Non Intrusive load monitoring

Rune A. Heick 11061

Aarhus University, Department of Engineering

March 15, 2016

Abstract

Write an abstract Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Abbreviations

ECO	Electricity Consumption and Occupancy. 24–26, 30, 35
EMD	empirical mode decomposition filling. 10
FHMM	factorial hidden Markov models. 21, 27, 32, 36, 39, 42
HMM	hidden Markov models. 19–22, 27
IMF	intrinsic mode functions. 10
NILM	non intrusive load monitoring. 6, 16, 17, 19–21, 26, 35, 36, 42, 44
NILMTK	NILM-Toolkit. 20, 22
P-G	Papoulis-Gerchberg. 8, 9, 12, 14
SSA	Spatio-Temporal filling. 9
UDP	user datagram protocol. 2

List of Figures

List of Tables

Contents

Introduction 1

Write an Introduction Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.1 SmartHG Dataset

Introduce the dataset, evt. statistics Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Data Quality 2

Various projects today is 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 are 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 [5]. Looking at the quality of the data can help you 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"* [6].

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, and 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 is citizen science projects, where it is the citizens who collect the data, and not the researchers [7]. This enables researchers to gather enormous amount of data, but they are no longer in control of the conditions the data is collected in, which introduces errors and other quality decreasing factors.

Non intrusive load monitoring 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 meters 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 a lot of packets 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 is 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 is 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 [8]. Method for defining quality in time series data have also been developed [9]. 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 [10]. 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 is dataset 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 [11].

In the article *Taking a big Data approach to data quality in a citizen science project*[7] 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 smartphones. 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.

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. The sample availability 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 since a good data set must contain both areas with activity and areas without.

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 are 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 measurable 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.

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.

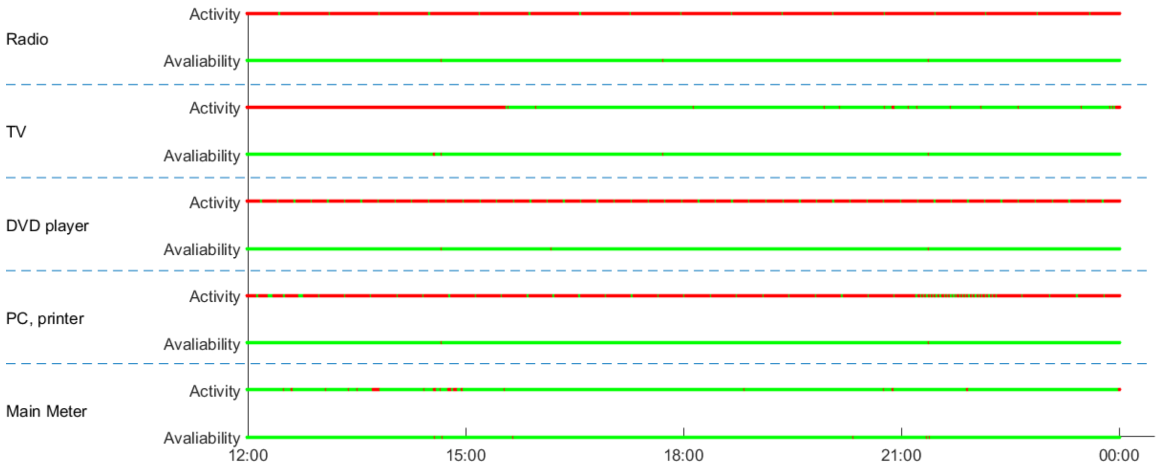

Figure 2.1: 12 hour overview of house 10

On 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 availability is shown as a line where green is indicating that a sample is received as expected, and red shows a missing sample. On 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 defined as a change in the signal, from prior values. From this we can see that the resident have a lot on 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 availability quality of the data is assessed. The 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.

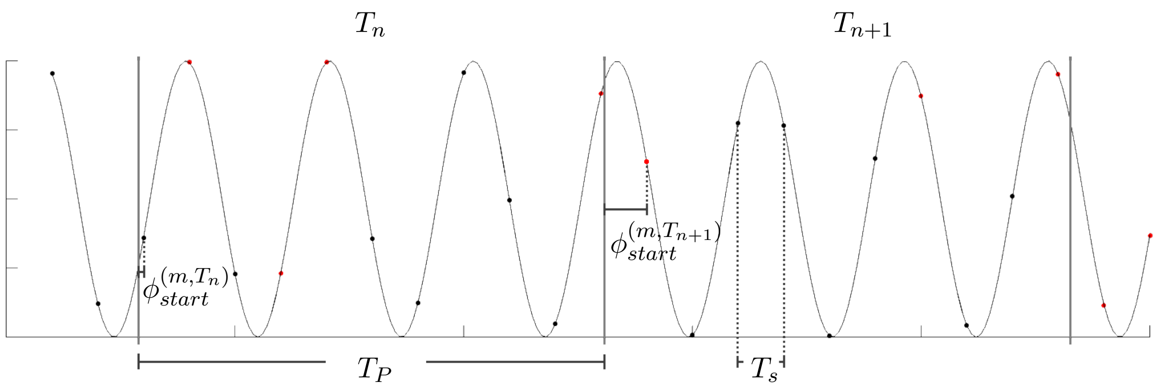

Figure 2.2: Illustration of availability analysis

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. On figure 2.2 a 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 on 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 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.

