Coursework Declaration and Feedback Form

The Student should complete and sign this part

Student Number: 2704250F	Student Name: Benjamin Frazer
Programme of Study (e.g. MSc in Electronics and Electrical Engineering): MSc in Electronics and Electrical Engineering	
Course Code: ENG5059P	Course Name: MSc Project
Name of **First** Supervisor: Dr Benoit Couraud	Name of **Second** Supervisor: Dr David Flynn
Title of Project: Large Scale Non Intrusive Load Monitoring	

Declaration of Originality and Submission Information

I affirm that this submission is all my own work in accordance with the University of Glasgow Regulations and the School of Engineering requirements Signed (Student) : Benjamin Frazer	
Date of Submission : 19/08/2022	

Feedback from Lecturer to Student – to be completed by Lecturer or Demonstrator	
Grade Awarded: Feedback (as appropriate to the coursework which was assessed):	
Lecturer/Demonstrator:	Date returned to the Teaching Office:

Abstract

The ongoing energy transition on the UK grid will lead to an increase in heat-pump load on the electrical network, presenting an opportunity to utilise this flexible load for demand-side response (DSR). Efficiently quantifying the available flexibility could be aided by developments in the field of non-intrusive load monitoring. Non-intrusive load monitoring (NILM) is the process of determining individual appliance loads from an analysis of an single aggregated point of measurement, such as a smart meter. The NILMTK is an open source repository of tools, algorithms and datasets targeted to wards appliance disaggregation research. NILMTK is structured towards appliance disaggregation at high sample rates (1-0.5Hz) and at only the individual household level, while the existing NILM research datasets have few heat-pumps. In this work, a synthetic NILMTK compatible dataset is presented, targeted towards heat pump load at both the individual and multi household level. A selection of contemporary disaggregation algorithms are then bench-marked using NILMTK on this dataset under a variety of power aggregation levels and sample rates including those reflective of A) typical household smart meter sample rates B) a feeder level measurement.

The full methodology, structure and assumptions entailing dataset synthesis are discussed, along with the necessary modifications to the NILMTK and the structure of the experiments. It is shown that with high sample rates (1-0.5Hz) and with just a single household, heat pumps can be disaggregated with a high degree of accuracy. However, with feeder measurements containing greater that four households in aggregate the performance of all algorithms is shown to have degraded and offer little advantage over simple heuristic methods [1]. It is also shown that the predictive disaggregation performance at decreasing sample-rates, up to and including those of household smart meters, remains accurate though degrades rapidly as more households are introduced.

[1] For example Mean Disaggregation (Section 2.1.1)

Objectives

The high level objective of this research is to establish if existing[2] NILM algorithms presented in the literature are suitable for determining available heat pump based flexibility on the distribution system. As such, the high-level project aims to test the efficacy of existing NILM algorithms under measurement scenarios which are considered to be feasible and cost effective for system operators to deploy or leverage. The following are considered:

1. Single household meter measurements at sample-rates realistic for domestic smart meters (Ts=15min)
2. Local feeder level measurements with up to five downstream households [3]

Secondary outcomes and objectives include:

1. Understanding the effect of both lower sample-rates and increasing number of household in aggregate on individual algorithms and discovering specific limitations and or mitigation
2. Testing the limits of the NILM tool-chain under atypical conditions such as A) high power levels B) low sample-rates
3. Providing an open source NILM dataset for future heat pump disaggregation research

The experiments conducted in this work belongs to the category of a supervised learning problem, where the prior load profile of a given appliance is known and used to train a device model prior to evaluation. It should be noted that this type of problem is ideal, in-terms of prior knowledge about a given measurement point, and thus does not reflect a realistic scenario for the deployment of such a use case. Thus the conclusions drawn in this work should be taken to represent only a limited exploration into a subset of the many parameters which would need to be balanced in a large-scale deployment of such a technique.[4].

[2]It should be noted that only a subset of all algorithms are tested for various reasons that are elaborate in Section 3.2.3.

[3]More households could be tested but as will be shown such work would be of very little value.

[4]One logical continuation of this work would certainly involve testing the ability of algorithms to generalise to multiple devices.

Acknowledgements

I would like to thank both my supervisors for the opportunity to work on a fascinating and extremely timely topic, as well as all of the guidance they have given. I would also like to thank my friends for the many thought-provoking conversations, words of encouragement and help given. Finally I thank my family for their patience and support though the course of this work.

Glossary

CO Combinatorial Optimisation. 41, 42

DECC Department for Energy and Climate Change. 26

FHMM Factorial Hidden Markov Model. 41

NILM Non Intrusive Load Monitoring. 3
NILMTK Non Intrusive Load Monitoring Toolkit. 36

Power aggregation level The total load upstream of a given point of measurement, can me measured in terms of number of households or average power. 38, 39, 41–44, 46

RHPP Renewable Heat Premium Payment. 26

Contents

Chapter 1

Introduction

1.1 Context

Transitioning our society away from fossil fuels is expected to stress the electrical network in the following ways:

- The increase in intermittent generators on the network means that ensuring consistent energy at low cost will be challenging.
- More electric vehicles (EVs) and heat-pumps, as well as distributed generation, is expected to cause increased congestion and voltage excursions on the network.

An attractive low cost path to mitigate these challenges could lie in incentivising consumers to shift demand through a so called flexibility market. Specifically targeted at enabling this consumer-driven participation, OFGEM has introduced the role of a Distribution System Operator (DSO) as a replacement for the existing Distribution Network Operator (DNO) role. The primary distinction is that the DSO's will now act as neutral market facilitators as opposed to asset managers. OFGEM notes [1]:

> We want to achieve four strategic outcomes from our DSO reforms:
>
> 1. Clear boundaries and effective conflict mitigation between monopolies and markets.
> 2. Effective competition for balancing and ancillary services, and other markets.
> 3. Neutral tendering of network management and reinforcement requirements, with a level playing field between traditional and alternative solutions.
> 4. Strongly embedded whole electricity system outcomes.

1.2 Utility of large scale disaggregation in demand response

Balancing the need to reinforce increasingly constrained regions of the network against the purchase of consumer-driven flexibility will likely involve a running cost calculation specific to each region of network. These decisions will likely be predicated on factors such as the projected reinforcement costs, future demand, available demand side response (DSR) capacity, and the likely cost to purchase this DSR. It seems likely that planners will benefit from understanding the disaggregated loads and thus the DSR potential of a section of network, however, gathering this information through consumer surveys or sub-metering could be prohibitively costly, unreliable or time consuming.

A further usecase the system operator might find for disaggregated consumer load data might be verifying the type of DSR being provided. Consider that a DSR response involves a specified decrease in net power at the point of common coupling (PCC). The consumer may thus be incentivised to switch on embedded generation such as a diesel generator rather than turn down true demand. It is unclear whether the economics of DSR will ever make this kind of behaviour attractive, but it demonstrates that not all DSR capacity is necessarily equivalent and should, in the authors view, not be treated equally.

1.3 Non Intrusive Load monitoring

Non intrusive load monitoring (NILM[1]) or energy disaggregation is the process of estimating load profiles from an aggregated measurement of current voltage data. A usecase for this disaggregated load data might be to consumers manage their energy consumption. The George Harts seminal 1992 publication marked the start of the NILM field, though his work on the topic dates back to 1984 [2]. On NILM Hart notes:

> It is called nonintrusive to contrast it with previous techniques for gathering appliance load data, which require placing sensors on individual appliances, and hence an intrusion onto the energy consumer's property.

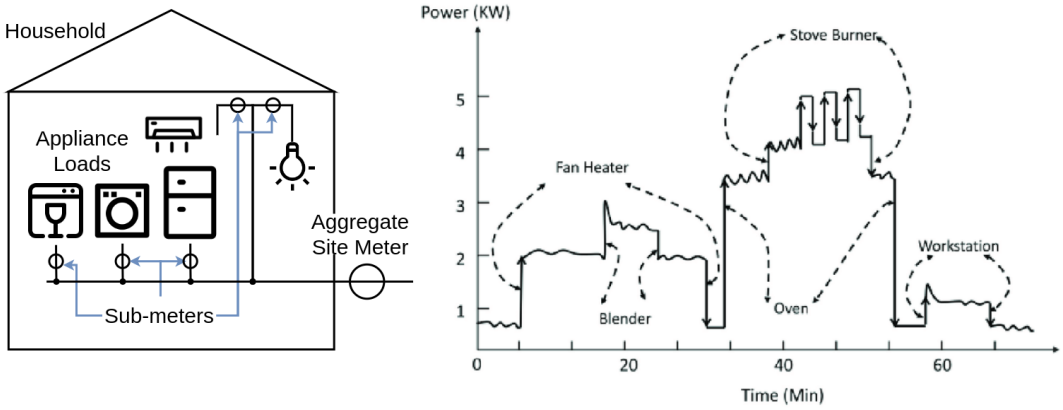

Figure 1.3.1: *Typical Household NILM problem, Adapted from [2]*

1.3.1 The NILM Problem

In its most general form NILM can be simply framed as a the aggregate measure $P(t)$ as a function of the individual appliance loads $P_i(t)$ summed with an error term.

$$P(t) = \sum_{i=1}^{I} P_i(t) + e(t) \tag{1.3.1}$$

As will be shown, the assumptions behind the $P_i(t)$ radically shape the approach to the NILM problem.

1.3.2 Appliance types

Hart groups, appliance types into one of three categories [2]:

On/off The device only occupies one of two states. Examples include toasters, lights etc.
Finite state machine (FSM) The appliance may occupy one of many finite states each with an asociated power level for example a washing machine with multiple cycles.
Continuously Variable Rather than having fixed states these devices have a continous range of outputs for example a power tool.

1.3.3 Signature Taxonomy

The approach used for any disaggregation task relies on the detection of one or more signatures. For example it might be possible to determine the on state of an appliance by measuring the amplitude of

[1] Also referred to as Non Intrusive Appliance Load Monitoring (NIALM)

a step in the agregate measure and comparing it a known value of it's rating. The size of this step step would thus be the devices a signiture. Figure 1.3.2 shows the taxomomy of signitures proposed by Hart.

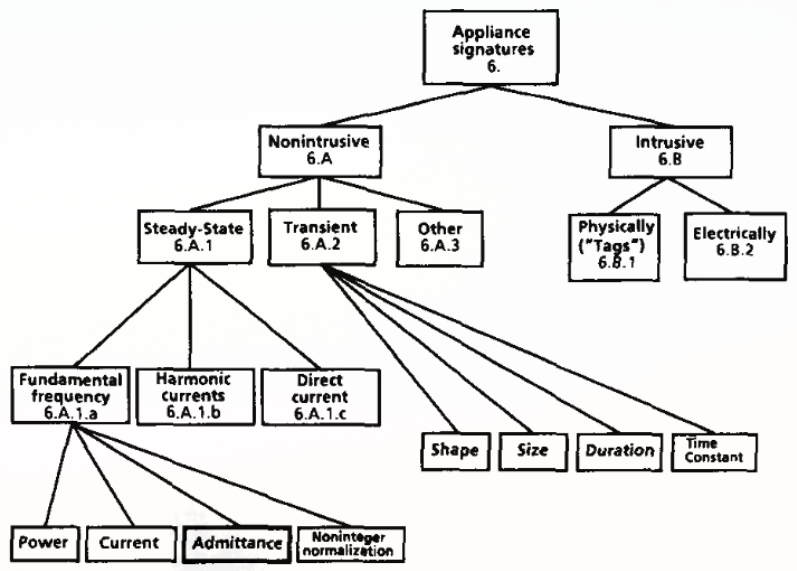

Figure 1.3.2: *Signature taxomomy proposed by Hart. Taken from [2].*

1.3.4 Disaggregation approaches

A broadest delineation of disaggregation approaches is likely on the basis of whether the appliance models allows/predicts:

1. Finite appliance states
2. Continuously variable

This delineation coincident also neatly separates between neural network based approaches since, to the author's knowledge, these are to date the only disaggregation algorithms that predict continuously variable appliances.

A second major delineation, is what type of signatures are leveraged for disaggregation. Combinatorial Optimisation, for example,[2] relies purely on additive steady state signatures, while FHMM[3] relies predominantly on transient signatures. With regard to neural network based approaches there is little to comment on beyond observing that the feature extraction occurs within the network and is thus hidden.

Finally the signatures which can be exploited are very much contingent on A) the type of physical units under measurement B) the sample rates of these measurements.

1.3.5 Learning Types

The learning refers to the nature in which the disaggregation algorithm builds the individual appliance models. In unsupervised learning, the algorithm must learn and label individual appliances based purely on the aggregate measurement. In supervised learning, the model is build up based on the sub-meter so called ground truth of each appliance load profile. Whereas supervised learning is often less complex and produces more accurate results, unsupervised reflects more closely a deployable solution given the likely changing combination of appliances within a typical household and impracticality of tailored sub-meter training data for every seperate household.

[2]See Section 2.1.2
[3]See Section 2.1.3

1.3.6 Switched Continuity Principle

The switched continuity principle posited by Hart states[2]:

> In a small time interval only a small number of appliances are expected to have changed state in a typical load.

Though difficult to encode explicitly, the switched continuity principle is an implicit assumption in any algorithm relying on transient event signatures. Furthermore it is a prerequisite of many unsupervised learning algorithms, since the clustering stage relies on grouping events resultant from only individual devices.

1.4 Non Intrusive Load Monitoring Toolkit (NILMTK)

The non intrusive monitoring toolkit (NILMTK)[4] is an open source software project written in Python aimed at providing a set of tools, test data and unified framework for benchmarking NILM algorithms [3]. Prior to the release of NILMTK, much of the NILM research was conducted on artificial datasets specific to the work. Further more many publications did not disclose source code for these algorithms again making results and variety of metrics are used to evaluate performance. All of this limited both the clear esablishment of what the leading approaches were and the ability to build on existing work. The main goals of NILMTK were thus:

1. to allow disaggregation algorithm performance to be easily reproduced
2. Enable newly algorithms to be easily and uniformly benchmarked against their contemporaries
3. Centralize common tools such as dataset parsers, preprocessed and inspection tools to lower the barrier to energy for new NILM research.

1.4.1 NILMTK terminology

Both the NILMTK and the NILM field more broadly use a controlled set of terminology with specific well defined meanings. This section attempts to highlight and define some of these terms:

Site-meter A measurement point upstream of all appliance loads in a given household

sub-meter A measurement point down stream of the site meter usually, but not exclusively, upstream of a single appliance.

Household In the context of the NILMTK this is a set of meters which are grouped downstream of a given site meter.

Ground Truth (GT) The true appliance load profile for a given time period, used for evaluation and training. Given by a sub-meter measurement just upstream of the given appliance.

Aggregation This refers to the combination of individual appliance loads into a single upstream point of measurement. In this work this is extended to also mean combined household loads, and combined samples (i.e. down-sampling).

disaggregation The process of splitting a single point of power/reactive power measurement into the individual appliance load profiles

1.4.2 Features

1. Standardised schema and datasets parsers for widely used NILM datasets, allowing datasets to be used interchangeably
2. Standardised schema for new disaggregation algorithms allowing any two algorithms to be used interchangeably

[4] `http://github.com/nilmtk/nilmtk`

3. Standardised set of performance metrics
4. Implementation of three benchmark NILM algorithms [5]
5. An API, which allows for rapid definition of NILM experiments across datasets and algorithms [4].

1.4.3 NILMTK Contrib

The NILMTK-contrib repository[6] acts as a location where researchers can contribute new NILM algorithms for others to test [4]. As such the algorithms it contains here are typically the most cutting edge, however with this addition content also comes an increased complexity in dependencies that must be navigated.

[5] Combinatorial Optimisation, Mean and Factorial Hidden Markov Model
[6] `https://github.com/nilmtk/nilmtk-contrib`

Chapter 2

Literature Review

2.1 NILM Algorithms

This section aims to provide an overview of NILM disaggregation algorithms presented in the literature that have been used in this work.

2.1.1 Mean

This algorithm predicts by taking the mean of the power in the training set over all time and simply predicting a consistent on state at this magnitude. Though this is an exceedingly primitive approach it is included as a usefully benchmark for other algorithms.

2.1.2 Combinatorial Optimization

2.1.2.1 Use in energy disaggregation

Combinatorial optimization (CO) is a widely studied disaggregation algorithm primarily used a to benchmark other algorithms. The first appearance of combinatorial optimization in the NILM literature is in Hart's seminal 1992 paper where it was introduced as examination of potential formulations of the problem [2].

2.1.2.2 Theory

In CO, the aggregate power at time t is defined as the combination of a set of n appliances each with a finite set of states K where each state is controlled by a binary switching vector $a_{n,k}(t)$ of length K where all but one member of a is zero [1]. The optimization is thus formulated as the following minimisation problem:

$$\hat{a}(t) = \underset{a}{\operatorname{argmin}} \left| P_{agg}(t) - \sum_{i=1}^{n} \sum_{k=1}^{K} a_{i,k} P_{i,k} \right| \tag{2.1.1}$$

Intuitively this can be thought of as finding the combination of allowed appliance powers that gets the closest to the value of the aggregate power.

2.1.2.3 Limitations

The primary limitations relate to the assumptions inherent to the formulation such as:

1. The assumption that all appliances and their possible states are know

[1] Think of this term as a bit mask that selects only one of the K powers represented in $P_{i,k}$

2. The assumption that all devices only occupy a finite set of states
3. Inability to discriminate between appliances with the same power level
4. Not adherence to the switched continuity principle (see Section 1.3.6)
5. Exponential complexity as the number of appliances and/or the number of states increases

Arising from the lack of adherence to the switched continuity principle, with high numbers of possible states, through either large numbers of devices and/or large numbers of states per device, small changes in aggregate power will lead to simultaneous reconfigurations of states for multiple devices simultaneous. This results in decreasing disaggregation performance with higher appliance numbers and simultaneously a higher sensitivity to noise [5].

2.1.3 Factorial Hidden Markov Model

2.1.3.1 Use in Energy disaggregation

The FHMM was first proposed for use in energy disaggregation by [6], and has since been and studied extensively for this application [7, 4].

2.1.3.2 Theory

Hidden Markov Model Hidden Markov Models (HMM) have found used for stochastically modelling the output sequential systems which can be adequately modelled with a finite set of states. HMMs have been successfully applied in diverse fields such as economics, gesture recognition, computational biology and many more.

A hidden Markov model consists of a Markov chain of finite states each of which is characterized by a probability distribution of observations at that particular state. A graphical representation is shown in Figure 2.1.1.

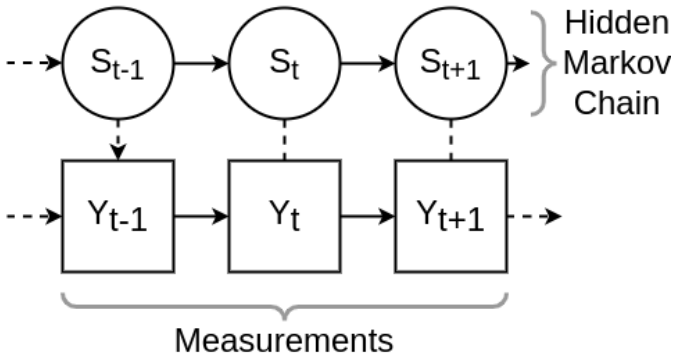

Figure 2.1.1: *Hidden Markov Model*

Factorial Hidden Markov Model The factorial HMM is introduced as an extension to the HMM by Ghahramani and Jordan [8] with the aim of modelling multiple hidden Markov chains. A graphical representation is shown in Figure 2.1.2.

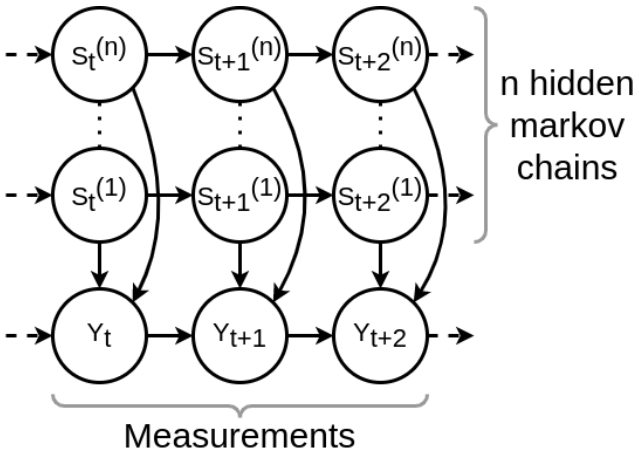

Figure 2.1.2: *Factorial Hidden Markov Model*

Each of the n hidden Markov chains has a discrete set of states $S_{i,t}$ at time t with a transition probability distribution P_i to move to the next state. At each time-step the state "emits" a power $y_t^{(n)}$ according to a probability distribution which is a function of the state. Emissions are summed with those emission of the other Markov chains emission at a given time. Only one state S_i may be occupied at any given time for a given chain. The output observation y_t is generated from a random variable whose probability distribution is given by a combination of each state occupied at time t.

