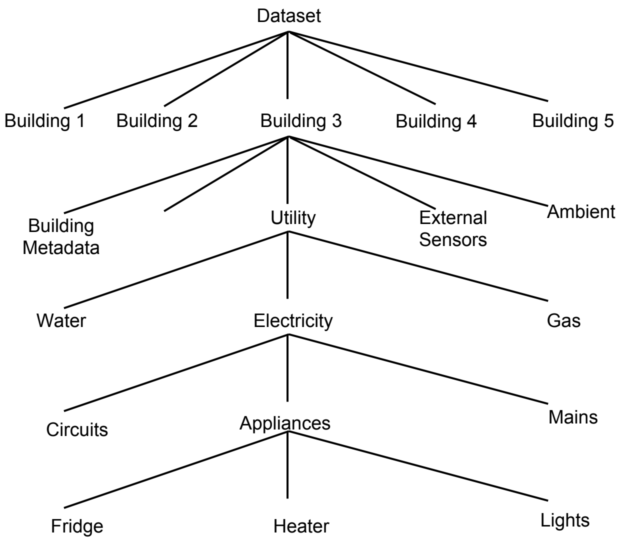

Figure 9: NILMTK-DF format hierarchy

D. ADDING A NEW NILM ALGORITHM

Every algorithm in NILMTK needs to define the following four functions:

train : Parameters of this function are the `building`; a list of `disaggregation features` (e.g. `[active power]` or `[active power, apparent power]`; `aggregate stream` (e.g. mains) and `sub metered stream`(e.g. appliances or circuits) The parameter style is inspired from R style formulae for linear regression.

disaggregate : This function takes as input a ‘building‘ and based on the ‘aggregate’ and ‘sub metered’ stream chosen in training phase, must produce a disaggregated stream for individual appliances.

import model : This function should imports from a JSON model to NILMTK disaggregator.

export model : This function writes the learnt model to a JSON file.

E. STATISTICS FUNCTIONS

The following paragraph should probably be in a table. Furthermore, the toolkit can calculate the distribution of appliance usage over repeating time periods, correlations between an appliance and weather conditions or across appliances, and the distributions of appliance power demands, on-durations and off-durations. Furthermore, NILMTK provides functions to calculate the proportion of time slices where a minimum percentage of the total energy has been sub-metered and also produces a list of the top-k appliances across the time slices.

F. QUERY EXAMPLES