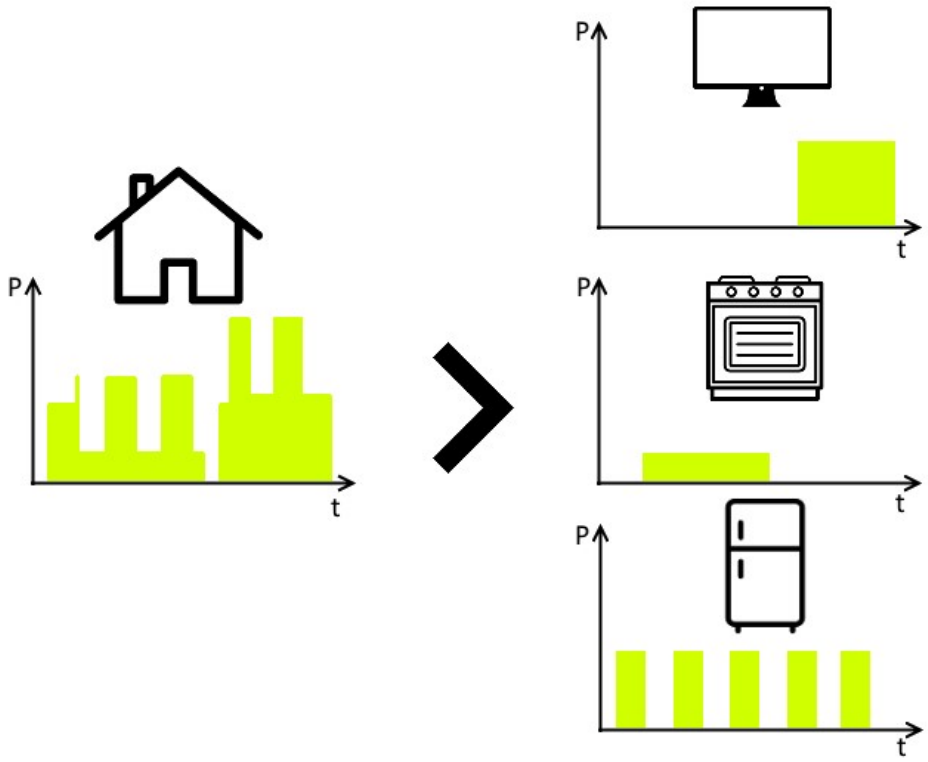
Capabilities And Limits

Of Non-Intrusive Load Monitoring

Aarhus University, Department of Engineering

13. April 2016



A Master's Thesis by Rune A. Heick
Student number: 11061
AU id: au478524
Card number: 544618

Supervisor:
Rune Hylsberg Jacobsen
rhj@eng.au.dk

Abstract

This master's thesis seek to investigate some of the common approaches used for non-intrusive load monitoring (NILM), in order to determine the capabilities and limits. It is assumed that the NILM application is deployed in a modern smart grid. The impact the smart grid infrastructure have on the data quality is investigated. It is shown that equipment and network errors are the major quality decreasing factors. Simple gap filling methods are evaluated in order to improve the quality. It is shown that simple gap filling can improve the performance. The performance is also shown to be depended on the number of appliances that are in a given environment, and the consumption requirements of the appliance. Interference from other appliances have a major effect on the performance. Higher sample rates and norm filters is shown decrease the interference. In general is a acceptable performance only obtained on the top consumers in the home. A small case study illustrates how a *service provider* in a smart grid can benefit from the information collected with a NILM application.

Resumé

Dette speciale undersøger mulighederne og begrænsningerne med NILM. Kvaliteten der kan forventes af data modtaget i en moderne smart grid infrastructure undersøges. Det er vist at fejl i måleudstyr og servere er nogle af de store kvalitets dæmpende aspekter. Forskellige "Gap filling" metoder er undersøgt, for at udbedre fejlene, og øge kvaliteten. Det er vist at simple "Gap filling" metoder kan øge kvaliteten. Det er vist at resultatet af NILM bliver væsentligt forringet hvis der er mange apparater i det samme miljø. Energikravene for de enkelte apparater har også meget at sige for resultatet. Interferens fra andre apparater er med til at forringe resultatet af NILM. Det er vist at højere sample hastigheder vil kunne og "norm filtre" vil kunne forbedre resultaterne markant. Generelt er det kun de apparater der bruger mest energi i husstanden, der får acceptable resultater. Et lille brugsscenarie viser hvordan en *service provider* kan benytte informationer fra NILM applikationer til at udvide deres forretningsmuligheder.

Abbreviations

TV House Dataset is a programmable machine.
Activity is a programmable machine.
Appliance Interference is a programmable machine.
Background Consumption is a programmable machine.
Completeness is a programmable machine.
Complexity is a programmable machine.
Customer is a programmable machine Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

.
Data Broker is a programmable machine.
Distribution is a programmable machine.
Gap Size Correction Capability is a programmable machine.
Generator is a programmable machine.
Knowledge is a programmable machine.
Markets is a programmable machine.
Norm Filter is a programmable machine.
Operator is a programmable machine.
Other Category is a programmable machine.
Purge And Merge Process is a programmable machine.
Sample Availability is a programmable machine.
Service Provider is a programmable machine.
Top Consumers Problem is a programmable machine.
Transmission is a programmable machine.

ECO Electricity Consumption and Occupancy.
EMD empirical mode decomposition filling.

FHMM factorial hidden Markov models.

HMM hidden Markov models.

IMF intrinsic mode functions.

NILL non-intrusive load leveling.
NILM non-intrusive load monitoring.
NILMTK NILM-Toolkit.
NIST National Institute of Standards and Technology.

P-G Papoulis-Gerchberg.

SSA Spatio-Temporal filling.

UDP user datagram protocol.

Glossary

TV House Dataset is a programmable machine. 52, 54, 55
Activity is a programmable machine. 7, 8, 11, 13, 62
Appliance Interference is a programmable machine. 46, 50, 54–56, 66
Background Consumption is a programmable machine. 43–45, 47–52, 54, 55, 59, 63, 66, 70
Completeness is a programmable machine. 51, 55, 56, 66
Complexity is a programmable machine. 51–56, 62, 66
Customer is a programmable machine Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

. 3, 4, 58, 61, 62, 65
Data Broker is a programmable machine. 57
Distribution is a programmable machine. 3, 58
Gap Size Correction Capability is a programmable machine. 16
Generator is a programmable machine. 3, 58, 62
Knowledge is a programmable machine. 17–20
Markets is a programmable machine. 3, 4
Norm Filter is a programmable machine. 50, 51, 59, 61, 64, 66, 71
Operator is a programmable machine. 3, 4, 22
Other Category is a programmable machine. 12, 43
Purge And Merge Process is a programmable machine. 50
Sample Availability is a programmable machine. 7–11, 13, 70
Service Provider is a programmable machine. 3–5, 57–61, 66
Top Consumers Problem is a programmable machine. 25, 26
Transmission is a programmable machine. 3

ECO Electricity Consumption and Occupancy. 30–32, 36, 43, 56, 66
EMD empirical mode decomposition filling. 16, 70

FHMM factorial hidden Markov models. 27, 33, 38, 42–44, 46, 47, 50–52, 56, 59, 61, 71

HMM hidden Markov models. 25–28, 33, 51, 56

IMF intrinsic mode functions. 16

NILL non-intrusive load leveling. 65
NILM non-intrusive load monitoring. i, 4–7, 11–13, 22, 23, 25–27, 32, 42, 44, 51, 54, 57, 58, 60, 62–66
NILMTK NILM-Toolkit. 7, 12, 26, 28

NIST National Institute of Standards and Technology. 1

P-G Papoulis-Gerchberg. 14, 15, 18, 20, 70

SSA Spatio-Temporal filling. 15, 70

UDP user datagram protocol. 6, 13

List of Figures

List of Tables

Contents

1 Introduction

In the past electricity production and consumption was localised to small communities. The communities often had one electricity utility company supplying the required energy. As the energy demand increased beyond single production facilities, the solution was to add more production facilities that could support in peak hours. This approach is costly, since such support facilities must be kept in a standby state, ready to deliver additional energy at a moments notice [1]. As the electric grid expanded, it began to interconnect the communities. The distribution of energy in the grid also became a challenging task. Today's electricity grids are a more global grid that span many communities, and even countries. The consumer plays a passive part in the grid. It is the operators of the electricity grid that must ensure that energy is delivered at the different consumers, at correct quantities, without overloading the grid. Furthermore must the impact errors and damages have on the grid be minimised [2].

1.1 Smart-grid

In order to combat some of the problems in the grid today, is a new type of electricity grid being deployed. As sensor technology became cheaper and more reliable the electricity utility companies started to integrate sensors in the grid. These sensors delivered information about the performance of key sections of the grid. This information was used to improve the performance of the grid. More types of energy producing devices, such as solar power and wind turbines, got connected to it. This created a need for more control of the grid and the energy production. This laid the foundation for the new grid type called smart-grid [2].

The National Institute of Standards and Technology (NIST) is tasked with the job of creating and maintaining the standards used for the American smart-grid. The concept behind the smart-grid is to create an electricity grid that enables two way communication. Besides electricity must information about the usage also flow between the consumers and the utility companies. Usage patterns must flow to the engineers tasked with controlling the grid, and control information must flow back to the consumers or control points in the grid. This can ensure the delivery of electricity more efficiently, reliably, and securely [3].

There are many different kind of stakeholders in today's smart-grid. In order to better understand the needs and relationships between the stakeholders have NIST split them in to seven domains. This helps to identify the different user interest in the grid, and their needs.

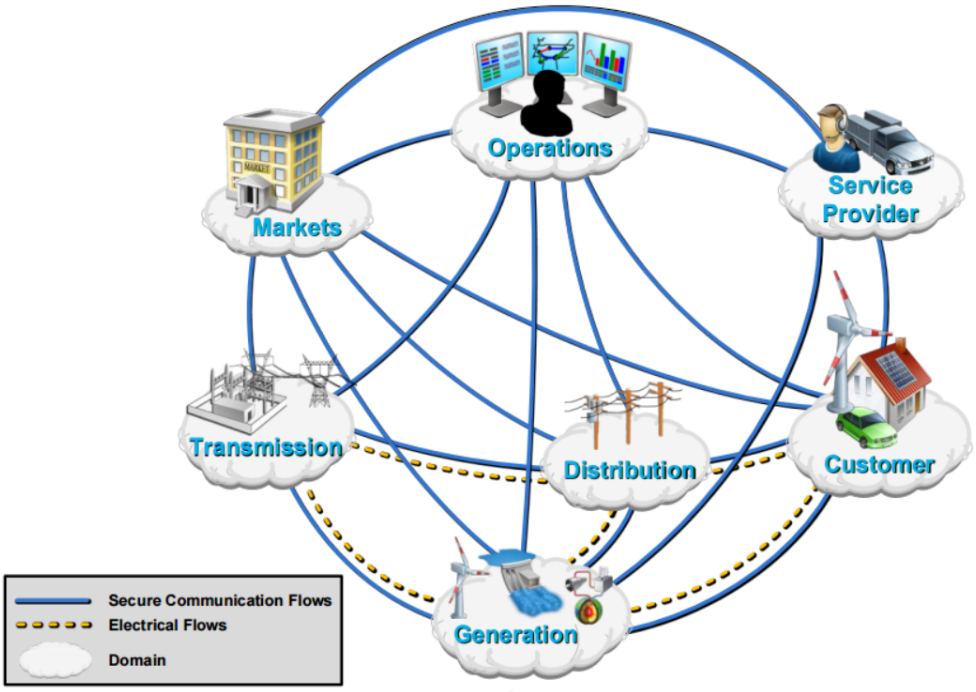


Figure 1.1: Conceptual model of smart-grid architecture. Source [4]

Figure 1.1 illustrates the seven domains. The seven domains each have a set of functions that is characteristic for the domain [4].

Generation: The *generator* role is the generators of electricity. This is typically coal, oil or nuclear power plants or large-scale hydro generators. This can also include energy storage facilities that stores energy for later distribution.

Transmission: *transmission* and transformers that hold the purpose of transporting the electricity over large distances.

Distribution: The *distribution* role are the units responsible of electricity distribution to and from the customers.

Customer: The *customer* role is the consumers of electricity. They may also generate electricity and sell it to the grid.

Operator: The *operator* role is managers of the movement and production of electricity. They are the group that ensures that the energy is efficiently delivered where it is needed.

Market: The *markets* is the instance responsible for sale and purchase of electricity between the different instances of the grid. Not just the *customer* uses this instance. Utility company can also purchase power from each other to keep up with demand at peek hours.

Service Provider: The *service provider* role are the organisations providing services to the utility companies or the *customer*. This could be home automation or information services.

1.2 Non-Intrusive Load Monitoring

The *operator* have the responsibility of controlling the flow of energy on the smart-grid. This requires the knowledge of the current consumption and a prediction about the future power requirement. This prediction is generality based on statistical data collected in the grid. If the prediction of the future power requirements can be improved, it is possible to create a better and more efficient power distribution. To accomplish this is a method known as non-intrusive load monitoring (NILM) developed.

The concept of NILM is to use the consumption information collected at household level, and use machine learning techniques to make a qualified guess on what appliances in the household is responsible for the current consumption. By using the information from the active appliances more precise estimates of future consumption can be made.

In today's smart-grid are all *customers* equipped with smart-meters. The smart-meter measures the consumption of the household, and reports it back to the smart-grid. This information can be used by the *markets* to automatically create billing information, or by the *operator* to regulate the power distribution. Since this information is used for distribution regulation it is send in real-time [4]. It is this information NILM applications hopes to use in order to create better predictions.

NILM can also create many opportunities for the *service provider*. Using NILM it is potentially possible to gesture about the usage of each appliance in a household. This can be used to better inform the resident about power saving opportunities. Studies have shown that detailed information about energy usage, makes the resident use less energy [5]. Other business opportunities, like selling the statistical information about the usage of specific appliances might also exists.

1.3 The Problem Statement

This master's thesis seek to investigate some of the common approaches used for NILM today, in order to determine the capabilities of such approaches. It is assumed that the NILM approaches are used in a modern smart-grid setting. In order to answer this broad question, the following questions must be investigated.

- **What quality of data can be expected from a smart-meter infrastructure?**
  In a smart grid is a lot of information collected from various points. For a system that needs the smart-meters data to be collected on a server, what quality can be expected? And where in the infrastructure that collects and transports the information over the Internet, is the quality being compromised?

- **How does sample errors and poor quality affect NILM applications?**
  If the system contains errors, due to samples being lost in the network, how does this impact the NILM applications? Does the quality effect the performance of NILM disaggregation?

- **What behaviours can be detected using NILM and where are the limits?**
  How detailed is it possible to gesture about the behaviour of the appliances in a household?

Where are the limits of susses? And what does these limits depend on?

- **What can be done to improve the performance of NILM technology?**
  Is it possible to clean data to improve quality and performance? What aspects is the bottlenecks for NILM technology today, and how does one improve on this basis?

If these question is answered it creates a strong foundation for gesturing about the the capability of NILM. The main motivation for this study is not to improve grid load predictions, but to get a better basis for validate different NILM related business opportunities for *service providers*.

1.4 The Approach

In order to investigate the questions in the problem statement is real data needed. Research on the popular methods and definitions in various areas of NILM, quality and machine learning has been analysed. The result is used to create the basis of the experiments conducted in this master's thesis.

1.4.1 The Data Approach

As dataset is the SmartHG dataset used. The SmartHG dataset is a collection of smart-meter data collected from 25 residential households in Denmark. The data is not filtered or cleaned in any way prior usage. This means that the data closely resembles what would be collected in a smart-grid. Furthermore does the dataset contain information about the usage pattern of selected devices. This information can be used to compare inferred usage patterns to the actual.

1.4.2 The Experiment Approach

Some of the experiments are conducted on real data, as it would be observed in a real deployment scenario. The behaviours observed in this types of experiment is used to design a series of experiments in controlled conditions, using artificial constructed datasets. These experiments serves the purpose of exposing the properties responsible for the observed behaviour. This improves the understanding of what properties creates wanted and unwanted behaviour.

Data Quality 2

Various projects today are focused on gathering data and analysing it. The gathered data is used for obtaining behaviours, habits and properties of the observed objects. This is done by using powerful statistical leaning algorithms, that are able to deduce these properties from the data. This approach is called data driven development, since the success is mainly determined by the data and not the algorithm.

When data is the central role of the system, the quality of the data is very important. Poor data can lead to wrong assumptions, and have a negative effect on the application. Choosing the correct dataset is therefore a key factor [6]. Looking at the quality of the data can help chose what dataset to use. Data quality can be described many ways, one of the more formal is from the ISO 8402 standard that describes quality as:

> *"The totality of characteristics of an entity that bear upon its ability of satisfy stated and implied needs"* [7].

This indicate that data quality is something that is very depended on the intended application, and is therefore hard to generalize.

Quality of data is a subject that is gaining more and more attention due to the fact that the quantity of data available is larger than ever before. This forces researchers to choose between datasets. A notion of quality in the dataset can help them choose. The heavy growth in available data is a result of projects that are moving data gathering from controlled labs, to the public. Many of these projects are citizen science projects. In citizen science projects are the citizens responsible for collecting the data, and not the researchers [8]. This enables researchers to gather enormous amount of data, but they are no longer in control of the conditions the data are collected in. This introduces errors and other quality decreasing factors.

Non-intrusive load monitoring (NILM) is a topic that have been in focus in the past years. This is due to the rise of the smart-meters, that makes it possible to measure at faster intervals, and collect the data on online services form real environments. But the smart-meter architecture is designed after billing and regulation purposes, and not load monitoring. The network architecture is therefore often based on the unreliable user datagram protocol (UDP), since it is more important to get the current information fast, than get all information. This courses some of the measurements to be lost in transition, which can degrade the completeness quality of the signal. The missing data can be a problem for load monitoring, since many methods of load disaggregation are based on learning techniques. The quality of the signal can also help identify if the collected data is suitable as a training set.

2.1 Quality Criteria

It is not uncommon that different areas of research has its own quality criteria. This is due to the fact that quality is a very domain specific subject. One of the areas that have been dealing with citizen data for many years are the Geographic information area, that are used for maps, weather prediction and climate research. They have come up with several ways of describing quality in spatial data [9]. Method for defining quality in time series data have also been developed [10]. To better define quality criteria in the smart-meter data inspiration from related work is used.

2.1.1 Related Work

Data quality is an area that recently have become a hot topic, due to the wast quantity of data. Many researchers strive to make tools that better can analyse data quality in different areas. In the area of spatial data is a *"Quality and Workflow tool"* being developed by the University of Wageningen [11]. The objective is to help researchers select the best suited data for a given data driven project. It does this by looking at different quality criteria, given by the user or found in standards for spatial data.

In bioinformatics is a tool named QCScreen developed to help create better dataset to metabolomics studies. In metabolomics studies are datasets often created by joining information from several different experiments of various quality. By using tools that can check the data quality and consistency to determine if a dataset is suitable for further processing, they are able to greatly improve the test results [12].

In the article *Taking a big Data approach to data quality in a citizen science project* [8] they talk about how quality assessment can be used to rate the believe on your data, and how to improve data collected in citizen science. The project focuses on bird observations, done by users on their smart-phones. They improve the quality by disallowing the user to send incomplete datasets to the database, and in this way forcing the user to only deliver high quality information. They then cross check the information with information from people in the same area, to see if it varies greatly.

In the NILM research area have data quality also emerge when talking about comparison between dataset. One of the tools working with this is the NILM-Toolkit (NILMTK) project [13]. In this they look at a series of quality parameter that can be used specifically for NILM. They point out that looking at the energy consumption observed by the main meter in relation to what is observed at the sub-meters tells a lot about the training set. This is since the success ratio of disaggregating appliances consuming lot of energy is higher than once that only consume a little.

One of the things all the methods have in common is trying to look at the completeness of the data. Some of the most low level criteria is the "sample availability". This metric describes how many samples there is collected in relation to the expected collection amount, and look at how the samples are distributed in the measurement period. It is also seen that the "activity" is a good quality metric. This metric describes the amount of changes in the signal, over time. A good data set must contain both areas with high *activity* and areas without *activity*.

2.2 Quality In SmartHG Citizen Data

As a part of the SmartHG project 25 households have been equipped with meters on selected appliances and the main meter. The data collected from this experiment is prone with errors due to malfunctioning test equipment or unexpected interference from the resident which have resulted in offline measurement equipment for periods of time. Examples on unexpected interference could be if the resident is unplugging the measurement equipment, or turning OFF the power socket that supply's it. Unstable network does further degrade the signal, since the measurement equipment uses a lossy network.

2.2.1 Completeness And Activity

The SmartHG data is intended for appliance recognition, and the quality must be assessed with this in mind. The completeness and the *activity* in the data is therefore important.

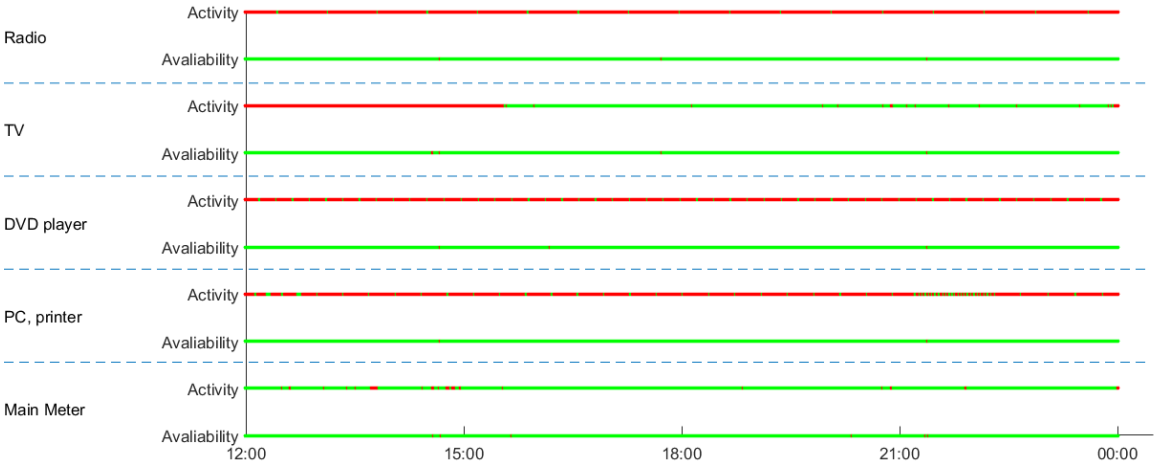


Figure 2.1: 12 hour overview of house 10

In Figure 2.1 is a 12 hour overview of the data in house 10 from the 16/8/2015. Here we see the *sample availability* and *activity* of the five different meters in the house. The *sample availability* is shown as a line where green is indicating that a sample is received as expected, and red shows a missing sample. In the figure it is shown that there is a few samples missing, which is to be expected due to the lossy network architecture. The *activity* is also shown as a line, where green indicates *activity* and red indicates no *activity*. *Activity* is measured as a change in the signal, from prior values. From this we can see that the resident have a lot of *activity* on the TV from around 16:00 to 00:00, which we can presume means that the television is turned ON in this period.

First the *sample availability* quality of the data is assessed. The *sample availability* quality for a specific period of time T_n for a specific meter m, is defined as the amount of samples observed in that timeslot over the expected sample amount. The resolution period T_P for each time period in T is chosen to be one hour.