Given all of the model parameters the most probable chain of hidden states can be inferred using the Viterbi algorithm from the set of observations Y_t [9]. The limitation here is that the complexity of the Verbiti algorithm grows exponentially with each additional appliance/state added. While manageable for small numbers of appliances, the inference computation becomes tractile for appliance numbers greater than ≈20 [7]. To create the FHMM for a given household, the individual model parameters must first be learnt. Parameters for each appliance can be learned from a sample of their sub-meter information using the Expectation Maximisation algorithm[10].

2.1.3.3 Variants and Subtypes

Given the FHMMs ubiquity in the field on NILM research it is unsurprising that many variants and additions to this algorithm exist. This section will give an overview of just a few of the more relevant of such algorithms.

Exact In this formulation the individual appliances Markov Chains are combined into a single equivalent 'super' Markov model. This allows the Viterbi algorithm to be applied for the inference stage. This has the disadvantage (discussed above) of incurring exponential inference complexity as appliance numbers increase. This is also the formulation provided in the NILMTK, and as is what is used in this work.

Approx As mentioned above, the computational intractability of large numbers of states and/or appliances has encouraged research into inference methods which are though only approximate remain computationally tractile with large numbers of devices [11].

SAC The signal aggregation constraint (SAC) places an additional constraint at the inference stage on the aggregate value of each appliance load over a given time window. This can be thought of as enforcing a maximum average power for the estimate of each appliance [12].

2.1.3.4 Limitations

While demonstrably a powerful tool, FHMMs have some inherent disadvantages:

- A Markov assumes that the state transitions from a given state are completely independent from the state history. Thus is not directly possible to encode intuitive behaviour about patterns in the time history of the states[2].
- The Markov model is inherently limited to approximating appliance behaviour with a finite set of states. This is fine if the device has a very square load-profile, but suffers when confronted with continuously variable appliances.

2.1.4 Sequence to Sequence

The term sequence to sequence (Seq2Seq) was first coined by Zang et all [13] as an umbrella term to describe a class of algorithm where neural networks learns a regression map between an input sequence of the aggregate site-meter measurements to an output sequence of predictions of the same length and time alignment. Algorithms grouped in this class are those put forward by Kelly and Knottenbelt [14] which include the DAE discussed separately in Section 2.1.6.

Zang et all also propose an alteration to the architecture proposed by Kelly and Knottenbelt which uses the sliding window approach for both training and disaggregation. Following the NILMTK naming convention, the sequence to sequence algorithm refers **not** to a class of algorithms, rather to the specific variant proposed by Zang et all.

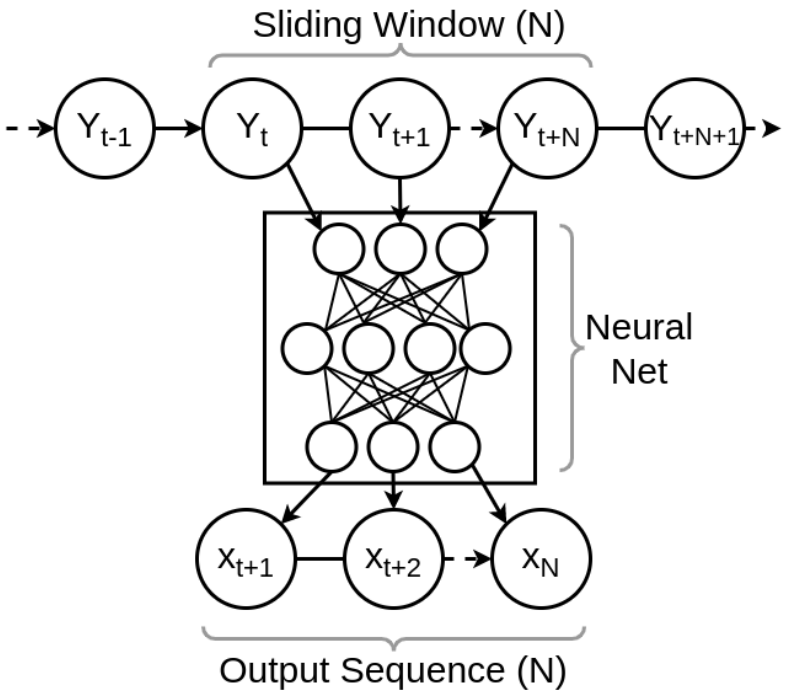

Figure 2.1.3: *Sequence to Sequence Disaggregation*

As the sliding window is moved a along the input sequence, the samples be subject to repeated predictions. To synthesized these multiple predictions both [14, 13] take the mean of all of the predictions.

[2]Consider an automatic light which comes on only for a fixed period: Human intuition would suggest that if the light is switched on, there is a high likely-hood, given prior observation, that it will turn off in a particular time-frame. This is impossible to directly encode in a Markov model

2.1.4.1 Known limitations

As shown by Zang et all. paper this architecture typically tends to perform worse than their proposed alternative: sequence to point (discussed in Section 2.1.5). Beyond this, seq2seq is still shown to preform in general substantially better than the vast majority of other contemporary algorithms tested in [4].

2.1.5 Sequence to point

As mention in Section 2.1.4, Sequence to point (seq2point) was originally proposed by Zang et all. as an alteration to the seq2seq disaggregation architecture. As before, a neural network operates on a sliding window of the aggregate site meter, this time outputting a single point prediction of the target appliance load profile in the centre of the time window.

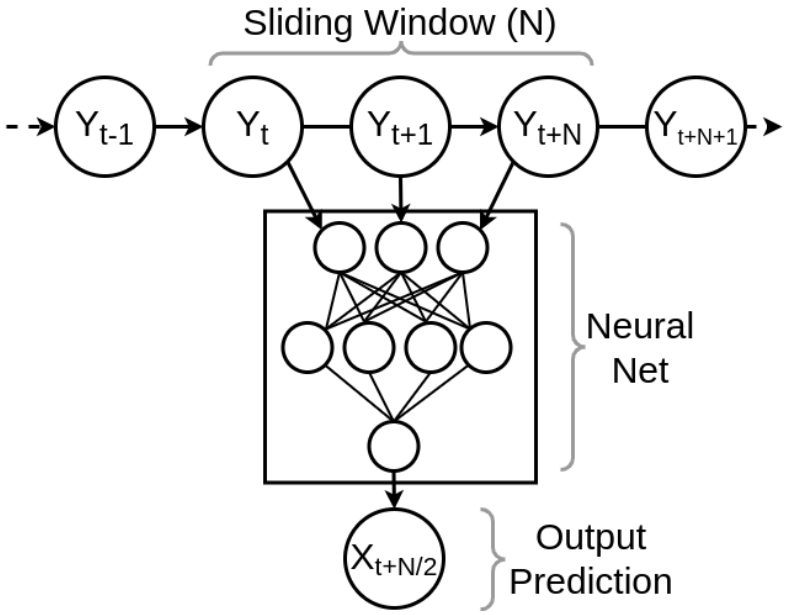

Figure 2.1.4: *Sequence to Point Disaggregation*

The intuition here is that the sample at the centre of the time window is naturally the optimal placement for estimation since the neural network has information about both the preceding and following sequence.

The Seq2Point algorithm has been shown to be the best performing contemporary disaggregation algorithm on a variety of appliance classes and datasets[4].

2.1.6 De-noising Auto encoder

Proposed by Kelly and Knottenbelt [14], the de-noising auto-encoder is a class of neural network which attempts to reconstruct its input given an assumed source of corruption. In the case of NILM, a fixed window of the aggregate measurement is assumed to be the corrupted signal while the target appliance load profile is the desired uncorrupted signal to be reproduced. While this algorithm falls under the definition of sequence to sequence, it will henceforth be referred to as De-noising Auto Encoder (DAE) in keeping with NILMTK.

2.2 Public NILM Datasets

The first publicly available NILM dataset Reference Energy Disaggregation Dataset (REDD) was presented in 2011, specifically targeted towards NILM research [7]. The dataset includes both aggregate site meter

and labelled sub-meter information for a variety of appliances. The dataset included detailed metadata on device type, number, rating etc. Given it's ready availability and well considered structure, this dataset very quickly emerged as the favoured dataset for NILM research and thereby set the standard for subsequent datasets. Indeed, much of the structure of this dataset originally dataset is still reflected in the current NILMTK dataset schema. Since REDD's release and, in-part, due to decreasing costs of meter devices, the number of available datasets has expanded drastically. Given the objectives of this work is the evaluation of NILM algorithms on specifically heat pump loads, a dataset is required which contains these devices. Ideally this dataset would contain greater than five households each containing a heat pump. An objective of this work is to test these algorithms with increasing numbers of households in aggregate to simulate feeder level measurements. Given A) specificity of the dataset required to meet this goal [3] B) The fact that all NILM datasets[4] are structured towards disaggregation of single households, it seems likely that some test data will need to be at least synthesised. The ideal dataset would thus provide a single source for five heat pump and household load profiles with sub-meter measurements for at the very least all heat pumps which would then form the basis for a synthesized dataset. In an attempt to find a dataset satisfy the requirements listed above, a comprehensive survey of 18 known NILM datasets has been undertaken. Though specifically searching for numbers of heat pumps, for the benefit of potential future work, a tally of other potentially flexible loads[5] was also recorded along with general dataset metrics such as:

- Compatibility with NILMTK
- Inclusion of sub-meter data
- Location
- Building type (Residential or Commercial)
- Number of properties
- Time period of study
- Sample rate/Resolution
- Channels (Types of measurement recorded)

Sources for these metrics included:

- Supporting publication(s) [6]
- Embedded dataset meta[7]
- NILMTK dataset converters stored metadata [8]

A subset of the survey results is presented in Table 2.2.2, filtered by A) The inclusion of sub-meter measurements B) Number of properties studied is greater than 1. A supporting reference key is shown in Table 2.2.1. Furthermore, only a restricted set of metrics are included. Both the full table as well as supporting notes on individual datasets are included in Appendix 7.11.

[3] Ideally a dataset with measurement points with various numbers of households in aggregate, and with one household containing a heat pump.

[4] To the Author's knowledge

[5] EV's, electric heaters, air conditioning (AC) units

[6] Supporting publication referenced are included within the Table 7.11.2

[7] Sources for download can be found for each dataset in Appendix 7.11

[8] In some cases `.yaml` files describing the dataset are stored with the dataset converter.

Table 2.2.1: *Table 2.2.2 Key*

Symbol	Description
R	Residential
C	Commercial
P	active power
Q	Reactive Power
V	Voltage
I	Current
S	Apparent Power
f	frequency
Θ	Current/Voltage Phase shift
+	Indicates count is incomplete

Table 2.2.2: *Sample of NILM Datasets Survey with more than one household and suitable sub-metering. Full Survey of 18 NILM datasets Found in Appendix 7.11.*

Name	NILMTK?	Properties	Period	Resolution	Channels	Heat-Pumps	cite
Dataport	Y	75	3.25y/1m	1min	P	0	[15]
Smart	Y	3-5?	90d	1min	PS	1	[16]
DEDDIAG	Y	15	3.5y	1Hz	P	1	[17]
REDD	Y	6	19d	15kHz/3s[9]	SVP	0	[7]
UK-Dale	Y	5	499d	1s/6s [19]	SPV	0	[18]
Ideal	Y	39	?	1Hz	S/P[10]	0	[19]
REFIT	Y	20	21m	8s	P$	0	[20]
HES	Y	225/26[11]	1m/1y[21]	2min	P	0	[21]
GREEND	Y	9	310d	1Hz	P	0	[22]
ECO	Y	6	244d	1Hz	PIVΘ	0	[23]

With no known NILM datasets including more than a single heat pump, and the total over all datasets amounting to only four, the criterion laid out above clearly cannot be satisfied. As will be discussed in subsequent sections, this finding has motivated the creation of a synthetic NILM which will be presented as a part of this work.

[19]Here the site and sub-meter are recorded at different resolution. Both resolutions are displayed as follows `<site meter resolution>/<sub-meter resolution>`.

[20]Site meter is measures apparent power while sub-meters measure real power

[21]In this study 255 houses were studies for 1 month and 26 were studied for a year.

Chapter 3

Implementation

The goal of the experiments conducted in this work is to test algorithm performance under a limited subset of variables which might affect the wide-scale deployment of NILM for the disaggregation of heat pumps. This section aims to give an overview of the methodology, procedure and assumptions comprising these experiments. This can be grouped into the following sections:

1. Experimental Design (Section 3.2)
2. Tools used (Section 3.1)
3. Creation of a synthetic test dataset (Section 3.3)

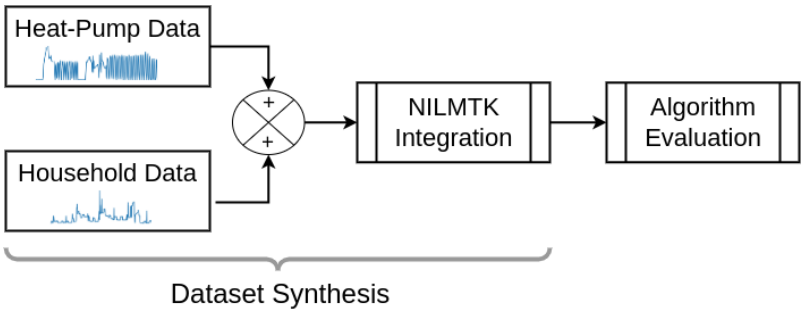

Figure 3.0.1: *High level Diagram showing the main algorithm evaluation stages.*

3.1 Tools/Environment

The NILMTK is written exclusively in Python, and beyond it's other advantages it makes this the obvious choice for tasks such as data processing and analysis and presentation. Table 3.1.1 lists the versions of the libraries used.

Table 3.1.1: *Development tools/libraries used in the dataset synthesis/algorithm testing*

Name	Version
Python	3.8
Pandas	0.25.3
Numpy	1.19.3
NILMTK*	0.4.3
NILMTK-contrib**	0.1.2

*It should be noted that difficulty was faced installing both NILMTK and NILMTK-contrib and though an anaconda installation appears to be the best supported approach, this package was not found to work on Manjaro Linux `5.15.50-1`. These packages were instead installed using `pip`, however this approach incurred several dependency problems which the `NILMTK-contrib` package, which were only fixed by editing the package source. A full write-up of the approach used for this work is included in Appendix 7.6.

**The author's fork of the NILMTK is used in this work which includes several improvements in the way in which the experimentation API handles result. These changes were necessitated by the combatively large volume of results data generated over the course of this work. These changes are non breaking and have been build with reintegration into NILMTK in mind.

3.2 Experimental Design

The parameters/factors affecting how the selected algorithms[1] will perform when deployment are numerous. Some examples include:

- The influence of increasing power aggregation level
- The influence of increasing temporal aggregation level
- Effects of generalised training vs training on a single device [2]
- Effects of different household energy usage patterns
- Presence of other large appliances such as EVs
- Presence of multiple heat pumps
- Presence of embedded generation
- Effects of Seasonal variability
- Variability in performance between heat pumps

Thus, the scope of this work has been deliberately restricted to the consideration of what are expected to be the two most critical parameters, namely: temporal and power aggregation level. The designation of 'critical' in this case stems from the idea that success on a critical variable is a precondition of testing other parameters. If, hypothetically, it is discovered that disaggregation beyond three households is nonviable, then there it makes little sens to test the effect of generalisation at this level since this would only make things worse. Thus the selected parameters critical parameters can be though of as the best way of quickly shrinking the search space into a tractable problem where these other parameters can be tested. This approach however necessitates the remain parameters to be set to values which are considered to be ideal from the point of view of disaggregation performance. The configuration of remaining parameters used in *all* subsequent tests are detailed explicitly in Table 3.2.1.

[1] Covered in section 3.2.3

[2] The distinction between generalised an non generalise training *here* is that, in the case of generalised training, the device is deliberately trained on a representative subset of the appliance class (heat pumps in this case) with the goal of having acceptable performance on the total population of heat pumps. Non-generalised training(of the type).

Table 3.2.1: *Configuration of additional experimental factors/parameters used for all experiments.*

parameter/factor identified	configuration used
Other large appliances	Household with known EV's and electric heating filtered
Generalisation	Test individually on each heat pump
Household usage patterns	Same households used for every increment of power aggregation (See 3.2.2)
Embedded generation	Households with known embedded generation filtered
Multiple heat pumps	Only a single heat pump aggregate measurement (households with known heat pumps filtered)
Variability between heat pumps	Each dataset contains 5 heat pumps combined with identical household load
Seasonal variability	All source data is from months January-August, Jan-May for testing & August for training

The experiments described in this section are designed to examine the isolated effects both of these variables exert on the disaggregation performance of each algorithm. This however leaves the decision of what configuration the remaining factors should be set to

To extract a trend for a given variable, all other variables must be kept constant, while the variable in question is incremented testing disaggregation performance at each step. This 'trend extraction' procedure will henceforth be referred to as a test-sweep, while the procedure for running a single test of disaggregation will be referred to as a test-case.

3.2.1 Chosen Test sweeps

Given the high computational cost of running even a single test-case (≈20mins[3]) not all permutations of temporal and power aggregation are tested. The space of possible permutations of power and temporal aggregation, hence forth know as the 'pt area' is instead probed with the four test sweeps detailed in Table 3.2.2.

Table 3.2.2: *Test sweeps designed to probe the 'power-temporal area'*

Sweep variable	Tncrement	Boundary	Constant variable	Constant variable Value
Power Aggregation	1 house	1-5 houses	Temporal Aggregation	Ts:2min
Temporal	1 min	1-15min	Power Aggregation	1× Households
Temporal	1 min	1-15min	Power Aggregation	3× Households
Temporal	1 min	1-15min	Power Aggregation	5× Households

These sweeps are chosen in an attempt to best explore the 'pt area' while keeping the computation tractable.

3.2.2 Dataset structure

NILMTK has support for *on the fly*[4] resampling of a dataset, however, does not support combining appliance/ site-meter data from multiple households into a single *super* household. Through modifications

[3]This drops off approximately linearly with sample rate thus 15 samples at increasing sample rate is substantially less than $15 \times 20 = 5$h

[4]In this case *on the fly* means a synthetic dataset created in RAM at the time of running the experiment from arbitrary appliance loads from other NILMTK datasets stored on disk.

of the NILMTK source code, the author was able to achieve this *on the fly* aggregation between both individual households and dataset[5]. This work was discontinued since the limited coverage of NILM datasets for heat pumps necessitated the creation of a new synthetic dataset anyway.

Given that household loads may not be combined *on the fly*, the test sweeps discussed in section 3.2.1, require new load profiles for each aggregation level. Thus five separate datasets are created, identical in all but the number of household load profiles which are combined with the individual heat pumps. Seen from the NILMTK schema each dataset contains five individual 'households' which in actuality contain the aggregate load profile for N household load profiles $H_n = \{h_{n,0}, h_{n,t}, ...\}$ as well as a single heat pump load profile $P_N = \{p_{N,0}, p_{N,t}, ...\}$ (Described formally in equation 3.2.1).

$$y_t^{(N)} = \sum_{n=1}^{n=N} h_{n,t} + p_{N,t} \tag{3.2.1}$$

Equation 3.2.1 is represented graphically in Figure 3.2.1.

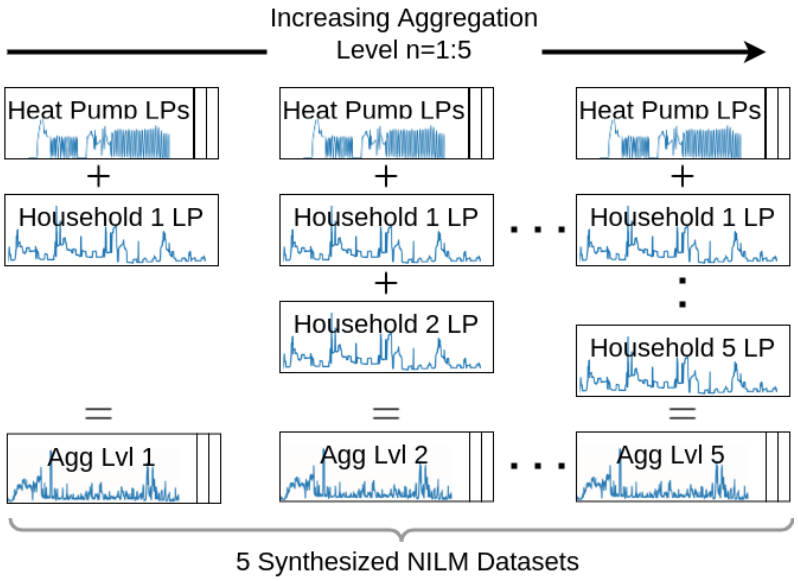

Figure 3.2.1: *Structure of the five datasets at increasing aggregation levels produced in as a part of this work. Here double bars represent a collection of five individual load profiles and addition occurs element wise for each load profile.*

3.2.3 Selected Algorithm Configuration

Where relevant, the specific configuration of each algorithm is listed.