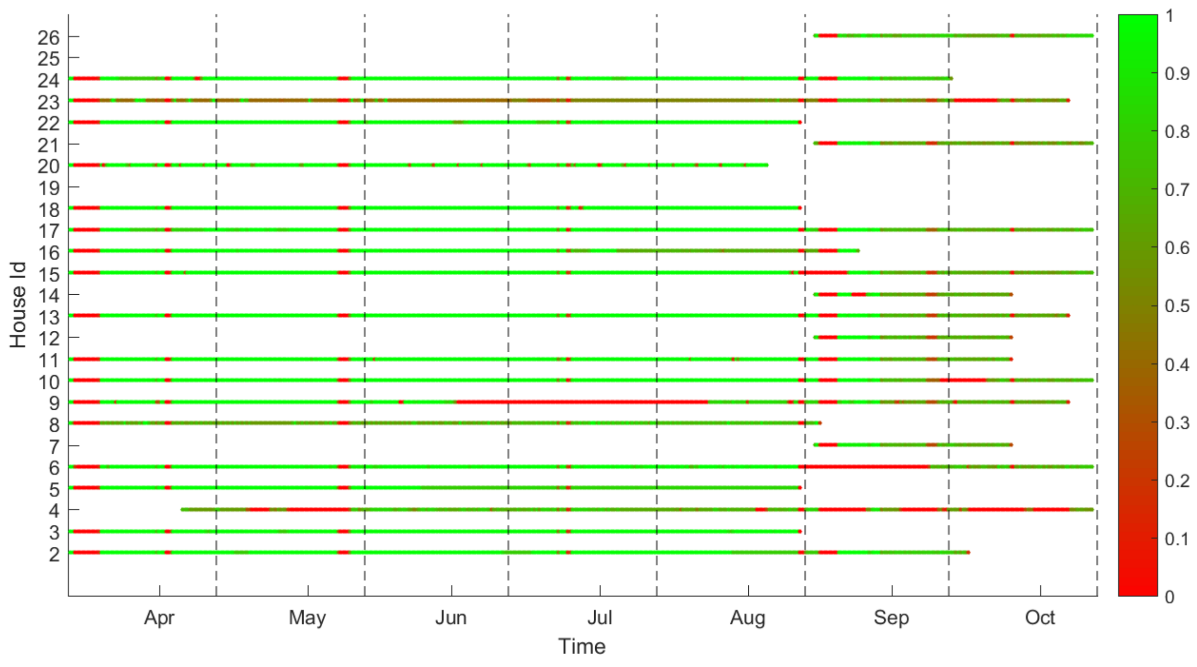

Figure 2.3: Quality of houses in SmartHG project

On figure 2.3 is the availability quality shown for the 25 houses in the SmartHG project. When seen on this scale with a 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.

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. On 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.

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 non intrusive load monitoring (NILM) application in training scenarios. We define activity as area in the data where there is change as described in equation 2.5.

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

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

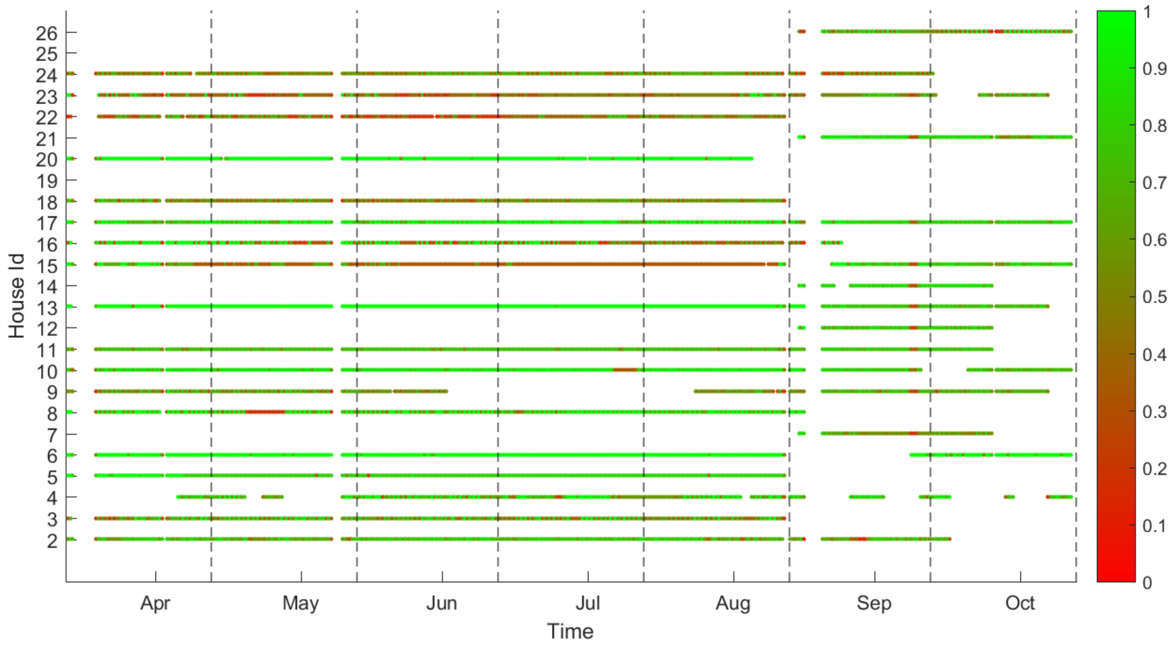

Figure 2.4: Activity of houses in SmartHG project

The activity shown on 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 read 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.

Gap Reconstruction 3

One of the more common problems in citizen science projects is 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 to deal 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 then gap filling is not recommended[12].

In the case of the SmartHG project the data can be seen to have a part that is depended on the previous and future data plus a stochastic part that are determined by the user and the appliance. Due to the stochastic part a perfect reconstruction is not possible, but 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 it is important to keep the frequency spectra as intact as possible, for other it is 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 it will be briefly described what makes the five algorithms unique, and how they work.

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[13]. 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 is 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 [14]. This approach is fairly good at estimating the true value of M, but it is time-consuming.

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 knowledge, it estimates the missing samples [15]. 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.

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 [16].

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.

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 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 [9]. When the new maxima and minima are found the points is connected by using spline [17].

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.

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 a empirical mode decomposition on a signal with a gap in is a non trivial process and can introduce errors [17].

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

On 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 figure 3.1.