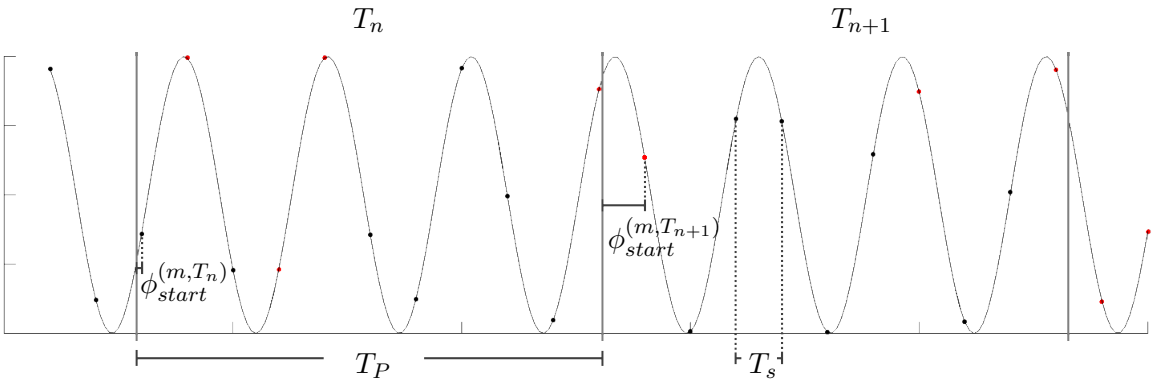


Figure 2.2: Illustration of availability analysis

To calculate the amount of samples expected to be in a specific period of time $N_{max}^{(m,T_n)}$ does the sample phase $\phi_{start}^{(m,T_n)}$ for the given period need to be known. In Figure 2.2 is an illustration of a signal, the black dots are the samples and the red dots are the ones that are missing. The sample phase is the time from the beginning of the period T_n and to the first expected sample. This is needed since for a timeslot T_n of a length of T_P the maximum expected sample amount can vary with 1. This is shown in Figure 2.2 where the period T_n has a potential of having 11 samples, where as period T_{n+1} only can have 10.

$$N_{max}^{(m,T_n)} = \lfloor \frac{(T_P - \phi_{start}^{(m,T_n)})}{T_s^{(m)}} \rfloor + 1 \tag{2.1}$$

$$q^{(m,T_n)} = \frac{N_{observed}^{(m,T_n)}}{N_{max}^{(m,T_n)}} \tag{2.2}$$

As shown in Equation 2.1 is the maximum number of samples for a meter m in the period T_n calculated by taking the period time T_P, corrected with the sample phase $\phi_{start}^{(m,T_n)}$ for the given period, and dividing it with the sample time T_s. The *sample availability* quality of the meter is calculated as the ratio of observed samples in the timeslot T_n to the maximum samples, shown in Equation 2.2.

To find the quality of a house in a given period T_n, that have a set of meters $\mathbf{M}$ with a cardinality of M, we take the mean value of all the meter quality's, as shown in equitation 2.3.

$$\mu_{q(\mathbf{M},T_n)} = \frac{1}{M} \sum_{m \in \mathbf{M}} q^{(m,T_n)} \tag{2.3}$$

A quality vector Q is constructed for each house. The quality vector contains the house quality

found with a period T_P on one hour. This have been done from March T_1 to October T_N.

$$Q^{(\mathbf{M})} = \{\mu_{q^{(M,T)}} | T \in \{T_1, T_2, ..., T_n, ..., T_N\}\} \tag{2.4}$$

This is shown in Equation 2.4 where $\mathbf{M}$ is a set of meters in a given house. This can be graphically shown in Figure 2.3 where all the houses Q vectors is shown. The color is a gradient running from light green for the best quality to red for bad quality.

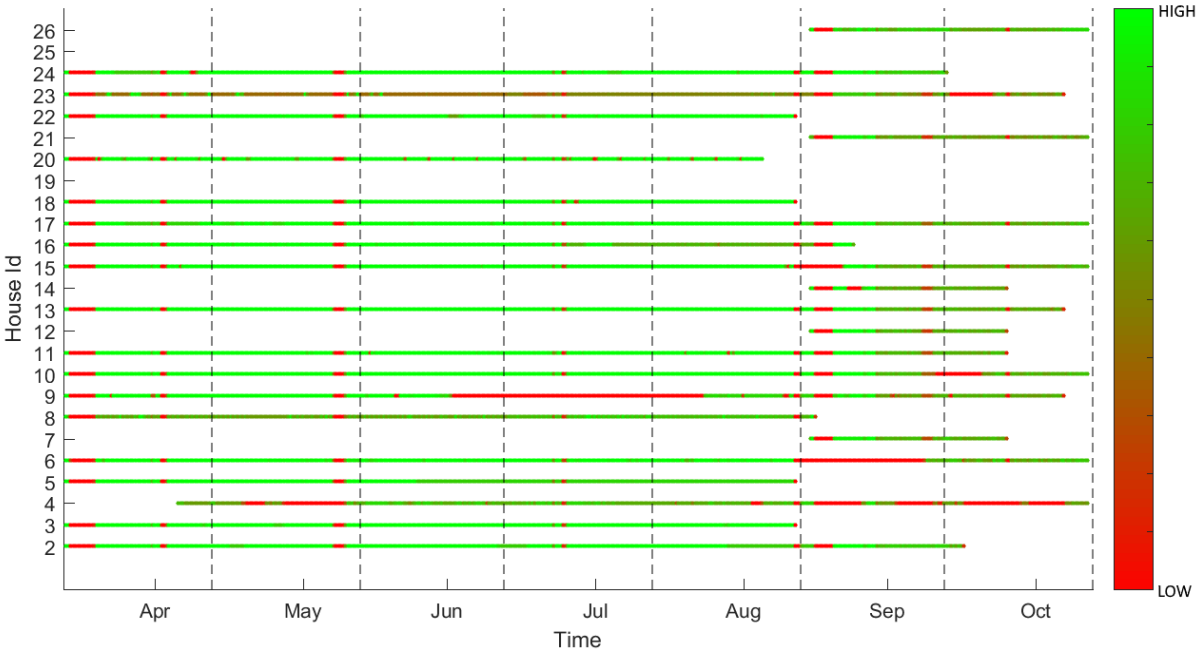


Figure 2.3: Quality of houses in SmartHG project

In Figure 2.3 is the *sample availability* quality shown for the 25 houses in the SmartHG project. When seen on this scale with an analysis resolution on one hour it is hard to see degradation coursed by single sample missing here and there, but meters that have malfunctioned over a longer times shows itself. Since the quality of a house is the mean of several meter quality's, will this most likely appear as darker green spots, since not all meters are malfunctioning at the same time. But there still are some red areas indicating that all the meters in a house is not working. This is probably due to network problems, preventing the meters from sending the information.

There are the red spots that goes through all houses on the exact same time. This indicates that the server receiving the data for all the houses have been down, since it is unlikely that all meters in every house is down at the same time. The conclusion being that red dots most commonly are coursed by the network being unavailable so the client can not sent to the server, or the server is unable to receive.

It is assumed that the first sample received from a meter happen at the time of meter installation, and the last sample received is the time of meter removal. The meter is assumed to be operating

in between these two points in time. In the figure does the coloured Q vector starts at installation time, and ends at removal time. This illustrates how some houses have been operational longer than others. A plot showing how many hours a certain percentage of meters where operating normally can be found in Appendix A. This illustrates that approximately 60% of the time where all meters operating, without any malfunctions.

Since the data is intended for appliance recognition it is of interest where in the data there is *activity*, and where there is nothing happening. Both areas are impotent for a NILM application in training scenarios. We define *activity* as area in the data where there is change as described in Equation 2.5.

$$f(x) + \epsilon < f(x+1) \vee f(x) - \epsilon > f(x+1) \tag{2.5}$$

Where ϵ describes a threshold to filter out changes caused by noise. This can also be described as the standard divination over an area is grater than the threshold. The *activity* is analysed in the available data, and is shown in Figure 2.4 where green is high *activity* and red is non *activity*.

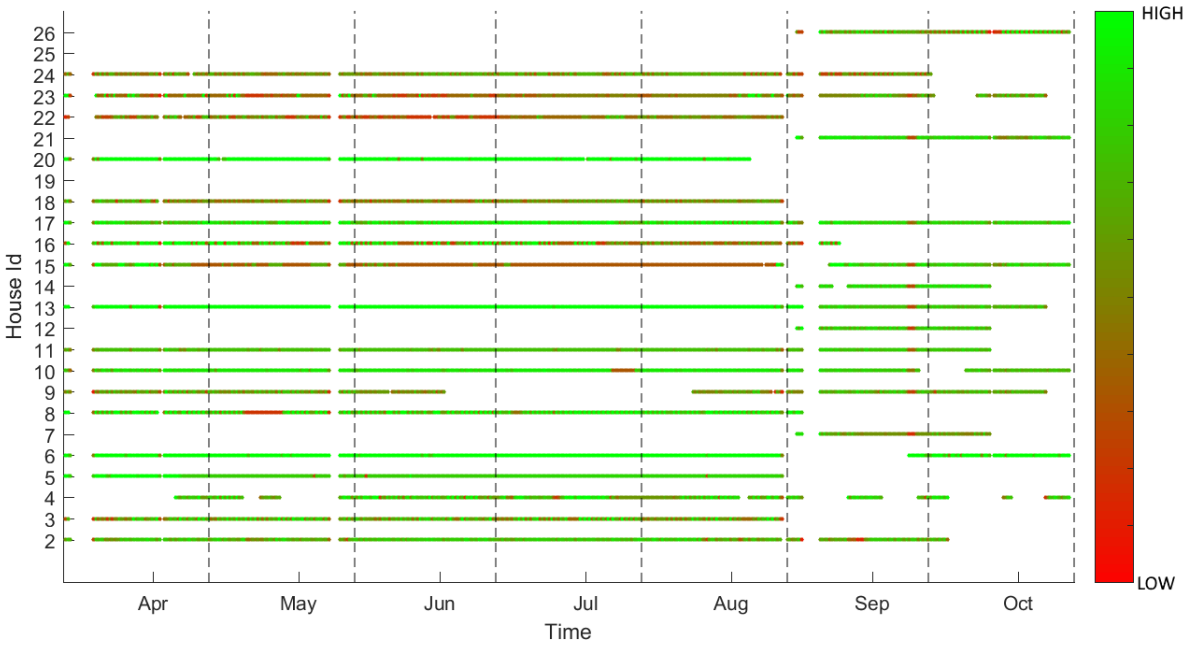


Figure 2.4: Activity of houses in SmartHG project

The *activity* shown in Figure 2.4 is the average of the *activity* of the meters in the house. There is almost never a meter that does not have at least a little *activity* in a house, so a complete red area is fairly rare. The most interesting areas in the *activity* map, is often the places where there is a lot of change in the amount of *activity* like for house 16. The activity plot in Figure 2.4 and Figure 2.3 is a compact overview of the period. For a more detailed overview of the *activity* and *sample availability* see Appendix B.

2.2.2 Main Meter In Relation To Sub-meter Consumption

As disused in the NILMTK project [13] is the relation between the energy consumption measured by the main meter and the consumption measured by the sub-meters an important quality metric. This can give two important indications about the dataset. The first indicator is how much knowledge about a house we have. A house where only a small fraction of the energy consumption is accounted for can give problems when applying NILM algorithms. This is due to a lot of unknown consumption from other appliances. This is further discussed in Section 5.3. The other is more appliance specific. If an appliance is not responsible for a significant part of the total consumption odds are that it is almost never ON, or that it usages so little energy that it is hard to detect. Both can have a significant impact on NILM algorithms. This will be further discussed in Chapter 4 and Chapter 5.

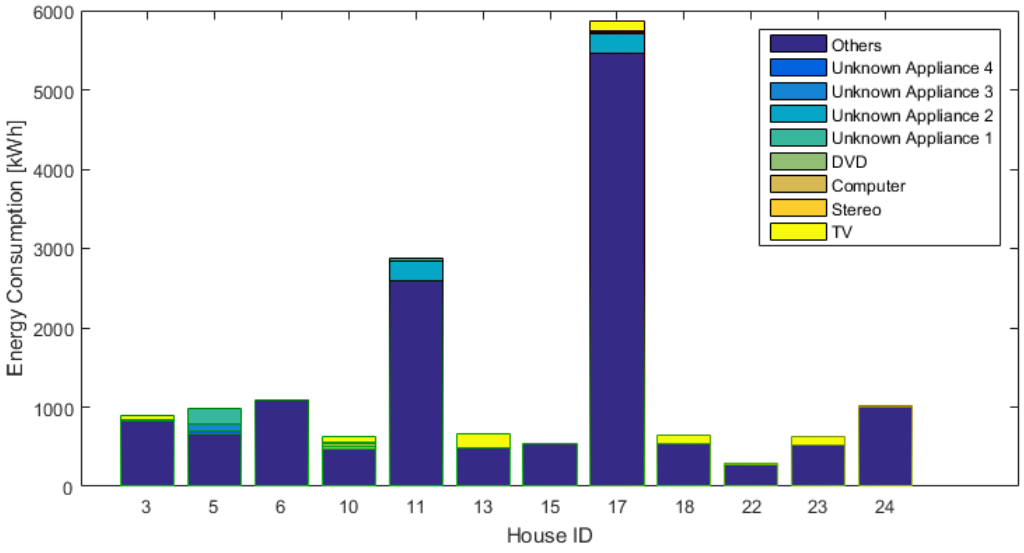


Figure 2.5: Usage of SmartHG houses

In Figure 2.5 is the total consumption of the houses shown, and split up in the the different sub-meters, and an "other category". The *other category* is the energy the main meter has used besides the energy accounted for by the sub-meters. In the figure is only shown the houses which main meter had observed an energy equal or greater than the sum of all the sub meters. This is not the case for houses that have malfunctioning main meters for a longer period of time, or for houses that have generated a lot of energy form solar cells.

As seen in Figure 2.5 is the appliances equipped with sub-meters only responsible for a minor part of the total house consumption. Only the TV's and some of the unknown devices, have a relative high energy usage. The unknown devices is devices that have a sub-meter, but the appliance on the sub-meter is unknown.

2.3 Chapter Summary

For NILM applications is the *activity* in the data, and *sample availability* in the data good indications of quality. Further more is the amount of energy observed by the main meter in contrast to the energy observed by the sub-meters in a house also a good indication of the dataset quality.

In smart-grid based on a classic UDP network architecture, can a packet loss be expected. The packet loss due to the network is shown to be relatively small. It is shown that the quality also can indicate problem in the data gathering phase of a project. problems such as malfunctioning equipment, network failure or server problems can be detected by quality monitoring systems.

When working and gathering big amounts of data it is important to be able to gesture about the data prior to analysing it. To look at different quality measurements researchers are able to get a good general overview of the data, and find better areas to focus on.

Gap Reconstruction 3

One of the more common problems in citizen science projects are gaps in data. This can happen either if the network connection is unstable or the test equipment gets prematurely turn OFF, as discussed in Chapter 2. This can greatly degrade the data quality, and lead to errors in the application. One way of dealing with this problem is to use mathematical gap filling techniques to come with a qualified guess on how the data would look like in the gap.

In order to use this methods we must assume that the missing data in the gap follows the same behaviour as the data on each side off the gap. If the signal is so stochastic that this is not the case is gap filling not recommended [14].

In the case of the SmartHG project the data can be seen as having a part that is depended on the previous and future data, plus a stochastic part. This part is determined by the user and the appliance. Due to the stochastic part is a perfect reconstruction not possible. It is the hypotheses that the non-stochastic part is still so dominant that a decent reconstruction is possible.

3.1 Gap Filling Methods

Various methods exists for gap filling. Five popular algorithms are selected for this project, and is validated on the SmartHG project data. These methods have been chosen since they all have a different approach on the gap filling process. For some of the algorithms is it important to keep the frequency spectra as intact as possible, for other is it the jitter power or the exact sample value they are trying to estimate. Some algorithms have also been designed for small gaps, while others have been designed for larger.

In the following sections will what makes the five algorithms unique, and how they work, be briefly described.

3.1.1 Papoulis-Gerchberg Algorithm

The Papoulis-Gerchberg (P-G) algorithm is a multi gap filling algorithm, meaning it is capable of correcting more than one gap at the time. This makes the algorithm preform good in conditions with many gaps and few available data points between gaps. This is due to its ability to collect information about the signal across multiple gaps [15]. The P-G algorithm works under the assumption that the signal is a periodic stationary signal with a known bandwidth. The signal will therefore consist of M frequency components, and everything outside the band is assumed to be noise. The signals in the SmartHG are not stationary, but for small snippets can approximately stationariness be assumed.

The true bandwidth is also unknown in the signal. The P-G algorithm is very depended on the

bandwidth for a correct reconstruction. A modified version of the algorithm that estimates the bandwidth, by varying the frequency components M and analysing the mean square error on the known signal is therefore used [16]. This approach is fairly good at estimating the true value of M, but is time-consuming. The implementation of the algorithm developed for this master's thesis can be found in Appendix C.

3.1.2 Wiener Filling Algorithm

The Wiener filling algorithm is an extension of a Wiener predictor. The Wiener predictor assumes that there exist a linear relationship between the next sample and the previous samples. By trying to predict the missing samples from both sides of the gap, and combining the results, it estimates the missing samples [17]. For larger gaps does this methods rely on earlier predictions to close the gaps. This result in errors being accumulated over the gaps. The method is fast, and is therefore suited for large data with small gaps. The implementation of the algorithm developed for this master's thesis can be found in Appendix D.

3.1.3 Spatio-Temporal Filling Algorithm

The Spatio-Temporal filling (SSA) algorithm uses singular spectrum analysis to split the signal into a series of sub-signals. The sum of the sub-signals is the original signal, and the sub-signals are ordered so the most dominant is first, and the least dominant is last.

The reconstruction philosophy is that the gap has introduced noise in the signal, but a sum of only the most dominant sub-signals must be close to the original signal without noise. But in order to know how many sub-signals to include in this sum, we introduce an other artificial gap. While the sub-signals are being accumulated the mean square error of the artificial gap is observed. When this mean square error hits its minimum peek, it is assumed that the reconstruction is as good as possible [18].

This method is very popular for gap filling. It has shown to be very noise resistant since it finds the overall trends in the data. It does require quite a lot of data to be known post and prior to the gap since an artificial gap must be introduced. It is based on singular spectrum analysis which assumes that the signal consist of stationary processes. This is a similar constraint to the P-G Algorithm in Section 3.1.1. The implementation of the algorithm developed for this master's thesis can be found in Appendix E.

3.1.4 Envelope Filling Algorithm

Unlike the previous described methods does the Envelope filling algorithm not depend on frequency analyses, but rather on the expected power of the signal. Looking at the envelope of the signal it assumes that all local maxima and minima must lie on the upper and lower envelope. It then looks at the data prior and post of the gap and try to estimate the number of local maxima and minima in the gap, and their locations. It does this by looking for patterns in the time series data [10]. When the new maxima and minima are found the points is connected by using spline [19].

The methods do not make any assumptions about the signals stationariness or bandwidth. The method can also be used on none equally spaced time series. The implementation of the algorithm developed for this master's thesis can be found in Appendix F.

3.1.5 Empirical Mode Decomposition Filling Algorithm

The empirical mode decomposition filling (EMD) algorithm uses empirical mode decomposition, to break the signal into intrinsic mode functions (IMF). The sum of all IMF's is the original signal. The IMF's is all more low frequent and simpler in structure than the original signal. The hypothesis is that it is easier fixing a gap in a simple signal than a complex one.

The envelope filling algorithm in Section 3.1.4 is used to fix the gaps in the IMF's. The IMF's can now be accumulated to get the original fixed signal. Like the envelope filling algorithm does it not make any assumptions about the signals stationariness, bandwidth and can be used on none equally spaced time series. But making an empirical mode decomposition on a signal with a gap in is a non-trivial process and can introduce errors [19]. The implementation of the algorithm developed for this master's thesis can be found in Appendix G.

3.2 Gaps In SmartHG Dataset

The gaps in the SmartHG project dataset is caused by a lot of different sources such as bad network connection, unplugged measurement equipment or server breakdown. This makes the type of gaps different from case to case. Three aspects of a gap is important for the gap filling: The size of the gap, the data known before the gap, and the data known after the gap.

3.2.1 Gap Size

In Figure 3.1 is the quantity of different gaps shown. The different gap size is measured as samples missing, e.g. a gap size on 3 means that there is 3 samples missing in a row, between two received samples. Looking at the different gaps in the dataset we see that the normal gap is relatively small. Most of the gaps are between 1-5 samples as seen in Figure 3.1.

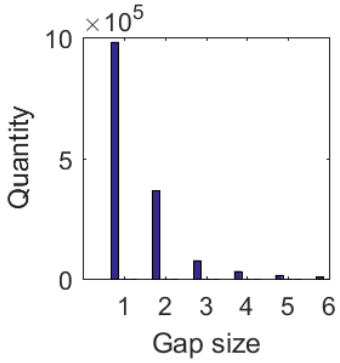


Figure 3.1: Gap quantity

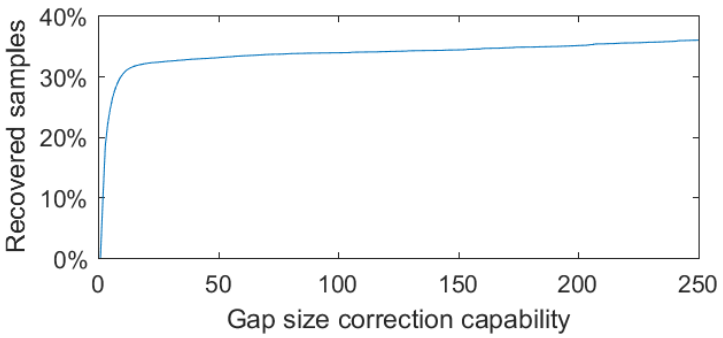


Figure 3.2: Error recovery capability

The aim of data reconstruction is to recover the missing samples. The "gap size correction capability" is a metric telling how big gaps it is possible to correct. If an application has a *gap size correction capability* on 5 it means that is is capable of correcting gaps that have the gap size of 5 samples or smaller.

In Figure 3.2 is shown how much of the missing signal that can be reconstructed with different *gap size correction capability*. It is shown that if a *gap size correction capability* of approximately

20 samples can be achieved, it is possible to recover 30% of the missing data in the SmartHG dataset. Since the signal is partly stochastic, and it is not possible to recover the stochastic part, the complete signal can newer be reconstructed. The greater the gap, the greater influence does the stochastic part have on the signal. Smaller gaps can therefore be fixed with greater success. It is therefore unlikely that recovery of more than 30% will be possible.

3.2.2 Post And Prior Knowledge

The reconstruction process works by looking at the samples available prior and post of the gap. Common for all reconstruction methods is that they assume that the signal in the gap must behaved in relation to the samples prior and post for the gap. Therefore is the samples prior and post for the gap called "knowledge", since it grants the knowledge used for reconstructing the gap.

But in a signal with lots of gaps it can be interesting to see how much *knowledge* is available for reconstructing a gap. This is done by seeing how many samples that are available prior and post to the gap. This is important since much data allows for detailed models, that can improve gap reconstruction and little *knowledge* gives the stochastic part dominance which will lead to error prone reconstruction.