CO Model configured with three power states
FHMM Mode configured with two power states
DAE No non-default configuration parameters
Seq2Seq No non-default configuration parameters
Seq2Point No non-default configuration parameters

[5]This feature however has yet to implement dataset alignment thus several limiting it's utility. This work can be found under the author's fork of NILMTK (See Section 7.1)

3.2.4 Training

In the interest of all other factors being 'optimal', training is conducted on only the heat pump which will be tested on. The algorithms are then also retrained ate every test-case to allow for optimal performance at each power/temporal aggregation level. As stated in Table 3.2.1, the effects of seasonal variability are mitigated by requiring that the source data for both household and heat pumps load profiles be taken from a consecutive calender month span of January to -August. Training is always conducted on months January-May while the evaluation always occurs on the month of August.

3.2.5 Seasonality and Data Alignment

Given the seasonal and daily variability inherent to heat pumps, it seems likely that the strength of signature and therefore the 'detectability' of a given appliance device will vary accordingly. Furthermore, it seems possible that the distribution of loads within the households relative to the activations of the heat pump may act to either obscure or reveal the heat pump's activity thus making it harder or easier to detect. This implies that care must be taken when aligning the heat pump and household data prior to combining them into a synthetic aggregate. For this reason the dataset should be created with both heat pump and household load profiles aligned to the same time of year and hour of the day prior combination. Additional, to make performance between the five individual heat pumps compatible, a minimum span of the months of January-April should, at minimum, be present within all datasets.

3.2.6 Unknown Embedded Household Loads

Ideally, the household data chosen to combine with the separate heat pump data sub meter data should be assumed to contain no embedded generation or loads such electrical heating and EV's, since the effects of such devices should be studied explicitly.

3.3 Dataset Synthesis

A hard requirement for any NILM dataset is the presence of both the aggregate 'site-meter' measurement as well as a sub-meter measurement just upstream of the heat pumps. This sub-meter measurement forms the ground truth against which the algorithms will be trained and evaluated. Further more, the design of the experiments discussed in Section 3.2 places additional specifications onto the synthetic heat pump datasets (covered in Section 3.3.3).

This section will cover each stage, of the data synthesis pipeline presented in Figure 3.3.1, however, for brevity this will be mostly limited to the broad approach and the assumptions inherent therein [6]. In Figure 3.3.1, data that is stored on disc, including intermediary and metadata, is denoted by a cylinders, filter stages are denoted by an inverted trapezoid and finally process stages are represented with double barred rectangles.

[6] Code has been made publicly available (See Appendix 7.1)

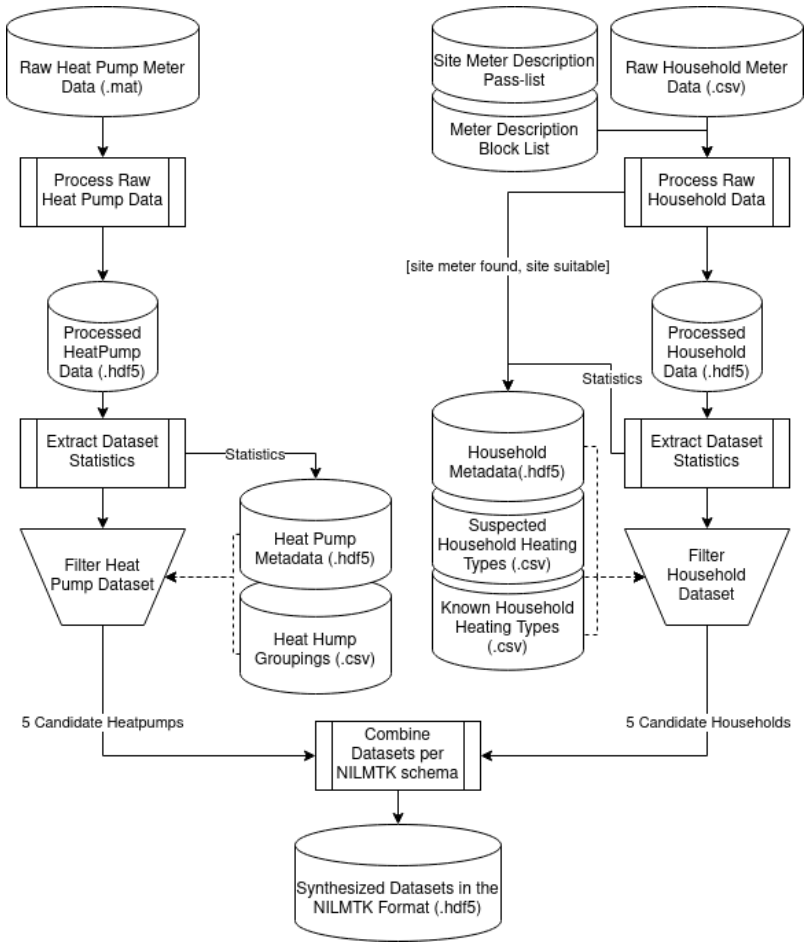

Figure 3.3.1: *High Level Data Synthesis Pipeline*

3.3.1 Strategy

The high level strategy take in the absence of real world data that exactly meet the requirements, is to take the required components from separate real world sources and combine them to create a synthetic dataset. The two major components here are of course the household and heat pump load profile. To create the new synthetic site meter these two load profiles can, with some consideration, be brought onto a common time base and then subjected to an element-wise summation. The pre-summation load profile of the heat pump then forms the ground truth for the NILM dataset.

3.3.2 Terminology

Subsequent sections rely on specific terminology, these terms are be defined explicitly here.

Span Defined as a range of time, in this case used in conjunction with load profile data
Valid Span Defined as the period of time between the first and last valid sample of a given load profile.
Completness Defined as the number of samples in a given span

3.3.3 Specifications

This section summarises the parameters/factors relevant to the dataset structure discussed in Section 3.2, which form the specifications against which the datasets are created.

1. Different household usage patterns - Heat pump and household load profiles are taken from the same calendar months of Jan-Aug, and aligned based on A) day of the year B) time of the day.
2. Variability between heat pumps - Household with known or suspected heat pumps, EV's, Storage Heaters and Other electric heating are removed from consideration (filtering stage)
3. Seasonal variability - Households with known or suspected embedded generation are removed for consideration (filtering stage)
4. Large electric appliances - The same households are used for each increment of power aggregation level.
5. Embedded generation - Five heat pumps with identical additional household load are tested with the performance averaged

In total five datasets are created, identical in all but the number households in aggregate per heat-pump. This is done in to satisfy specifications 1 and 2, as well as the requirement to test under increasing power aggregation increments.

3.3.4 Source Data Sets

This section will give an overview of the heat pump and household datasets which will form the basis of the synthesised dataset. The aim is to give context to the subsequent sections on data processing and filtering (3.3.7, 3.3.5).

3.3.4.1 Heat pump Dataset

Origin The Heat pump dataset used in this work has been sourced from the second of two major UK field trials of heat pumps conducted by the Department for Energy and Climate Change (DECC) and the Renewable Heat Premium Payment (RHPP). Of the approximately 14,000 heat pumps installed in this trial, around 700 were monitored primarily with the aim of studying the efficiencies of heat pumps operating in the UK. These measurements include measurements of heat pump electrical power consumption at 2 minute intervals [24]. It is noted in [24] that though the second field trial lasted between 2011-2014, the actual data ranges from 2013-2015.

Structure The data has been made available in both a raw and pre-cleaned state by the DECC, though for this work only the pre-cleaned set is used. Broadly speaking the dataset cleaning involved the filtering of data with systemic flaws, removal of duplicate timestamps and removal of highly incomplete data. More details of the cleaning are available can be found in the DECCs detailed report [25].

The resulting dataset includes 391 sites and is provided as a single two dimensional array of signed integers in a Matlab (`.mat`) file format. Columns represent individual households, with the first row containing the household ID (a four number positive integer which corresponds to the house ID within the larger study).

Rows represent measurements at two minute time increments.where the measurement unit is defined as the energy in Wh over this increment. This schema results in multiplication by a conversion factor of `30` between the native unit (Wh/2 min) to W.

Finally the magnitude of missing values are represented as `-1`.

Dataset Completeness Figure 3.3.2 plots the distribution of data from a random sample of heat pumps over the entire range of the dataset.

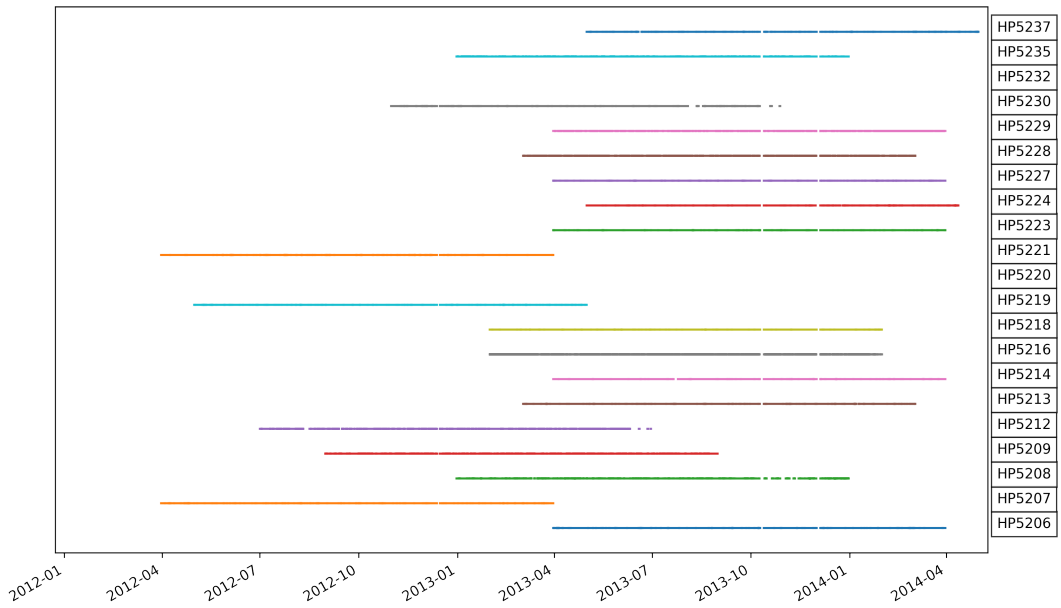

Figure 3.3.2: *Availability of data form a random sample in the heat pump dataset*

Though most of the datasets plotted in Figure 3.3.2 cover approximately the same span, the start and end times vary radically. Furthermore, certain datasets, such as `HP5232`, are fully empty, whilst others such as `HP5230` are heavily fragmented. The distribution of dataset metrics such as largest gap and completeness are studied in greater detail in Appendix 7.2.

3.3.4.2 Household Dataset

Source The household dataset used in this work is taken from an energy study of households in Scotland provided to The University of Glasgow by a third party. This dataset however is not publicly available therefore the subsection used for this work has been anonymised.

Structure The data has been provided as a collection of `.csv` files, where each file corresponds to a given household and month. Files are named based on a presumed "household ID" and the respective month [7] . Examples of file names include: `D1F8F_Nov.csv`, `DA79AJUL21.csv` and `DB382_Jan.csv`.

Each `.csv` consists of first a column containing timestamps, and subsequent columns containing the power data for each meter on the property, the content of a typical file is shown in Table 3.3.1.

Table 3.3.1: *Sample of a household dataset .csv file*

Timestamp	Power (Wm) - sid = 827412 (Turbine)	Power (Wm) - sid = 841327 (Whole House)
2021-07-01 00:00:00	31.0	646.0
2021-07-01 00:01:00	31.0	686.0
...	...	...

The column headings for each sub-meter follow the same format: `Power (Wm) - sid = <sid> (<descriptor>)`. The descriptor, henceforth column descriptor, contains some information about the downstream load [8]. A list of all unique column descriptors can be found in Appendix 7.9).

[7] exact naming convention varies between months
[8] For example the presence of an EV

Of these columns it is assumed that one will always be the site aggregate measurement and is be named accordingly, though it appears no strict convention has been followed when naming site-meter columns.

Finally, a single dataset was found within the dataset, with a sample rate of one hour rather than the one minute. This is raised only to give context to filtering applied in Section 3.3.7.2.

Dataset Completeness The time distribution of valid data for the household dataset is plotted in figure 3.3.3.

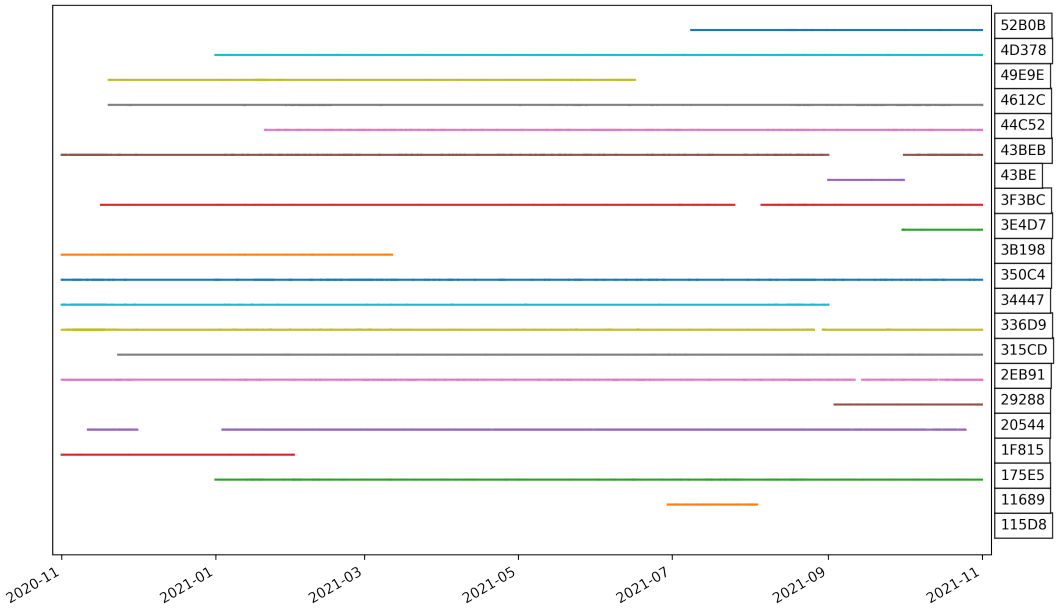

Figure 3.3.3: *Availability of data for a random subset of the household dataset*

The dataset as a whole spans approximately one year from ≈01/11/2020 to ≈31/10/2021, however the start and end times vary substantially for each household. Furthermore, the data is highly fragmented with, in many cases, only one month of data is present. Another issue highlighted by Figure 3.3.3 is the problems faced with the file naming, since household `443BEB` and `43BE` are likely the same, yet, due to inconsistencies in naming have not been recognised as such. Thus the missing section from `443BEB` is the exact region present in `43BE`.

3.3.5 Processing Raw Data

While the individual files comprising each dataset can be parsed with relative ease, factors relating to the storage schema of both datasets make retrieving data directly from source whenever needed an impractical proposition. The household load profile data for example splits individual households by month meaning multiple files must be parsed and combined to form a continuous time-series. Furthermore not all households start and end in the same calendar month and some have gaps in certain months. Though contained within the same continuous dataset, the heat pump datasets also has start and end times staggard through the year. Additionaly, given the size of the complete datasets (on the order of several Gb each) performing operations with all of either set loaded in memory becomes challenging.

To address some of these problems the data is first loaded, processed and stored in an intermediary `.hdf` file. This format allows for efficient tabular data storage with multiple data-types at arbitrary length, while also allowing quick retrieve of individual data objects. Each household/heat pump load profile is stored as it's own tabular data object.

Given the different storage schema employed by the heat pump and household datasets, the details of processing will be treated individually in the next sections.

3.3.5.1 Household Data Specific Processing

Since a given households worth of data is split amongst many `.csv` files, a process of stitching must occur to form one continuous set. Ideally individual files would be loaded and each household-month would be appended to it's appropriate column as it's loaded. This presents a the problem of determining which dataset to a append a given household-month to. As discussed in 3.3.4.2 the column descriptors contain both a unique 'sid' number denoting a unique sub-meter, while the files themselves are named according to month and a 'household ID'. In this case the 'household ID' derived from the filename will be used to uniquely identify newly loaded household-months and thus which dataset to append to.

Since we wish only to extract the whole house aggregate measurement (site-meter), the column containing this information must be specifically selected. A further problem arises here due the fact that there are not only site-meter but also sub-meter columns within each `.csv` with an unrestricted naming convention used for the column descriptors (see Section 3.3.4.2). As such a pass list of search terms is constructed (Table 3.3.2), the occurrence of any of which within a column descriptor would indicate that this column is a site meter. As a fail-safe if either no or multiple column matches are made, this household will be flagged under 'site meter not found' in the dataset metadata (discussed later). To avoid a scenario in which some months contain additional column descriptors which might lead to a failure to identify site-meter by it's descriptor, the first encountered instance of a given site will record both the 'Household ID' and the 'sid' and, in subsequent encounters, only the 'sid' matched in the prior encounter will be searched for.

The other columns and their descriptors, though not used directly, still expose useful information about a given household, namely the types of load and embedded generation (if any) present. As discussed in Section 3.2.6, a vital stage of filtering involves removing households known to contain particular types of load. A block list of expressions is therefore created where any occurrence of which in any column will flag a household under 'isSiteUnsuitible' in the dataset metadata.

Programmatically extracting 'sid' and 'household ID' from the column descriptor and file name respectively is done using regular expressions. In the case of 'household ID', the `.csv` extension and the three letter month abbreviation with an optional underscore are matched with look-ahead term returning the 'household ID', while the 'sid' is matched by searching for the 'sid' keyword followed by an equality sign surround with optional spaces and returning the proceeding number.

Table 3.3.2: *Sample List of search terms that will act as a either a block list to a household with a matching descriptor, or a pass list to identify the site-meter*

Embedded devices Block list Expressions	Site-meter Pass list Expressions
Car	Whole house
car	Whole House
Solar	whole house
solar	Whole Property
Heatpump	whole property
Heat pump	House Supply
Wind	house supply
Turbine	
generation	

Household data is thus loaded by iterating through all raw `.csv` files and appending loaded household-month to it's appropriate "household ID" in the data-store as identified through the process described above. Where a first occurrence of a household is encountered an entry is created in the dataset metadata recording the site's suitability, and the "sid". Finally each household dataset is sorted by increasing time index, and duplicate timestamps are removed.

3.3.5.2 Heat Pump Specific Processing

As discussed in 3.3.4.1, the heat pump dataset is natively stored in a `.mat` file as an array of N heat pumps and M individual samples. This array is small enough (3.8 Gb) that it may all be loaded and operated on memory at one time. The first step is to convert the datatype from signed integer to floating point. Though this incurs an increase in stored size, the advantage is that missing values, represented as `-1` in the source data, may now be represented as `Nan`, which is not supported with integer types. This is vital since the final stage of dataset synthesis will be an element-wise sum the household and the heat pump dataset along the time axis. Where a missing value exists in either dataset, this element must also be treated as missing in the synthesized set. If missing elements are represented as `-1` then the sum of this and it's corresponding household element will most likely still return a positive yet incorrect value meaning this data-point cannot be distinguished as missing. When `Nan` values, however, are summed with any other number, the result will always inherit the value `Nan`, thus nicely encoding the desired behaviour described above.

Since power is natively measured in Wh/2min a scaling factor of `30` is applied to convert to W. The resultant array is then stored, as in the previous section, to an intermediary `.hdf` file.

3.3.6 Dataset Metrics

In the subsequent filtering stage, individual load profiles from both household and heat pump dataset will be filtered by dataset metrics. This section gives an overview of these metrics and their relevance.

Some metrics such as 'site suitable', and 'site-meter found', are best suited to extraction as the raw data is loaded and processed while others, such dataset completeness, can only be evaluated once the full dataset is assembled. A persistent object accompanying each dataset is created at the raw data processing stage and passed the subsequent metadata extraction stage. A sample of this table for the household dataset is shown in Table 3.3.3.