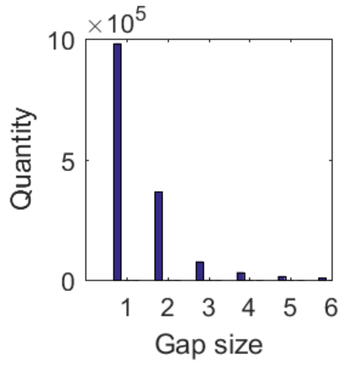

Figure 3.1: Gap quantity

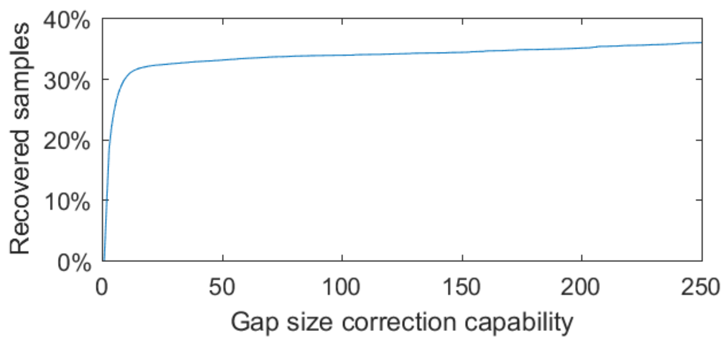

Figure 3.2: Error recovery capability

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 a 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.

On 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 a 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 have behaved in relation to the samples prior and post for the gap. Therefore is the samples prior and post for the gap called knowledges, since it grants the knowledges 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.

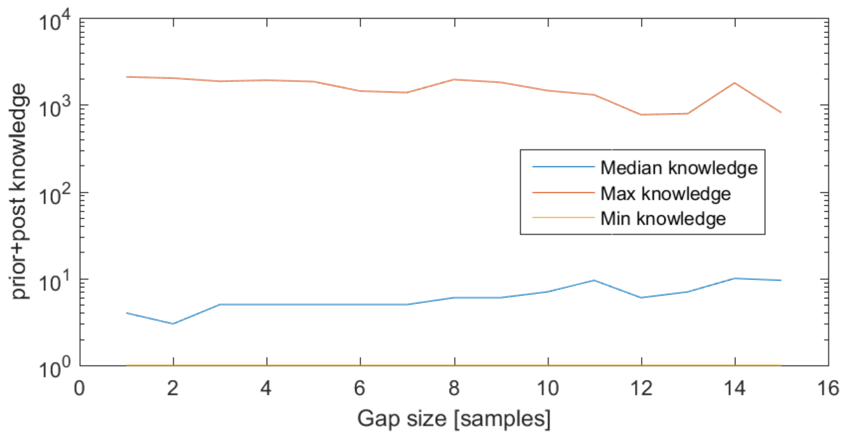

Figure 3.3: Available samples for recovery

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. On 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 on section 3.1 have been tied on the SmartHG dataset. First have 6 unique error free areas in the data been found, and a 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 3 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, where we see that the power of the jitter is the same, 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.

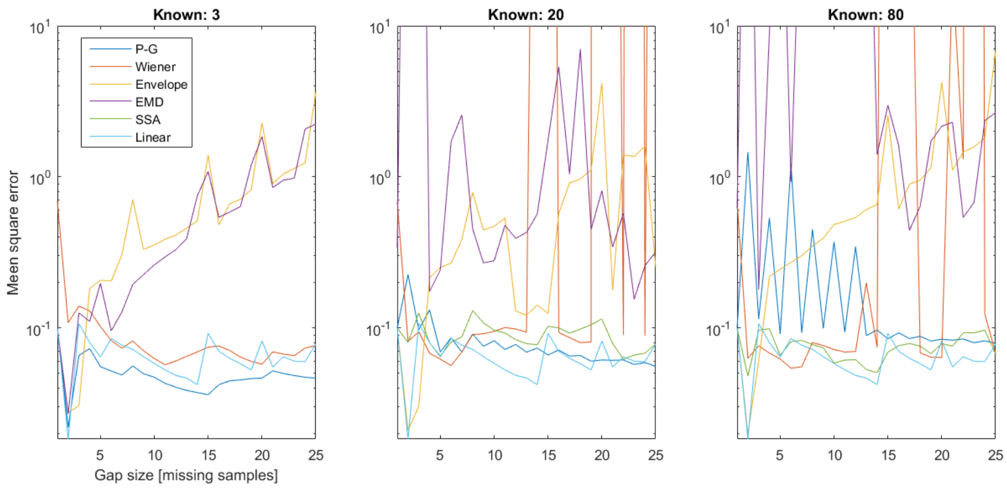

Figure 3.4: Sample comparison of the reconstruction methods

On 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 on the figure the best reconstruction is 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 information or knowledge we have 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 rate. 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 grows the more likely is it the a stochastic event will bring false information to the model. This can be illustrated as on figure 3.5.

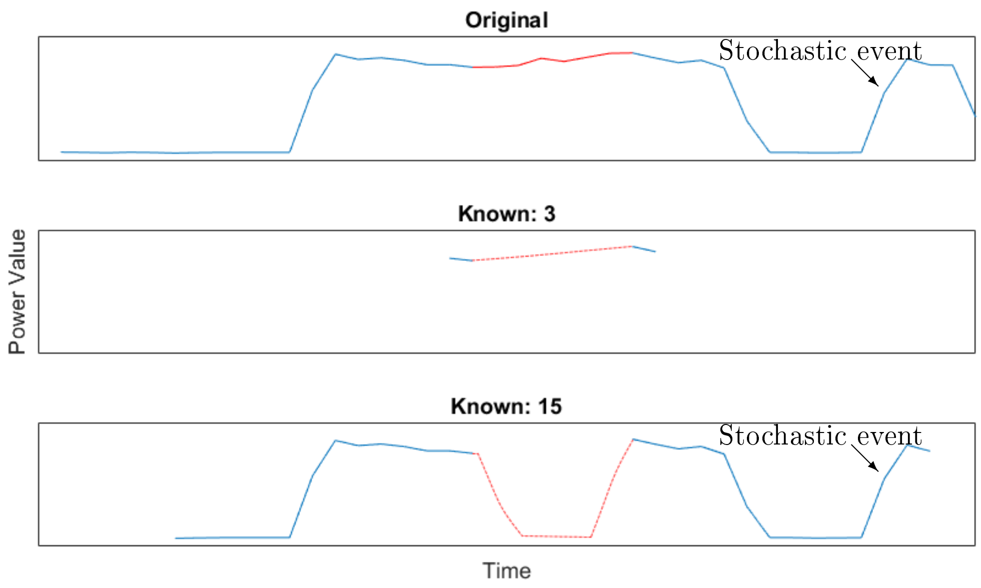

Figure 3.5: Stochastic effect on prediction.