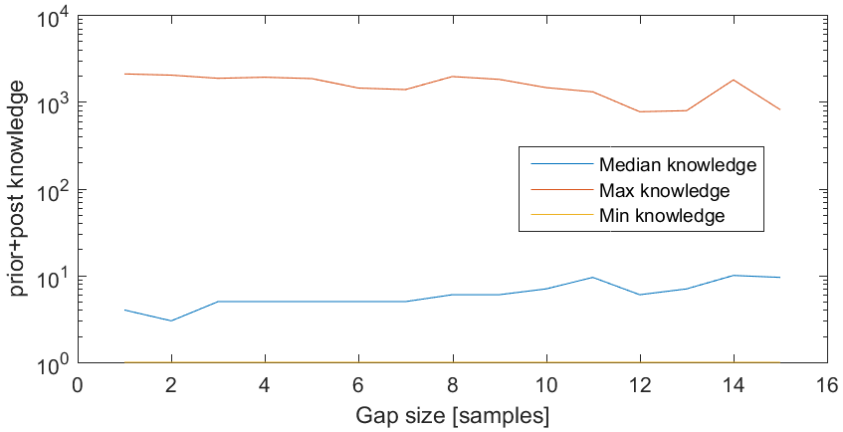


Figure 3.3: Available samples for recovery

In the case of the SmartHG project dataset the data available prior and post to a gap varies greatly but is around the same max and min values for every gap size. The median samples available to fix a gap is around 6 samples as shown in Figure 3.3. In the figure is shown that no matter the gap size is the expected *knowledge* about the same. This can be problematic since lager gaps offend needs more *knowledge* than smaller gaps. This further indicates that complete recovery of all gaps is impossible.

3.3 SmartHG Dataset Reconstruction

The data reconstruction algorithms mentioned in Section 3.1 have been tied on the SmartHG dataset. First have 6 unique error free areas in the data been found and an artificial gap have been introduced in the 6 scenarios. The gaps have now been reconstructed and a comparison to the true value is made.

The 6 scenarios have been randomly chosen under the constraints that there where no missing data in them and they all are different in activity level from the other chosen scenarios. Both accumulative scenarios and non-accumulative scenarios have been chosen.

There are several ways of comparing the different algorithms. In this report three have been selected based on the likelihood of the importance in a learning algorithm, which is the target application for the data. The first is a simple sample by sample comparison, where it is seen how much each sample differs from the true value. The other is the frequency, where it is analysed how much the frequency response is different. The last is a method called jitter compression. In this method is the power of the jitter compared. More on this in Section 3.3.3.

3.3.1 Sample Comparison

The sample by sample comparison is created by finding the mean square error of the samples compared to there true values.

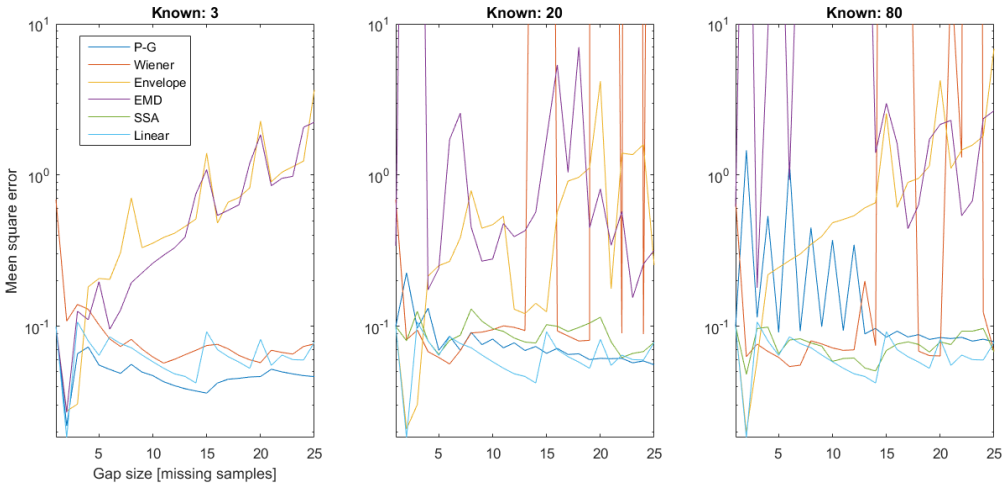


Figure 3.4: Sample comparison of the reconstruction methods

In Figure 3.4 is the result shown for 3 different cases. The Known attribute indicates how much prior and post *knowledge* in samples is used to reconstruct the gap. On the x axis is shown the gap size. As expected does the prediction grow worse the greater the gap size. As a baseline is a linear predictor also added. The linear predictor makes a simple line between the two samples at each edge of the gap, and places all missing samples on the line.

As seen in the figure the best reconstruction made by the P-G algorithm, with relatively few samples as *knowledge*. The Wiener algorithm is also close to the linear base line. The logical statement is the more *knowledge* that are available the better should the prediction become.

This does not seems to be the case, since a lot of the algorithms amperes to preform better at lower *knowledge*. This can be explained by the stochastic part of a signal. When the *knowledge* is small it is more likely that the information around the gap is relevant, the more *knowledge*

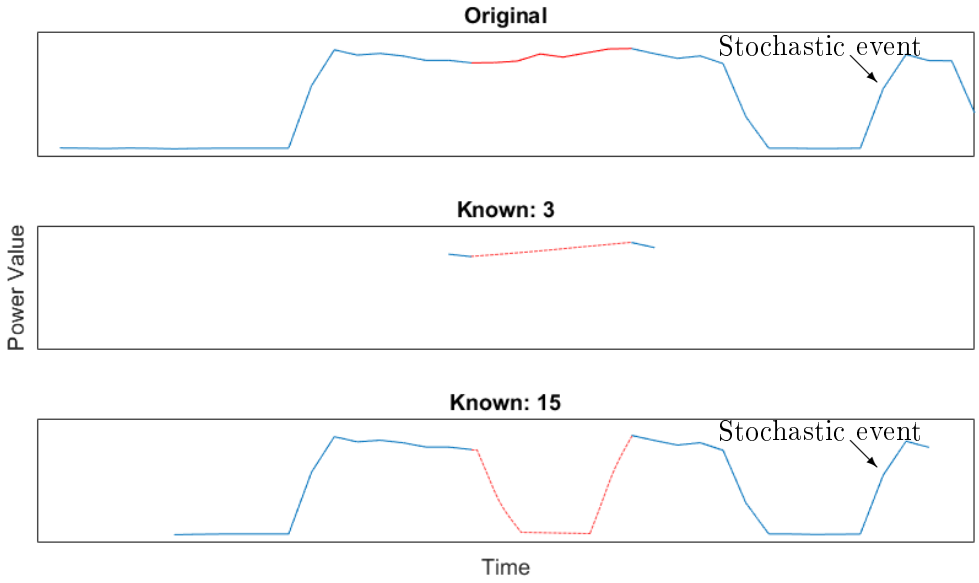


Figure 3.5: Stochastic effect on prediction.

In the Figure 3.5 is shown an original signal with a gap, indicated by the red area. The true value of the gap is illustrated by the full red line in the original signal. The two subsequent graphs is the reconstruction, at different *knowledge*. The blue area is the known area and the dotted red line is the predicted values. When the known area is small there is not enough frequency information to make any grant changes, so the prediction will look more or less like a linear interpolation. This is compared to the true value not a bad guess. When there is more samples known it is possible to make more complex estimates of the signal. In the figure is shown that a *knowledge* of 15 will include the stochastic event in the model. Taking this in to the model, changes the prediction to a less correct value, due to the frequency's added by the stochastic event.

Since the results shown in Figure 3.4 at higher *knowledge* shows that the reconstruction methods for the most part preforms worse than the linear base case, it can be concluded that the signal is quite stochastic, and not as periodic as one would hope. For many smart-meters it is possible to get both the instantaneous current power and accumulated power over time, as the samples. By looking at the accumulated power before and after a gap it is possible to gesture about the changes in power during the gap. This information can be used to further improve the gap filling.

3.3.2 Frequency Comparison

An other metric to compare is how well the frequency response is preserved in the reconstruction, since this is often more important that the true value itself for smaller devices with many states, like computers. Such devices is often recognised by there usage pattern and not the true power draw. By looking at the Fourier transform of the original signal and the reconstructed power was the frequency components compared.

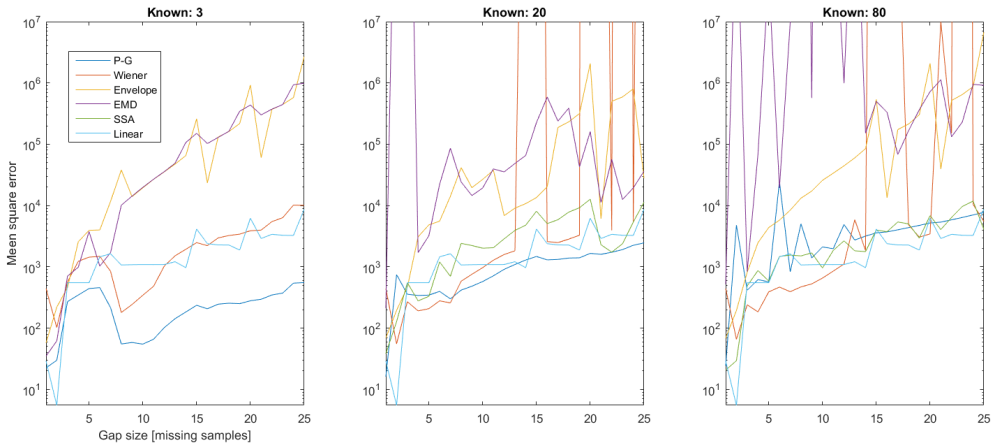


Figure 3.6: Frequency comparison of the reconstruction methods

As shown in Figure 3.6 both the Wiener and the P-G algorithm seems to preform well. Like for the sample comparison the most successful reconstruction seems to be at low *knowledge*.

3.3.3 Jitter Comparison

The jitter is a different metric as it tells how well we have modelled the noise of the signal. In a jitter analysis the signal is thought of as a slowly changing signal with a faster noise signal embedded in it.

The slow signal is assumed to be found by taking a low pass filter over the signal to filter out the noise. The jitter analysis measures the power in the jitter from this signal as illustrated in Figure 3.7.

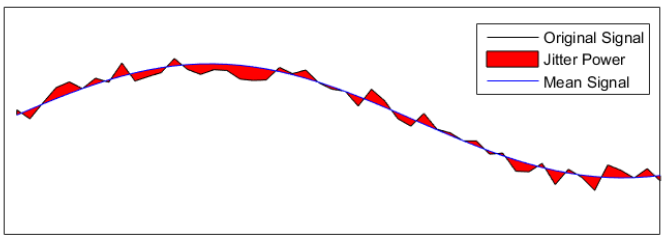


Figure 3.7: Jitter Power Illustration

It the estimated jitter power have been compared with the real jitter power for all the reconstruction methods.

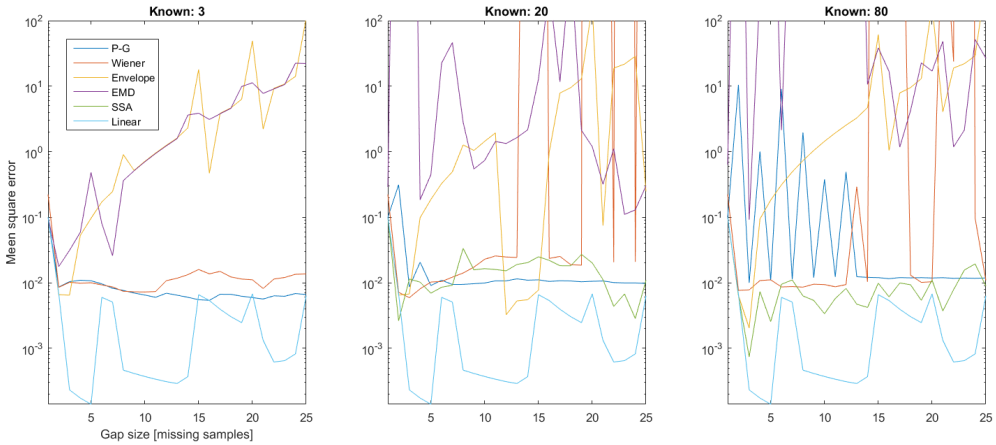


Figure 3.8: Jitter Power comparison of the reconstruction methods

As seen in Figure 3.8 does non of the methods model the noise very well. This is to be expected due to the nature of the selected methods. This is will probably not be a problem for the SmartHG data, since it is sampled at a very low sample rate, and noise therefore is not likely to be used as a parameter. For data there have been sampled with high frequency it have shown that the noise created by the different appliances can be used to detect them [20].

3.4 Chapter Summary

In the chapter the errors of the smartHG dataset is analysed. The dataset is a raw dataset, that have not undergone any cleaning prior to the process. It is shown that the most common errors in this types of data is single sample errors, and minor gaps of up to 5 missing samples.

different kind of gap filling techniques is used on a series of artificial introduced gaps. It is shown that the small gaps, that are the most common, can be fixed relativity easy. For larger gaps it is harder do to, due to the stochastic nature of the signal. The stochastic element of an electric signal is often greater than the periodic part, which is a challenge for the gap filling. This combined with the low sample rate makes it hard to gesture about the true frequency content of the signal, since it is sampled way below the Nyquist rate.

Appliance Recognition 4

Appliance recognition and load disaggregation are some of the key aspects in non-intrusive load monitoring (NILM). The main purpose of the NILM approach is to better understand the power usage in the home. This information is used by the *operator* to better plan the energy production and distribution. It also have the potential to help the different households to more optimal power savings. This is done by informing the residents about the power consumption of individual appliances. It is shown that informing a household about its usages pattern can lead to significant savings [5]. This makes the basis for load disaggregation, where appliance specific load is guessed based on the main meter load. This enables the user to know what uses the energy and when.

One other aspect of NILM is event detection. Sometimes the actual consumption of an appliance is not as important as when it was used, and for how long [21]. This is a simpler task than guessing the true consumption, and for some applications is a better approach. As an example is this kind of information sufficient if we want to track the activity of an elderly person in an assisted living situation [22].

4.1 NILM Concepts And Challenges

The aim of NILM is to partition the whole house consumption data in to appliance specific consumption.

$$P(t) = p_1(t) + p_2(t) + ... + p_n(t) \tag{4.1}$$

This can be seen as in Equation 4.1 where $P(t)$ is the total consumption at time t, and $p_n(t)$ is the consumption of appliance n at time t. The aim is to partition $P(t)$ back to the different appliance specific consumptions. In order to structure the problem we categorise the appliances in 4 groups [20]:

Type-I: Appliances that only have 2 states corresponding to ON/OFF. This could be a lamp or a water boiler.

Type-II: These are appliances with multiple states. The appliance in this category can be modelled as a finite state machine. Many modern devices such as TV's, computers and washing machines fall in this category.

Type-III: These appliances are referred to as "Continuously Variable Devices". These devices have a variable power draw and is impossible to model as a finite state machine. This could be appliances like power drills, and dimmer lights. These are by far the hardest for the NILM algorithms to detect.

Type-IV: These are a special kind of appliances that are always ON and consume energy at a constant rate. Such devices could be smoke detectors and TV receivers.

Some of the different appliance types is illustrated in Figure 4.1.

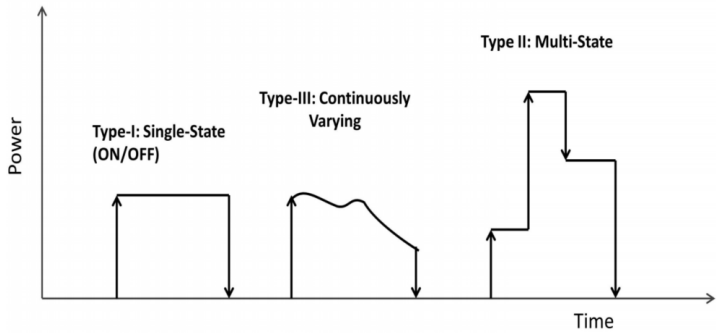


Figure 4.1: Appliance types. Source [20].

The appliances in Type-IV is not a particular researched area, since it does not give the user much information to calculate savings from. These appliances are often using very little energy, and must not be turned OFF at will.

Type-I and Type-II are very used, and can cover almost any appliance. In the area of event detection is the most common approach to see all appliances as Type-I and detect ON/OFF events. Most appliances today is Type-II due to the growing amount of complicated electronics that are embedded in them. Then there are devices that are Type-II but most of the time act like Type-I appliances. An example of this is the vacuum cleaner. On most vacuum cleaners you are able to change the suction intensity, which makes it Type-II. But most people do not change this very often, so the energy usage looks more like a Type-I appliance.

4.1.1 NILM Features

For data disaggregation it is common to use method based on machine learning and optimization. The approach is to first extract features from the dataset, and then train or validate advanced statistical models by using these features.

In NILM the features can be sub categorised in steady state, transient state and non-traditional features. These categories can be further expanded as shown in Figure 4.2.

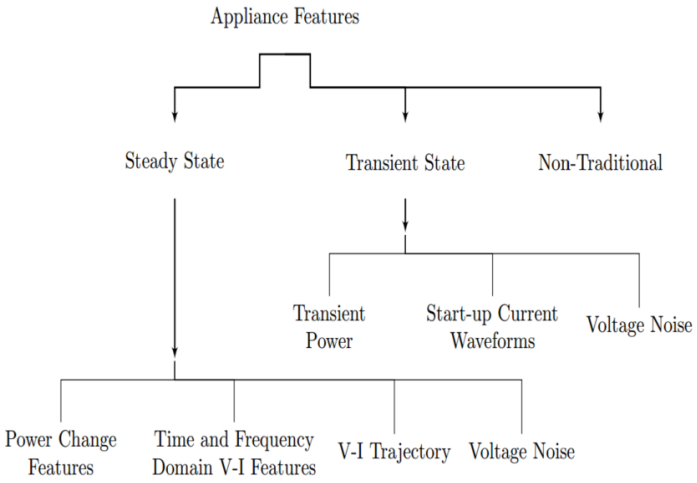


Figure 4.2: Feature types. Source [20]

Steady state features is features that can be extracted when the signal is in the stable state. Features that are often extracted in this state is "Power Change" which is the jump in power or reactive power usage. Time and frequency analysis is often used on the voltage and power signals of the meters. It is shown that analysing the harmonics in signals is a powerful way of making appliance detection [23].The trajectory between voltage and power usage is a good feature. This can be used to detect the reactive power of a signal, which is a give-away for inductive loads. Some researchers have shown that different appliances makes different noise profiles on the main line. Unfortunately does this kind of recognition require an extremely high sampling rate in the kHz or MHz area [24].

The transient state of an appliance is a short state that comes when an appliance switches between steady states. In this state there often is appliance semi-unique waveforms in the current and voltage domain. The noise generated on the mainline in this state is also a good indication of the appliances type.

There also exists a series of non-traditional features, that is used in special types of algorithms. One of them is based on matching the power usage profile to geometrical figures, and using the series of figures to recognize an appliance [20].

Many of the above features requires a high sample rate in kHz or MHz span in order to be effective. This kind of resolution is often not available since it is a normal smart-meters that are providing the data. For a smart-meter sample rate at 1 Hz would be considered fast, and for many applications this is down to 0.03 Hz or slower. It is therefore common only to use the steady state features, and most of the time only the power change feature, which looks at changes in power, reactive power, current or voltage.

4.1.2 Learning Strategy

When you have extracted features from the signal there are several algorithms that can be selected for the training process. They can roughly be split up in two categories optimization algorithms and pattern recognition algorithms. The optimization algorithms relies on a database

of appliance info, and tries to find the subset of the database that are most likely to be in use.

$$M = \underset{x \in \mathbb{P}(D)}{\operatorname{argmin}} \left(\mid \sum_{i=0}^{I} x_i - \hat{y} \mid \right) \tag{4.2}$$

This is illustrated in Equation 4.2 were a set of appliances M that correspond to the measured signal $\hat{y}$ needed to be found. This is achieved by searching the known database of all appliances D. This is done by taking the powerset of the database and finding the subset that minimises the error. It can therefore be seen as a minimization problem, where the desired set M is the arguments to the minimum solution. This approach is fine for a small amounts of appliances, but for large databases and many appliances in use at the same time this would take far too long to be practical.

Due to this, most researchers use pattern recognition techniques instead. These have the benefits of being more scalable, and be usable even if only parts of the device database is known. The basis concept is to create a statistical model that can be used to classify the appliances. There are generality two approaches to pattern recognition: supervised and non-supervised. In supervised training it is assumed that the ground truth of your system is known. For a NILM application this means that the true consumption of the appliances needs to be available for the training process. This means that sub-meters must be placed to measure the appliances individually, which often is costly and time consuming. In non-supervised methods only the main meter readings are available, and no information about the appliances are known. This type of algorithms will usually try to cluster the readings in distinct groups cosponsoring to a guess of what appliances that correspond to the reading. Fore this kind of training it is easy to collect a big training set, but it requires the groups to be manually labelled afterwards.

Many algorithms are based on clustering techniques and hidden Markov models (HMM) [25], [26], [21], [20], [27]. HMM have proven very effective when classifying Type-I and Type-II appliances. This is due to its ability to model state changes as an imported part of the classification. It is shown that using clustering techniques can greatly improve the classification, and reduce the training time. Artificial neural networks are also showing great promise, and is a topic that is currently in focus by many researchers. More on this can be found in Section 4.2.

4.1.3 NILM Challenges

There are many challenges in NILM. One is the Type-III appliances. Due to their varying, and at times random, energy consumption it is hard to fit them to a behavioural model. This is made further harder by the low sample rate. Many great features of the appliances is hidden due to the low sample rate most data is collected with[20].

A other problem is the "top consumers problem". In a household there are some appliances that require a lot of energy when they are on, such as stoves, air conditioners and refrigerators. These are often refereed to as the top consumers. Then there is the appliances that consumes very little energy like laptops, DVD players and routers. The *top consumers problem* is that

the top consumers is so dominant that they make it hard to track the power signature of little consumers. This is why many researchers only focus on the top 10 consumers when during NILM applications [13].

4.2 Related Work

There are several interesting problems and approaches being researched in the NILM community in the moment. One is the lag of a good bases of comparison, both to compare the performance and the training time of a solution. It was a common practise that each researcher uses his own dataset, or an artificial lab setting to validate the results. This made it hard to compare the algorithm to others, and raised questions about the correctness of the findings.
To accommodate this problem the NILM-Toolkit (NILMTK) was created. This is a collection of python libraries, that create a framework to evaluate NILM solutions. It supports a wide range of known datasets that can be used to train or validate an algorithm. It also comes with two benchmark algorithms to be used to compare a new algorithm against. The NILMTK will help researchers to easily share and compare algorithms in the future[13].

Another framework to accomplish better comparison and sharing is the NILM-eval framework. This is based on Matlab and supports algorithms to be written in either Matlab or in python. The NILM-eval and the NILMTK is created to solve the same problem, and is compatible with each other in the sense that the interface for algorithms are the same[26].

Deep neural networks have last years improved the area of computer vision and speech recognition remarkably. NILM researchers are starting to look into if similar techniques can be used to improve data disaggregation. Neural networks have like HMM the ability to model state changes. Furthermore they are capable of automatically finding important features in the data. Preliminary tests shows that a deep neural networks can outperform the classic HMM approaches. The downside with the deep neural network is that it takes weeks to train, and there are thousands of parameters there can be changed. This makes the task of finding the global minimum hard, and the method is very prone to be stuck in a local minimum. The method is currently being used on top consumers[28]. Current research is looking that its potential in solving the *top consumers problem*, but no definite conclusions is yet to be reached.

Many approaches is supervised, since this kind of approach is better at separating the appliances. But this require that you first build a database with information about all the appliances. To have this prior knowledge of the system seems unrealistic in an application setting, where it is most likely you do not know anything about the house. To combat this, various unsupervised methods have been developed. The focus in this kind of studies have been load separation for the sake of making power savings, or to inform the electric companies to help them create a more stable electric grid. There are several good algorithms for this today that support this purpose. For event detection and user habits analysis it is still most common to use the supervised approaches [29]. This report will therefore focus on the supervised approaches.