Table 3.3.3: *Sample of the household dataset metadata table. Each row represents a heat pump/household, and each column represents a metric. Dots indicate that this table is truncated for illustration purposes.*

ID	siteMeterFound	siteSuitible	sid	start	end	Comp_M1	...	Comp_M12
115D8	True	True	846174	2021-09-02	2021-10-31	nan	...	1.0
175E5	True	False	828389	2020-12-31	2021-10-31	1.0	...	1.0
...	...	...	...	...	...	...	...	...

The following sections will define some of the key the metrics extracted in the metadata extraction phase, their relevance and any assumptions and limitations. The significance of other metadata metrics such as, 'sid', 'siteMeterFound' and 'siteSuitible' are discussed in Section 3.3.5.1.

3.3.6.1 Start/End

Defined as the first and last valid sample observed in a given dataset. This is completely invariant to the level of completeness between those two points, however is still relevant as will be discussed in Section 3.3.6.2.

3.3.6.2 Completion by Month

Given the specification of a minimum January-August span for all synthesized load profiles, filtering by completeness over the entire dataset is superfluous. Given, however, that the intent is to create this tool-chain in a flexible and scalable manner, it seems unwise to 'hard-code' a metric for only a specific date range. Thanks to the date span of both datasets being less than one year, the 'completenes' of the dataset

for each calender month, can instead be evaluated. Thus evaluating the dataset for completeness within a given date range simply involves taking the average of the completeness for all the months of interest[9]. This method must be used in conjunction with the start and end methods to avoid a corner case with datasets spanning close to a year, where the individual months are complete but are not sequential[10].

3.3.6.3 Sample Period

Due to the presence of outlier sample-rates within the dataset (see Section 3.3.4.2) the sample-rate is extracted as a metric for future filtering. This is evaluated by taking the time differences between each adjacent samples in a given load-profile and taking the mode of the resultant array.

3.3.7 Filtering and Selection

Prior to the synthesis stage, the datasets comprising 300+ individual heat pump and 70+ individual households must be filtered down to just five of each. This is done based on:

1. Metrics relating to the datasets themselves (i.e. completeness)
2. The membership of the load profile in question to a blacklist category (i.e. 'known heating types' = heat pump)

This section aims to give a brief overview of what filters were applied and, if relevant, what thresholds were set and why.

3.3.7.1 Dataset metrics

Sample period Set to less than 180s to filter extreme outliers such as those described in Section 3.3.4.2.
Completion by Month Months January-August are greater than 90%

3.3.7.2 Household Dataset Specific filters

Site Suitable The suitability of the site as determined by the column descriptors in the raw processing stage.
Known Heating Type households All heating types known to include electric heating of any kind are excluded.

3.3.7.3 Filter by Heat pump Types

Examination of a random sample of the heat pumps within the heat pump dataset reveals a wide variety of different load profile shapes (Figure 3.3.4). Resorting again to human intuition, without the prior knowledge about Figure 3.3.4, the fact that they all represent the same class of device would not be apparent. This presents an immediate challenge to an inherent assumption present in this work: that the ability to disaggregation on one heat pump can generalize to many.

[9] Assuming this range is an integer number of months
[10] This is elaborated further in the Appendix

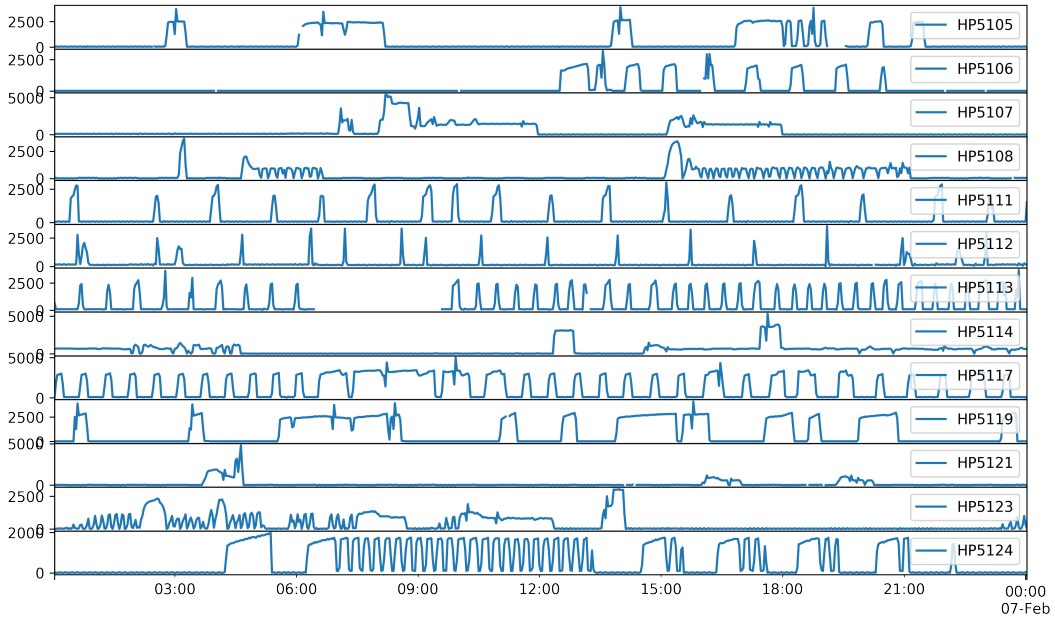

Figure 3.3.4: *Random selection of heat pump load profiles over a full day*

Though there appear to be individually substantially different load profiles, it is can be seen that there are three groupings with very similar features[11]:

1. Characteristics of a dead-band controller, resulting in a long continuous pulse and then relatively periodic narrow pulses at the same approximate magnitude (`HP5105`, `HP5124`)[12]. These are henceforth referred to as the "dead-band" group.
2. Similar to "dead-band" however, the on state is however continuously variable, resulting in what appears to be a modulated square wave(`HP5108`, `HP5123`). Referred to as "dead-band cv".
3. Activations occur only in narrow approximately periodic pulses (`HP5112`, `HP5113`, `HP5111`), referred to as "narrow low freq".
4. Activations occur for long and durations at relatively consistent magnitude (`HP5114`, `HP5107`), referred to as "wide consistent".

It should be noted that the experiments conducted in *this* work are currently limited to training and testing disaggregation algorithms on individual heat pumps and, as such, the variability between heat pumps would be *unlikely* to affect the experimental outcome. Given however, the high-level aim of this project[13], it seems prudent not to disregard the substantial impact that this *will* likely have on algorithm generalisation. With this in mind, it was decided to proceed by explicitly testing *only* on heat pumps within the dominant "dead-band" group[14].

Studying underlying mechanisms causing these differences, and thus creating a more rigorous and systematic grouping would, undoubted, be preferable to the heuristic categorisations proposed above. Given the lack of available information about the heat pump models, is was decided that the heat pumps would be grouped by conservative manual inspection.

[11]Groups which do not fit well into any of these categories, or include signatures that could be grouped under either category are grouped under "other/combination".

[12]There are others in Figure 3.3.4 which also fit this pattern but with lower frequencies (`HP5106`, `HP5117` for example)

[13]Investigating the viability of contemporary NILM algorithms to heat pump disaggregation (see Objectives).

[14]This group is selected both as it represents the dominant grouping within the dataset and also it's characteristics are well defined.

3.3.7.4 Filter by Household Heating Types

As discussed in Section 3.2.6, the synthetic dataset will be created under the assumption that each household contains only one form of electric heating, namely a heat pump. Thus a vital selection criterion for the household load profiles will be the absence of any such load. In section 3.3.4.2 it is discuss that heating information for a subset of the households has been distributed with the dataset, henceforth referred to as 'known heating types', and can thus be easily filtered. Further information extracted from the raw processing stage (Section 3.3.5.1) reveals additional information on another subset of households based on the column descriptors within individual `.csv` files. Together this information covers only a fraction of the dataset and, with the exception of the know heating types[15], this information only informs of what loads **are** in the household as opposed to what aren't. The remaining households should still be suspected of containing unwanted appliances.

Two methods were developed in an attempt to problematically discriminate households containing unwanted loads which were tested on the subset of households with known loads:

Average site-meter power This naive approach attempts to discriminate between households containing undesired appliances by simply discarding sites with a site meter average power is greater than some threshold.

Night time usage bias A feature commonly observed on the load profiles of households known to be clear of undesired loads is comparatively limited night time energy usage. This is exploited by calculating a so-called 'night time usage biias' (NTUB) which represents the relative weighting of load towards night time hours.

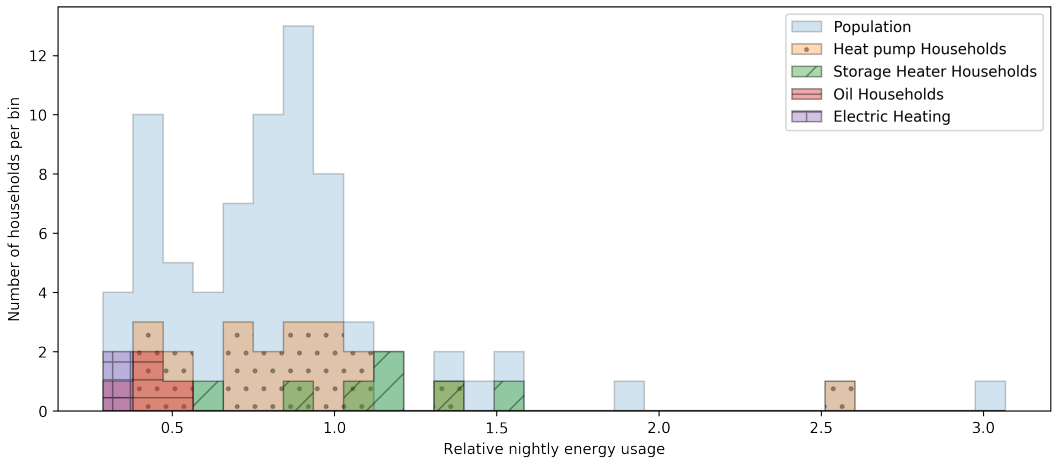

Figure 3.3.5: *Histogram of Night Time Energy usage Bias broken down by heating type, demonstrating the difficulty in separating desired from undesired heating types. NTUB=1 means equal energy used at night while NTUB=0 is no energy used at night.*

A full write up of both methods is included in Appendix 7.4.

Though both approaches were able to discriminate the majority of the household containing undesired loads [16], enough remained to cause concern. The decision was thus made to manually classify the remaining unknown household datasets based on intuition from the existing known datasets. It should be noted that this is an undesirable step since it subjects the dataset to human selection bias however, no other

[15] We assume that where a heating type is known that this is the exclusive or by far the dominant heating type in use by this household.

[16] Average site-meter power correctly discards `17/29` of the known undesired heating types, while retaining `32/68`, NTUB correctly discards `21/29` of the undesired heating types while retaining `19/68` of the whole population.

recourse was apparent. As a labour saving step, this manual classification stage was only conducted on the household dataset which passed the other filtering stage.

3.3.8 Synthesis

In total, five datasets have been created, each containing five synthetic load profiles each with a singe heat pump at a particular same power aggregation level in accordance with the experimental requirement laid out in Section 3.2.2. The data for a given synthetic load profile is generated, according to equation 3.2.1, by aligning and element-wise summing the heat pump load profile set P with N household load profiles H where N defines the power aggregation level.

Conceptually, the process of synthesis is relatively unassuming - however, to satisfying the aliment criterion laid out in Section 3.3.3, some consideration must be given to how the datasets must be shifted to be brought into alignment:

1. Which data will be shifted - this affects the dates of the output dataset.
2. How are time of the year and wall time retained during the shift

The choice of data to shift is arbitrarily selected to be the heat pump dataset, thus the months of interest are Jan 2021 - August 2021. The time of year is retained by shifting by the number of samples that comprise an integer number of years. If working with arrays this would not trivial given the variability in lengths of years etc, using the Python `datetime` library however makes this task relatively straight forward.

Calculating the number of years to time-shift again sounds trivial but has the hidden pitfall that it is a function of not only the start times of both datasets. This is illustrated in Figure 3.3.6, where two possible multiple time-shifts exist but only one maximises the alignment of the months of interest.

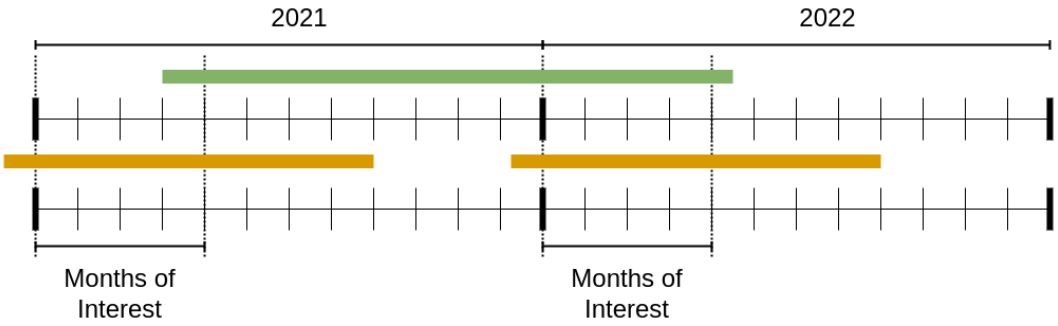

Figure 3.3.6: *Two possible time shifts of the orange dataset, where the right most shift maximizes the alignment of the months of interest*

This is solved programmatically by first selecting only the spans of datasets that include the months of interest. In the case of the green dataset this would result in one span in 2021 and one in 2020, while the orange dataset would only contain one span. For each span in both datasets the start year and the degree of completeness of this span is calculated for example the first span of the green dataset would start in 2021 and would have a completeness of 0.25%, while for span 2 it would have a completeness of 100% and a start year of 2022. Each completeness for a given dataset is put in an arrays ordered by start year. These array are then shifted past each other in a very similar operation to a convolution, however rather than multiplication the min operator is used as oppose to multiplying elements. The index of the largest total overlap is taken to be the optimal alignment. This is formalised in equation 3.3.1:

$$\text{shift} = \underset{n}{\text{argmax}} \left[\sum_{m=0}^{m=M} \min\{a_{n-m}, b_m\} \right] \tag{3.3.1}$$

Finally the dataset is saved in accordance with the NILMTK dataset schema as a `.hdf` file accompanied by the required supporting metadata.

Chapter 4

Results

4.1 Performance Metrics

There are myriad ways to quantify the performance of a disaggregation algorithm, many of which cater to a particular target insight about an algorithm and the suitability and interpretation of results can vary substantial as a result. It is thus vital that the reasoning and assumptions behind the metrics used in any such work be explicit. By default the NILMTK provides several metrics by default such as:

- Root Mean Squared Error (RMSE)
- Mean Average Error
- Proportion of energy correctly allocated
- F1 score

The choice of metric used in this work depends on nuances specific to the objectives of this work, which may not exist in this combination in other NILM literature:

1. The device class being tested have a wide range of power ratings (2.5-8kW observed)
2. The devices being tested are continuously variable as opposed to Finite state
3. The performance on several devices will be aggregate[1] in the presentation of results

Ideally the metric would be invariant to the rating of the device to allow for like for like comparison and aggregation between devices. None of the metrics supplied by the NILMTK meet this criterion, largely since most of the existing work focuses on evaluating algorithm performance on individual device classes such as fridges with very similar loads between models. This is not, however, the case with heat-pump ratings ranging from 2500-8000W and, as such, a new metric is required.

4.1.1 Mean Normalised RMSE (MNRMSE)

The Root Mean Squared Error (RMSE) is a widely used metric for measuring the difference between the predictions or output of a model or estimator and the observations themselves. The root mean squared error is defined in Equation 4.1.1.

$$RMSE = \sqrt{\frac{\sum_n^N (\hat{y}_n - y_n)^2}{N}} \tag{4.1.1}$$

Where $\hat{y}_n$, y_n are the estimate and observation at sample n respectively and N is the total length of the samples under test. The physical interpretation of this results is that it represents the average standard deviation of the true device from the ground truth it is attempting to estimate. Given that the standard deviation will tend to grow as a function of the typical rating of the load this must be normalised against another metric proportional to device rating to render it scale invariant. Some potential candidates are:

[1] The mean of performance on all five heat pumps is taken

1. Ground truth mean power
2. Disaggregation performance of a standard benchmark algorithm

In this case 2. is particularly attractive since it not only grants scale invariance, but also provides consistent and intuitive scale. For this the mean disaggregation algorithm performance is ideally suited since due to it's simplicity it is highly consistent between devices and also easily understood. The normalisation defined in 4.1.2.

$$\text{MNRMSE}_{alg} = \frac{\text{RMSE}_{Mean} - \text{RMSE}_{alg}}{\text{RMSE}_{Mean}} \tag{4.1.2}$$

With a MNRMSE=1 as the maximum attainable value, 0 being the same as Mean and, negative values represent poorer performance than Mean.

This metric is unit-less and has little physical meaning beyond the comparison of aggregation performance. As such all results presented in this section are supported by the pre-normalised RMSE in Appendix 7.8.

4.2 Results Summary

The average performance of each algorithm at a representative subset of test cases is shown in Figure 4.2.1, giving an overview of performance at power aggregation levels 1,3,5 and temporal aggregation levels of both 2 and 15min. The variability observed between performance on each heat pump is shown with by plotting the upper and lower quartile of performance with error bars.

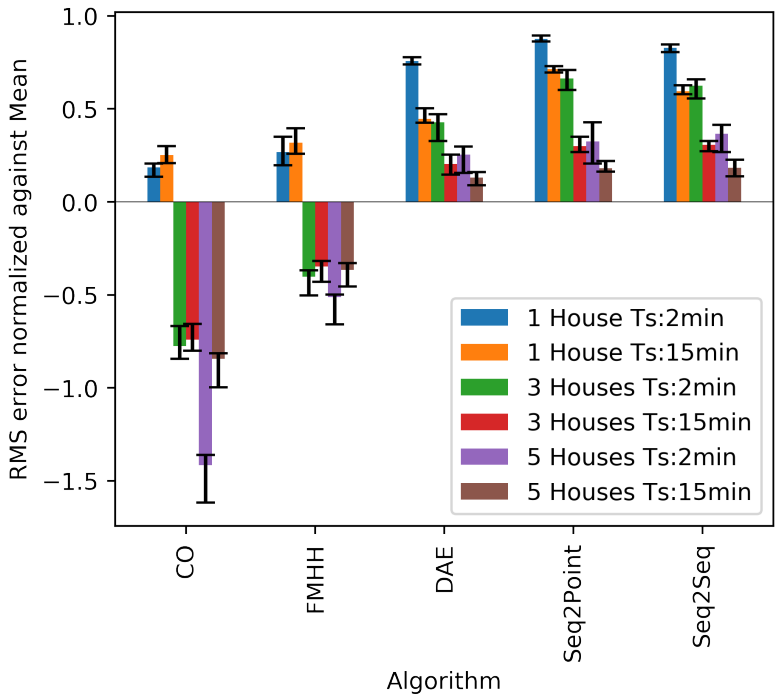

Figure 4.2.1: *Selection of results Summarizing algorithm performance. Performance is given in unit-less MNRMSE.*

For brevity, the observations from Figure 4.2.1 discussed in this section are entirely qualitative with units given in MNRMSE which has little physical meaning beyond the comparison of algorithm performance. A non normalised tabular version of all results obtained is included in Appendix 7.8.

A the following qualitative observation are made from Figure 4.2.1:

1. Finite state base algorithms (CO&FHMM) are extremely sensitive to power aggregation level with performance well below that of mean aggregation.
2. Unintuitively, finite state base algorithms appear to marginally improve performance under decreased aggregation levels.
3. Neural Network based algorithms (DAE,seq2seq,seq2point) as a whole appear to perform substantially better than finite state algorithms, with seq2seq performing the best at low aggregation but overtake at extreme power aggregation by seq2seq.
4. Neural network bases algorithms again display high sensitivity to both increase power and temporal aggregation level though the sensitive to the latter appears to be greatest at moderate feeder aggregation levels.
5. Variability in performance between individual heat pumps is in most cases minor relative to the effects of increased aggregation however grows under high power aggregation.

The isolated effects of both power and temporal aggregation are discussed in detail in the subsequent sections.

4.3 Increasing Power Aggregation Level Discussion

Intuitively, it is expected that the aggregation of additional household load will act to obscure the heat pump depending on the number, rating and distribution of loads present. To quantify this effect on algorithm performance, the algorithms were each tested on five datasets each representing an increasing number of households in aggregate with the same single heat pump load profile. The load profile content for a given heat pump at each increment is shown in Table 4.3.1. Each dataset contained five separate load profiles derived from a heat pump load profile in aggregate with between 1-5 household load profiles, representing increasing Power aggregation levels. Given the differences between the finite state, and neural network based algorithms, the analysis of this section will be broken down accordingly.