On the figure 3.5 is shown a original signal with a gap, indicated by the red area. The true value is illustrated by the full red line in the original signal. The two subsequent graphs is the reconstruction, 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 we are able to make more complex estimates of the signal. On the figure it is shown how when we have a knowledge of 15 we also see the stochastic event. 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 basecase, we can conclude that the signal is quite stochastic and not as periodic as one would hope.

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 whit 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 of the different frequencies was compared.

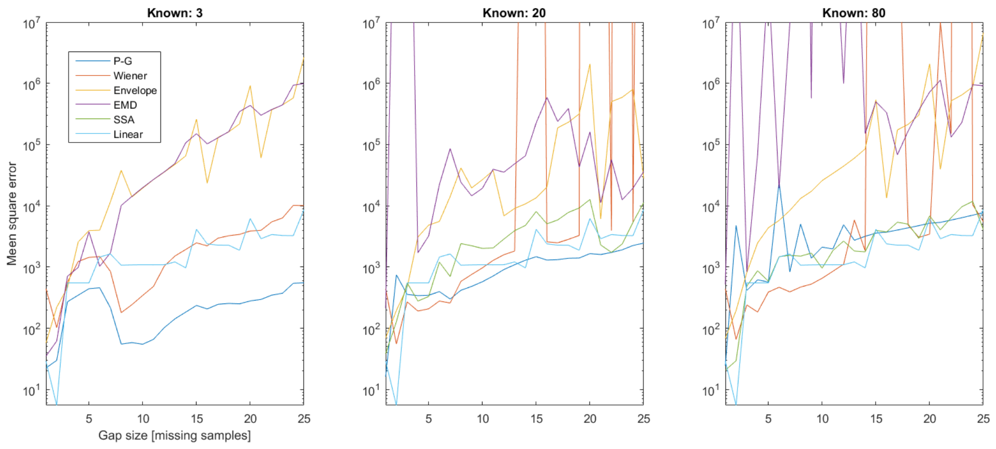

Figure 3.6: Frequency comparison of the reconstruction methods

As shown on 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 on figure 3.7.

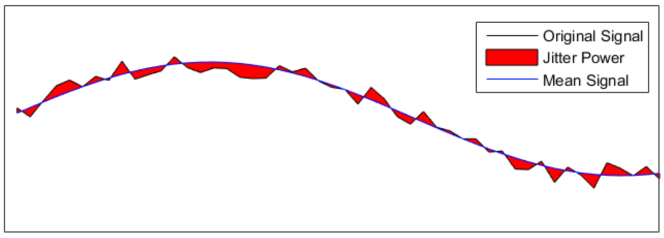

Figure 3.7: Jitter Power Illustration

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

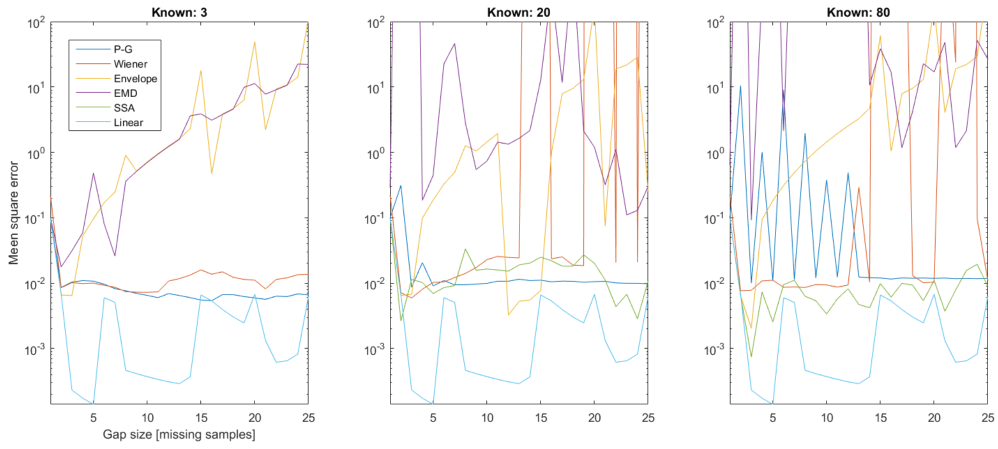

Figure 3.8: Jitter Power comparison of the reconstruction methods

As seen on figure 3.8 does non of the methods model the noise very well, as 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 [1].

Appliance Recognition 4

Appliance recognition and load disaggregation is 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, and 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 [18]. 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. Sometime the actual consumption of an appliance is not as important as when it was used, and for how long. This is a simpler task than guessing the true consumption, and is for some applications a better approach. As an example is this kind of information sufficient if we want to track the activity of a elderly person in a assisted living situation.

4.1 NILM Concepts And Challenges

The aim of NILM is to partition the hole 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[1]

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, and 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 is 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 on figure 4.1.

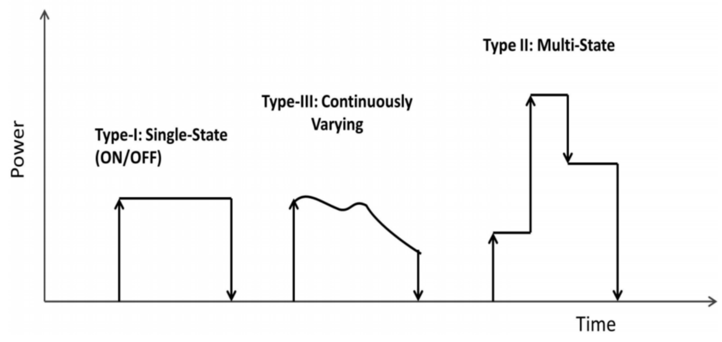

Figure 4.1: Appliance types. Source [1]

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 is often using very little energy, and must not be turn off at will.