4.3 Recognition Methods

Methods based on HMM are currently the most popular method for NILM. Three method is selected to be validated in this paper. The methods are selected since they are often referenced in the literature, and is often used as benchmark algorithms. All of the algorithms are based on HMM or clustering in different configurations. The following subsections will briefly introduce the algorithms.

4.3.1 Factorial Hidden Markov Models

The HMM has proven to be one of the most used tools to create probabilistic models based on time series data. This is done by seeing the system as a series of observable states Y_t, and a series of hidden states S_t [30]. It is assumed that there exists a probabilistic relation between the observations Y_t and a sequence of hidden states S_t. This is implemented as a first order model, which implies that the current hidden state S_t is only depended on the previously state S_{t-1} and the current observed state Y_t.

$$P(\{S_t, Y_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{4.3}$$

The joint probability for a specific hidden state S_t that has caused the observed output Y_t can be seen in Equation 4.3. $P(Y_t|S_t)$ is the emission probability, which shows the conditional probability for Y_t being outputted by a given hidden state S_t. The probability for a state change is given by the transition property $P(S_t|S_{t-1})$.

Factorial hidden Markov models (FHMM) is an extension of this classic HMM where there exists several hidden state chains that all contribute to the observed output. This can be seen as an extension to Equation 4.3 where the hidden states S_t can be seen as a set of states as shown on Equation 4.4 where M is the number of chains.

$$S_t = \{S_t^{(1)}, ..., S_t^{(m)}, ..., S_t^{(M)}\} \tag{4.4}$$

This can also be shown graphically in Figure 4.3. It is illustrated how several hidden states from different chains contribute to the output.

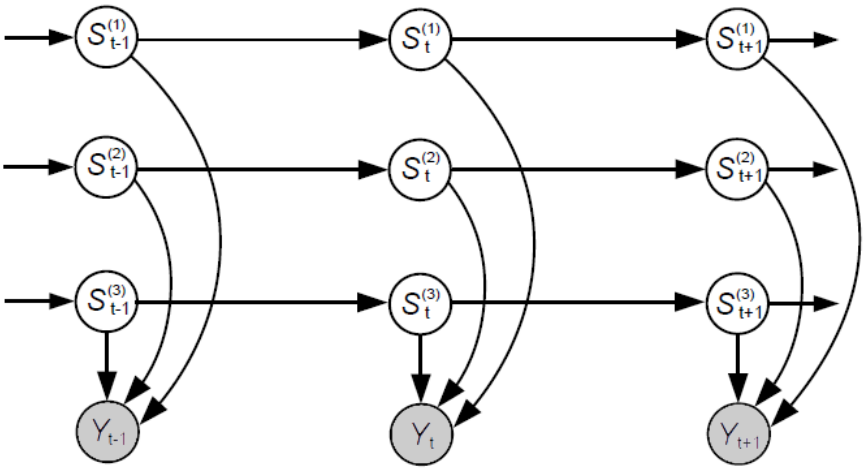


Figure 4.3: Factorial hidden Markov model illustration. Source [30]

As an extra constrain can each hidden state, in each chain, only depended on the previously state in the chain, and not on the hidden state of other chains. This makes the state transition probability to be defined as in Equation 4.5.

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P\left(S_t^{(m)}|S_{t-1}^{(m)}\right) \tag{4.5}$$

The observed stated is therefore created by the contribution of many different hidden states.

In this implementation there is normally used only two or three states, since most appliances are seen as Type-I or Type-II appliances. One chain for each of the appliances plus one that is called the other chain is created in the model. The other chain is to allow for the other appliances that are not taken in to the model and noise. The model can therefore be used to validate for a given output what is the probability that a given appliance is contributing to the output. The concrete implementation for this algorithm has been borrowed from the NILMTK [13].

4.3.2 Parson

The Parson algorithem is a variant of the Kolter and Jaakkola algorithm[31]. The algorithm is designed to be a semi-supervised learning algorithms. This means that the algorithm does not need to train on the data from the concrete house prior to deployment, but it still requires some prior knowledge. The idea is to have a database of general appliance models, that tells how specific appliances most likely will behave. This model will be able to detect the device, and do some unsupervised special training to better fit the concrete appliance.

The algorithm is based on a kind of HMM called "difference HMM" since they take the step difference in the aggregated data and use as the observed output Y_t. In a normal HMM there is

only emitted one output for each hidden state in a chain. In the Parson algorithm the model is changed to output two states from the hidden states X_t and Y_t. This creates a model structure that can be graphically illustrated as in Figure 4.4 [25].

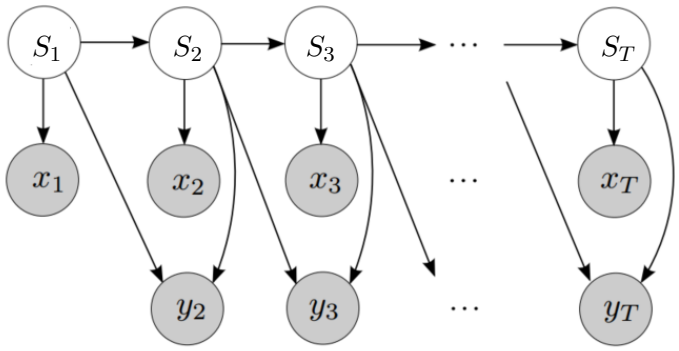


Figure 4.4: Parson hidden Markov model structure. Source [25]

The Y_t is corresponding to the step in power that is observed since the last sample. It is assumed that the only thing that can cause a step in power is if an appliance changes states from one to another. The X_t is the constant power consumption the appliance would have in the state. This is used to filter out the appliance if the power consumption observed is smaller than this threshold.

It is assumed that the power consumption of a state in the general model of an appliance can be modelled as a normal distribution. This can be illustrated as in Equation 4.6.

$$w_t|_{S_t} \rightarrow \mathcal{N}(\mu_{S_t}, \sigma^2_{S_t}) \tag{4.6}$$

This shows how the power consumption of an appliance is modelled as a normal distribution. The step Power Y_t, can also be modelled as a normal distribution that is created from the change between two states as shown in Equation 4.7.0

$$Y_t|_{S_t,S_{t-1}} \rightarrow \mathcal{N}(\mu_{S_t} - \mu_{S_{t-1}}, \sigma^2_{S_t} + \sigma^2_{S_{t-1}}) \tag{4.7}$$

To model the constraint that the meter only can be on if the power consumption present in the signal is higher than the requirement for the state is an additional constraint added to the

emission probability. This constrain is mathematically described in Equation 4.8.

$$P(w_{S_t} \leq x_t | S_t) = \int_{-\infty}^{x_t} \mathcal{N}(\mu_{S_t}, \sigma_{S_t}^2) dw \tag{4.8}$$

Where x_t is the measured power consumption and w_{S_t} is the appliance power consumption constraint. This makes the probability going towards zero if the total power draw is to little and towards 1 if there is more than enough power. This is a loose constraint since too high power draw also could be created by other appliances. To minimise the impact this have for the algorithm is the mean power draw subtracted from the data before searching for other appliances.

In the implementation validated in this report are all appliances modelled as Type-I. The algorithm is allowed to train on the house data, to better fit the general models to the appliances and create the initial database.

4.3.3 Weiss

The algorithm proposed by Weiss called "AppliSense" is a other approach that do not use Markov models[21]. The philosophy is that some appliances like lamps and kettles are purely resistive, where others like washing machines and air conditions are more inductive and some appliances like laptops are more capacitive. By measuring not only the power consumption, but also the amount of reactive power it is revealed if there is inductive or capacitive appliances.

The algorithm is supervised since it requires a signature database to be created prior to usage. The signature database consists of how the changes will be in reactive and real power for a specific device when it is turned ON and OFF. In their paper the authors show a method where this database can be created of a user, by using a smart-phone application and turning ON and OFF appliances in the house, and inputting the information in the mobile application[21]. In this paper the database is created by data collected from the sub-meters measuring only the devices.

The algorithm looks for significant power changes in the data. If a change is found it is assumed to be an ON or OFF event. The change is now compared to all the values in the database. If a close match is found it is assumed to be this device, if not it is a unknown device. This limits the approach to Type-I appliances, and the algorithm requires the reactive power to be measured.

4.4 The ECO Dataset

The experiments in this chapter will be done on the Electricity Consumption and Occupancy (ECO) dataset [26], [32]. The dataset consists of electricity consumption measured from six households that have been equipped with sub-meters on selected meters and the main meter. The data is collected in Switzerland in the period June 2012 to January 2012.

The data is sampled with a resolution on 1 Hz on both the sub-meters and the main meter. The main meter data contains information about real and reactive power, where on the sub-meter

only real power is measured.

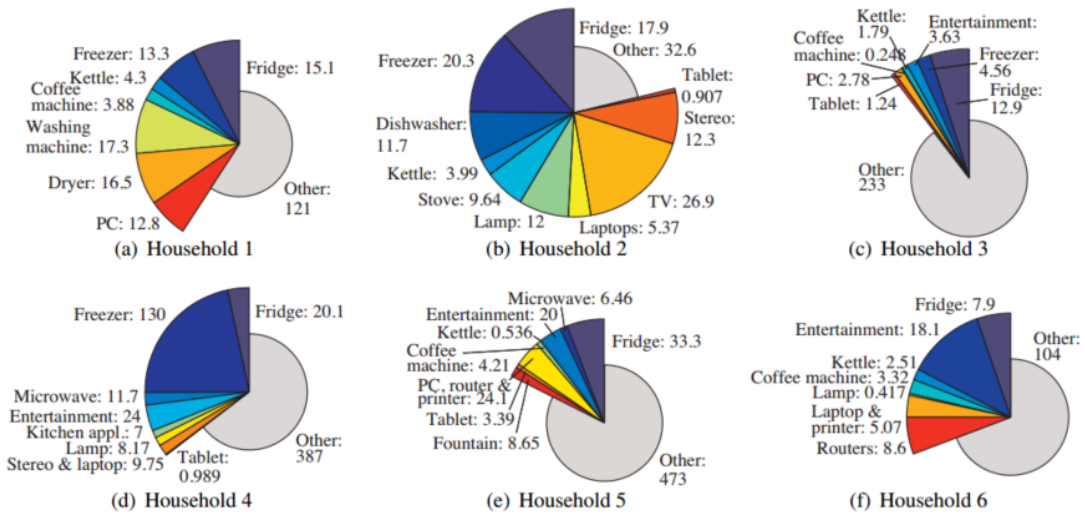


Figure 4.5: House energy distribution. Source [26]

In Figure 4.5 is shown how different appliances are responsible for the energy consumption in the different houses in the ECO dataset. For house two, more than 75% of the energy usages is accounted for by the sub-meters. This makes house two a preferred candidate for experiments, since we know who the the top consumers are and who the little consumers are.

4.5 Validation of Methods

This section describes the experiments conducted using the three methods mentioned in Section 4.3. The experiments is validated using F1 and accuracy sensitivity and specificity metrics's. Since the focus is event detection a sample can fall in one of four categories.

True Positive	This is when the guess says that there is an event, and there really is an event. Corresponding to the appliance is ON.
True Negative	This is when the guess says that there is no event, and there really is no event. Corresponding to when an appliance is turned OFF.
False Positive	This is when the guess says that there is an event, but there really is no event.
False Negative	This is when the guess says that there is no event, but there really is an event.

The F1 and accuracy score is calculated as shown in Equation 4.15 and Equation 4.16

$$TP = \text{number of true positives} \tag{4.9}$$

$$TN = \text{number of true negatives} \tag{4.10}$$

$$FP = \text{number of false positives} \tag{4.11}$$

$$FN = \text{number of false negatives} \tag{4.12}$$

$$recall = \frac{TP}{TP + FN} \tag{4.13}$$

$$precision = \frac{TP}{TP + FP} \tag{4.14}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.15}$$

$$accuracy = \frac{TP + TN}{TN + TP + FN + FP} \tag{4.16}$$

The F1 score is mainly focused on how well the system guesses the the events, and do not score the true negatives. The accuracy is slightly different in that it looks at how many correct guesses there are in relation to the total amount of guesses. For a NILM application the accuracy score on its own is not a very representative metric since some appliances have way more OFF time than ON time [33]. To guess that a appliance is OFF is just as good at guessing that it is ON for the accuracy score. Appliances that are mostly OFF can therefore get a high accuracy score even though it is hard to determine when they are ON. Therefore is it necessary also to look at the F1-score. This score tells how accurate it can be guessed when a appliance is ON. This is done by combining the recall, that tells how many true positive was guess correctly in relation to the total amount of positives, as described by Equation 4.13, and the precision that is the ratio of correct guessed positives events in relation to all positive guesses, as shown in Equation 4.14.

Five appliances have been selected from the ECO dataset house 2. The appliances have been selected so there is some top and some little consumers. The appliances selected are: water kettle, fridge, TV, stereo and Laptop. All the methods will be traind using a training period of 15 days and validated on 75 days of data.

4.5.1 Sample rate experiment

Common for the three methods is that they are designed for signals sampled at low sample rates in the sub 1 Hz rate. To determined the sensitivity, on the sample rate, is an experiment conducted where the dataset is downsampled to simulate different sampling rates. The downsampling is done by applying a mean filter, so the new sample will be the average values of the samples in a specific period. This method is selected since it is the assumed behaviour of a meter that transmits samples at low rates.

The results shown in Figure 4.6 illustrates the average score of the different algorithms when recognising the five appliances discussed in Section 4.5.

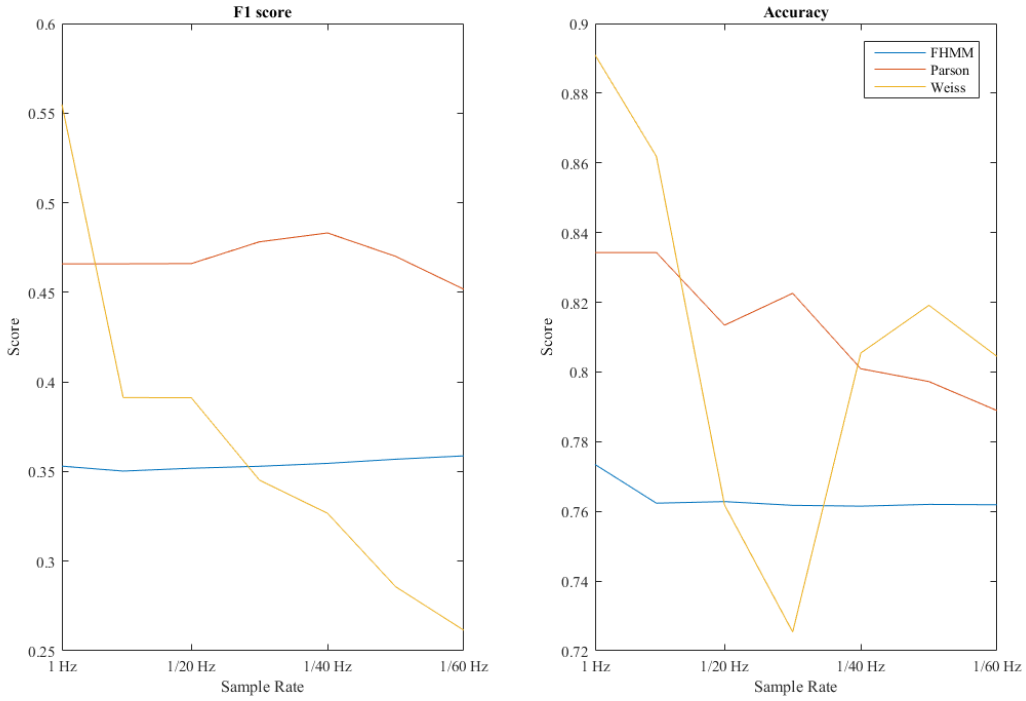


Figure 4.6: Algorithm validation at different sample graduality.

Both the accuracy and F1 score is shown. It is shown that for the Weiss clustering algorithms the overall performance is decreasing for both scores. This indicates that it is both harder for it to determine when an appliance is ON and when it is OFF. There is all most no change in the F1 score for the Parson and FHMM algorithms that both builds on HMM. Where the accuracy is decreasing for the parson as well. This tells that it finds the areas where the appliance is ON just as well at high sample rate as for low. But to correctly guess the OFF periods is getting harder for the parson algorithm, where the FHMM algorithm seems to be unaffected.

It seems that for high sample rates does the Weiss algorithm preform superior to the others, where for lower is the Parson algorithm that preforms well. The FHMM algorithm seems to preform worse that the others, but is more robust to changes in the sample rate.

The results shown in Figure 4.6 is the average for all appliances. The different appliances individual scores shows that some appliances are easier to find than others. The accuracy score for the different appliances is shown in Figure 4.7.

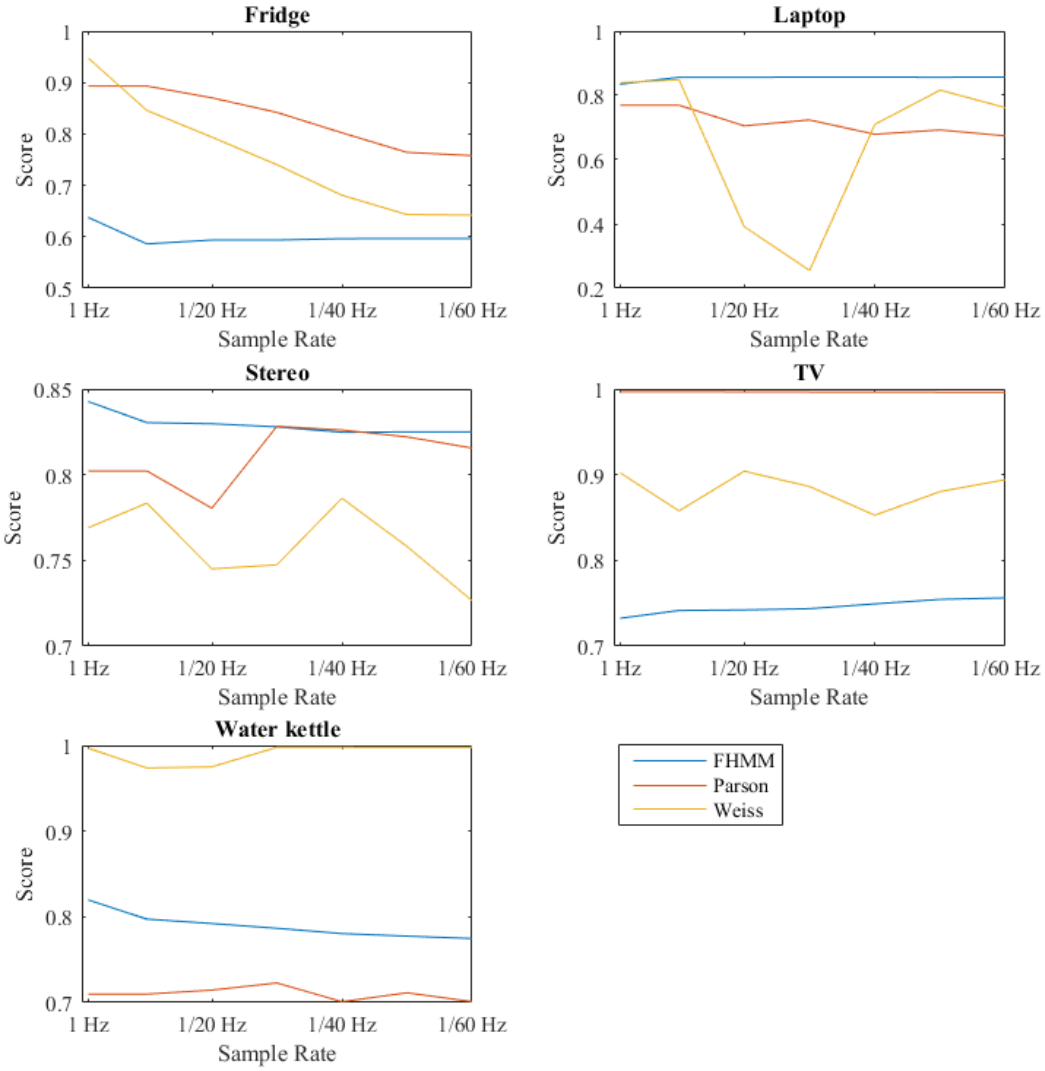


Figure 4.7: Appliances accuracy score

Since the accuracy score tells how well all events are guessed it can be heavily dominated by the OFF events. Most of the appliances selected is mostly OFF appliances. Especially the water kettle is a good example, as it is only used a few minutes each morning, and is OFF rest of the time. If it was assumed that this device was always OFF it would get a fairly high score since it is OFF most of the time.

This is the reason why the score is fairly high for all the different appliances. It is also shown that the top algorithm is different form appliance to appliance. The fridge and the stereo is the only appliances that are greatly affected by the change in sample rate, the rest seems to be relatively stable.

Since the accuracy score is dominated by the OFF periods, the F1 score is also shown in Figure 4.8. This once again indicates how the different appliances has different algorithms that

best detect them.

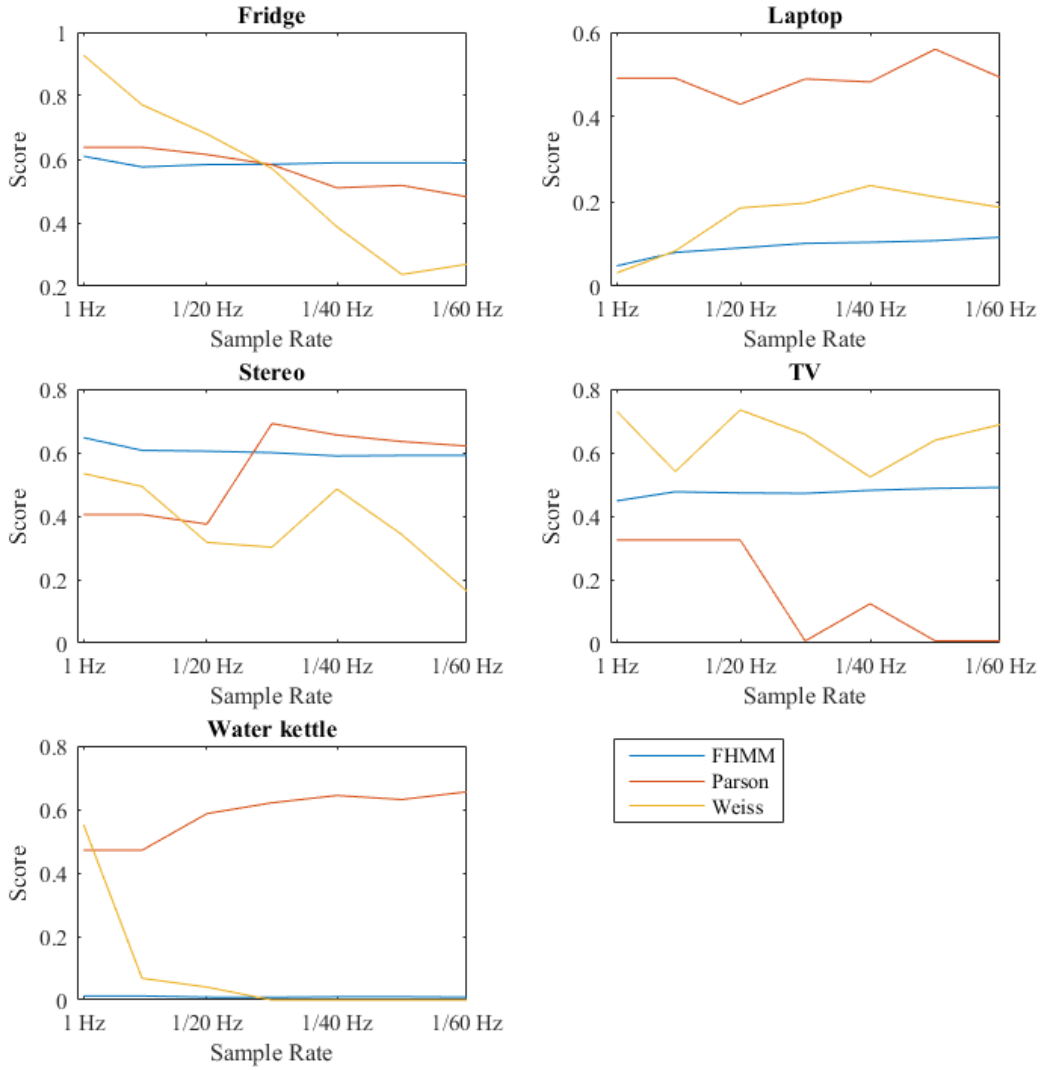


Figure 4.8: Appliances F1 score

4.5.2 Error Tolerance Experiment

As discussed in Chapter 2 is the data collected often filled with errors. This degrades the quality of the signal which can lead to worse performance. To validate the errors impact on the algorithms is an experiment conducted where the error rate is increased, and the F1 and accuracy score is measured.