Table 4.3.1: *Content of aggregate load profile for an individual NILMTK 'household' at each power aggregation level*

Power Aggregation	Single Heat pump load	House 1	House 2	House 3	House 4	House 5
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	F	F
4	T	T	T	T	T	F
5	T	T	T	T	T	T

4.3.1 Neural Network Based Algorithms (DAE, Seq2Seq, Seq2Point)

Figure 4.3.1 plots the average algorithm performance of neural network based algorithms at increasing Power aggregation levels.

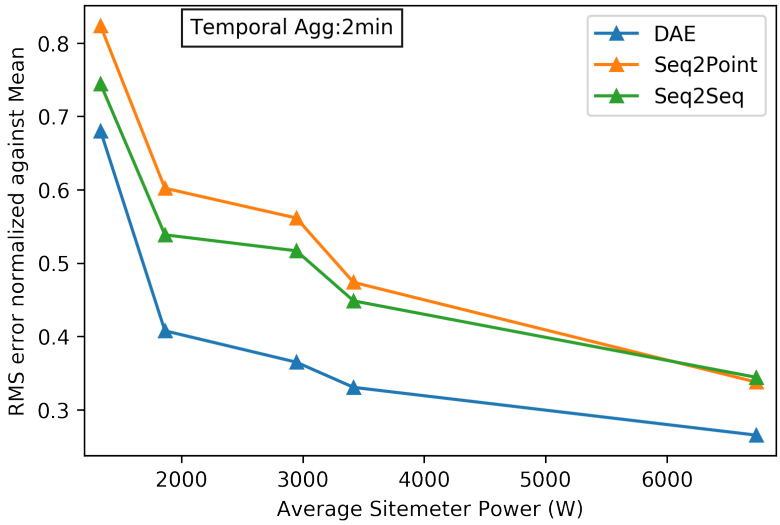

Figure 4.3.1: *Disaggregation Performance Relative to mean benchmark against average site meter power*

The choice of the dependent variable (x axis) to represent Power aggregation level must be a close analogue of this 'obscuration'. The two approaches considered are the number of households in aggregate or the (as has been chosen) the total average site-meter average power. The argument in favour of using site meter average power is that is compensates for the large variability in the individual household load and thus the degree of "obscuration", while the counter argument states that "obscuration" might relate more to the number of individual load signatures which might scale more linearly with number of households in aggregate. Testing these arguments would represent a body of work unto itself and as such the choice to go with a site meter average power is somewhat arbitrary. It should be noted that the number of households in aggregate can still be seen from Figure 4.3.1, since each sample of increasing Power aggregation level (marked with a diamond) still correspond to the an increment of a single households worth of power.

As expected, the performance of all algorithms decreases with increasing Power aggregation level. This decrease in performance is initially rapid before but appears to slow with higher aggregation. It is also noted that the individual algorithms start initially slightly spread out, but, with increasing aggregation, appear to grow tighter. Both of these observations hint towards algorithm performance generally settling to some value slightly above Mean, though confirming this would require additional data-points. The existence of such a settling point does make intuitive sense since, as demonstrated by Mean disaggregation, there will be metrics that can be applied based just on historic observations of the target device's load profile. As an example if the load is highly periodic, it might be quite effective to predict the load based purely on time of day. Such non-observation driven prediction heuristics will likely preform better than mean disaggregation, and can be expected to be the asymptote towards which these algorithms broadly tend. Rather than normalizing RMSE against Mean disaggregation, it should be normalized against the errors incurred by a neural network trained on without any aggregate load profile information, thus producing predictions only based on historic data.

A second observation is that the decrease in MNRMSE between aggregation level 2-3 is lower than that from both 1-2 and 3-4, despite the assumed trend to be a decaying exponential, and 2-3 involving a substantially larger step in site-meter average power. This lends some weight to the argument that the importance of a site's average power is secondary to the types of loads in that site.

The effect of Power aggregation level on the predictions of the algorithms themselves is shown in Figure 4.3.2, by plotting sample of predictions at the same section of load profile under each level of aggregation.

Figure 4.3.2: *Sample of several hours of predictions vs ground truth of each neural net based algorithms for household one at increasing power aggregation levels.*

As expected we observe a general trend across algorithms that as power aggregation increases, there is:

1. Poorer resolution of detail
2. decreased accuracy (though not globally)
3. Responses towards events not present in the ground truth.

Point one is very intuitively obvious, the more noise in the algorithm's input the less likely a given feature in the aggregate measurement is attribute to the heat pump. Again point makes intuitive sens, however with an interesting exception that can be observed across all algorithms after 10:00. Here the first aggregation level tracks the ground truth well however level two departs substantially with a positive DC offset, and subsequent aggregation levels 3 and 4, each providing a respective increase in accuracy. This initial departure from ground truth can be directly attributed to a DC load present in the site meter at aggregation level 2. This additional load remains present with the subsequent aggregation levels, so the question begs, how are these loads less affected by this offset? An intuitive explanation here is that algorithms trained at a lower aggregation tend to "trust" the site meter measurement more since it is generally less noisy, thus a DC offset appearing on the ground truth was assumed to be attributable to the heat pump. The algorithms trained at a higher aggregation conversely place "trust" in the ground truth and are therefore less likely to misattribute the dc offset.

Finally, on point three, it is observed that at increasing aggregation levels we begin to see patterns within the predictions which look like time shifted versions of the ground truth in what will henceforth be referred to as "ghost predictions". This can be seen especially clearly on the De-noising Auto Encoder prediction at Power aggregation 5 between 6-7AM, though is present to a lesser degree in the sequence to sequence between 1-2PM. Intuitively it follows that as the algorithm receives noisier information it must rely more on purely historic load profile information which are then "played back".

4.3.2 Finite State Algorithms (CO, FHMM)

In Figure 4.3.3, the average performance of each disaggregation algorithm is plotted at increasing Power aggregation levels.

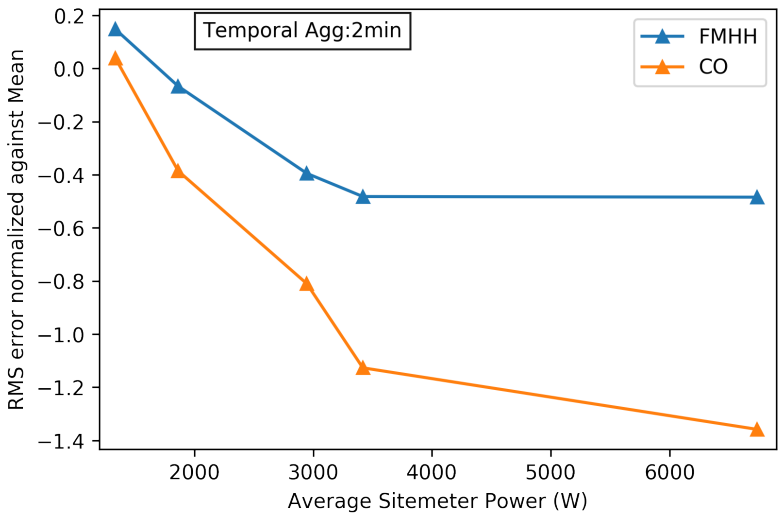

Figure 4.3.3: *MNRMSE Disaggregation Performance Plotted against site-meter average power*

Both FHMM and CO, have drop below the performance of mean disaggregation[2] by the second Power aggregation level, and continue to drop before appearing to start settling. An examination of a sample of the individual algorithm predictions under each aggregation level (Figure 4.3.4) gives an insight into:

1. Why the performance for both falls below that of mean
2. Why the performance tends to settle where it does

[2] As indicated by negative MNRMSE (see 4.1)

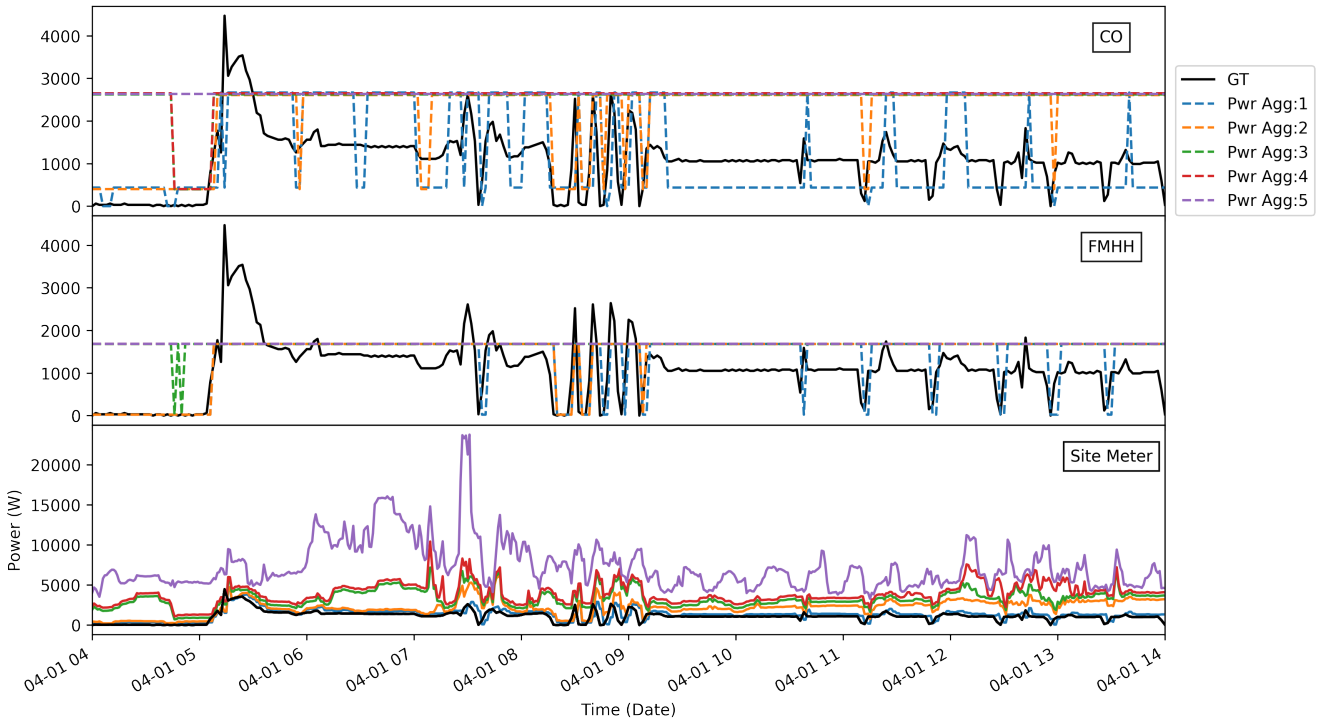

Figure 4.3.4: *Predictions vs ground truth of each finite state algorithms for household one at increasing power aggregation levels.*

As discussed in Section 2.1.2.3, CO does not adhere to the switched continuity principle[3], and thus the state estimates will frequently be confused by the addition of additional noise[5]. A second critical flaw relates to the fundamental structure of CO problem. Since CO relies on attempting to find an optimal combinations of loads with known finite states to minimize an error term, the lack of state information about any other loads in the systems means that all household load is treated as the error term to be minimized. Thus at any power larger than half of the heat pump's on state, the algorithm will always predict that it is on since this minimizes the residual error term. Intuitively this can be though of as the algorithm attempting to select the heat pump state that is as close as possible to aggregate power measurement as possible. It is thus intuitive that, with Power aggregation levels a preference towards high power states will emerge since for any aggregate power higher than half of the highest heat pump state, the minimal error state estimate will always be the largest.

It is as yet unclear to the author why the FHMM appear to be exhibiting the same characteristic trend towards the permanently on state at higher aggregation, since the additional noise should introduce equally both spurious positive and negative signatures. Perhaps the on state represents a lower average error than the off stage and so as noise increases the expectancy maximisation algorithm is increasingly "cautious" about leaving this state. A test to confirm this would be to run a FHMM trained on lower aggregation on the high aggregation dataset. The hypothesis is that this model would exhibit much more frequent spurious state changes and incur a greater RMSE. The author has yet to find any discussion of this phenomenon in the NILM literature.

Given the trend of both algorithms towards the permanently on state, the upper bound of settled performance can thus be explained by considering that the algorithm can spend no more than 100% of it's time at this state, and thus cannot get any more wrong. The different values both algorithms settle to is simply a byproduct of the maximum states allocated to each appliance during the model traning stage.

[3]The idea that devices in all probability will not all happen to switch in the same time instant

4.4 Increasing Temporal Aggregation Level Discussion

As in Section 4.3, the intuition predicts that under increased temporal aggregation the algorithms will generally perform less well since there is less information available to exploit. To examine this effect, each algorithm is tested with a fixed Power aggregation level and from 1-15min in increments of 1min[4]. This is repeated on three separate power aggregation levels (1,3,5). As in Section 4.3, this analysis will be separated into neural network and finite state based algorithms, given the substantial difference in both performance and structure.

4.4.1 Neural Network Based Algorithms (DAE, Seq2Seq, Seq2Point)

Figure 4.4.1 shows sweeps of neural net based algorithm performance against temporal aggregation at Power aggregation levels 1,3 and 5.

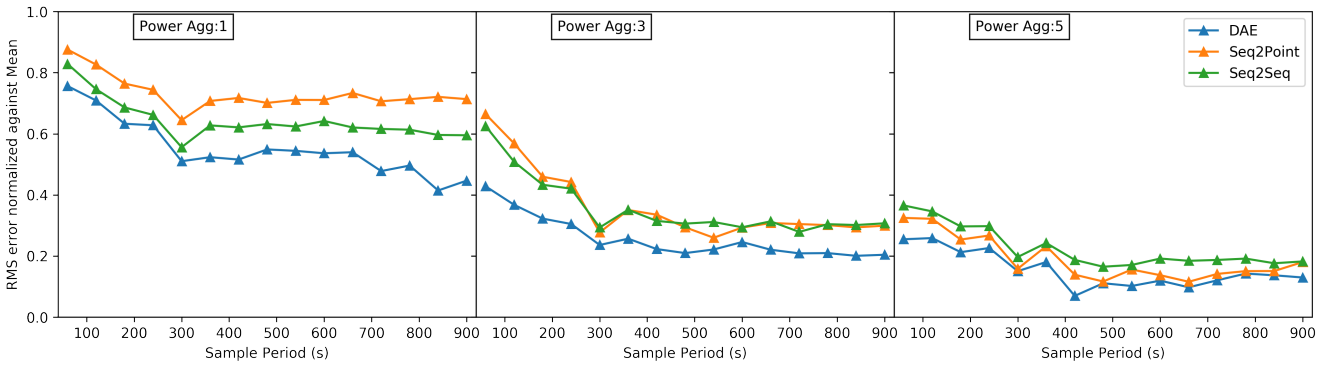

Figure 4.4.1: *Average disaggregation performance for each neural-net based algorithms vs increasing temporal aggregation levels at (left to right) power aggregation levels 1, 3 and 5*

As intuited, a general downward trend is visible in all algorithms with all eventually appear to settling at some positive nonzero MNRMSE. Though these results generally conform to expectation, several observations warrant further investigation:

1. Though a general downward trend is followed a point of negative inflection is observed at 300s (5mins) that is visible on all three plots and common to all three algorithms. Given that each successive disaggregation attempt involves a model predictions made with less information, how is it possible that an algorithm with more information available performs more poorly than that with less? It seems unlikely that this is due to random chance favouring one algorithm over the other given A) the common location on all plots and with all algorithms B) random variability should be minimised by the average of results over five heat pumps.
2. The performance of each algorithm appears to be substantially different and tends to diverge at power aggregation level 1, while converging in both levels 3 and 4.
3. The performance of the algorithms appears to settle at different accuracies depending on the power level.

A potential hypothesis to explain the inflection point in performance at 300s is that it arises from aliasing related effects due to down-sampling. An investigation of the NILMTK source code reveals that the down-sampling functionality exposed by the experimentation API relies on the pandas data-frame resample method, which itself does not apply any anti alias filtering. According to the Shannon Nyquist Sample criterion: No frequency above half that of the sample rate may be uniquely distinguished [26]. It

[4]Increasing temporal aggregation is achieved through the down-sampling feature of the experimentation API (See section **??**)

should also be noted In the case of a sample period of 300s this leaves us with a critical sample period of 600s a sample period lower than which would appear as an alias at a different frequency. This effect can clearly be observed within the test results as plotted in Figure 4.4.2.

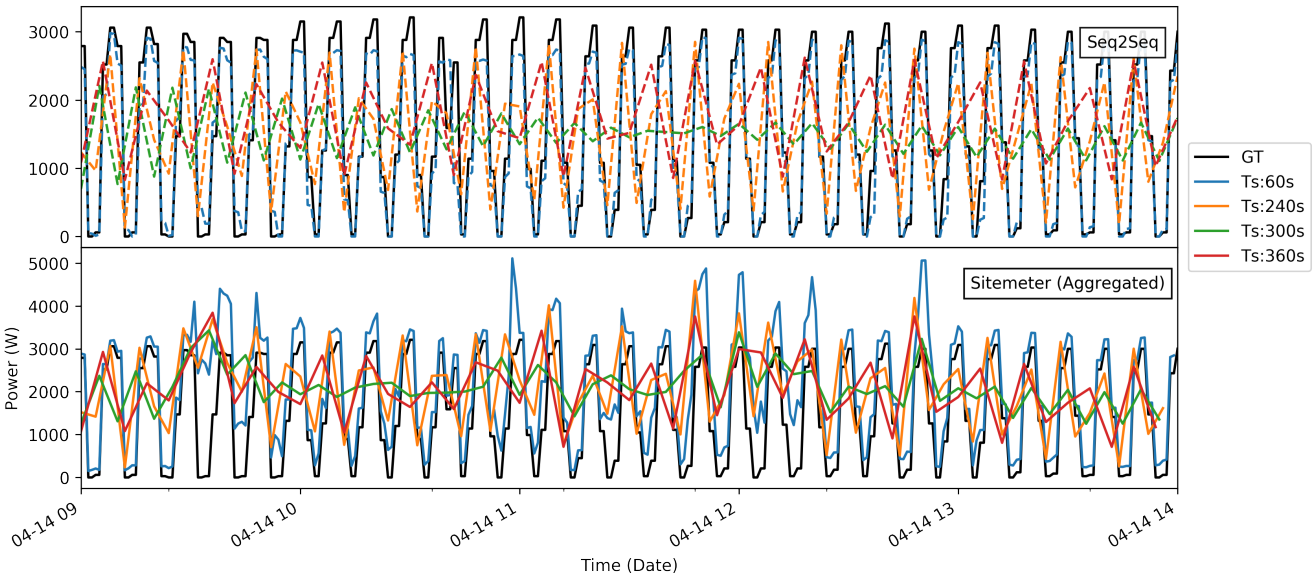

Figure 4.4.2: *sample of several hours of highly periodic heat pump activity with a period of ≈600s(10min), and aliasing effects visible at higher temporal aggregations.*

In this instance the dominant 600s(10min) component of the heat pump activity is exactly at the Nyquist limit of the 300s sample period trance (green) while that at Ts=240s is comfortably below and that at Ts=360s is above. It can be seen, that though imperfect, the Ts=240s trace represents a wave, at the same frequency and in phase with that of the ground truth. The trace at Ts=360s however appears to be representing a lower frequency waveform meaning that high and low predictions often appear out of phase with the ground truth. Finally the trace at Ts=300s, though at approximately the correct frequency appears also to drift in and out of phase with respect to the ground truth, as a function of minor differences between the frequency of the ground truth and that of the down-sampled site-meter.

The fact some heat pumps exhibit periodicity at exactly the critical frequency of the sample period where the anomalous inflection point was observed cannot be taken as proof that aliasing is the root cause. The observed presence of aliasing related effects within the predictions should, however, be a compelling reason to consider the use of anti-alias filtering in similar future works. A more rigorous test of this hypothesis would be alter the NILM source code to apply decimation using a library such as Scipy which include support for anti-alias filtering.

To further examine points 2 and 3, a sample of the predictions of each algorithm at increasing temporal aggregation levels are plotted against time at A) Power aggregation level 1 - Figure 4.4.3, B) Power aggregation level 3 - Figure 4.4.6.