Type-I and Type-II are very used, and can cover all most any appliance. In the area of event detection are the most common approach to see all appliances as Type-I appliances and detecting on/off events. Most appliances today is Type-II appliances due to 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 two. But most people don't 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 by using this 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 on figure 4.2.

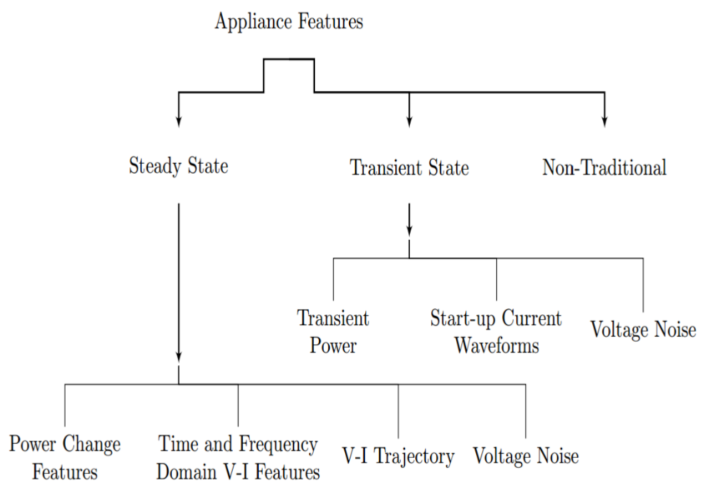

Figure 4.2: Feature types. Source [1]

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 [19].The trajectory between voltage and power usage is a god 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 a extremely high sampling rate [20].

The transient state of an appliance is a short state that comes when a 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 [1].

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 normal smart meters that are providing the data. For a smart meter a 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 there can be selected for the training. 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. This approach is fine for a small amounts of appliances, but for large databases and many appliances in use at the same time would this take far to 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. 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 appliance need to be available for the training process. This means that a sub meter 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.

Most algorithms are based on clustering techniques and hidden Markov models (HMM) . HMM have proven very effective when classifying Type-I and Type-II appliances. This is due to its ability to model state changes as a 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 is also showing great promise, and is a topic that is currently in focus by many NILM researchers. More on this in section 4.2.

4.1.3 NILM Challenges

There are many challenges in NILM . One is the Type-III appliances. Due to there varying, and at time 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[1].

A other problem is the "heavy consumer problem". In a house there are some appliances that require a lot of energy when they are on, such as stoves, air conditioners and refrigerators. These is often refereed to as the heavy consumers. Then there is the appliances that consumes very little energy like laptops, DVD players and routers. The heavy consumer problem is that the heavy 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 heavy consumers when during NILM applications [21].

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 library's, 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 easer share and compare algorithms in the future[21].

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[4].

Deep neural networks have last yeas improved the area of computer vision and speech recognition remarkably. NILM researchers are starting to look in to if similar techniques can be used to improve data disaggregation. Neural nets 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 net can out preform 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 heavy consumers[22]. Current research is looking that its potential in solving the heavy consumer 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 a 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 [23]. This paper will therefore focus on the supervised approaches.

4.3 Recognition Methods

Methods based on HMM is 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 refereed to in literature, and is often used as benchmark algorithms. All of the algorithms is based on HMM or clustering in different configurations. In the following subsections will the algorithms be briefly introduced.

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 [2]. It is assumed that there excites 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 have 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 series 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 on figure 4.3. It is illustrated how several hidden states from different chains contribute to the output.

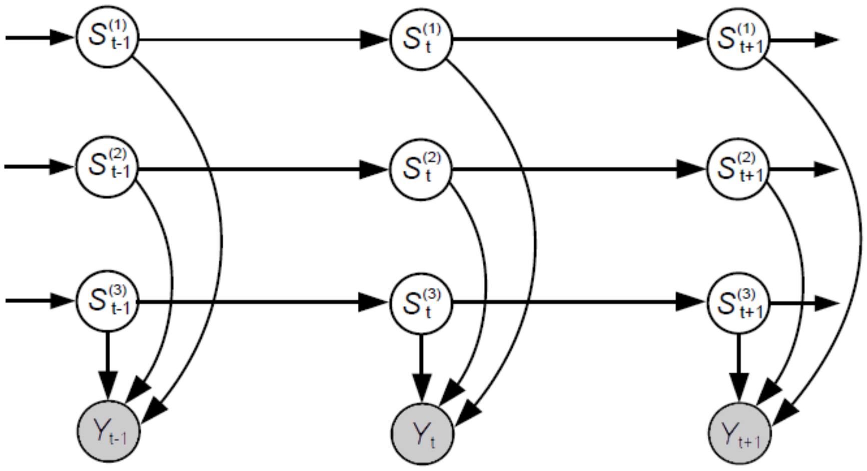

Figure 4.3: Factorial hidden Markov model illustration. Source [2]

As a extra constrain must each hidden state, in 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 is seen as Type-I or Type-II appliances. There is trained one chain for each of the appliances plus one that is called the other chain. This 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 [21].

4.3.2 Parson

The Parson algorithem is a variant of the Kolter and Jaakkola algorithem[24]. 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. This creates a model structure that can be graphically illustrated as on figure 4.4[3].

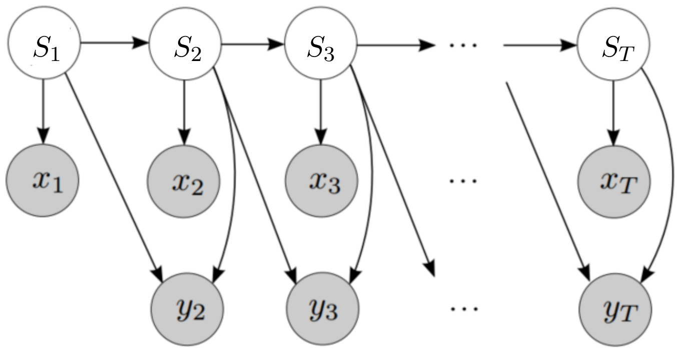

Figure 4.4: Parson hidden Markov model structure. Source [3]

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 mean constant power draw of the state. This is used to filter out the appliance if the power draw 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 Gaussian. 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 draw of a appliance is modelled as Gaussians. The step Power Y_t, can also be modelled as a Gaussian that is created from the change between two states as shown in equation 4.7.