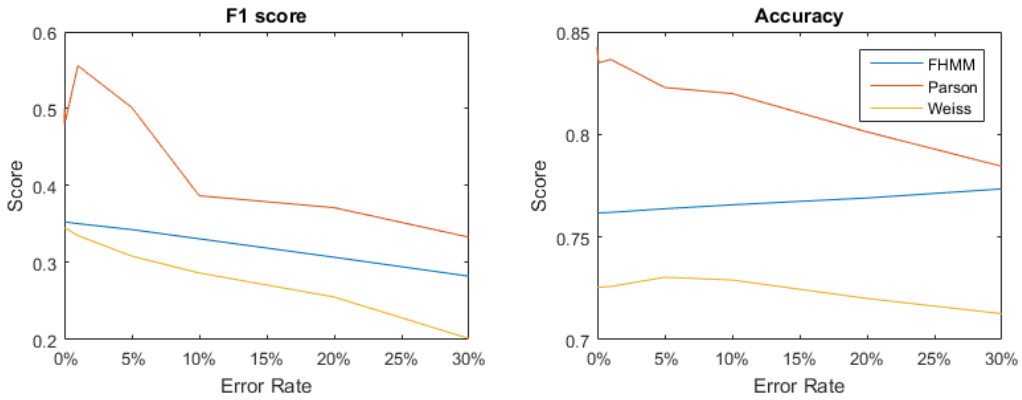


Figure 4.9: Algorithm validation at different error rate

The experiment was preformed by randomly introducing errors with a probability given by the error rate. The errors was introduced only on the validation dataset, to simulate a situation where a model was created in an error free environment, and then deployed in an error full environment. The errors is introduced independently of each other, simulating network packet loss error. This is the most common error type in the SmartHG dataset, as discussed in Chapter 2. This was done up to 30% where approximately $\frac{1}{3}$ of the samples are missing. The results is shown in Figure 4.9. It is shown that all methods is impacted by the errors, and have a decrease in performance. The experiment is conducted at a simulated sample rate on $\frac{1}{30}Hz$.

The algorithms tries as default to make a mean filtering to try to remove errors and noise. This will in situations with many errors remove some of the steps from the signal, and decrease the performance.

4.5.3 Gap Filling Experiment

In Chapter 3 it was described how mathematical gap filling methods could be used to estimate values for the missing gaps. It was mentioned that due to the stochastic nature of the signal, it was impossible to make a perfect reconstruction, but a qualifiedly guess could still be estimated.

In Chapter 3 six methods of gap filling was evaluated. It turned out that the simple linear method performed quite well in comparison to the others. The envelop method did also perform acceptable for smaller gaps. Therefore are these two method evaluated on the ECO dataset, to see if it is possible to increase performance when preforming gap fixing.

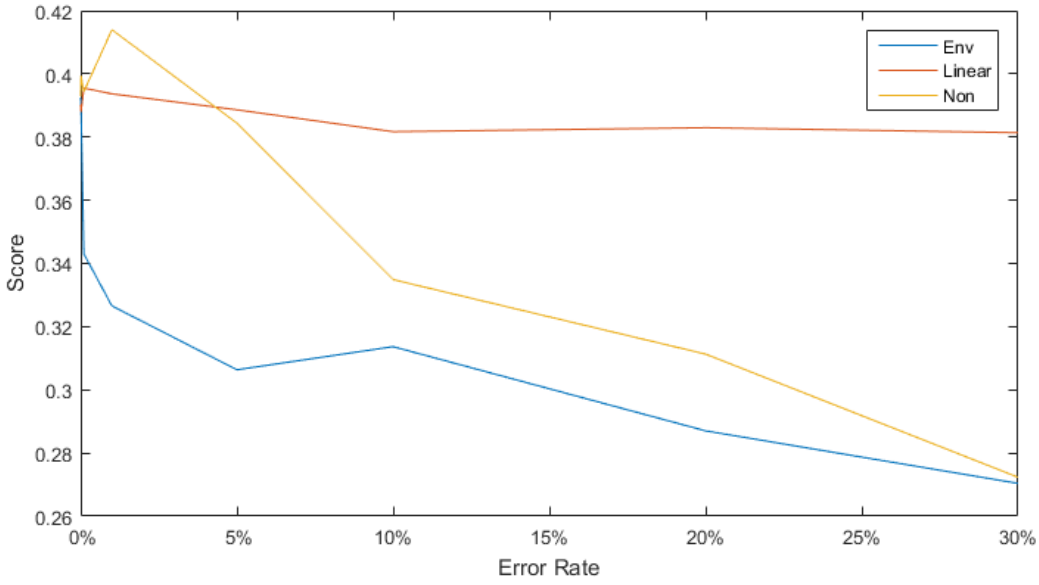


Figure 4.10: Average recognition after recovery of the 3 methods

Like the error tolerance experiment described in Section 4.5.2, is errors artificially introduced in the dataset. After this process is an attempt to removed the errors being done by using the gap filling techniques.

In Figure 4.10 is the average F1 score shown for all 3 recognition methods applied on all 5 appliances. This indicates that by using a simple reconstruction method it is possible to keep the high performance. Where for the more advanced Envelope method seems to make it worse. This is typically because the more advanced types of reconstruction methods tries to model the frequency components, which for greatly stochastic signals is more likely to introduce errors.

To ensure that it is not one recognition method that courses the bad performance for the envelope is the results inspected for the individual appliances and methods.

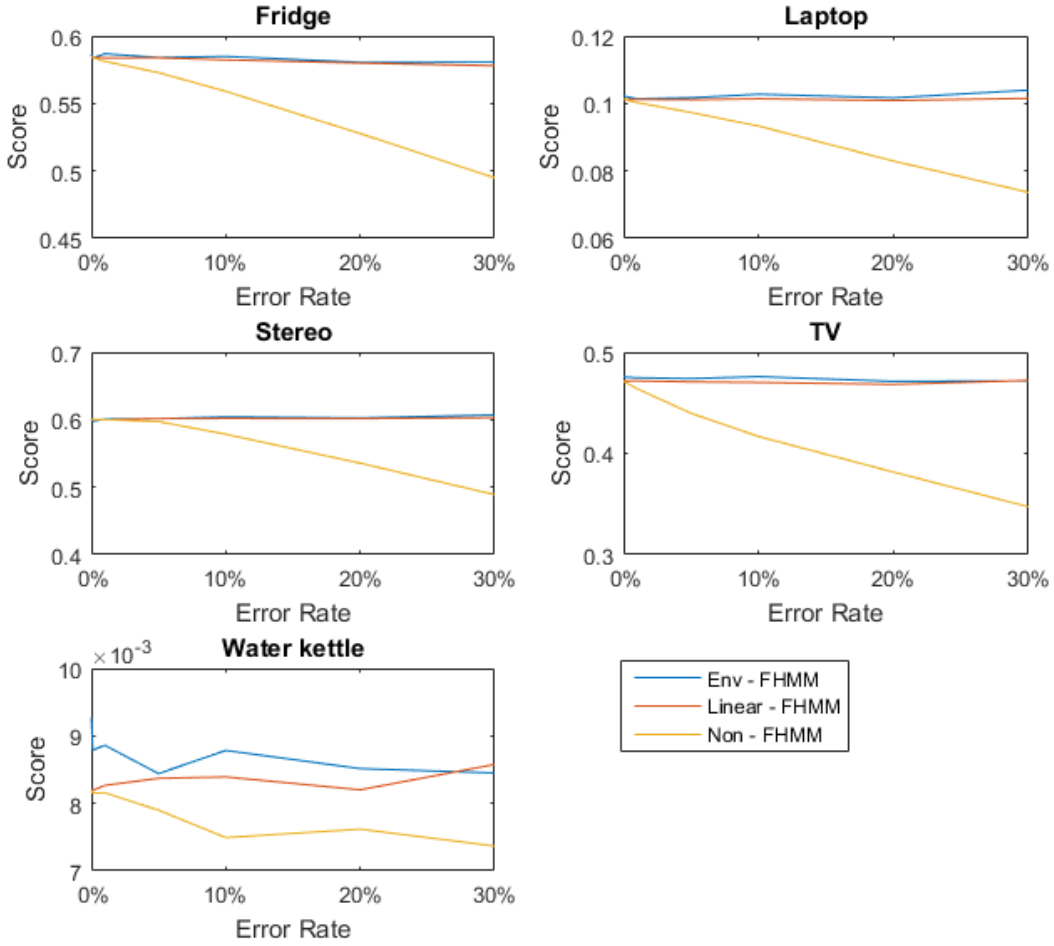


Figure 4.11: Recovery in FHMM algorithm

In Figure 4.11 is the result from the reconstruction with the FHMM algorithm shown. Here we see that the performance is prevented from decaying by the two reconstruction methods. It is also shown that the Envelope reconstruction actually preforms marginally better for the FHMM algorithm. In all cases is the reconstruction better than non-reconstruction.

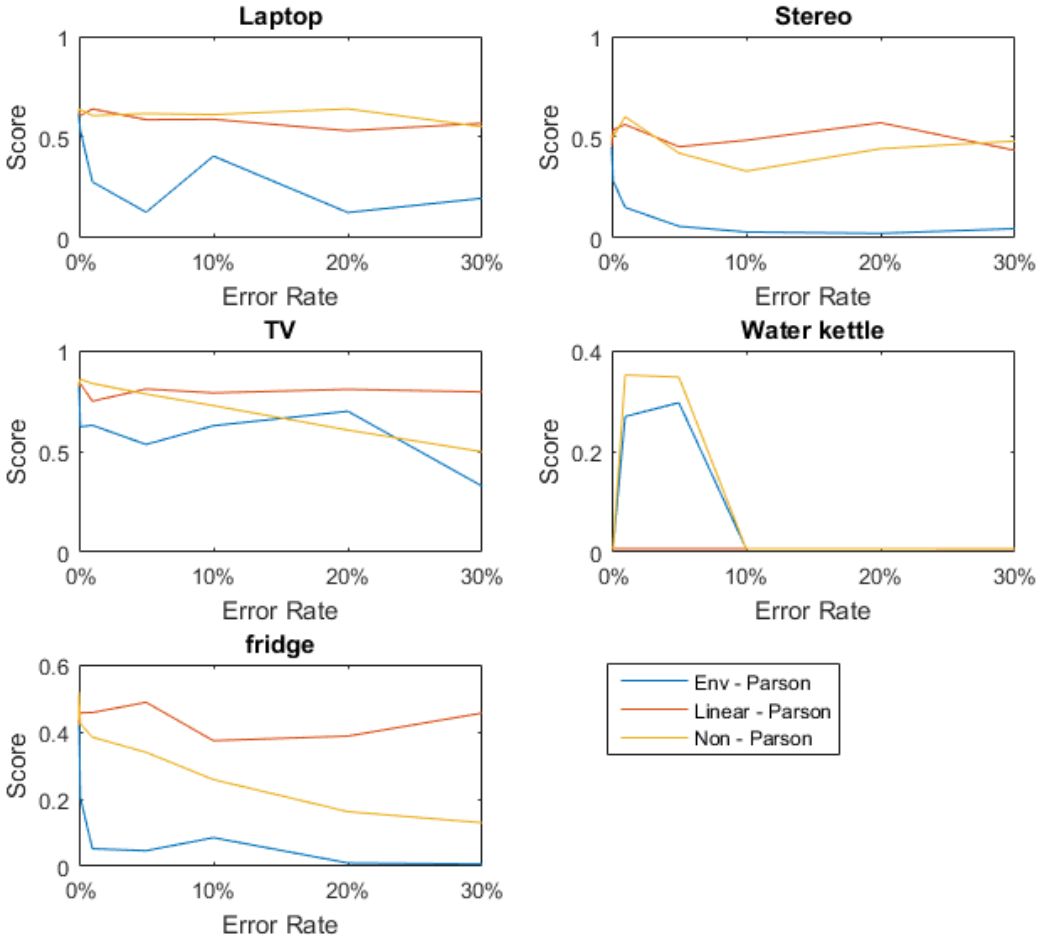


Figure 4.12: Recovery in Parson algorithm

In Figure 4.12 is the result from the reconstruction with the Parson algorithm shown. Here we see that the Envelope method is always worse than no reconstruction. For some of the appliances is the linear reconstruction a help, but fore some does it preform worse than no reconstruction.

The parson algorithm looks at jumps in the power, and uses a moving window mean filter to noise reduction and to ensure that only the significant jumps is detected. This behaviour can explains why a linear reconstruction gives almost no performance gain compared to no reconstruction for some of the devices.

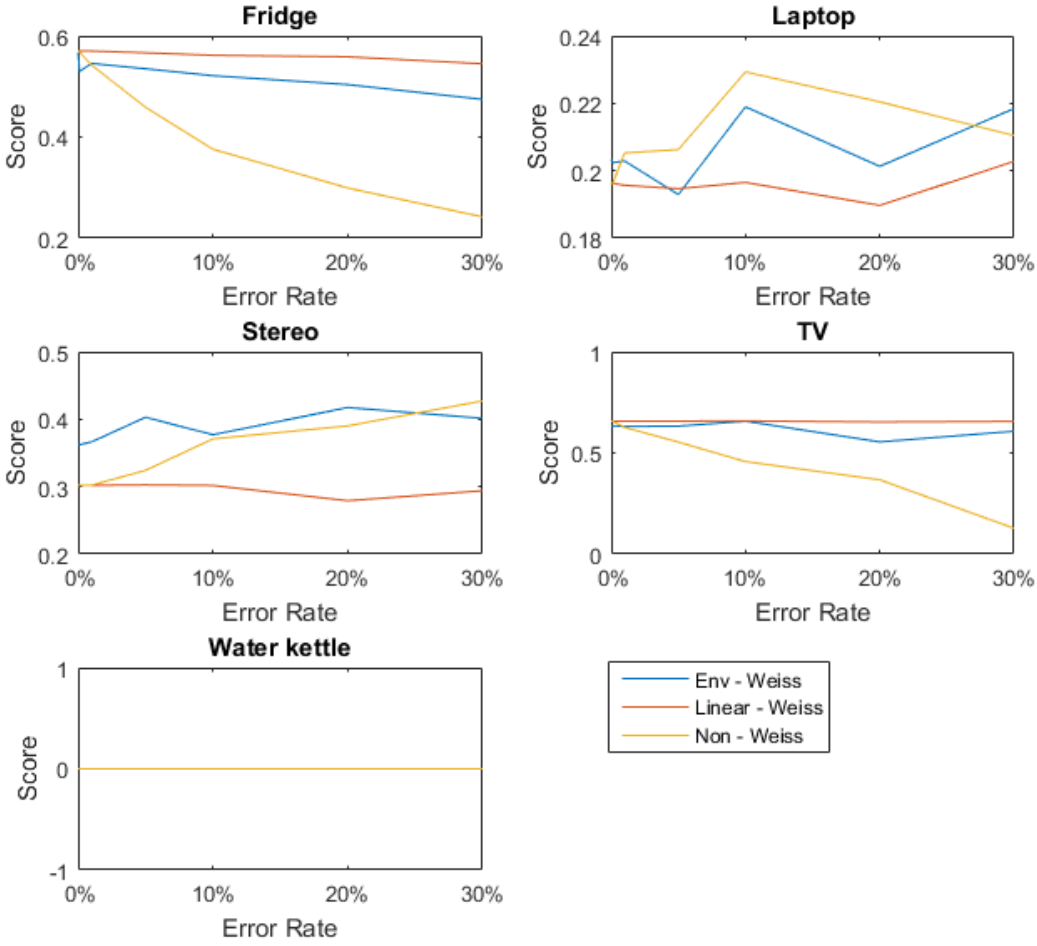


Figure 4.13: Recovery in Weiss algorithm

In Figure 4.13 is the result from the reconstruction with the Weiss algorithm shown. For the Weiss algorithm it is appliance depended what reconstruction method is the best. Even though it seems like reconstruction is not preferred for some of the appliances like the laptop, is it worth noticing that the score is close to 0.2 which is a low score. For appliances that have a higher score like the TV and fridge reconstruction seems to work as intended.

The low score indicates that the method in it self is struggling to identify the appliance, and the data error is therefore not the primary problem.

4.6 Chapter Summary

In the chapter the effects of errors and gaps in the dataset is investigated. It is shown that a system that simply ignores that errors will perform worse than one that ties to do some compensation. The amount and type of compensation is shown to be very signal depended. In general is a simple linear interpolation enough when using algorithms that are designed for low frequency signals.

The resolution of the samples seems to have an effect of the performance. The algorithms in focus are designed to low frequency signals, even so it is shown that the small variations created in higher resolution can improve the performance. This is working with the assumption that the instantaneous power consumption received from the smart meter is a average of the power consumption since the last sample.

There is also indications of that some appliances is more easily detected by some algorithms, and other appliances by other algorithms. This indicates that it is very likely that a greater performance can be achieved by mixing algorithms depended on the appliances.

Environment Influence 5

Not just the sample rate and error rate are determining factors when designing a non-intrusive load monitoring (NILM) application. The environment which the application is deployed and trained in is critical for the performance of the system. The "environment" is the parameters describing the static conditions of the household which the algorithms are deployed in. An example of an environment parameter can be number of known devices, number of unknown devices or number of simultaneously active devices.

In order to investigate some of the environment parameters effect on a NILM application the SmartHG dataset is used. In the chapter is appliance disaggregation conducted on the SmartHG dataset, using the factorial hidden Markov models (FHMM) and Parson algorithms. The Weiss algorithm uses the power factor information, which is not available in the SmartHG dataset. The disaggregation process consists of a training phase and a validation phase.

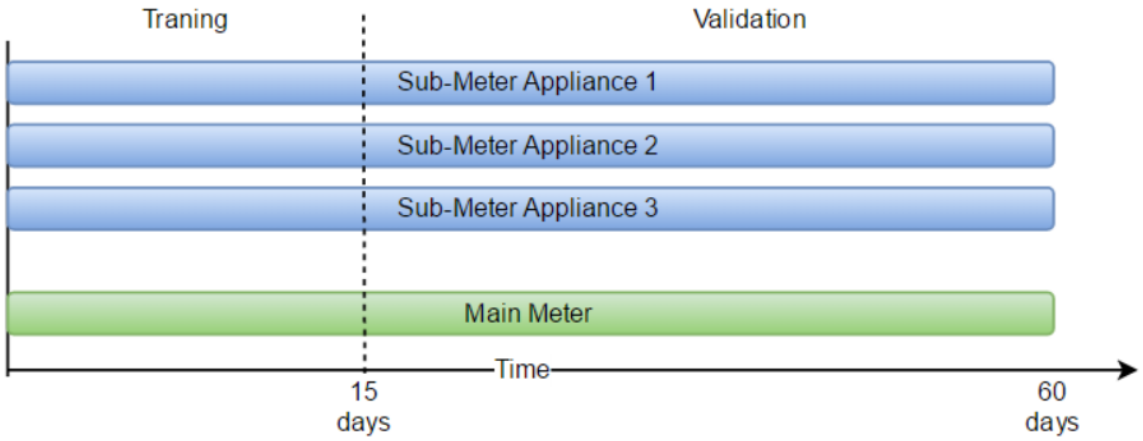


Figure 5.1: Disaggregation phases

All experiments conducted in the chapter will start with a training phase of 15 days. During this period the sub-meter and main meter data will be used to by the FHMM and Parson algorithm to learn statistical disaggregation models. After this initial training phase will the algorithms be tasked with the job of disaggregating the main meter signal in a 45 day validation period. The algorithms will only have access to the data supplied by the main meter, and the statistical disaggregation models found in the initial training phase. The sub-meter data in the validation phase be used to validate the results of the disaggregation. The basis setup is illustrated in Figure 5.1 where the green color symbolises data from main meter, and blue symbolises data

from sub-meter.

5.1 Challenges In The SmartHG Dataset

The Electricity Consumption and Occupancy (ECO) dataset used in the last chapter, consists of 6 households and the data is sampled at a rate of 1 Hz over a period of 8 months. The SmartHG dataset is different in many aspects. The data is sampled at a slower rate of $\frac{1}{30}$ Hz, but on 25 households over a period of 8 month. Each house have only a small number of sub-meters. The sub-meters are mainly placed on little consumers such as televisions and stereos, which presents an interesting challenge for load disaggregation. The SmartHG dataset contains both the accumulated power, and the instantaneous power usages of the different households.

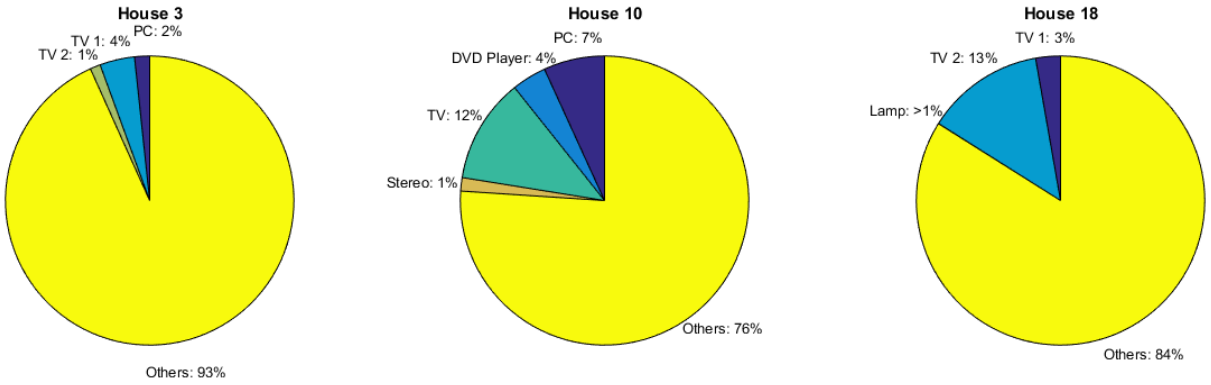


Figure 5.2: Energy distribution in a household

The power distribution of three households is shown in Figure 5.2. This illustrates how much energy each appliance uses, in relation to the households total energy consumption. The *other category* shows the energy consumption not accounted for by the sub-meters.

Household 3, 10 and 18 are some of the houses that have the smallest *other category*. They are therefore selected for further study, since they provide the most information about the house. For the energy distribution in the other households in the SmartHG dataset see Appendix H.