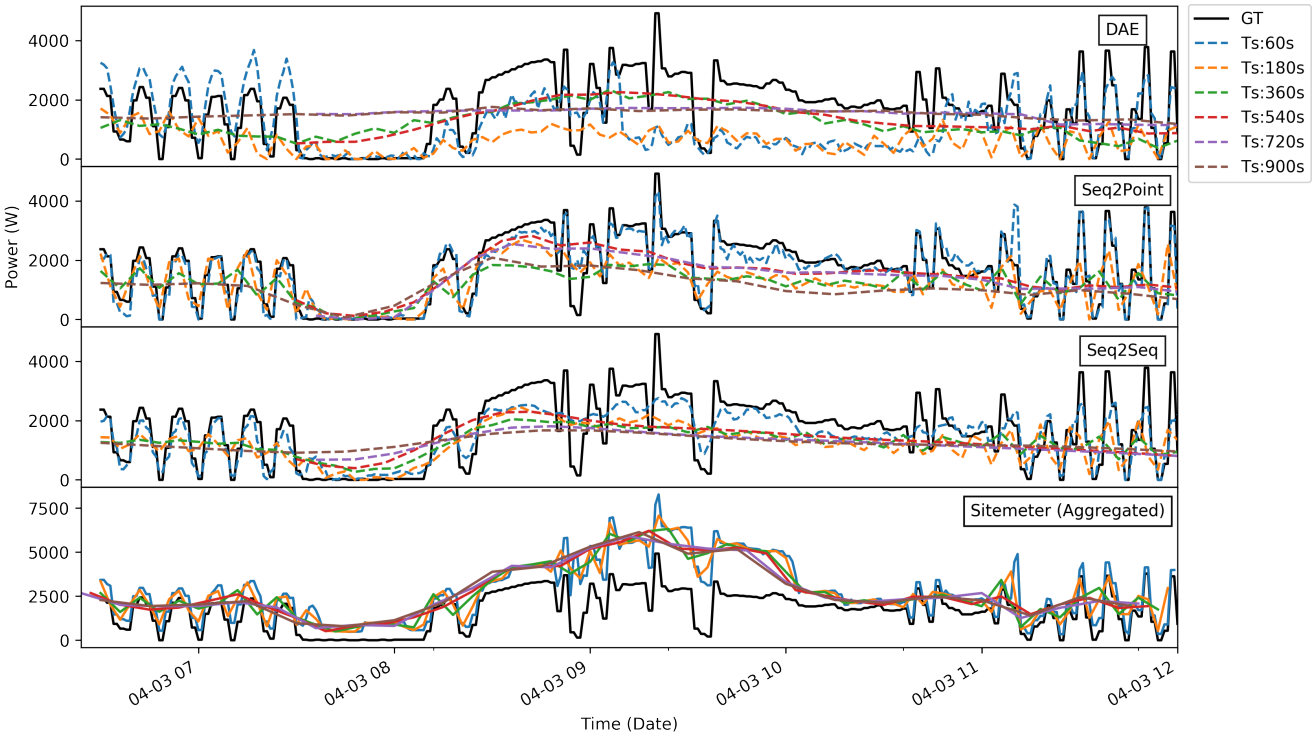

Figure 4.4.3: *Sample of several hours of disaggregation predictions for neural-net based algorithms at increasing temporal aggregation levels. Load profile for household 2 at power aggregation level 1.*

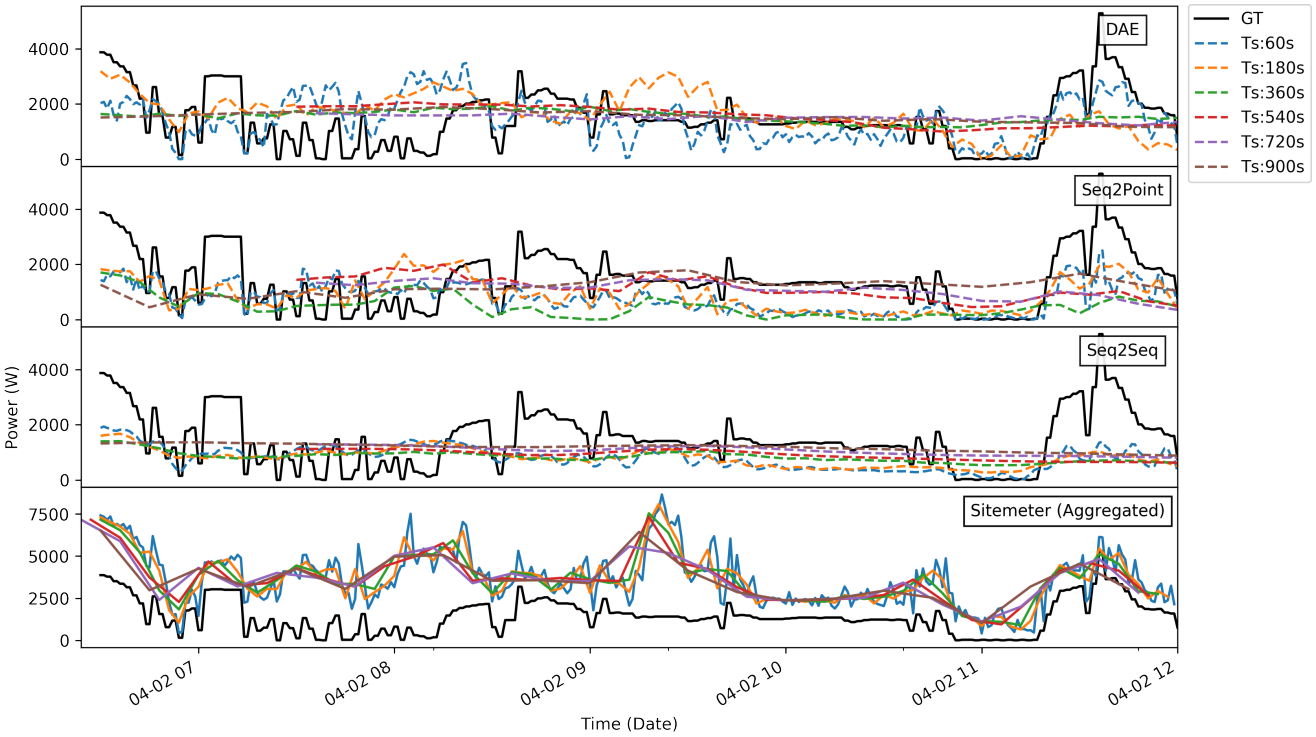

Figure 4.4.4: *Sample of several hours of disaggregation predictions for neural-net based algorithms at increasing temporal aggregation levels. Load profile for household 4 at power aggregation level 3.*

With regards to the diverging behaviour of the algorithms, deriving the root cause behaviour of neural networks is often impossible, however with seq2seq the characteristic divergence may stem from how the algorithm's output is interpreted. As discussed in Section 2.1.4, the output of this algorithm makes multiple prediction passes on a single data-point. Given that the sliding window is of fixed length, it will increase in absolute length as temporal aggregation level increases. It seems likely that beyond a particular time separation there is very little valid information a particular sample contains about the ground truth. Through the seq2seq averaging process however, the neural network is still forced to make predictions from these temporally distant regions, which are then treated with equal weight to those produced much closer in time to the sample. This would likely force the neural net to treat the estimations at the periphery as best guesses which minimally detriment the high validity forecasts in the centre of the window one assumes by simply outputting a smooth periodic approximation of historic load behaviour. This would explain the comparatively smooth prediction produced by seq2seq relative to seq2point observed in Figures 4.4.3, 4.4.6.

The convergence of seq2seq and seq2point at higher power aggregation, is simply due to seq2point being forced to reverting to the same strategy of approximating a best fit of the ground truth based on learned historic data.

4.4.2 Finite State Algorithms (CO, FHMM)

Figure 4.4.5 shows sweeps of neural net based algorithm performance against temporal aggregation at Power aggregation levels 1,3 and 5.

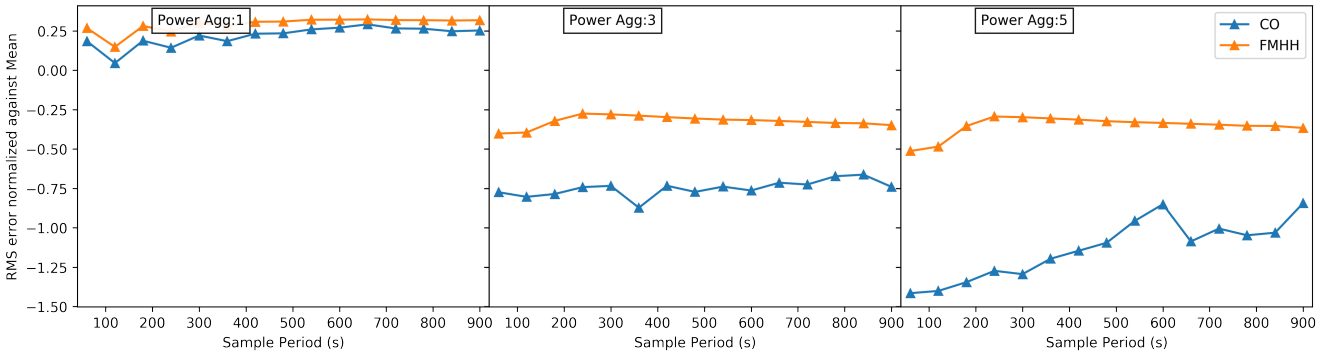

Figure 4.4.5: *Sweep of average disaggregation performance for finite state algorithms vs temporal aggregation levels at three individual power aggregation levels.*

Examination of Figure 4.4.5 reveals that, in breaking with the intuition, there does not appear to be any substantial downward trend observable in-fact to the contrary at Power aggregation level 5 there appears to be a substantial upward trend exhibited by CO.

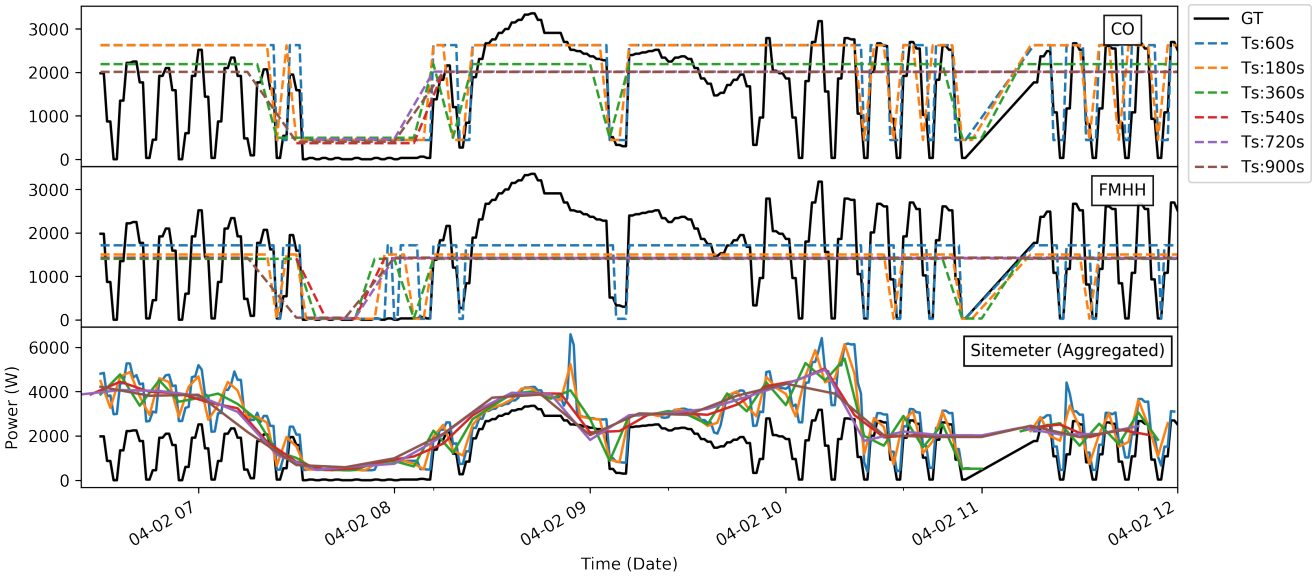

Figure 4.4.6: *Sample of several hours of disaggregation predictions for finite state algorithms at increasing temporal aggregation levels. Load profile of sample household at power aggregation level 1.*

A number of key observations are made:

1. Predictions made at higher aggregation level appear (blue and orange), in this limited sample, to be a closer approximation to ground truth.
2. Predicted on and off state magnitudes vary between different aggregation levels, which was not observed with models trained at different power aggregation levels. It would appear that the clustering approach used to derive the respective appliance states varies as a function of the down-sampling applied to the training data.
3. As with increasing power aggregation levels (Section 4.3), both algorithms exhibit a tendency to increasingly favour the on state, it is assumed for the same reasons discussed in Section 4.3.2.

Assuming that Figure 4.4.6 is representative, then observation 1 appears to be in direct conflict with the results observed in Figure 4.4.5 which appear to show increasing performance between Ts=2min and Ts=3min. While some random variability appears to be due to the chosen appliance on-state power (observation 2) this cannot be used to explain the large *increase* in performance in FHMM between Ts=180s and Ts=360s since the on-state magnitude is almost exactly the same between these traces. A subsequent investigation into the NILMTK source code reveals that the error metrics are calculated using a the down-sampled ground truth. This of course the wrong approach since, in reality, the fact that the algorithm sees a restricted view of the time-series in no way changes the ground truth against which it's predictions are measured. It seems likely that this down-sampled ground truth would might prove an easier target for disaggregation.

Chapter 5

Conclusions

In anticipation of demand for knowledge of available flexibility on the distribution network, this work presents a proof of concept study of contemporary NILM applied to heat pumps under measurement scenarios realistically available to potential system operators. To enable this work, a publicly available synthetic NILM dataset target towards heat-pump disaggregation is presented, and the dataset synthesis procedure structure and underpinning assumptions are discussed in detail. Both the effect of increased power aggregation [1], and the effects of increased temporal aggregation [2] are then studied in detail with the following parameter sweeps:

- Increasing Power aggregation levels in increments of one household at a fixed sample period of 2min.
- Increasing temporal aggregation levels in increments of 1min at for 1 household
- Increasing temporal aggregation levels in increments of 1min at for 3 households
- Increasing temporal aggregation levels in increments of 1min at for 5 households

A systematic under-performance in MNRMSE[3] was observed on both CO and FHMM, with performance falling below than that of the baseline Mean algorithm at all power aggregation levels greater than one household. In both cases this was traced to an increased preference to the on state at higher power aggregation. For CO this was found to be due to fundamental limitations with CO, whereby prior knowledge about the power levels of the majority individual loads is an implicit precondition. Given this cannot realistically be satisfies in any wide-scale deployment, it is concluded that CO is ill suited to this application. The underlying cause of this preference towards the on-state within FHMM is not fully understood. Both algorithms can entirely disqualified from consideration at feeder level measurement on basis of poorer performance than Mean under otherwise ideal[4] conditions.

Finite state algorithms were seen, unintuitively, to marginally improve performance under increased temporal aggregation. This is assumed to be an artefact in how algorithm performance is evaluated within the NIMTK experimentation API under down-sampling. The ground truth of individual appliance loads is down-sampled before performance is evaluated and thus favours the flatter predictions given at higher temporal aggregation levels. Given that correcting this would likely have a substantially negatively impact at higher temporal aggregation against the already poor performance it is inferred, though not concluded, that true performance actually exhibits a downward trend.

Neural network base algorithms (DAE, Seq2Seq, Seq2Point) collectively exhibit higher performance under all conditions than their finite state contemporaries, with seq2point fractionally leading with an average MNRMSE of `0.86` under nominal low temporal and power aggregation levels. Algorithm performance falls substantially at increased power aggregation where maximum performance observed was

[1] Emulating upstream feeder level measurements
[2] Emulating sample rate simulating measurement from domestic smart meters.
[3] Measured in RMSE normalised against RMSE of the Mean algorithm.
[4] The meaning of ideal is defined in Section 3.2

exhibited by seq2seq with a MNRMSE of `0.36`. At these higher power aggregation levels, algorithms appeared to rely increasingly on playing back learned patterns observed in the ground truth.

The Neural network based algorithms appeared to exhibit a surprising insensitivity to temporal aggregation with performance of MNRMSE of `0.72` exhibited at temporal and power aggregation levels representing to those expected for a typical domestic smart meter. It should be noted however that these results suffer from the same concerns regarding evaluation of error as those discussed with finite state algorithms.

Taking all of the above into consideration, it is the author's view that, while appearing promising, the relatively high performance observed may is predicated on several precarious assumptions which will likely act to substantially worsen these results. Though it remains for future work to fully quantify this assertion, the apparent high reliance of Neural network based algorithms, specifically at high aggregation levels, on specific learned prior knowledge of the loads it is attempting to dis-aggregate is antithetic to the high-level objective of this work, namely to infer information without *any* prior knowledge. It should be further noted that even the knowledge that there is a heat pump to disaggregation at all should be considered prior knowledge and not assumed. The author expects that when the algorithms are faced with the task of transfer learning, the performance will drop drastically.

In the author's view the focus of the NILM problem to specifically the task of extracting an exact load profile of individual appliances is perhaps not best suited to the requirements of the system operator. Perhaps this problem should be treated more as a task of separating types of load as opposed to individual appliances.

Chapter 6

Future Work

This study has stretched the conventional boundaries of the NILM field, operating with an unconventional load at extremes of sample-rate and power aggregation level. It is unsurprising, therefor, that in the process corner-cases and bugs have been uncovered. This section aims to provide prioritised accounting of areas of refinement and future study on this topic. Additionally, less vital areas are listed in Appendix 7.10.

6.1 Direct follow on

These areas are identified as critical follow-on work:

Down Sampling invariant performance metrics As noted in Section 4.4, the NILMTK experimentation API is found to down-sample the ground truth prior to calculating performance metrics. This is clearly the wrong approach[1], since this is not reflective of the real world. Fixing this would require a non-trivial refactor of the experimentation API and should be implemented as made a non-default optional behaviour since it will likely incur additional computation to calculate metrics. A subset of the results at high temporal aggregation results should then be replicated with this new feature and compared to establish if this has a major impact on accuracy. Decreasing aggregation level tests may need to be rerun subject to this outcome.

Anti-Alias Filtering Prior to downsample As noted in Section 4.4.1 substantial aliasing effects have been observed at particular sample-rates, it is assumed arising due to the lack of anti-alias filters employed prior to down-sampling within the experimentation API. This again would involve a not trivial refactor of the API code, to use the `scipy.signal.decimate()` method instead of `pandas.resample()`. This should again be implemented as an optional non-default feature given the likeliness of increased computational burden.

6.2 Future study

The follow areas (listed by priority) are seen as the logical continuation of this work.

6.2.1 Generalisation

A critical barrier to a wide-scale deployment of these techniques is creating an algorithm that works over a wide range of devices without requiring training tailored to a particular device. Thus testing that this can be achieved, and if so by which algorithms is of high importance, if only to save wasted effort. A simple test of this could be run using the existing datasets presented in this work by simply training on four households, and testing on the remaining household in a basic transfer learning test.

[1] At least in the context of work in which the effect of this down-sampling on performance is deliberately being studied.

6.2.2 Study Heat pump types

As observed in Section 3.3.7.3, a wide variety of types of heat-pump load profiles have been observed. This work used a largely heuristic categorisation system to group these load profiles and used only the dominant group [2] within the test dataset. Understanding the reason behind these differences seems prudent and may inform better targeted disaggregation strategies.

6.2.3 Testing with reactive power

Certain smart meters come with the ability to measure both real and reactive power and it seems probable that the additional information of two measurement channels will aid in disaggregation performance. Quantifying this however, relies on access to a heat pump dataset with both measurements.

6.2.4 Algorithm optimisation

Many of the algorithms within NILMTK allow configured beyond their default settings, the FHMM for example, allows the number of states to be set. In this work however, no configuration was conducted beyond the default settings. It is possible that some of these algorithms have been sub-optimally configured for the given task and could be tuned to improve performance. Some examples of configuration options include:

[2] Believed to be governed by a dead-band controller

Bibliography

[1] "Position paper on Distribution System Operation," OFGEM, Tech. Rep., Aug. 2019.

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

[3] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "NILMTK: An open source toolkit for non-intrusive load monitoring," in *E-Energy 2014 - Proceedings of the 5th ACM International Conference on Future Energy Systems*, Jul. 2014.

[4] N. Batra, R. Kukunuri, A. Pandey, R. Malakar, R. Kumar, O. Krystalakos, M. Zhong, P. Meira, and O. Parson, "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*, ser. BuildSys '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 193–202.

[5] O. Parson, "NIALM as a combinatorial optimisation problem - continued," Apr. 2011.

[6] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised Disaggregation of Low Frequency Power Measurements," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2011, pp. 747–758.