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

To model the constraint that the meter only can be on if the mean power draw is present in the signal is a additional constraint added to the emission probability. This constrain is mathematically described in equation 4.8.

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

Where x_t is the measured power consumption and w_{S_t} is the appliance power draw constraint. This makes the probobility 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 paper 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[25]. 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[25]. 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 [4][26]. The dataset consists of six houses 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. On the main meter is both real and reactive power measured, where on the sub-meter it is only the real power is measured.

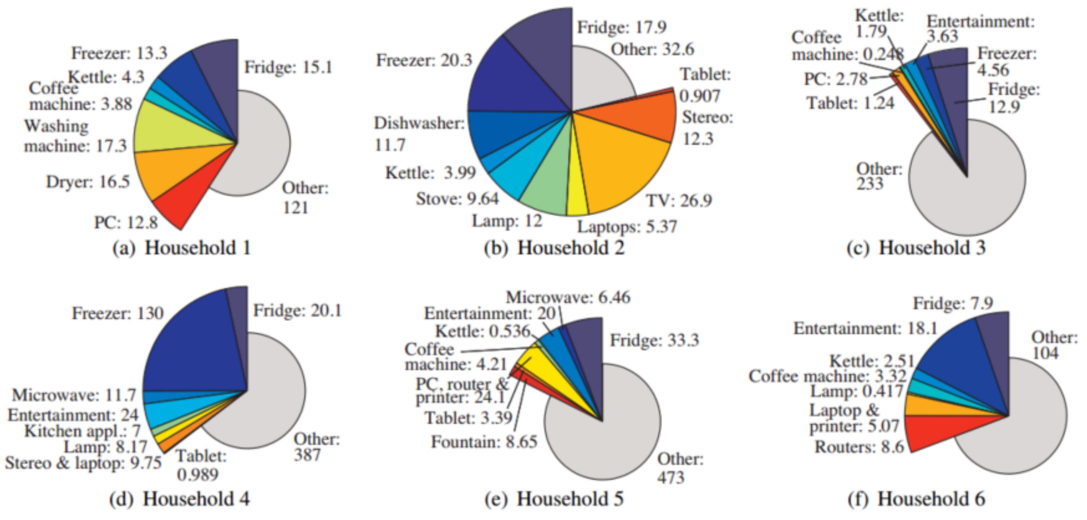

Figure 4.5: House energy distribution. Source [4]

On 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 heavy 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 a 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 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. Fore 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. 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 on equation 4.14.

Five appliances have been selected from the ECO dataset house 2. The appliances have been selected so there is some heavy and some little consumers. The appliances selected is: 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 rate. To determined the sensitivity, on the sample rate, is an experiment conducted where the dataset is downsampled to simulate different sample 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 shows the average score of the different algorithms when recognising the five appliances.

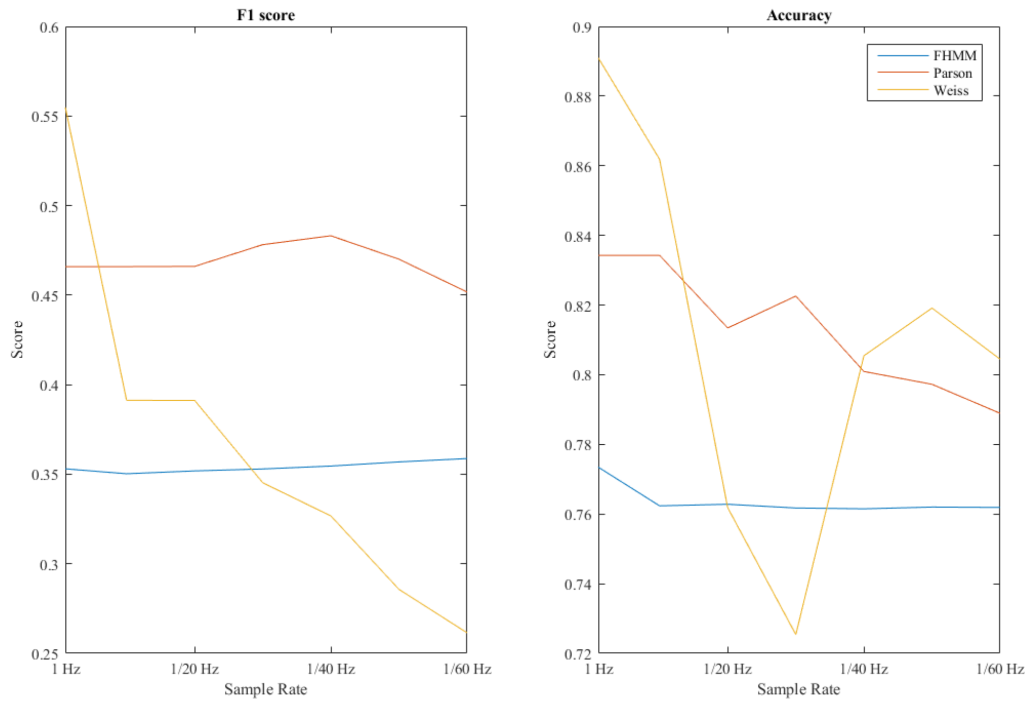

Figure 4.6: Algorithm validation at different sample graduality.

Both the accuracy and F1 score is shown. It is shown that for the Wiess clustering algorithms is the overall performance is decreasing for both scores. This indicates that is it both harder for it to determine when a 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 is easier to find than others. The accuracy score for the different appliances is shown in figure 4.7.