5.2 Background Consumption Influence On Disaggregation

The households in the SmartHG dataset have a relative big consumption created by appliances in the *other category*. The consumption in the *other category* is created by appliances that are unknown. In order to do disaggregation, statistical models like models based on the FHMM and Parson algorithms, must first be created. This requires knowledge about the appliances consumption. This kind of information is not available for appliances in the *other category*.

This consumption created by appliances that is not part of the statistical disaggregation models can be thought of as "background consumption". The appliances in the *other category* will always create *background consumption*. This does not exclude appliances that are not part of the *other category* from creating *background consumption*. All appliances that are not part of the statistical disaggregation models, contribute to the *background consumption*.

5.2.1 Detection In An Environment With High Background Consumption

To investigate the influence *background consumption* have on a NILM application, disaggregation of the known appliances of house 3, 10 and 18 are done using the Parson and FHMM algorithms.

		FHMM		Parson	
		F1	Accuracy	F1	Accuracy
House 3	TV 1	0.19	0.74	0.10	0.40
	PC	0.19	0.84	0.13	0.45
	TV 2	0.03	0.84	0.20	0.11
House 10	TV 1	0.60	0.76	0.40	0.25
	Stereo	-	1.00	-	1.00
	PC	-	0.99	-	0.99
	TV 2	-	0.99	-	0.99
House 18	TV 1	0.36	0.65	0.24	0.14
	Lamp	-	0.99	-	-
	TV 2	0.73	0.58	-	0.00

Table 5.1: Appliance disaggregation results of house 3,10 and 18 for the SmartHG dataset

As previously stated is the *background consumption* high for this dataset. Table 5.1 summarizes the results from the disaggregation. The low F1 scores indicates that it is hard for the algorithms to correctly disaggregate the meters. This is due to the many spikes there are in the data that can indicate a change in appliance usage. In Figure 5.3 is a snippet from the validation period shown. The blue lines shows the data as it is received from the smart-meter. The other lines indicates the appliances. In the sub plot named "True Signals", on the left side, is the true consumption shown for all the meters. This consumption is known from the sub-meters on the appliances. In the sub plot named "Inferred Signal", on the right, side is the inferred usages of the different appliances created by the FHMM algorithm.

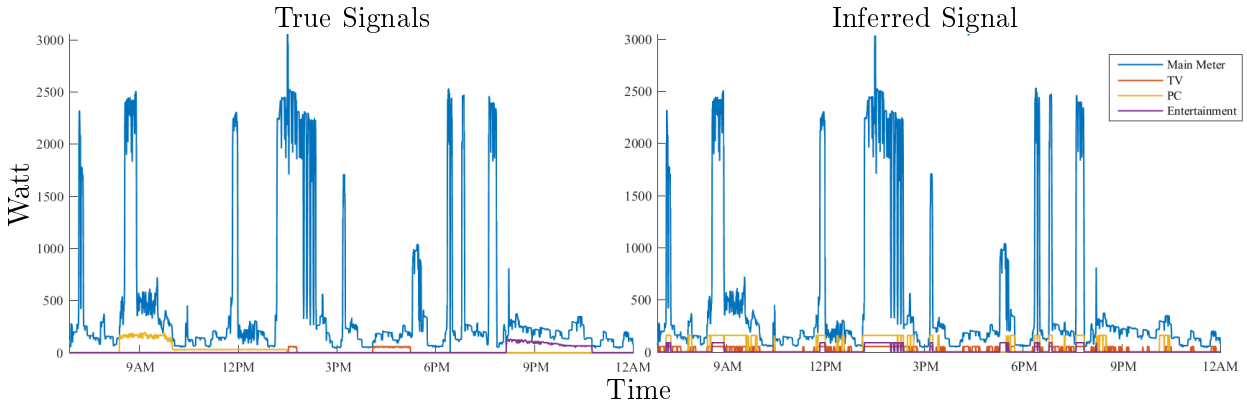


Figure 5.3: FHMM disaggregation snippet

In the figure does the blue lines illustrates how the main meter have many fluctuations. The algorithm tries to map each fluctuation to a change in an appliance state, or as *background consumption*. There are many of the fluctuations created by the *background consumption* that are similar to the power changes created by the TV, PC and entertainment system. The inferred

signal shows how the algorithm thinks the appliances is used. It is easily seen that this is quite different from the true values.

5.2.2 Detection In An Environment With No Background Consumption

The assumption that the *background consumption* is what troubles the disaggregation dictates that for a house where all appliances are known, and there are no *background consumption*, the disaggregation will perform better.

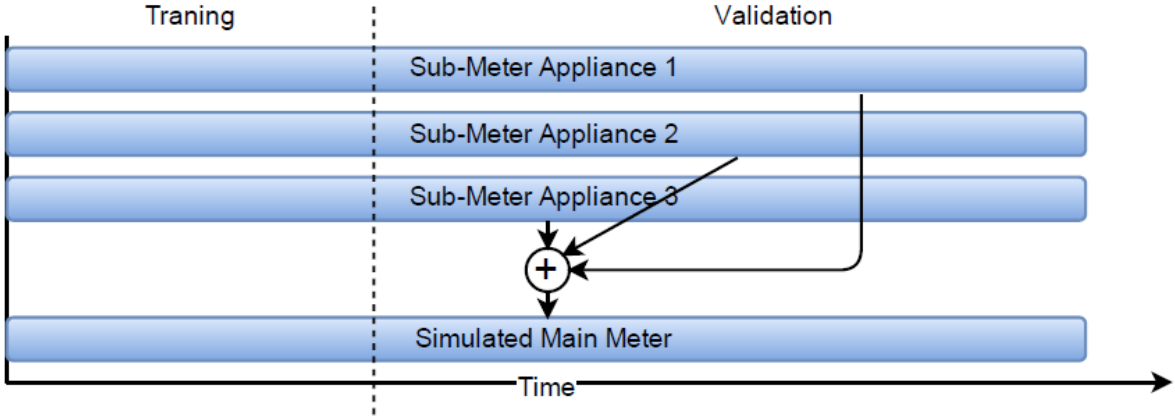


Figure 5.4: Artificially constructed main meter

In order to test this hypothesis is a main meter signal artificially constructed by summing the values of all the sub-meters in a house. This process is illustrated in Figure 5.4. In this manner is a new set of artificial houses created where there are no *background consumption*. By applying the same disaggregation algorithms, as for the environment with high *background consumption*, on the new environment with no *background consumption* is the performance much increased as shown in Table 5.2.

		FHMM		Parson	
		F1	Accuracy	F1	Accuracy
House 3	TV 1	0.73	0.96	0.26	0.86
	PC	0.74	0.96	0.30	0.91
	TV 2	0.83	0.96	0.20	0.11
House 10	TV 1	0.97	0.98	0.40	0.25
	Stereo	-	1.00	-	1.00
	PC	-	0.99	-	0.99
	TV 2	-	0.99	-	0.99
House 18	TV 1	0.95	0.98	0.24	0.14
	Lamp	-	0.99	-	-
	TV 2	0.73	0.58	0.73	0.58

Table 5.2: Disaggregation of appliances on artificially constructed main meters

The F1 and accuracy score are both improve, as shown in Table 5.2. Even though the artificial house only contains three or four appliances, and all are accounted for by the statistical

disaggregation model created by the FHMM and Parson algorithms, there are still some of the appliances that are hard to find. This is due to the appliances power usage pattern in OFF/standby mode combined with the rare usage of the devices. When the standby pattern of a device is complex and the appliance is only rarely used, the model will try to fit to the standby pattern and not the usage pattern. This happens if the overall best consumption disaggregation results is achieved by fitting to the small energy fluctuations in the OFF state. This is often the case for appliances that are rarely used int the training period. The downside to this is that it will be incapable of detecting change between the ON/OFF states.

One other aspect to note is that even though the F1 scores are higher, only a few is higher than 0.9. This could be due to "appliance interference". This happens when appliances are similar. When a house contains two or more appliances that are similar it gets harder for the algorithm to tell if it is the one or the other. This is could be what happened for TV 1 and TV 2 in house 3. The algorithms will wrongfully assign some of the events belonging to TV 1, to TV 2 and vice versa. This happens because the power draw signature are similar. This can be a problem if the goal is to determine exactly which appliance there is responsible for the power consumption.

Appliance interference has a profound effect on the Parson algorithm. The parson algorithm is designed to split up appliances based on appliance categories. This is achieved by utilizing a set of general models. Since a lot of the appliances are similar in their consumption changes from state to state are the model used by the Parson algorithm almost identical. This leads to the events being wrongly categorised. The usages of general models is one of the benefits of the Parson algorithm, since it makes the algorithm very portable between unknown domains. Unfortunately does this feature come whit the downside of increased sensibility to *appliance interference*. The influence of *appliance interference* is further studied in Section 5.3.2.

5.2.3 Background Consumption Effect On Training And Validation

The experiments in the two previous Section 5.2.1 and Section 5.2.2 suggest that *background consumption* has an impact on the performance of the system. But for a system based on machine learning techniques like the FHMM and the Parson algorithms it raises the questions, is it to hard for the system to train the correct statistical disaggregation models in the signal with much *background consumption*? or are the *background consumption* just interfering with the disaggregation process?

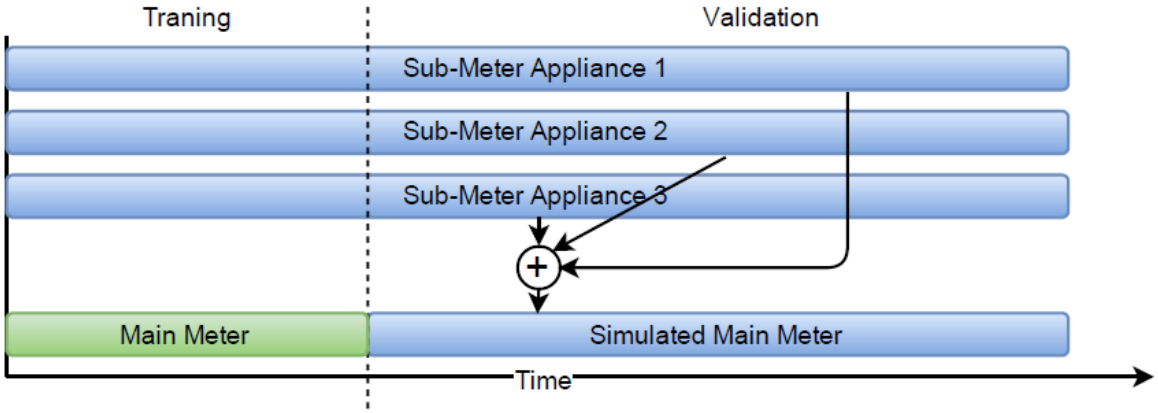


Figure 5.5: Illustration of dataset creation by combining real and constructed data

In order to investigate these questions was a new dataset created that combined real data collected from the house main meter, and an artificially constructed main meter. The new dataset was created by using the real data in the training period of the house, and constructed main meter data in the validation, as illustrated in Figure 5.5.

This creates a environment with a lot of *background consumption* in the training phase and no *background consumption* in the validation phase. If the *background consumption* did not influence the creation of the statistical disaggregation models that represent each appliance, then the results were expected to be just as good as for the environment with no *background consumption* as conducted in Section 5.2.2.

		FHMM		Parson	
		F1	Accuracy	F1	Accuracy
House 3	TV 1	0.02	0.94	0.23	0.86
	PC	0.29	0.94	0.32	0.92
	TV 2	0.00	0.88	0.20	0.11
House 10	TV 1	0.00	0.74	0.40	0.25
	Stereo	-	1.00	-	1.00
	PC	-	0.99	-	0.99
	TV 2	-	0.99	-	0.99
House 18	TV 1	0.00	0.86	0.24	0.14
	Lamp	-	0.99	-	-
	TV 2	0.73	0.58	0.73	0.58

Table 5.3: Disaggregation in real and constructed main meters combined

If the results from the experiment, shown in Table 5.3, is compared with the results from Table 5.2 from Section 5.2.2 it is clearly seen that the performance of the system trained in real data is much lower than the one trained on artificially constructed data. This indicates that the disaggregation models obtained in the real data is influenced by the *background consumption*, and is therefore not fitting accurately enough to the true disaggregation models. This indicates that the low performance in the real data shown in Table 5.1 in Section 5.2.1 is caused by the models not fitting correctly to the true values, and not by the models being triggered by application noise from other appliances.

In order to validate this hypothesis was a similar experiment conducted, where the training data was artificially constructed, and the validation data was the real data from the main meter as illustrated in Figure 5.6.

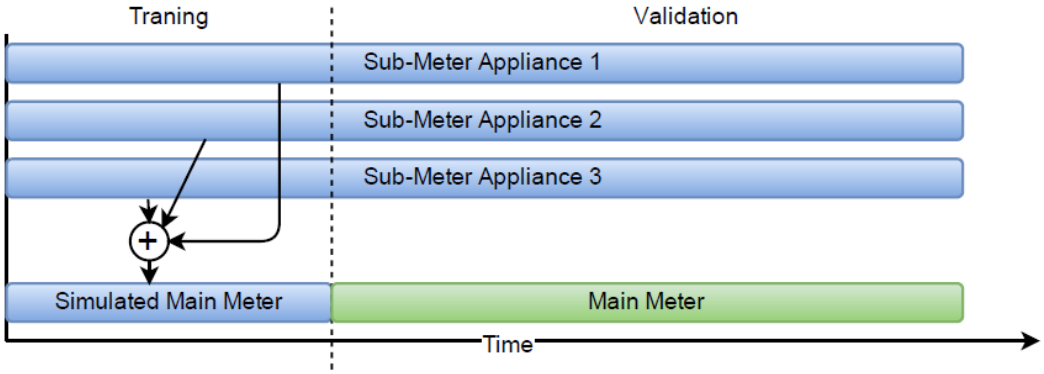


Figure 5.6: Illustration of dataset creation by combining constructed and real data

By training the model on the constructed data, is the true model for the appliances more easily found. If the environment with high *background consumption* and real data is non-interfering with the disaggregation models can a performance like for the artificially constructed environment, with no *background consumption*, described in Section 5.2.2 be expected for the validation phase.

		FHMM		Parson	
		F1	Accuracy	F1	Accuracy
House 3	TV 1	0.10	0.42	0.10	0.45
	PC	0.13	0.39	0.12	0.44
	TV 2	0.21	0.35	0.20	0.11
House 10	TV 1	0.40	0.25	0.40	0.25
	Stereo	-	1.00	-	1.00
	PC	-	0.99	-	0.99
	TV 2	-	0.99	-	0.99
House 18	TV 1	0.27	0.26	0.24	0.14
	Lamp	-	0.99	-	-
	TV 2	0.73	0.58	0.73	0.58

Table 5.4: Disaggregation in constructed and real main meters combined

The results are shown in Table 5.4. The performance of this experiment is lower than for the

one in the environment with no *background consumption*. This lead to the conclusion that the *background consumption* is both affecting the models, and interfering in the validation process.

The results from the disaggregation of the data in the real environment from Section 5.2.1 are compared with the results from the the previous experiments, where the models were extracted from constructed data and validated on the real data. The results is summarized in Table 5.5.

		FHMM Real		FHMM Constructed	
		F1	Accuracy	F1	Accuracy
House 3	TV 1	0.19	0.74	0.10	0.42
	PC	0.19	0.84	0.13	0.39
	TV 2	0.03	0.84	0.21	0.35
House 10	TV 1	0.60	0.76	0.40	0.25
	Stereo	-	1.00	-	1.00
	PC	-	0.99	-	0.99
	TV 2	-	0.99	-	0.99
House 18	TV 1	0.36	0.65	0.27	0.26
	Lamp	-	0.99	-	0.99
	TV 2	0.73	0.58	0.73	0.58

Table 5.5: Trained in real vs. constructed data

The table shows that the performance of the system is better when trained in the real environment, as in relation to when the the system is trained in a artificial environment and then deployed in a environment with *background consumption*. This is in contrast to what one might think, since the environment with no *background consumption* should have supplied more correct models. But when the *background consumption* is removed from a house, in the training process, is a offset error introduced in the models.

Appliances such as refrigerators and freezers that are always on will create a local house offset. This will vary from house to house, and is depended on the types and number of appliances in the house. If only a small number of devices are modelled in each house, as in the SmartHG case, is the offset almost exclusivity a part of the *background consumption*. When the models are learned in the training phase in the real data, is the offset learned into the model, and further improves the model. This is not the case when the model are trained in the constructed dataset.

If not all appliances in a house are known, a better performance can therefore be obtained by training in the intended environment of deployment.

Some algorithms are designed to not be affected by the house offset, and is therefore moved from different environments more easily. The Parson algorithm is designed with this in mind. The performance of the Parson algorithm is therefore the same, or in some cases improved, when using models trained in constructed data. This advantage unfortunately comes with the problem of generalization, which lead to harder source separation, as discussed in Section 5.2.1.

5.2.4 Improving Performance Using Norm Knowledge

When an appliance is in a environment with a lot of *background consumption*, some of the *background consumption* can be interpreted as the appliance. This *appliance interference* creates a scenario where it seems like the appliance is used more than it actually is. Often does the *appliance interference* create usage patterns that is in contrast to that one would expect. This can make it seems like a appliance is used in a unrealistic manner. By improving the method with the knowledge of how a specific appliance is most commonly used, also called the norm usage, it is possible to filter the signal and improve the performance.

In Figure 5.7 is the disaggregation of TV 1 from house 10 shown for an one day period. The blue line indicate the FHMM algorithms suggestions on when the TV are ON and OFF. The green line is the actual ON and OFF periods, collected from the sub-meters.

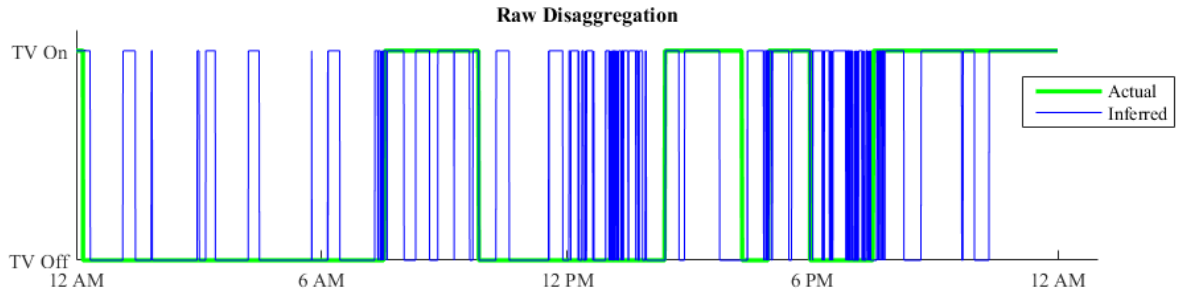


Figure 5.7: Disaggregation of TV 1 from house 10

As seen in the figure is there a lot of small spikes that are less than 10 min, and sometime is the TV ON for 15 minutes OFF in 2 and then ON again for another 15 minutes. This behaviour does not fit the normal behaviour one would expect for a person. The norm is to see TV more than 20 minutes at the time, and when a TV is turned OFF it is common that it is turned OFF for atleast 15 minutes before turned ON again. These numbers are a loose estimate, found by observing the sub-meter data.

One approach of applying the norm knowledge to the disaggregation result is to use a "norm filter". This is a filter that takes the results from the disaggregation and removes unrealistic behaviour. This is achieved by using a method that iteratively purges events from the signal and merges events that are too close to each other. The "purge and merge process" runs two iterations. First is all events shorter than 10 minutes are purged from the signal. Next is events separated with less than 15 minutes of OFF time merged. The purge and merge cycle is repeated one more time where events shorter than 25 minutes are purged and events closer than 30 min are merged.

The values used in the *purge and merge process* is found experimentally. These parameters gave the best results for the TV's in the experiments. In a real deployment scenario where a *norm filter* was applied, the coefficients must be selected to fit the normal behaviour of the residents.

The downside of this approach is that it assumes that the devices is used in a certain manner. For our example of values found for a TV will all watch sessions shorter than 25 minutes be missed.

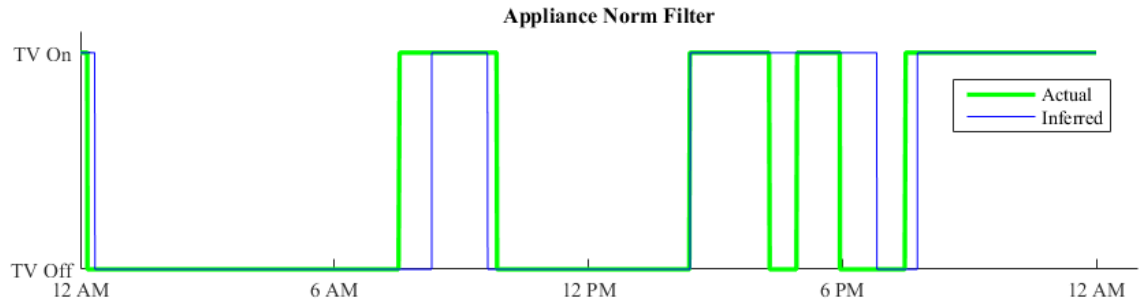


Figure 5.8: Purged and merged signal from TV 1

In Figure 5.8 is the same signal as in Figure 5.7 filtered by using the *norm filter*. As seen in Figure 5.8 does the signal inferred by the filtered FHMM algorithm match more closely to the actual signal.

In this example is the norm knowledge applied as a filter, that can filter the disaggregation result. Another approach could be to integrate the norm knowledge as a component in the state transition probability of the hidden Markov models (HMM).

The *Norm filter* is only recommended if the accuracy and F1 score of the disaggregation is low. It is undesirable to force the system to enforce the normal usage as the truth. This can filter away accentual events, that lay outside the norm. But for systems that have a low accuracy and F1 score can this approach help clean the signal.

5.3 Model Complexity And Completeness Influence

Some of the parameters that often differ in the environments where NILM applications are deployed, is the number of appliances. As previously disused is not all appliances included in the statistical disaggregation models used by the FHMM and Parson algorithms. These are the appliances creating the *background consumption*. The statistical disaggregation models often contain the information needed to do disaggregation on multiple appliances. The number of appliances in the environment is defined as the "complexity" of the model. The *complexity* does not tells how many appliances there is in a given model. As the *complexity* in a NILM application is increasing the detection rate is deceasing [34]. This is one of the major problems and is why many researchers only focus on a small subset of appliances that have a fairly unique consumption signature. The *complexity* is the sum of appliances creating the *background consumption* and the appliances in the statistical disaggregation model.

An other metric looks at the ratio between the number of appliances in the statistical disaggregation model and the *complexity*. This ratio is called the "completeness" of the model, since it indicate how many of the total appliances that are include in the model.

5.3.1 Test Set Creation

In order to experiment with the *completeness* and *complexity* of the statistical disaggregation models in the SmartHG dataset is a more controllable environment required. In order to obtain this is a series of datasets artificially constructed from the SmartHG dataset.

First an artificial house dataset where created called the "TV house dataset". The *TV house dataset* is created by picking the relative dominant TV signals from the houses 5, 10, 11, 13, 18 and 23 in the SmartHG dataset. By combining the signals to one artificial signal can a artificial main meter for a house with 7 TV's be created.

The *TV house dataset* consists of dominant appliances, and there is no *background consumption* from unknown devices. It is still worth noticing that all the 7 appliances are a Type-II appliances, and they are all TV's, which make there usage pattern some what similar. Since the Parson algorithm is greatly effected by this, is it excluded from the experiments. Therefore will the focus mainly be on the FHMM algorithm.