[7] J. Z. Kolter and M. J. Johnson, "REDD: A Public Data Set for Energy Disaggregation Research," p. 6.

[8] Z. Ghahramani and M. I. Jordan, "Factorial Hidden Markov Models.:," Defense Technical Information Center, Fort Belvoir, VA, Tech. Rep., Jan. 1996.

[9] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[11] J. Z. Kolter and T. Jaakkola, "Approximate Inference in Additive Factorial HMMs," p. 11, 2012.

[12] M. Zhong, N. Goddard, and C. Sutton, "Signal Aggregate Constraints in Additive Factorial HMMs, with Application to Energy Disaggregation," p. 9.

[13] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for nonintrusive load monitoring," Sep. 2017, comment: 8 pages, 3 figures.

[14] J. Kelly and W. Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. Seoul South Korea: ACM, Nov. 2015, pp. 55–64.

[15] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers, "Dataport and NILMTK: A building data set designed for non-intrusive load monitoring," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 210–214.

[16] S. Barker, A. Mishra, D. Irwin, E. Cecchet, and P. Shenoy, "Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes," p. 6.

[17] M. Wenninger, A. Maier, and J. Schmidt, "DEDDIAG, a domestic electricity demand dataset of individual appliances in Germany," *Sci Data*, vol. 8, no. 1, p. 176, Dec. 2021.

[18] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Sci Data*, vol. 2, no. 1, p. 150007, Dec. 2015.

[19] Goddard, Nigel, J. Kilgour, Pullinger, Martin, Arvind, D.K, Lovell, Heather, Moore, Johanna, Shipworth, David, Webb, Jan, Berliner, Niklas, Brewitt, Cillian, Dzikovska, Myroslava, Farrow, Edmund, Farrow, Elaine, Mann, Janek, Morgan, Evan, and Zhong, Mingjun, "IDEAL Household Energy Dataset," 2021.

[20] D. Murray, J. Liao, L. Stankovic, V. Stankovic, C. Wilson, M. Coleman, and T. Kane, "A data management platform for personalised real-time energy feedback," p. 15.

[21] "Powering the Nation Household electricicty-using habits revealed," Energy Savings Trust, Tech. Rep. 1, 2011.

[22] A. Monacchi, D. Egarter, W. Elmenreich, S. D'Alessandro, and A. M. Tonello, "GREEND: An Energy Consumption Dataset of Households in Italy and Austria," May 2014.

[23] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "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*. Memphis Tennessee: ACM, Nov. 2014, pp. 80–89.

[24] R. Lowe, "Renewable Heat Premium Payment Scheme: Heat Pump Monitoring: Cleaned Data, 2013-2015." *Department of Energy and Climate Change*, 2017.

[25] Alex Summerfield, Phillip Biddulph, Andrew Stone, Chris Grainger, Paolo Agnolucci, Colin Gleeson, Eleni Oikonomou, and Robert Lowe, "DECC_RHPP_160112_Detailed_analysis_report_v5.3.pdf," UCL Energy Institute, DECC RHPP Detailed Analysis Report 5.3, Mar. 2017.

[26] D. R. Bull and F. Zhang, "Signal processing and information theory fundamentals," in *Intelligent Image and Video Compression*. Elsevier, 2021, pp. 59–105.

[27] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, "It's Different: Insights into home energy consumption in India," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. Roma Italy: ACM, Nov. 2013, pp. 1–8.

[28] M. Maasoumy, "BERDS-BERkeley EneRgy Disaggregation Data Set," 2013.

[29] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *2013 IEEE Electrical Power & Energy Conference*, Aug. 2013, pp. 1–6.

[30] N. Batra, O. Parson, M. Berges, A. Singh, and A. Rogers, "A comparison of non-intrusive load monitoring methods for commercial and residential buildings," Aug. 2014.

[31] Kyle D. Anderson, Adrian Ocneanu, Derrick R. Carlson, and Anthony G. Rowe, "BLUED : A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research," 2012.

[32] A. S. Uttama Nambi, A. Reyes Lua, and V. R. Prasad, "LocED: Location-aware Energy Disaggregation Framework," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. Seoul South Korea: ACM, Nov. 2015, pp. 45–54.

[33] L. Pereira, F. Quintal, R. Gonçalves, and N. J. Nunes, "SustData: A Public Dataset for ICT4S Electric Energy Research:," in *ICT for Sustainability 2014 (ICT4S-14)*, Stockholm, Sweden, 2014.

[34] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the Accuracy of Appliance Identification Based on Distributed Load Metering Data," p. 10, Jan. 2012.

Chapter 7

Appendices

7.1 Links

NILMTK `https://github.com/nilmtk/nilmtk`
Fork Of NILMTK `https://github.com/BenjaminFrazer/nilmtk`
Code and Data `https://github.com/BenjaminFrazer/thesis_tools`

7.2 Household Dataset Metrics

Figure 7.2.1 presents a the distribution of a selection of metrics within the heat pump dataset [1]

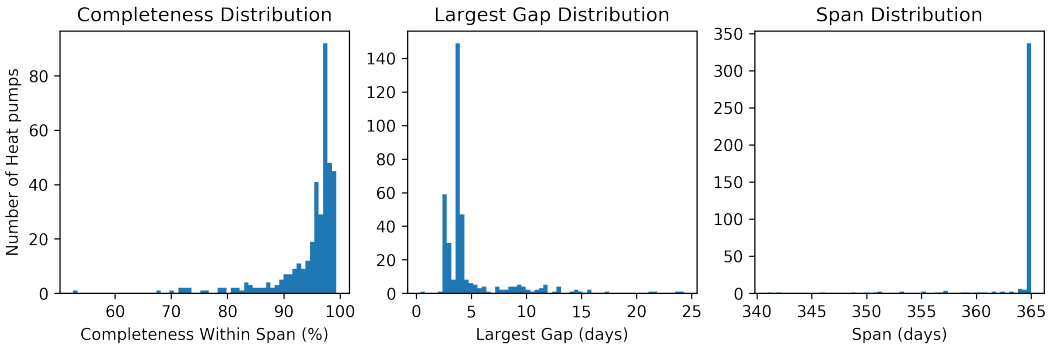

Figure 7.2.1: *Distribution of selected metrics within the Heat pump dataset (empty datasets omitted).*

Finally the completeness of the dataset is high with over 40 datasets in excess of 98% completeness. The vast majority of the non-empty datasets seem to be exactly one year with none exceeding this length and the worst outliers at no fewer than `340` days. The second observation is that only a single dataset has a largest gap of less than two days, with the next grouping at around `2.5` days [2]. Thus none of the heat pump datasets are fully complete over their entire span. It can also be seen that the vast majority (>80%) of datasets have a largest gap lower than `7` day with none having greater than `25`.

7.3 Heat pump Dataset Metrics

Figure 7.3.1 plots the distribution of selected metrics throughout the dataset.

[1] It should be noted that all empty datasets have been omitted from Figure 7.2.1.
[2] It is assumed that this can be attributed to the synchronised dataset wide gaps discussed in the prior section, though why just one dataset was affected has not been investigated.

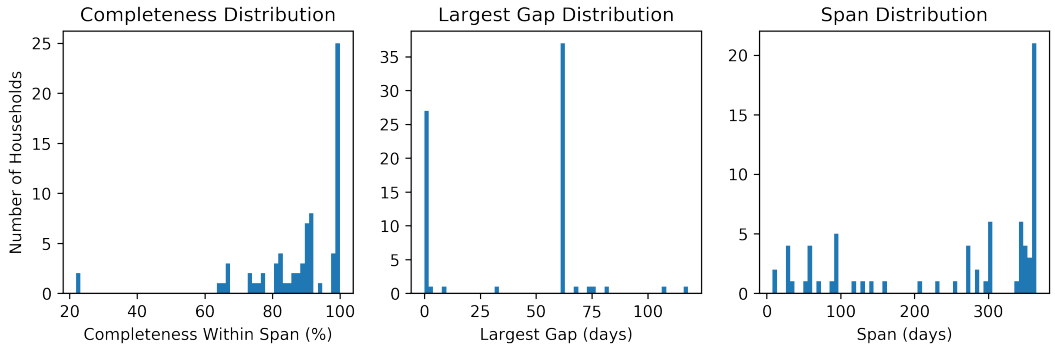

Figure 7.3.1: *Distribution of selected metrics within the household dataset (empty datasets omitted).*

As seen in the prior section, the household dataset is highly fragmented, largely as a result of missing one or more files each containing exactly one months worth of data. This can be clearly observed in the "span" metric's distribution with regular groupings at ≈ thirty day intervals [3]. The dataset span distribution confirms the observations from the prior section that the dataset spans at most one year with only ≈ 20 households appearing to span this entire period. Figure 7.3.2 plots only data without missing months discarding the data with "largest gap">=30.

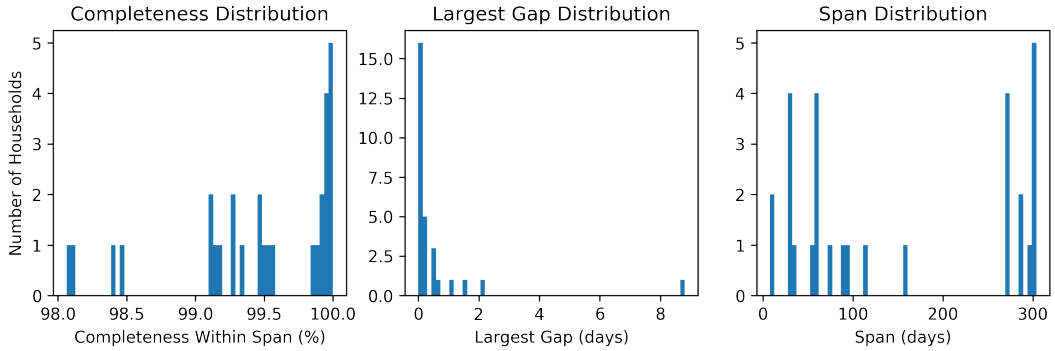

Figure 7.3.2: *Distribution of selected metrics within the household dataset (datasets missing months omitted)*

Here we see that of the remaining data, the completeness is extremely high with all exceeding 98% completeness within their spans.

7.4 Programmatically separating Household heating types

A naive approach to discriminate between households containing undesired appliances might be to simply discard sites with a site meter average power is greater than some threshold. This approach may be tested on the known heating types, with the assumption that households using oil heating do not make heavy use of electrical heating. Though imperfect, testing against the limited 'known heating types' set, give an idea of how effective it might be at discriminating the as-yet unknown heating types[4]. Figure 7.4.1 shows the population distributions of site meter average power by heating type.

[3] Such groupings also exist within the other metrics but are less visible

[4] We assume that where a heating type is known that this is the exclusive or by far the dominant heating type in use by this household.

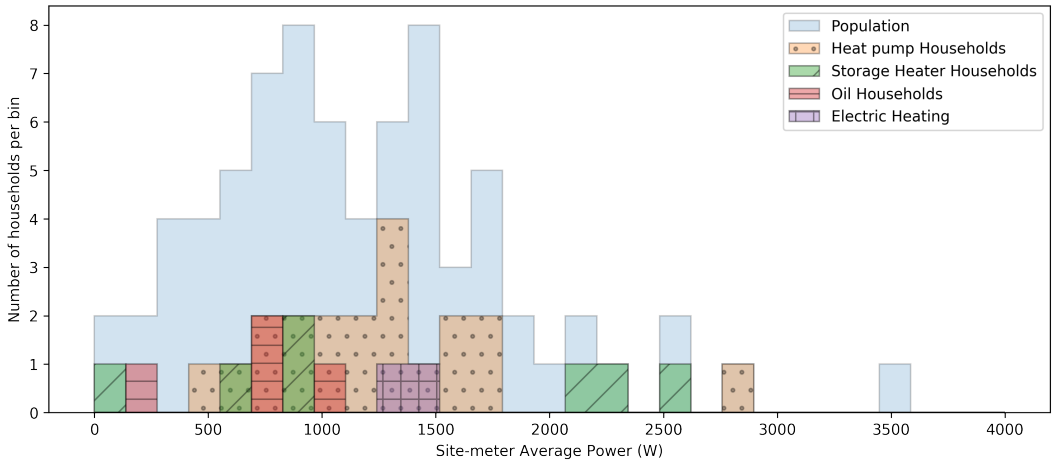

Figure 7.4.1: *Histogram of average site meter power (W) for households with known heating types as well as the total dataset (population)*

It can be seen that, though there appears to be a slight difference in population mean between the datasets the overlap is such that no clean separation of the sets can be achieved. A threshold set to `1200W` would successfully discard `17/29` of the known undesired heating types, while retaining `32/68` of the total population. A secondary issue with this approach however, is that the lower this threshold is set, the greater the explicitly introduced bias towards low-power households and thus, likely, easier disaggregation.

An examination of the load profiles of known heating types however does reveal certain differences between the heating types, which could be exploited (Figures 7.4.2, 7.4.4, 7.4.3).

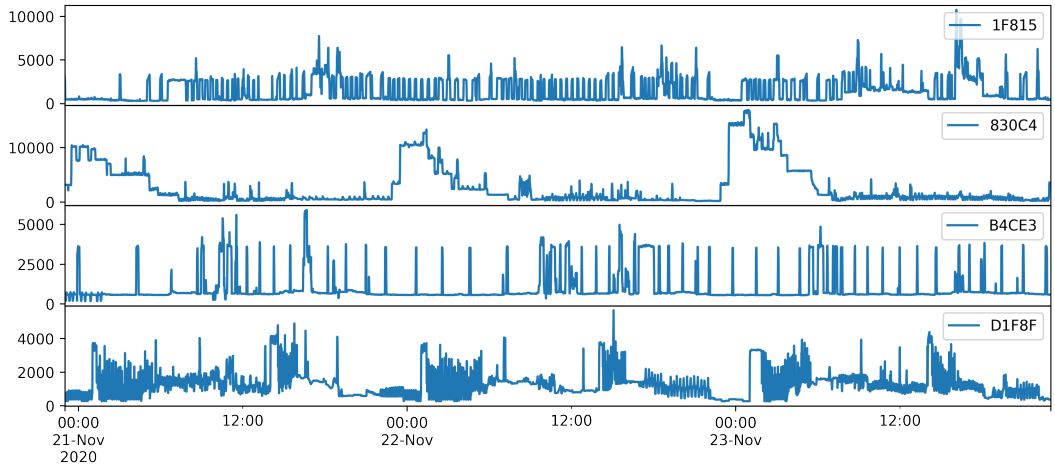

Figure 7.4.2: *Representative Sample of Households known to contain Heat Pumps over a three day period*

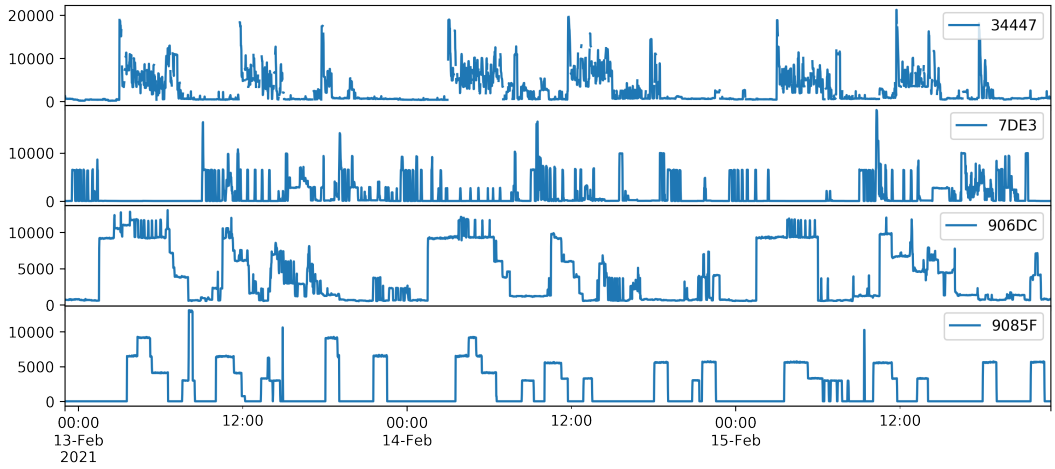

Figure 7.4.3: *Representative Sample of Households known to contain Storage heating over a three day period*

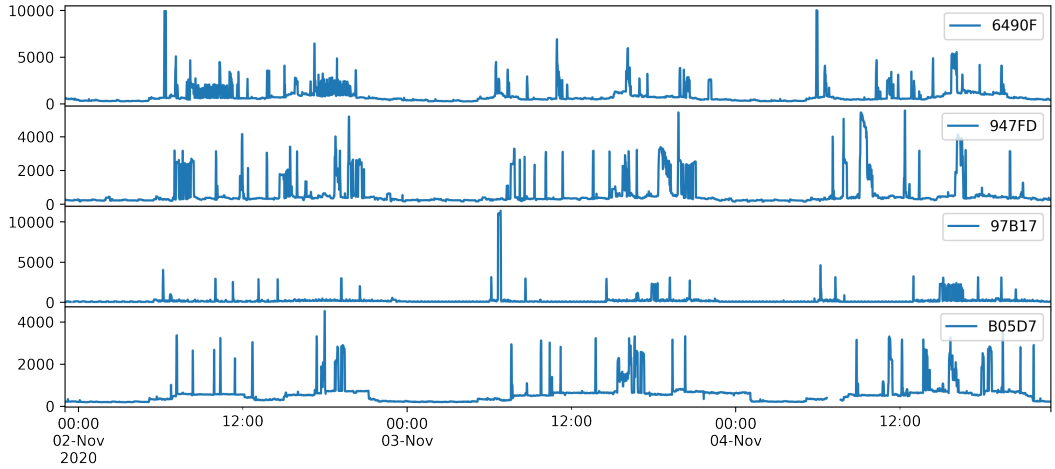

Figure 7.4.4: *Representative Sample of Households known to contain Oil heating over a three day period*

The feature common to the electrically heated households is the relatively random energy usage thought the day. The households heated with oil have the dominant electricity use almost exclusively between the hours of 7:00 and 19:00, whilst all of the heat pump/storage heater load profiles shown above either show no predominant time of day, or their daily use falls at least partially outwith this range. This also makes intuitive sense since the dominant electrical loads excluding heating will be appliance loads whose use depends predominantly on human activity during waking hours (oven, toaster, washing machine etc.).

A crude metric to quantify a given load profile's relative weighting towards daytime usage can be defined by taking the average power drawn between the hours of 0:00 and 6:00 vs overall average. For households we would expect this ratio to be lower than unity, given the expected low activity during the night while we would expect that houses with heat pumps would be closer to unity, while storage heater households, which preferential heat at night would be weighted greater than unity. This metric will henceforth be referred to as the 'night-time usage biias'.

$$\text{NTUB} = \frac{\mu_{\text{night}}}{\mu_{\text{total}}} \tag{7.4.1}$$

Where μ_{night} is the mean site-meter power for all time periods between 0:00 and 6:00, and μ_{total} is the mean over the entire dataset span.

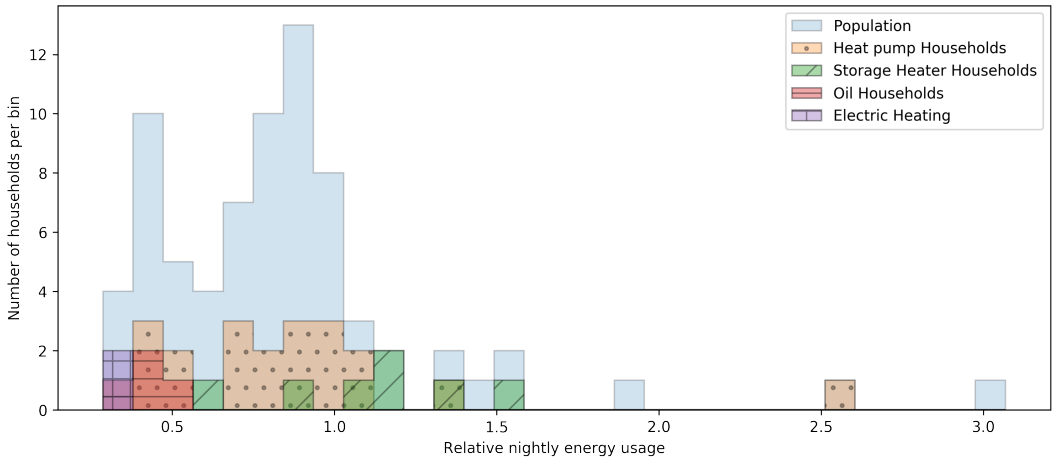

Figure 7.4.5: *Histogram of Night Time Energy usage Bias broken down by heating type, demonstrating the difficulty in separating desired from undesired heating types.*

With this metric we see a distinct change in the relative distributions of partial heating types. Though the overlap between heat pump and oil households still exists it is to a lesser degree with the tail of the heat pump distribution no-longer fully extending past the oil distribution. Secondly the storage heater distribution is now grouped more tightly and has shifted to the right, making possible to separate them fully from the oil distribution. The electric heating households however appear to have moved closer to the oil households making them impossible to separate. With a threshold of `0.5` this approach successfully discriminates `21/29` of the undesired heating types while retaining `19/68` of the population as a whole. Though only a fractional improvement over the prior approach, this method no-longer explicitly biasses the remaining set towards lower power levels and is thus preferable.