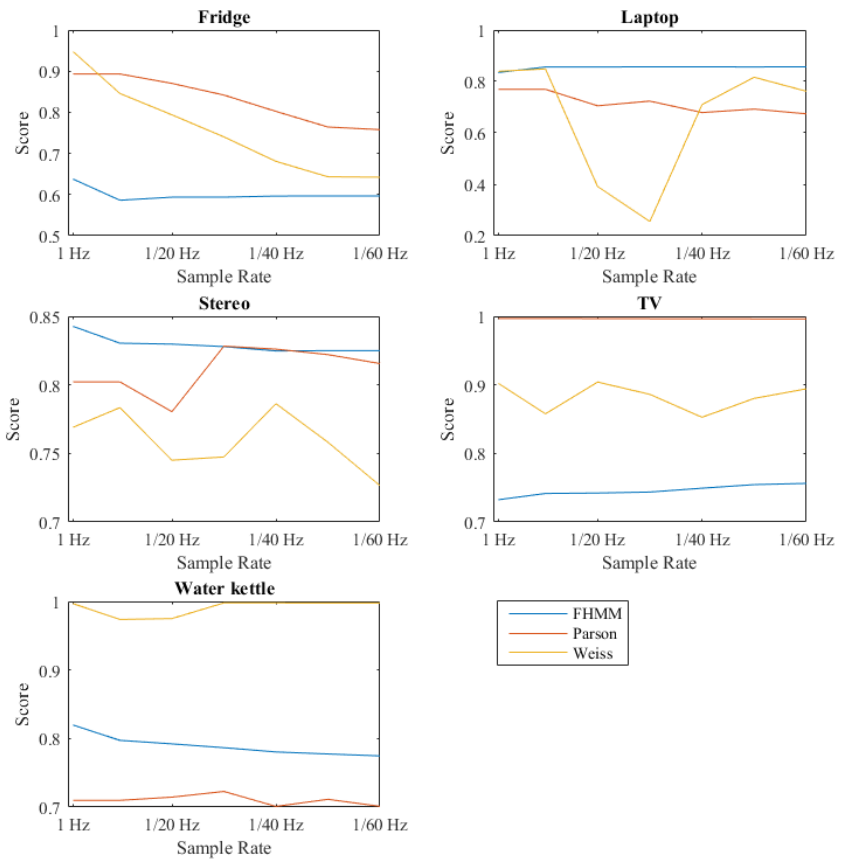

Figure 4.7: Appliances accuracy score

Since the accuracy score tells how well the 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 minuses 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 detects them.

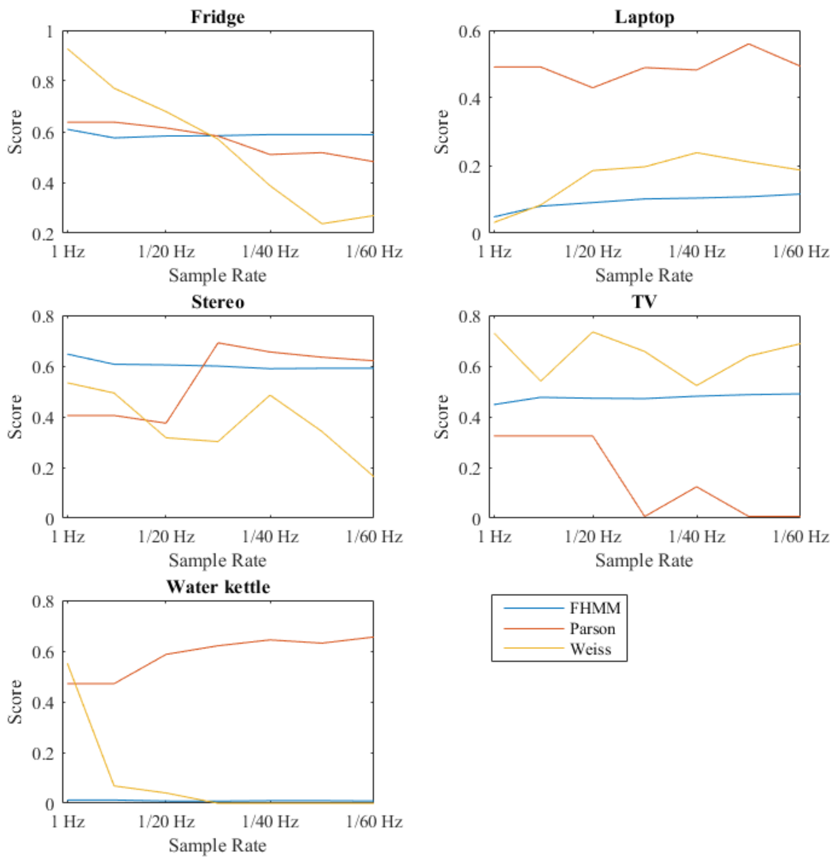

Figure 4.8: Appliances F1 score

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 a experiment conducted where the error rate is increased, and the F1 and accuracy score is measured.

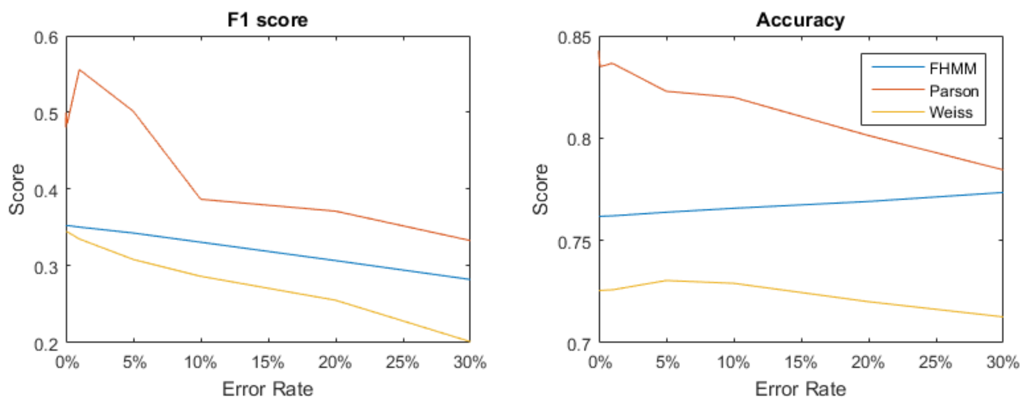

Figure 4.9: Algorithm validation at different error rate

The experiment was preformed by randomly introducing errors with a probability given by the error rate. 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 preformed quite well in comparison to the others. The envelop method did also preform 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.