If the *TV house dataset* was seen as a set of sub-meters as mathematically shown in Equation 5.1. Where TV_n is a vector containing the consumption data collected for a specific TV.

$$\mathbf{S}_{full} = \{TV_1, TV_2, ..., TV_7\} \tag{5.1}$$

Since there is no *background consumption*, can the main meter data be artificially created as the sum of sub-meter data as shown in Equation 5.2. Where X is the vector of sub-meter measurements from the televisions. M_{main} is a vector containing the artificial main meter consumption data.

$$M_{main} = \sum_{X \in \mathbf{S}_{full}} X \tag{5.2}$$

Using this approach it is possible to create a set with a specific *complexity*. This can be achieved by taking a subset of the $\mathbf{S}_{full}$ set with the cardinality of the desired number of appliances. The main meter can now be found for this subset, using the same principle as in Equation 5.2. Using the Equation 5.3, it is possible to find a set of all sets with a desired *complexity c*.

$$\mathbf{H}_c = \{x | x \in \mathbb{P}(\mathbf{S}_{full}), \forall |x| = c\} \tag{5.3}$$

This is done using a set comprehension that iterates over the power set of $\mathbf{S}_{full}$ and picking out all sets that have a cardinality equal to the desired *complexity c*. $\mathbf{H}_c$ is therefore a set of sets with a given *complexity*. This can be graphically shown as in Figure 5.9.

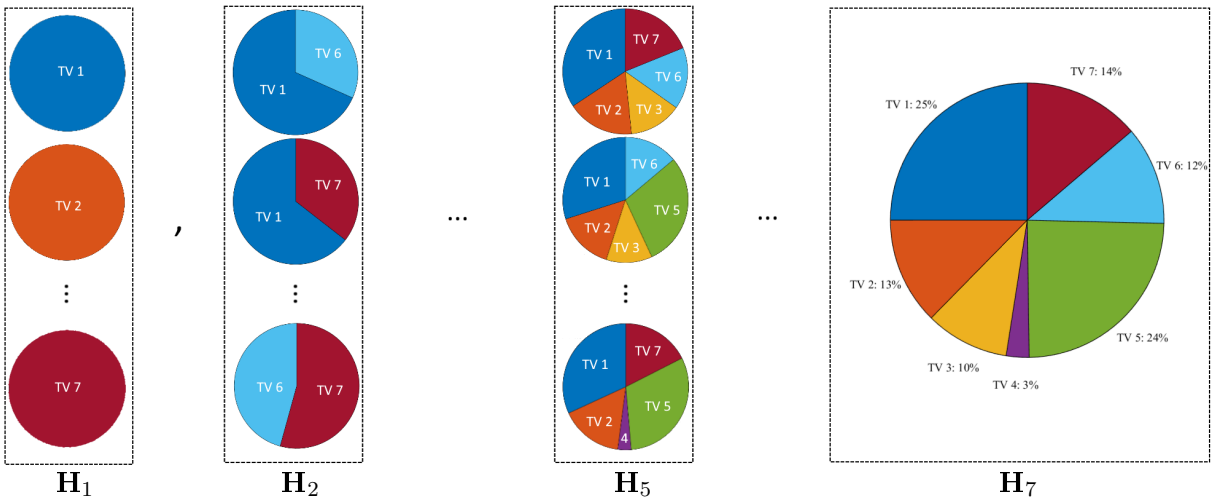


Figure 5.9: $\mathbf{H}_c$ sets from the TV house dataset

Where the $\mathbf{H}_7$ only contains the $\mathbf{S}_{full}$ set. In the figure is illustrated the relative energy usage of each appliance in a dataset. In this manner is it possible to artificially create one or more datasets with a desired *complexity*, equal to or less than the number of appliances in $\mathbf{S}_{full}$. The amount of artificiality created houses with a desired *complexity* can be found by using the simple combinatorial equation shown in Equation 5.4. Where $|\mathbf{H}_c|$ is the cardinality of $\mathbf{H}_c$. The desired *complexity* is c, and the size of the pool for which appliances can be selected from is given by the cardinality of $\mathbf{S}_{full}$.

$$|\mathbf{H}_c| = \frac{|\mathbf{S}_{full}|!}{c! \times (|\mathbf{S}_{full}| - c)!} \tag{5.4}$$

$$A_c = |\mathbf{H}_c| \times \frac{c}{|\mathbf{S}_{full}|} \tag{5.5}$$

As illustrated in Figure 5.9 can an appliance appear in multiple sets in a given $\mathbf{H}_c$ collection. The number of sets containing a specific appliance in a $\mathbf{H}_c$ collection is A_c. This can be calculated as in Equation 5.5. In order to ensure that the combination of appliances does not have an effect on the experiments, are the experiments conducted on all combinations in $\mathbf{H}_c$. The results for each experiment is an average of the performance for the appliance in the experiment.

	$c=1$	$c=2$	$c=3$	$c=4$	$c=5$	$c=6$	$c=7$
$\lvert\mathbf{H}_c\rvert$	7	21	35	35	21	7	1
A_c	1	6	15	20	15	6	1

Table 5.6: TV House dataset complexity

In the case of the *TV house dataset* can the cardinality of $\mathbf{H}_c$ and A_c values for the different number of appliances c, be seen in Table 5.6. The cardinality of $\mathbf{H}_c$ tells how many artificial houses that can be constructed with the desired *complexity* c. The A_c values tells how many artificial houses contains a specific TV.

5.3.2 Model Complexity Test

To investigate the effect the *complexity* of the statistical disaggregation models have on the performance of a NILM system, is disaggregation done on every artificial house that can be created from $\mathbf{H}$. The experiment is conducted by training the models to be able to disaggregate all the appliances in a given scenario. This creates an environment where there is no *background consumption* that can interfere.

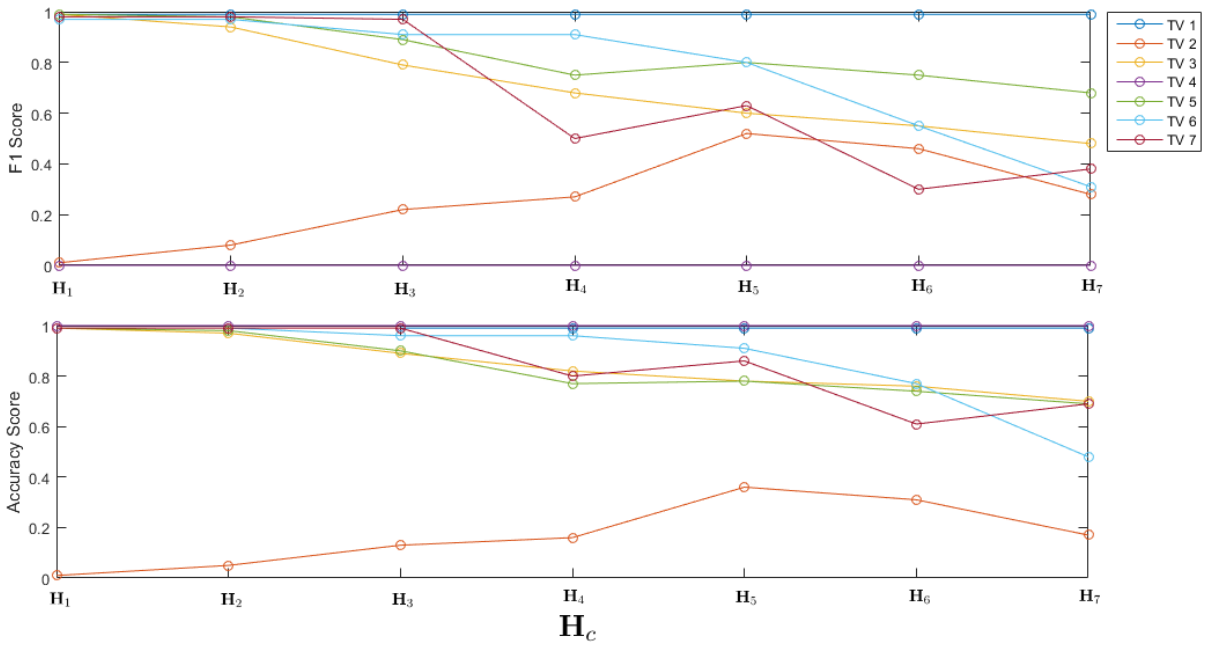


Figure 5.10: Average score at different complexity

The test is performed on every combination, to ensure that there is not some combination that are more favourable than others. The results shown in Figure 5.10 are the average of the results from the many combinations as described in Section 5.3.1.

For the most part it is fairly easy to correctly classify the TV's when the *complexity* is only one. As a general trend does the F1 and accuracy score of the TV's decrease as the *complexity* of the house increase. This is due to the *appliance interference*. The *appliance interference* effect is when there actually is an event, but the system classifies it to the wrong device. An effect of this is most clearly seen on TV 2 signal. TV 2 is a very complex signal, and is therefore hard to track by the algorithm, as seen at the $\mathbf{H}_1$ *complexity* that have an F1 score on almost zero. When the *complexity* is increased it looks like it is easier to detect the signal, and the F1 score is increasing. What is actually happening is *appliance interference* from the other TV's. Since all the appliances are TV's they have a lot of overlap in their usage. A lot of people watches TV at the same time of day e.g. between 18:00-22:00. This makes the algorithm encounter signal's

that are similar in structure for the different TV's. It has a hard time deciding which appliance is responsible. Therefore can some events generated by the other TV's be assigned as TV 2. If by chance TV 2 actually was on when the events from the other TV's was wrongfully assigned, the F1 and accuracy score of TV 2 would improve.

For other appliances that are not as hard to classify as TV 2 will the *appliance interference* effect decrease the F1 and accuracy score as seen in the figure.

5.3.3 Model Completeness Influence

In the experiments conducted on the the SmartHG dataset in Section 5.2 it was shown that the *appliance interference* made it hard to correctly disaggregate the appliances of the houses. In the last experiment in Section 5.3.2, it was shown that the performance of the disaggregation was higher when the *complexity* of the house was low. This is much the same results as seen in the experiments on the SmartHG dataset. The completely simulated houses that had a low *complexity*, preformed better, than the real house with a high *complexity*.

But in the SmartHG dataset was the *complexity* of the real dataset unknown since the amount of appliances that generated the *background consumption* is unknown. It is therefore hard to conclude if the poor performance in the real data is due to the fact that the appliances is unknown by the disaggregation model, or because the *complexity* of the house is greater for the real data.

To further investigate this was an experiment conducted on the *TV house dataset*, where the statistical disaggregation models where trained on the $\mathbf{H}_7$ dataset. The models trained did not contain the full knowledge of the house. The models was only trained to find a subset of meters in the house. Like for the experiments in Section 5.3.2, was the experiment conducted on all subset combinations. This creates a test where the amount of *background consumption* is artificially controlled. There is a lot of *background consumption* when there only is one appliance in the disaggregation model, and non if all seven appliances is in the disaggregation model. This can also be seen as a change in the *completeness* of the model.

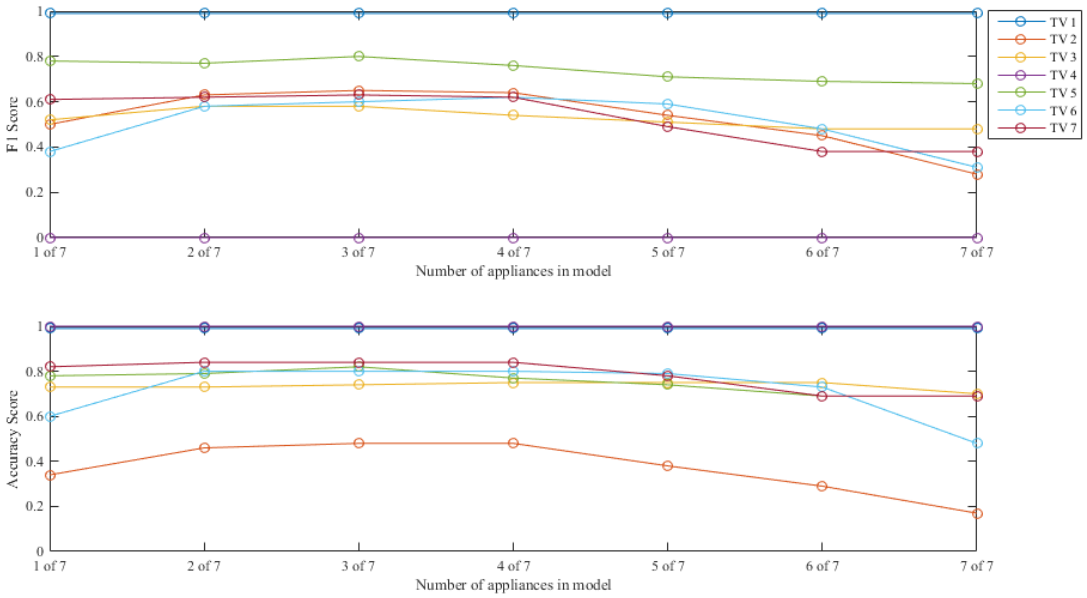


Figure 5.11: Completeness test of disaggregation of appliances in the TV house dataset

In Figure 5.11 is the result for each appliance shown. The result is an average of all tests at a given model *completeness*. It is seen from the figure that even though the disaggregation algorithm obtains more information about the complete house, it does not improve the recondition.

This indicates that 7 instances of the algorithms working in parallel, each disaggregating one meter, would obtain the same results as one model capable of seeing the house as a whole. It is worth mentioning that these tests are based on the FHMM algorithm and other algorithms might preform different in this aspect.

The results indicate that the performance is more depended on the *complexity* of the model, rather than the *completeness*.

5.4 Chapter Summery

The findings in this chapter indicates that the *complexity* of a dataset has a profound impact on the detection performance. A simple way to increase the performance in a house is to look at each phase in the house separately. This is one of the techniques used in the ECO dataset to improve the performance in the system. This does require additional knowledge of which appliances that are connected to which phases. This knowledge can be analytically extracted from the dataset, or supplied by the household.

It is also indicated that the methods based on the HMM does not greatly improve by knowing the entire house model, in relation to only knowing a sub part. This can indicate that a system that focus on detecting each appliance individuality, potentially can detect the appliance as well as one that tires to see the house as a whole.

The many problems that can occur when using small consumers and once that are similar in behaviour is nicely illustrated in the chapter. The *appliance interference* effect for similar appliances is greatly decreasing the overall performance. This makes the small consumers almost undetectable. In order to better this performance, more unique features is needed. This could either be higher frequency features, as discussed in Chapter 4, or by using reactive features. Furthermore it is shown that the best performance is obtained by training in the same environment as the solution will be deployed in. This is a subject that is also discussed by the developers of the Parson and Weiss algorithms [25], [21], where they discussed ways to train or fit the models to the desired deployment environment.

NILM As An Application 6

The last couple of years have the number of smart-meters installed in residential houses increased drastically [35], [36]. The motivation for installing smart meters in the different houses have been to better understand energy consumption, in order to better plan energy distribution and production, and increase security [2]. This on of steps in creating a modern smart grid. With the smart grid came a opportunity for non-intrusive load monitoring (NILM), since the equipment and infrastructure needed to measure the consumption and transfer the results through the Internet will be available in many households.

The applications proposed in many of the articles published about NILM focuses on energy management. Either it is for the electricity producers, that needs it to better predict consumption, or it is for the resident of the house that can optimize power usage to obtain savings [20]. Even though these are fine examples of the potential usages of NILM there might also exist an opportunity for the *service providers* in the smart grid to sell the NILM information. A instance that collect information with the purpose of selling it is defined as a "data broker". This chapter contains a small case study to illustrate how a *service provider* could obtain TV information from a city and act as a *data broker* to a local TV station.

6.1 The TV Viewing Habits Case

The case setting is a small town, that have a number of households and an electricity utility company supplying energy to the city. For this case data from the SmartHG dataset is used to simulate a small town with 4 houses. The houses selected for this case are 10, 13, 18 and 23 since they contain a TV that are relatively dominant, as discussed in Section 2.2.2.

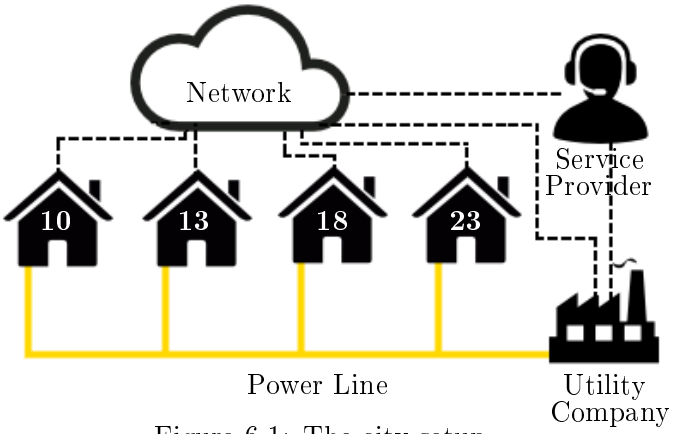


Figure 6.1: The city setup

The city setup is illustrated in Figure 6.1. The figure illustrates how the utility company have a *generator* role and supplies energy to the houses. All the houses have smart meters, and are informing the utility company and the *service providers* about the current consumption of each house. This is done using some network. This is a fairly simplified example of a smart grid, where only the *customer*, *generator* and *service provider* roles are used. The power line is illustrated as a single feeder, but could just as well be a complex *distribution* network. In this case is it assumed that the *customer* sends information about the consumption of the houses every 30 seconds, as in the SmartHG dataset. Even though many smart-meters today are capable of delivering information at this speed, it is more common to only get information each hour. This is mainly because the information is used to regulate distribution and production, which is a slow process that would not improve by faster update rates.

The *service providers* collect the information from the smart meters, and by using NILM can calculate statistical information about the town. In the fictive case is a local TV station interested in knowing what time a day the citizens of the town watches TV, in order to improve some aspect of there product.

6.2 Methodology

The aim of this case is to illustrate the potential of this kind of application for a *service provider*. Therefore is the data analysis conducted in a manner, that illustrates how a *service provider* could do it. This means that only the main meter information is used in the analysis, and the sub-meter information is only used to validate the results. The goal is for the *service provider* to disaggregate the TV signals from the main meter signal, in order to use it in some statistical analysis.

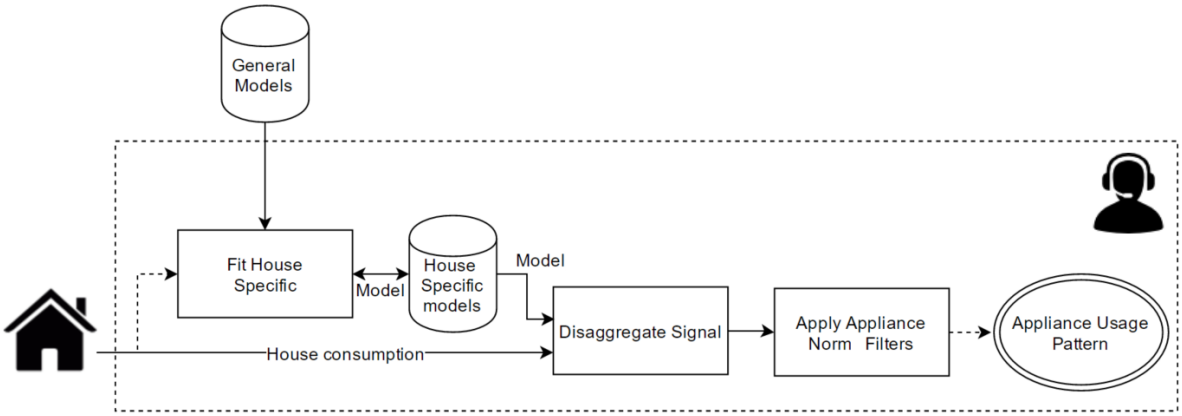


Figure 6.2: The disaggregation process

This process is illustrated in Figure 6.2. It is assumed that the *service providers* have access to a very general statistical model of the TV. This model can come from some outside database, or be one that the utility company or *service provider* developed themselves. The first step is to create a household specific model, that is a specialisation of the general model. This could be done with sub-meter data. To make the experiment more realistic is the training of the statistical disaggregation models done using only the accumulated data from the main meter. This is done

using the same method as developed for the Parson algorithm [25]. This method finds periods where an appliance is turned ON and OFF without any other appliances changing states, is detected. These ON/OFF single-event fragments are then mapped to specific appliances using the general models. When enough ON/OFF single-event fragments have been collected, can this information be used to train a more specific model. This can happen in a contentiously manner where the specific model is improved as more data is collected.

Using the specific models can disaggregation of the main meter signal be achieved. The statistical disaggregation models selected in this case are found using the factorial hidden Markov models (FHMM) algorithm. As discussed in Section 5.2.1, the disaggregation alone gives a very error prone signal. The signal is therefore feed through a *norm filter*. As discussed in Section 5.2.4 can the information about how a user normally would use the appliance be used to filter the disaggregated data to obtain better results.

The result of this process is an appliance usage pattern, that can be used by the *service providers* to derive various statistical properties.

The experiment is conducted with four weeks of data been used to train TV specific models for each house. The analysis for the TV station is conducted over a period of six weeks, that is not overlapping with the initial four training weeks.

6.3 Results

The viewing habits of the small town obtained by the disaggregation analysis is compared with the actual viewing habit obtained from the sub-meter data. In Figure 6.3 is the results shown for the first week. Plots showing the disaggregation of the entire period for each appliance can be found in Appendix I.

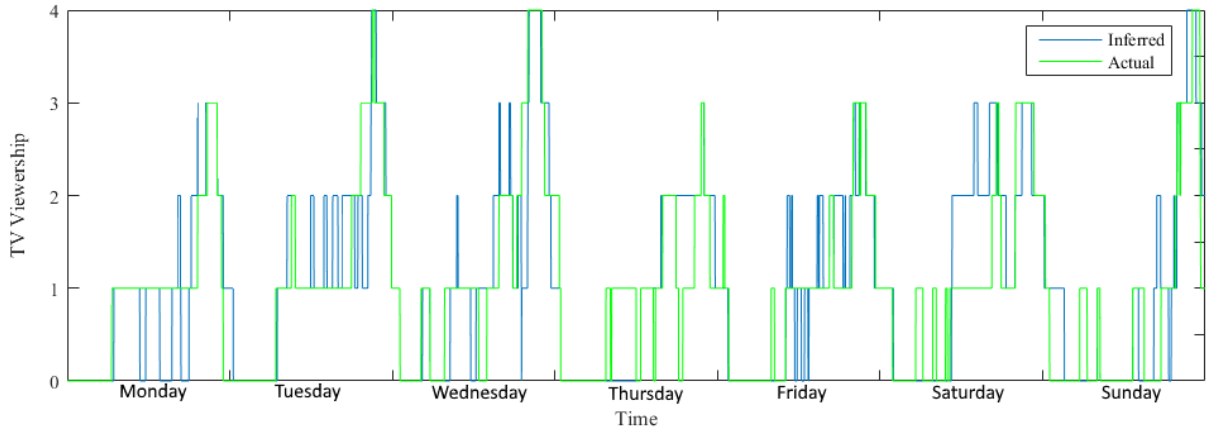


Figure 6.3: Viewership in week one

The figure shows the viewership, that tells how many in the city is watching TV at a given time. The green line is the actual viewership, where the blue is the viewership reported by the utility company. The results differer, as one would expect, since the disaggregation process in a environment with high *background consumption* is a hard task. The general trends that many are watching TV in the evening, and not that many in the morning, is still represented nicely. This

illustrates that the general trends of the viewership is maintained in the disaggregated signal, and this is commonly what is important for this kind of costumer statistics.

In Table 6.1 is the average number of viewers in 3 hour timeslots shown. The results shown in the table is the average of all 6 weeks, split up in the different weekdays.