7.5 Selected Datasets

7.5.1 Selected Heat Pumps

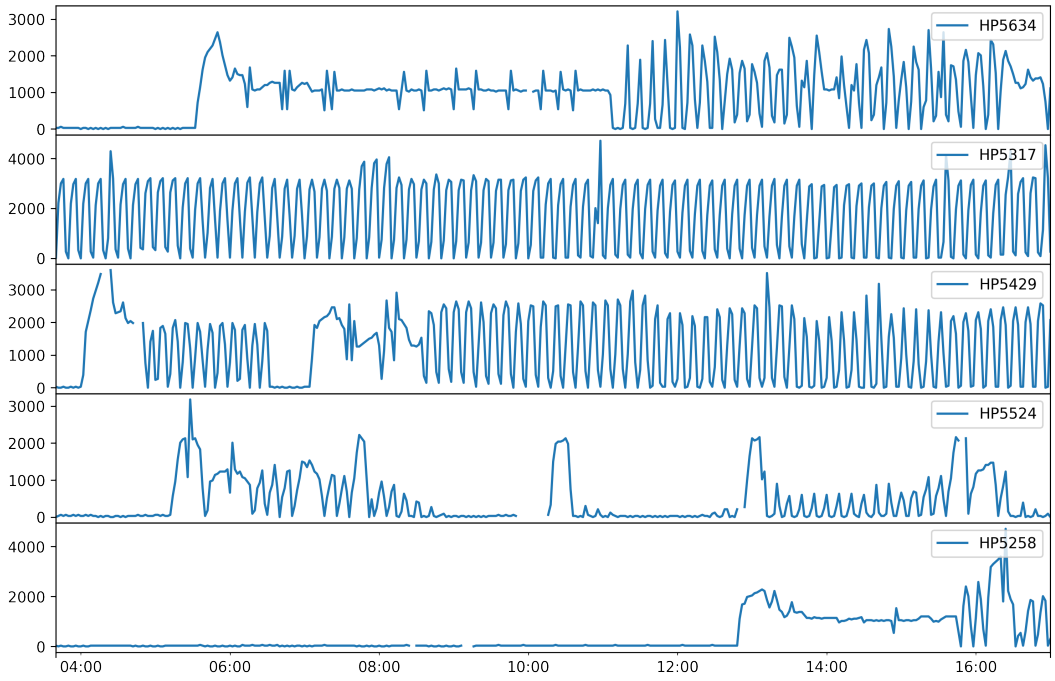

Figure 7.5.1: *Sample of the selected heat pump load profiles*

7.5.2 Selected Households

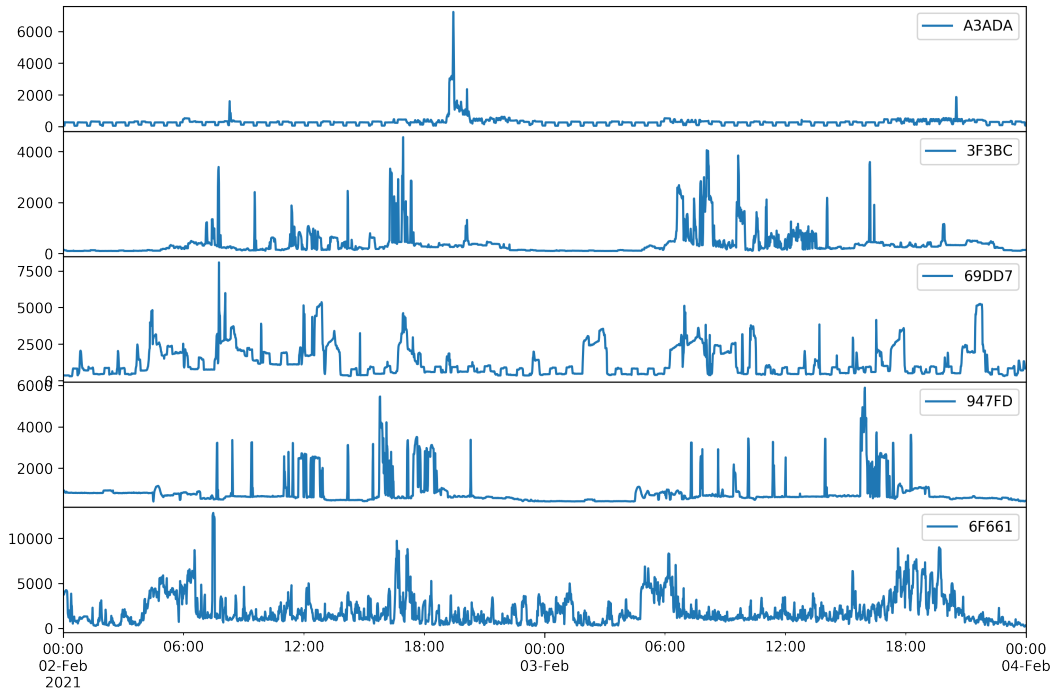

Figure 7.5.2: *Two day sample of the five selected household load profiles*

7.6 NILMTK Install Guide

7.7 Installation

Overview: This guide will hopefully take you through the NILMTK install using python virtual env. The suggested method of installation via conda didn't work (at least on linux), but this **issue** was found on the github page and suggested just using pip directly with a virtual environment. For this the virtualenv tool was used in conjunction with the `virtualenvwrapper`.

From the issues page:

> The solution from @TristanMenzinger I think should work, however I usually install it another way and so far I have had no problems.
>
> I recommend creating an environment with python 3.8 or lower and install via pip, the following line should work to install nilmtk: pip install git+`https://github.com/nilmtk/nilmtk@0.4.3`

Dependencies

- $\text{nilmtk}_{\text{metadata}}$
- <=Python 3.8

I found that when I tested the installation it complained about lacking the `nilmtk_metadata` package which can be installed as follows:

```
pip install git+https://github.com/nilmtk/nilm_metadata/@0.2.4
```

Instructions:

- download the `virtualenv` and `virtualenvwrapper` tools.
- Create a virtual environment with a python version < 3.8 with the `mkvirtualenv` command see managing-python-versions for details.
- Activate that virtual environment with the `workon` command.
- Install `nilmtk` and `nilmtk_metadata` using pip
- test the install by loading the package

The above boils down to:

```
sudo pacman -S virtualenv virtualenvwrapper
yay pthon3.8
mkvirtualenv --python=/usr/bin/python3.8 nilmtk-env
workon nilmtk-env
pip install git+https://github.com/nilmtk/nilm_metadata/@0.2.4
pip install git+https://github.com/nilmtk/nilmtk@0.4.3
python -c "import nilmtk"
```

7.7.1 Make a virtual environment with a particular version of python

- Download the desired version of python
- make the new virtualenv with

```
mkvirtualenv --python=/usr/bin/python3.6 nilmtk-env
workon <name>
```

7.7.2 Install with contrib

Overview The contrib package contains multiple more "bleeding edge" disaggregation algorithms which have been contributed by the community. The installation of NILMTK using venv is however slightly more complicated than before due to:

1. a problem with the versioning of the NILTK package.
2. Dependency troubles

Steps

1. Create and enable a virtual environment for NILMTK
2. install nilmtk-metadata though pip from github
3. [optionally fork &] clone nilmtk to local file system
4. checkout the version tag you wish to use (in my case the most recent `0.4.3`)
5. edit `setup.py` for the correct version number (`0.4.3`) see section 7.7.2.1
6. comment out imports in `nilmtk_contrib/disaggregate/__init__.py` see section 7.7.2.2
7. Pip install from your local copy of the repo
8. install nilmtk-contrib from github using pip

This amounts to:

```
mkvirtualenv nilmtk-env
workon nilmtk-env
pip install git+https://github.com/nilmtk/nilm_metadata/@0.2.4 # install
↪ nilm_metadata
git clone https://github.com/nilmtk/nilmtk ~/python/nilmtk/ # clone nilmtk to
↪ local filesystem
git clone https://github.com/nilmtk/nilmtk-contrib ~/python/nilmtk-contrib/ #
↪ clone contrib to local filesystem
git checkout tags/0.4.3 # get the correct version of nilmtk
git branch <yourBranch> # create and checkout your own branch to make edits
↪ (optional)
git checkout -b <yourNewBranch>
# make edits to local repos now
pip install ~/python/nilmtk/ # install NILMTK from local filesystem
```

7.7.2.1 The Problem With NILMTK Versioning

When installing via pip, no matter what version you try install by either pointing to a specific version tag (say 0.4.3) or commit hash, pip always thinks it's installing version `0.4.0.x`. This is because pip takes it's version number directly from the `version` variable set by `setup.py` which, is always set as `0.4.0.x` for some reason.

All of this is only becomes a problem if you wish to install the contrib package which has a requirement for `nilmtk>=0.4.3` which pip cannot resolve, because it thinks you have `0.4.0.xx`.

Solution To solve this you will need to clone the base NILMTK repo into your local file system and change the `setup.py` file to give you a version string of `0.4.3.xx`.

7.7.2.2 The Problem With Contrib Dependencies

When I attempted to get contrib working in 2022 using pip, I ran into numerous problems resolving dependencies. Despite getting my python packages to the versions specified in the documentation, the issues persisted. Since the issues often related to imports of non existent sub-modules, the import of any one algorithm would throw an error even if the import itself wasn't being used.

Solution To solve this download contrib onto the local file system and comment out the imports of the broken disaggregation in `nilmtk_contrib/disaggregate/__init__.py` install locally with pip.

```
from nilmtk.disaggregate import Disaggregator
from .dae import DAE
from .dsc import DSC
from .afhmm import AFHMM
# from .afhmm_sac import AFHMM_SAC
from .seq2point import Seq2Point
from .seq2seq import Seq2Seq
from .WindowGRU import WindowGRU
from .rnn import RNN
# from .rnn_attention import RNN_attention
# from .rnn_attention_classification import RNN_attention_classification
# from .resnet import ResNet
# from .resnet_classification import ResNet_classification
# from .bert import BERT
```

7.8 Supplementary experimental results

Table 7.8.1: *RMSE results for power aggregation sweep with Temporal aggregation Ts=2min*

Household	Aggregation	DAE	Seq2Point	Seq2Seq	CO	FMHH
1	1	300.56	163.71	245.33	899.41	753.28
2	1	306.66	170.65	234.2	1395.24	1296.29
3	1	289.98	123.78	213.91	974.71	780.77
4	1	276.13	167.81	216.3	671.71	693.5
5	1	300.81	181.79	265.8	684.62	567.24
1	2	542.2	329.1	408.0	1274.22	854.74
2	2	452.72	195.1	274.34	1450.14	1279.62
3	2	548.2	381.38	435.12	1285.54	994.78
4	2	545.68	373.48	422.6	1221.63	948.59
5	2	593.57	491.22	527.88	1181.93	922.56
1	3	581.68	389.0	429.26	1560.47	1148.89
2	3	440.02	245.74	318.93	1562.66	1297.47
3	3	576.22	381.65	413.31	1648.6	1322.09
4	3	582.33	418.36	458.66	1775.8	1256.36
5	3	682.05	518.98	548.1	1679.31	1388.36
1	4	612.32	435.26	483.76	1828.21	1251.67
2	4	499.5	337.94	409.91	1699.43	1313.76
3	4	594.64	457.99	453.91	1982.66	1424.07
4	4	599.27	500.23	516.16	2062.87	1304.19
5	4	726.61	630.76	627.61	2060.57	1507.09
1	5	711.18	619.01	631.17	2068.2	1255.29
2	5	620.11	473.91	496.97	1791.18	1313.73
3	5	684.27	573.36	576.26	2235.34	1426.26
4	5	613.85	601.96	564.07	2203.16	1304.48
5	5	731.68	720.24	707.92	2379.6	1511.66

Note that temporal aggregation sweep Tables 7.8.2, 7.8.3, 7.8.4 contain the average over all five heat pumps at each aggregation level, given the tables length otherwise. The full results can be found with other supplemental material at `https://github.com/BenjaminFrazer/thesis_tools`.

Table 7.8.2: *Average RMSE results for Temporal aggregation sweep with power aggregation level 1*

Sample Period (s)	CO	FMHH	Mean	DAE	Seq2Point	Seq2Seq
60.0	774.66	698.08	958.24	222.59	113.49	157.89
120.0	918.85	818.22	958.72	267.4	155.71	232.42
180.0	749.97	666.3	923.0	327.71	210.8	278.02
240.0	760.88	669.98	884.06	315.93	218.35	286.19
300.0	662.84	592.34	849.9	415.98	305.04	375.78
360.0	666.89	582.11	815.71	379.59	232.73	295.68
420.0	605.6	540.04	791.13	380.07	220.59	294.65
480.0	581.22	518.26	764.96	340.5	225.5	276.73
540.0	553.92	501.29	754.1	339.06	215.94	279.85
600.0	535.11	492.79	743.41	339.67	212.86	262.43
660.0	518.63	491.71	743.78	337.92	196.14	278.46
720.0	541.92	495.76	745.65	384.7	216.24	282.68
780.0	541.28	495.0	743.91	370.34	210.33	283.86
840.0	553.36	496.26	743.45	431.45	205.65	296.23
900.0	543.21	490.63	737.68	404.4	209.93	295.68

Table 7.8.3: *Average RMSE results for Temporal aggregation sweep with power aggregation level 3*

Sample Period (s)	CO	FMHH	Mean	DAE	Seq2Point	Seq2Seq
60.0	1605.59	1285.94	958.24	512.6	300.12	338.21
120.0	1639.42	1282.63	958.72	571.43	383.43	439.73
180.0	1568.42	1179.16	923.0	602.65	470.82	495.93
240.0	1476.18	1096.73	884.06	595.18	472.88	490.95
300.0	1421.45	1062.55	849.9	641.53	607.2	592.73
360.0	1483.77	1030.26	815.71	596.09	516.81	518.55
420.0	1337.36	1009.91	791.13	606.7	518.75	534.19
480.0	1328.39	987.45	764.96	599.1	536.06	525.7
540.0	1286.45	979.73	754.1	581.73	553.57	515.47
600.0	1287.11	969.42	743.41	555.15	521.24	520.95
660.0	1250.18	972.44	743.78	574.34	511.27	506.48
720.0	1260.23	978.45	745.65	584.67	514.66	533.6
780.0	1217.04	979.39	743.91	582.34	514.57	512.99
840.0	1208.68	979.78	743.45	588.64	520.1	515.82
900.0	1253.41	980.21	737.68	581.57	510.7	507.89

Table 7.8.4: *Average RMSE results for Temporal aggregation sweep with power aggregation level 5*

Sample Period (s)	CO	FMHH	Mean	DAE	Seq2Point	Seq2Seq
60.0	2184.9	1385.84	958.24	675.81	607.35	574.91
120.0	2173.3	1362.28	958.72	684.46	610.67	594.28
180.0	2057.37	1207.68	923.0	699.17	656.27	622.28
240.0	1923.42	1112.29	884.06	665.28	626.78	601.75
300.0	1880.99	1077.13	849.9	718.81	710.3	677.69
360.0	1741.86	1044.42	815.71	663.16	617.35	610.41
420.0	1659.71	1022.5	791.13	733.14	676.98	638.75
480.0	1576.35	1000.53	764.96	677.89	672.76	636.98
540.0	1453.75	992.44	754.1	673.47	635.08	624.52
600.0	1358.43	983.03	743.41	651.78	639.79	599.76
660.0	1528.73	986.16	743.78	667.67	655.21	605.26
720.0	1471.22	991.89	745.65	653.14	639.04	605.73
780.0	1495.74	992.91	743.91	635.99	629.91	600.97
840.0	1481.87	992.49	743.45	639.26	628.44	611.62
900.0	1334.4	993.42	737.68	639.63	602.29	603.27

7.9 Unique Column Descriptors

Table 7.9.1: *All Unique Column Descriptors Found in the household load dataset*

Unique Column Descriptors
<blank string>
Solar
Twl rail+Store htrs+oven+hob
excluding car ch
left unit
Whole house electricity
Whole House
Domestic
Whole Property + Solar generatio
Whole house
Heat pump
Car Charger
whole house test
Whole House
Heating and hot water
+ve to meter
originally THTC cir
Other
House Supply
THTC
and solar generatio
originally domestic
Sub Circuit
Turbine
1x Turbine generation
Heating
+ve from meter
Shower + hot water
New extension + solar thermal
Wind generation + car charger
Heating and hot wat
Water heater
Domestic loads

7.10 Additional Improvements

7.10.1 Experimentation API

- better specification of experiments (multi dimensional sweeps)
- Better automation of test sweeps to get better coverage of the "power-temporal area"
- storing of experimental results in a standardized fashion (partially implemented in `https://github.com/BenjaminFrazer/nilmtk`)

7.10.2 Structural Changes towards Large Scale disaggregation

- On the fly dataset synthesis (partially implement in `https://github.com/BenjaminFrazer/nilmtk`)

7.10.3 Contrib

- unit tests for to verify 'supported' library versions
- better documentation of dependency version

7.10.4 Miscellaneous

- Better Pip integration (pip takes wrong version of NILMTK, see Section appNILMTK$_{\text{InstallGuide}}$)

7.11 NILM Dataset Survey

7.11.1 Survey Overview

Table 7.11.1: *Table 7.11.2 Key*

Symbol	Description
R	Residential
C	Commercial
P	active power
Q	Reactive Power
V	Voltage
I	Current
S	Apparent Power
f	frequency
Θ	Current/Voltage Phase shift

Table 7.11.2: *Summary of Survey results*

Name	NILMTK	Sub-meter	Location	Type	Properties	Period(d)	Resolution	Channels	EV	AC	HT	HP	cite
REDD	Y	Y	Boston,USA	R	6	19d	15kHz/3s[5]	SVP	0	2	6	0	[7]
UK-Dale	Y	Y	UK	R	5	499d	1s/6s [52]	SPV	0	0	1	0	[18]
Dataport	Y	Y	TX&NY&CA,USA	R	75	3.25y/1m	1min	P	2+	1+	5+	0	[15]
iAWE	Y	Y	India	R	1	73d	1Hz	PQSfVI	0	2	0	0	[27]
Smart	Y	Y	MA, USA	R	3-5?	90d	1min	PS	0	0	2	1	[16]
BERDS	N	Y	USA	C	3	7d	20s	PQS	0	0	0	1	[28]
DEDDIAG	Y	Y	Germany	R	15	3.5y	1Hz	P	0	0	0	1	[17]
AMPds	Y	Y	Canada	R	1	730d	1min	PQSfVI	0	1	1	1	[29]
Ideal	Y	Y	UK	R	39	?	1Hz	S/P[6]	0	0	1+	0	[19]
REFIT	Y	Y	UK	R	20	21m	8s	P$	0	0	3	0	[20]
HES	Y	Y	UK	R	225/26[7]	1m/1y[54]	2min	P	0	1	72	0	[21]
GREEND	Y	Y	Austria/Ita	R	9	310d	1Hz	P	0	0	1	0	[22]
ECO	Y	Y	Switzerland	R	6	244d	1Hz	PIVΘ	0	0	0	0	[23]
COMBED	Y	Y	India	C	1	30d	30s	?	0	0	0	0	[30]
BLUED	N	Y	PA, US	R	1	8d	12kHz	PIV	0	1	0	0	[31]
DREDD	Y	Y	Netherlands	R	1	6m	1Hz	P	0	1	0	0	[32]
SustData	N	N	Portugal	R	50	504d	1min	?	0	0	0	0	[33]
Tracebase	N	Y	Germany	R	[8]	[55]	1-10s	P	0	0	0	0	[34]

[52]Here the site and sub-meter are recorded at different resolution. Both resolutions are displayed as follows `<site meter resolution>/<sub-meter resolution>`.

[53]Site meter is measures apparent power while sub-meters measure real power

[54]In this study 255 houses were studies for 1 month and 26 were studied for a year.

[55]The Tracebase dataset consists of a collection of power consumption traces rather than a whole household dataset