		Time of day							
		00-03	**03-06**	**06-09**	**09-12**	**12-13**	**15-18**	**18-21**	**21-00**
Monday	Inferred	0.12	0.00	0.66	1.19	1.02	1.31	2.39	2.08
	True	0.13	0.28	0.98	0.87	0.68	1.28	2.49	1.72
Tuesday	Inferred	0.13	0.28	0.98	0.87	0.68	1.28	2.49	1.72
	True	0.19	0.09	0.39	0.88	0.73	1.24	2.23	2.26
Wednesday	Inferred	0.19	0.09	0.39	0.88	0.73	1.24	2.23	2.26
	True	0.09	0.30	0.58	0.79	0.77	1.42	2.72	2.11
Thursday	Inferred	0.09	0.30	0.58	0.79	0.77	1.42	2.72	2.11
	True	0.12	0.09	0.31	0.96	1.12	1.69	2.86	2.30
Friday	Inferred	0.12	0.09	0.31	0.96	1.12	1.69	2.86	2.30
	True	0.10	0.18	0.66	0.91	0.74	1.65	3.17	2.35
Saturday	Inferred	0.10	0.18	0.66	0.91	0.74	1.65	3.17	2.35
	True	0.28	0.17	0.95	0.77	1.39	1.36	2.26	2.10
Sunday	Inferred	0.28	0.17	0.95	0.77	1.39	1.36	2.26	2.10
	True	0.12	0.00	1.01	0.96	0.96	1.33	1.93	1.65

Table 6.1: Average viewership in a 6 week period

This illustrates that the true trends in the information is maintained. The results indicates that it is to some degree possible to obtain user information from the disaggregated data. The TV is a relative hard appliance to detect, since there are many different types. The success in this experiment is partly due to the fact that the TV was some of the main consumers of the houses selected, and they had a pure Type-I behaviour.

At the current state of NILM is the information only limited to a small amount of devices, since it is hard to detect other appliances than the top consumers in a household. One of the techniques shown to greatly improve the effectiveness of NILM is to switch from the sub 1 Hz sampling range to a higher sampling rate in the kHz range. In the future there might be smart-meters capable of delivering information at high frequency or the NILM techniques have advanced so it is possible to detect all appliances. This will greatly improve the possibility for the *service providers* to deliver very detailed user behavioural statistics. This information can be of interest for companies developing and selling electronics, since they are able to get accurate user reports about their products.

6.4 Chapter Summery

This case is a very simple example of how the *service providers* can extend their product range to statistical information. This information could potentially also be used to extend their product range to the *customers*. One example of this could be a fridge surveillance system that would send warning messages to the resident if their fridge stopped working.

The cases focused around getting information from a TV. This information was used to extract some general statistical properties about a small city's viewing habits. This is done by using the FHMM algorithm and a *norm filter*. The approach is relatively successful. This indicates that such approaches can be used to extend the *service providers* product range. All TV appliances was relatively large energy consumers in the house. This makes them easy to detect. For appliances that are small consumers it is unlikely that the approach will have as big an effect.

Discussion 7

Non-intrusive load monitoring (NILM) is an area of research that is gaining a lot of attention due to the more complex smart-grid structures. Some of the challenges in the smart-grid are that there is not just one *generator* of electricity, and a series of *customers*. There can be multiple of both, and some households contains solar panels, or other energy producing technologies, that makes them bout produce and consume energy at different times. This creates challenges for some of the NILM applications, as it can camouflage the true power consumption of the houses. The same problem can be created by battery devices, that can mask the an appliance power demand. Some of these problems can be observed by monitoring the data quality.

7.1 Creating High Quality Measurements

When monitoring the the data quality as in Chapter 2 a range of problems when collecting consumption data can be detected. When observing the collection of SmartHG data there were two error groups that reviled itself. The one was equipment failure. Both failure for a specific meter, or the server was detected. The second is the houses that exhibit masking behaviour. By looking at the information collected by the sub-meters and comparing it with the main meter data it is possible to detect the masking behaviour created by devices such as solar cells or backup batteries. This will often make it look like the house uses less energy, than recorded by the sub-meters.

By detecting the errors during the measurement period it is possible to correct some of the errors. This creates better and more clean measurements. By looking at the measurements *activity*, it is possible to find inactive and active areas. This can be used to find active and inactive periods of the devices. These periods can be used to create a more direct training of the statistical models used to do appliance disaggregation. Finding periods in the data where only one single appliance have an ON/OFF cycle, without any other appliance changes state, have shown to be an important training technique. This technique is mostly relevant if sub-meter information is not available. Utilizing the information obtained about the *activity* in the quality analysis can help find such spots.

One of the big challenges in a NILM application is the amount of devices that are operating on the same power line. A model that needs to disaggregate a signal containing many devices preforms worse than one where there only are few devices. It is therefore beneficial to look at the power usage of each phase individually. By using this separation can the *complexity* of the models be reduced greatly, which is shown to increase performance.

Having these points in mind when measuring data at the *customer*, can potentially create a better dataset. A dataset could also improve by supplying information about what devices that

are not measured in a research study. If a list of devices in the household was supplied with the measurements it is possible to better gesture about the *background consumption* created by the other devices. This kind of information would also be beneficial for methods based on unsupervised learning [29].

7.2 Capability Of NILM

In cases focused around selected appliances have NILM shown to be quite effective. But when seeing NILM as a tool to disaggregate a whole household appliances, there are still many challenges. Generally it is easy to detect the top consumers in a household. The big consumption increases the chances of an unique power draw pattern. The more unique power draw pattern the easier it is to detect the appliance. The majority of devices in a household only consume a small amount of the total energy. These devices are hard to disaggregate. This is because the power draw pattern for the devices is relatively similar. This causes the device disaggregation models to interfere with each other under the disaggregation process. The result indicates that sometimes some of the events from one device are classified to the wrong devices.

Much research have been focused on solving this problem. The general concept behind all the solutions are to find other features than the static power draw that can create uniqueness. One approach is to use the the difference between the voltage and current usage of the devices, known as the power factor. The power factor can indicate if a load is inductive, capacitive or resistive in nature. One other approach is to sample at high rates in the kHz range. This can supply information about transients, harmonics or noise that can be used to detect uniqueness.

The solutions most commonly used today focus on techniques that uses low sample rates, in the sub 1 Hz range. This is due to the fact that most smart-meters are capable of delivering data at these rates. A new type of meter must be developed in order to utilize the benefits that higher sample rates would create. This meter must either sample/transmit the data at higher rates, or process the data on sight and transmit the results. This is a costly process, since a new meter must be developed and installed in the households. By including this type of capabilities in the next generation of smart-meters it would be possible to advance the NILM research area. This would increase the businesses opportunities for consumer applications that utilizes NILM techniques.

7.3 NILM Deployment

In the current state can NILM only deliver a very general picture of the top consumers. What type of devices that is the top consumers is different from country to country. In countries that get hot in the summer, is the air condition at the top of the list. In countries that are cold and have a rocky underground uses a lot of power on electric heading. The difference in consumption culture around the world makes it hard to cross compare results between datasets. This makes it hard to determine which solutions are most suited for a desired country.

When validating only by looking at the F1 and accuracy scores the results seems promising. When an appliance have a 0.60 F1 score, it generally means that the ON periods was guessed correctly 60% of the time. This still leaves 40% of the ON time to be wrongly allocated. The wrong allocations can be both be ON events that are missed and ON events wrongly introduced.

This gives the same F1 score, but can make a difference in an application. This is illustrated in Figure 7.1, where two versions of the same event signal is shown.

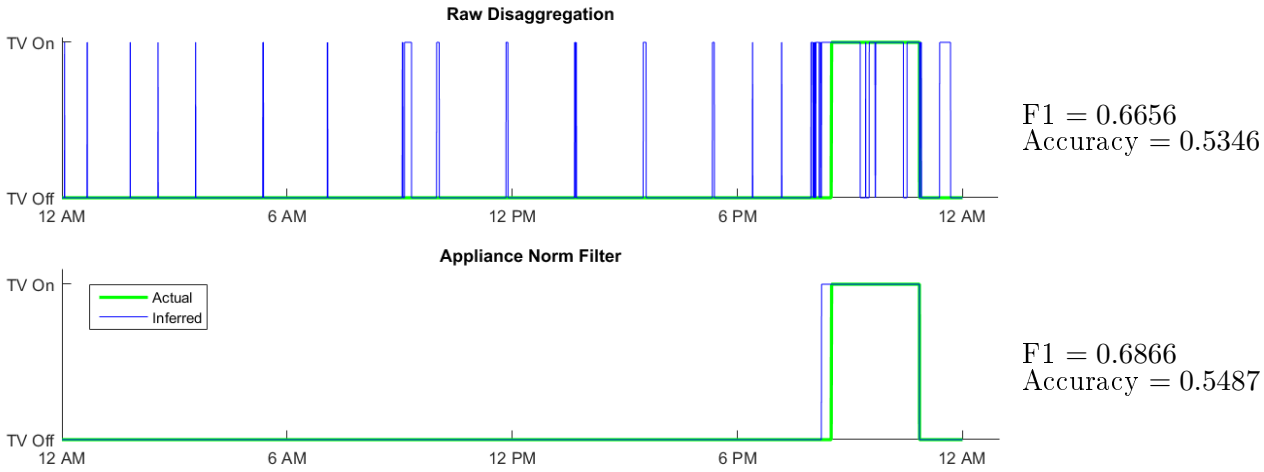


Figure 7.1: TV event detection

The "Raw Disaggregation" is before applying a simple *norm filter*. The second illustration is after the *norm filter*. The *norm filter* used is the same as in in Section 5.2.4. Visually the difference between the two signals is huge. The signal after the norm filter is much better then the signal prior to the norm filter. When looking at the F1 and accuracy scores of the two signals, the norm filter have only improved the signal with approximately 2%.

This raises the question: *is the F1 score a good indicator, and how high does the score need to be before it is application ready?* The F1 score seems to be generality adopted due to its simplicity and and acceptance in other areas of binary classification [33]. The F1 score required for an acceptable application can be dependent on many aspects. During the experiments, applications with a F1 score lower than 0.8 was generality influence much by interference from other devices. Where F1 scores higher than 0.8 could be filtered more easily, and gave more intuitive correct results.

The conclusion being that a functioning NILM solution, capable of detecting even little consumers, requires commitment. A successful system will require the metering company to develop, or at least install, a meter capable of high speed data collection or processing. Furthermore must the residents have an interest in helping improving the accuracy, in order to get the best results. This could be done by registering appliances on a smart-phone or tablet as suggested by Weiss [21].

7.4 NILM Privacy Concerns

Chapter 6 shows that it is possible to gesture about the TV habits of a household. Potentially can the usage pattern for any appliance be derived using NILM techniques. Researchers have shown that even the current movie shown on a TV can be deduced by looking at the power profile [37]. With NILM techniques it is more or less possible to track the behaviour of a residential household [38].

This kind of possible surveillance raises some privacy concerns. Some simple techniques can be used to mask the appliances from a NILM applications, to improve privacy. It have been shown that down-sampling the consumption signal makes it harder for the NILM algorithms. Quantifying the signal by rounding it to a multiple of a pre-defined quantization factor also helps mask signals. Averaging the consumption signal over a time period is also shown to have a masking effect [39]. Common for all these techniques are that they must be actively build in the meter and transfer mechanisms to mask the appliance data. This will only be the case if the utility companies have the same privacy concerns as the *customer*.

An other approach is to use non-intrusive load leveling (NILL) to mask the appliance signals. NILL is a technique where a capacitive device like a battery is used to store energy. This energy is used to camouflage power signatures from the household [40].

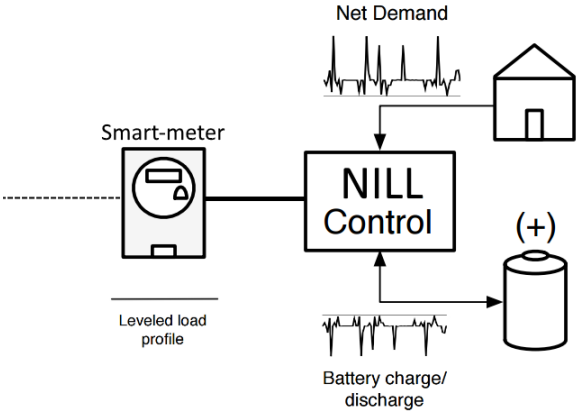


Figure 7.2: NILL illustration. Inspired by [40]

This principle is illustrated in Figure 7.2. A NILL solution can be installed after the smart-meter and can provide privacy, without the consent of the electricity utility providers.

At the current state of NILM is the technology not accurate enough to make reliable detailed surveillance of the household residents. But still advanced enough to raise some privacy concerns. There are many benefits with advancing the possibilities of NILM. Many services can be developed that can support and help the resident. Some of the possibilities are in saving systems and home automation. This systems could greatly benefit the resident of the household. Electronic companies could also benefit from getting accurate user statistics for there devices. But advancing the technology in this direction would also serve to expose the privacy of the home. This raises the questions about who are allowed accesses such sensitive information, and for what purposes?

Conclusion 8

This master's thesis seek to investigate some of the common approaches used for non-intrusive load monitoring (NILM), in order to determine the capabilities and limits. The SmartHG and the Electricity Consumption and Occupancy (ECO) dataset have been used in order to test some of the hypothesis raised in the study. It is shown that the quality of data collected form a smart meter over a lossy network is high. The greatest quality diminishing factors in the setup is shown to be malfunctioning equipment. Either on the reviving server or at the equipment in the house. It was investigated if a gap filling approach could improve the quality. Small errors on less than 5 samples is fairly easy to correct. Larger gaps is harder to correct. When using the corrected data in appliance recognition algorithms it is shown that more advanced methods of gap filling does not improve the performance compared to a simple linear interpolation. That said does the linear interpolation improve the performance, in comparison to no effort of correction.

It is shown that devices that have a relative high consumption is easier to detect than appliances that does only consume a little amount of energy. This is due to the greater chance of uniqueness for these types of appliances. The *completeness* and *complexity* of statistical disaggregation models where investigated. This showed that one of the factors that had a big impact on the performance was the number of devices in the environment. This was due to *appliance interference* between the devices. One way of minimizing the impact of *appliance interference* is to look at each phase individually, to minimize the number of appliances in each environment.

In general is a acceptable performance only obtained on the top consumers in the home. Therefore can the relative consumption of a appliance in compensation to the whole house consumption be used as a quality metric. In the report is a small case project illustrating how a *service provider* would be able to sell information about the usage behaviour of televisions in a city. It is shown that for this type of application is the performance acceptable, where detecting small devices like lamps and stereos is way more error prone.

The NILM disaggregation algorithms based on machine learning have a training period prior to deployment. In the training period is statistical disaggregation models learned, that makes the algorithms capable of appliance disaggregation. It is showed that the best performance is achieved when learning in the deployment environment. This is since model learned in a other environment is missing information about the *background consumption* in the deployment environment.

The NILM technology seems limited by the low sample rates expected from a smart grid. It is shown that the faster the sample rate, the better the performance. On approach that showed promise in improving the performance is the *norm filter*. The *norm filter* enforces the disaggregation with the rules of normal expected use of a appliance. This filters unrealistic event from the disaggregation.

Bibliography

[1] "The electricity grid: A history," 10/11/2011 2011. [Online]. Available: http://burnanenergyjournal.com/the-electricity-grid-a-history/

[2] J. A. Momoh, "Smart grid : fundamentals of design and analysis," pp. 1–29, 2012.

[3] N. S. Grid and C.-P. S. P. Office, "Smart grid," December 23, 2015 2015. [Online]. Available: http://www.nist.gov/smartgrid/

[4] C. Greer, D. A. Wollman, D. E. Prochaska, P. A. Boynton, J. A. Mazer, C. T. Nguyen, G. J. FitzPatrick, T. L. Nelson, G. H. Koepke, A. R. H. Jr, V. Y. Pillitteri, T. L. Brewer, N. T. Golmie, D. H. Su, A. C. Eustis, D. G. Holmberg, and S. T. Bushby, "Nist framework and roadmap for smart grid interoperability standards, release 3.0," pp. 123–146, oct 2014. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.1108r3

[5] D. Bonino, F. Corno, and L. D. Russis, "Home energy consumption feedback: A user survey," *Energy and Buildings*, vol. 47, pp. 383–393, 2012.

[6] N. Regnauld, "Generalisation and data quality," 2015-08-01T00:00:00Z. [Online]. Available: https://doaj.org/article/c22b25fbc7db468581f974b22d0f4640

[7] ISO 8402:1994, "Quality management and quality assurance – vocabulary," 1994-03-24.

[8] S. Kelling, D. Fink, F. A. L. Sorte, A. Johnston, N. E. Bruns, and W. M. Hochachka, "Taking a 'big data' approach to data quality in a citizen science project," 2015-10-27; 2015-11.

[9] ISO 19157:2013, "Geographic information – data quality," 2013-12-15.

[10] G. Pastorello, D. Agarwal, T. Samak, C. Poindexter, B. Faybishenko, D. Gunter, R. Hollowgrass, D. Papale, C. Trotta, A. Ribeca, and E. Canfora, "Observational data patterns for time series data quality assessment," vol. 1, pp. 271–278, 2014.

[11] M. Meijer, L. A. E. Vullings, J. D. Bulens, F. I. Rip, M. Boss, G. Hazeu, and M. Storm, "Spatial data quality and a workflow tool," 2015-08-01T00:00:00Z. [Online]. Available: https://doaj.org/article/2d94274972fe46a0b598b7f0569ee5b3

[12] A. Simader, B. Kluger, N. Neumann, C. Bueschl, M. Lemmens, G. Lirk, R. Krska, and R. Schuhmacher, "Qcscreen: a software tool for data quality control in lc-hrms based metabolomics," 2015-10-24. [Online]. Available: http://www.biomedcentral.com/1471-2105/16/341

[13] 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," 2014-04-15. [Online]. Available: http://arxiv.org/abs/1404.3878

[14] D. G. Manolakis and V. K. Ingle, *Applied digital signal processing : theory and practice*. New York: Cambridge University Press, 2011.

[15] P. J. S. G. Ferreira, "Interpolation and the discrete papoulis-gerchberg algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2596–2606, 1994.

[16] M. Marques, A. Neves, J. Marques, and J. Sanches, "The papoulis-gerchberg algorithm with unknown signal bandwidth," vol. 4141, pp. 436–445, 2006. [Online]. Available: http://dx.doi.org/10.1007/11867586_41

[17] D. J. Thomson, L. J. Lanzerotti, and C. G. Maclennan, "Interplanetary magnetic field: Statistical properties and discrete modes," *Journal of Geophysical Research*, vol. 106, no. A8, pp. 15 941–15 962, 2001.

[18] D. Kondrashov and M. Ghil, "Spatio-temporal filling of missing points in geophysical data sets," *Nonlinear Processes in Geophysics*, vol. 13, no. 2, pp. 151–159, 2006.

[19] A. Moghtaderi, P. Borgnat, and P. Flandrin, "Gap-filling by the empirical mode decomposition," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3821–3824, 2012.

[20] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[21] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, "Leveraging smart meter data to recognize home appliances," pp. 190–197, 2012.

[22] E. Kidmose, E. S. M. Ebeid, and R. H. Jacobsen, "A framework for detecting and translating user behavior from smart meter data," pp. 71–74, 2015.

[23] D. Srinivasan, W. S. Ng, and A. C. Liew, "Neural-network-based signature recognition for harmonic source identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, 2006.

[24] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd, "At the flick of a switch: Detecting and classifying unique electrical events on," pp. 271–288, 2007.

[25] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," pp. 356–362, July 2012. [Online]. Available: http://eprints.soton.ac.uk/336812/

[26] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The eco data set and the performance of non-intrusive load monitoring algorithms," pp. 80–89, nov 2014.

[27] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in nialm systems," pp. 3462–3468 vol.4, 2004.

[28] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," 2015-07-23; 2015-09-28. [Online]. Available: http://arxiv.org/abs/1507.06594

[29] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, "Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview," pp. 1175–1180, 2015.

[30] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, no. 2, pp. 245–273, 1997.

[31] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," 2012.

[32] W. Kleiminger, C. Beckel, and S. Santini, "Household occupancy monitoring using electricity meters," sep 2015.

[33] S. Makonin and F. Popowich, "Nonintrusive load monitoring (nilm) performance evaluation," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, 2014. [Online]. Available: http://dx.doi.org/10.1007/s12053-014-9306-2

[34] A. H. Kazmi, M. J. O'Grady, and G. M. O. Hare, "Towards low-cost energy monitoring," pp. 2965–2970, 2015. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-5888-2.ch289

[35] J. J. Spencer, "Denmark – all homes to have smart electric meters by 2020," vol. 2016, no. 04/04, 19/09/2013 2013. [Online]. Available: http://www.metering.com/denmark-all-homes-to-have-smart-electric-meters-by-2020/

[36] L. M. Maggs and A. Painter, "Smart energy gb response to decc quarterly report on smart meter installations," vol. 2016, no. 04/04, 31/3/2016 2016. [Online]. Available: https://www.smartenergygb.org/en/the-bigger-picture/about-smart-energy-gb/press-centre/press-releases/press-release-folder/decc-report-march-2016

[37] U. Greveler, P. Glosekotterz, B. Justusy, and D. Loehr, "Multimedia content identification through smart meter power usage profiles," *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, p. 1, 2012.

[38] G. W. Hart, "Residential energy monitoring and computerized surveillance via utility power flows," *IEEE Technology and Society Magazine*, vol. 8, no. 2, pp. 12–16, 1989.

[39] A. Reinhardt, F. Englert, and D. Christin, "Enhancing user privacy by preprocessing distributed smart meter data," *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*, pp. 1–7, 2013.

[40] S. McLaughlin, P. McDaniel, and W. Aiello, "Protecting consumer privacy from electric load monitoring," pp. 87–98, 2011. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046720

Appendix Overview

Appendix A : Overview of available data from SmartHG.

The amount of meters that sends information in a given hour. This is a series of plots that shows how many hours that 100% of the devices was sending information, and how many hours only 90% send information and so on. This information tells how the general *sample availability* is.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/Sample%20Availability`

Appendix B : The Quality vector of each house

The quality and *sample availability* for each of the house in the SmartHG dataset. The analysis is made with a one hour analysis period. A list of devices for each house is also shown.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/Quality%20Vector`

Appendix C : Implementation of Papoulis-Gerchberg (P-G) algorithm

The Matlab implementation of the P-G algorithm used in the project.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/P-G%20Algo`

Appendix D : Implementation of Weiner gap filler algorithm

The Matlab implementation of the Weiner gap filler algorithm used in the project.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/Wiener%20Algo`

Appendix E : Implementation of Spatio-Temporal filling (SSA) algorithm

The Matlab implementation of the SSA algorithm used in the project.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/SSA%20Algo`

Appendix F : Implementation of Envelope gap filler algorithm

The Matlab implementation of the Envelope gap filler algorithm used in the project.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/Env%20Algo`

Appendix G : Implementation of empirical mode decomposition filling (EMD) algorithm

The Matlab implementation of the EMD algorithm used in the project.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/EMD%20Algo`

Appendix H : Pie-chart of all houses appliance consumption and *background consumption*

A pie-chart that shows the distribution of energy for each house is shown in this Appendix. The values is relative to the total house consumption.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/SmartHG%20Consumtion%20Pie`

Appendix I : Detailed results from the disaggregation of TV's in Case study.

This shows detailed plots from the results of the case study. The results is the plots prior to the *norm filter*. This illustrates the TV signal as the factorial hidden Markov models (FHMM) algorithm would produce.
`https://github.com/RuneHeick/SpecialeAppendix/tree/master/Case%20Study%20Plot`