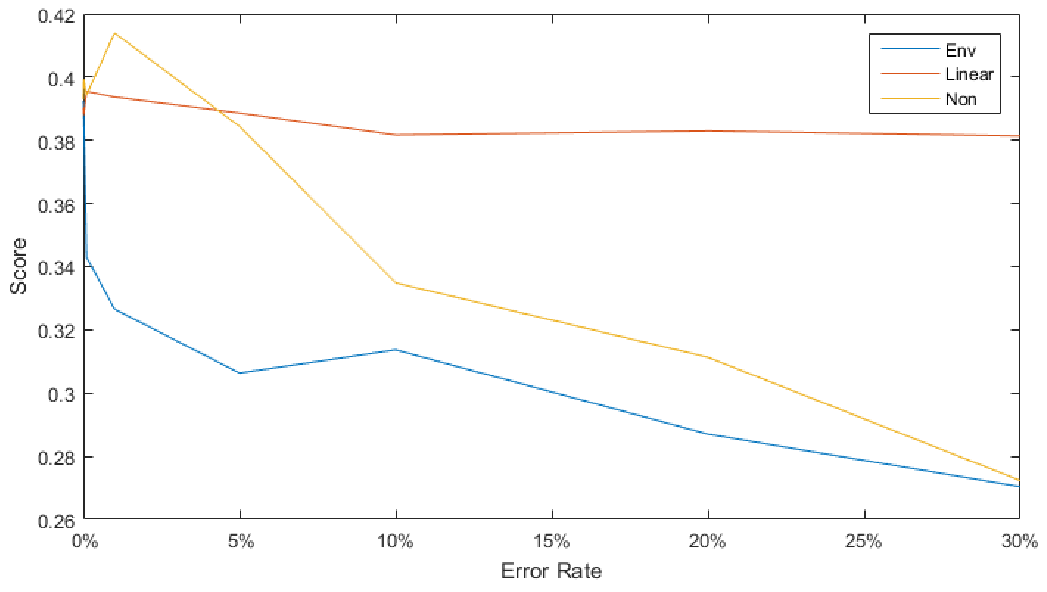

Figure 4.10: Average recognition after recovery of the 3 methods

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.

On 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, witch 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.

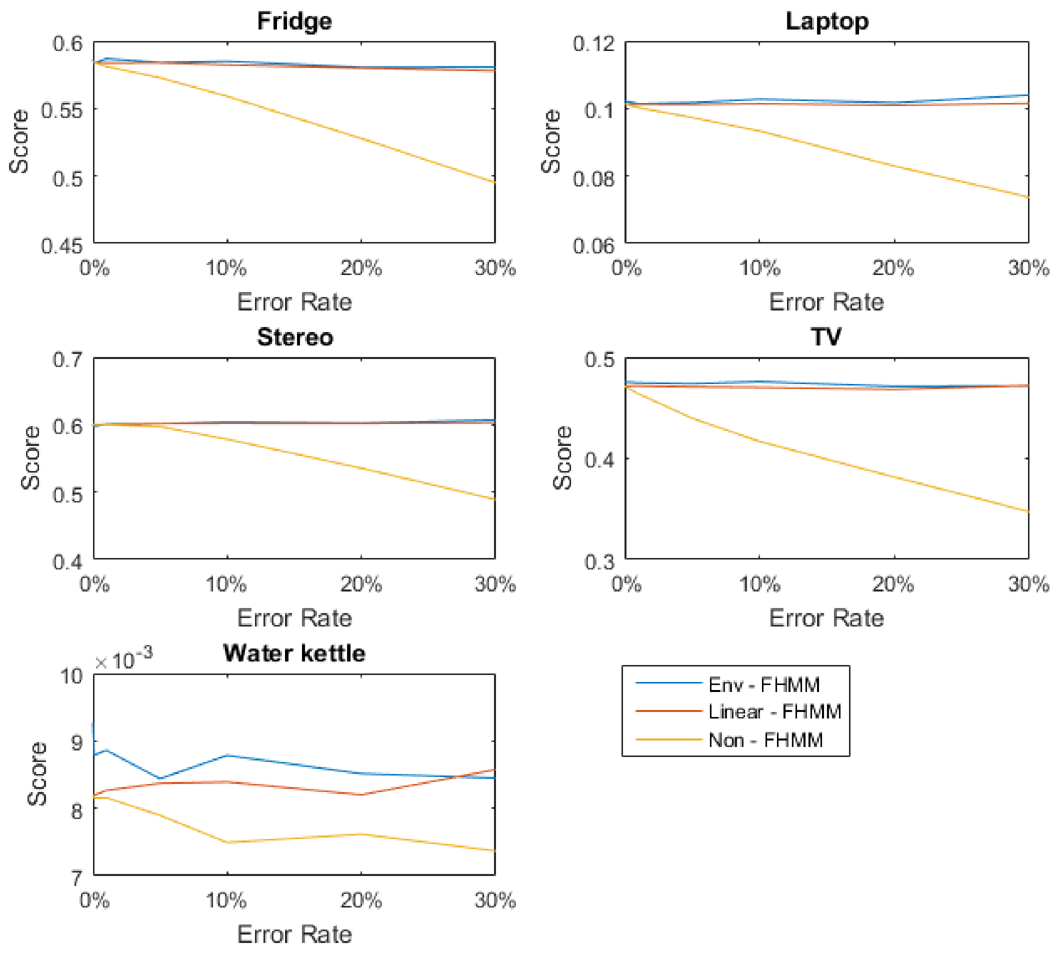

Figure 4.11: Recovery in FHMM algorithm

On 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.

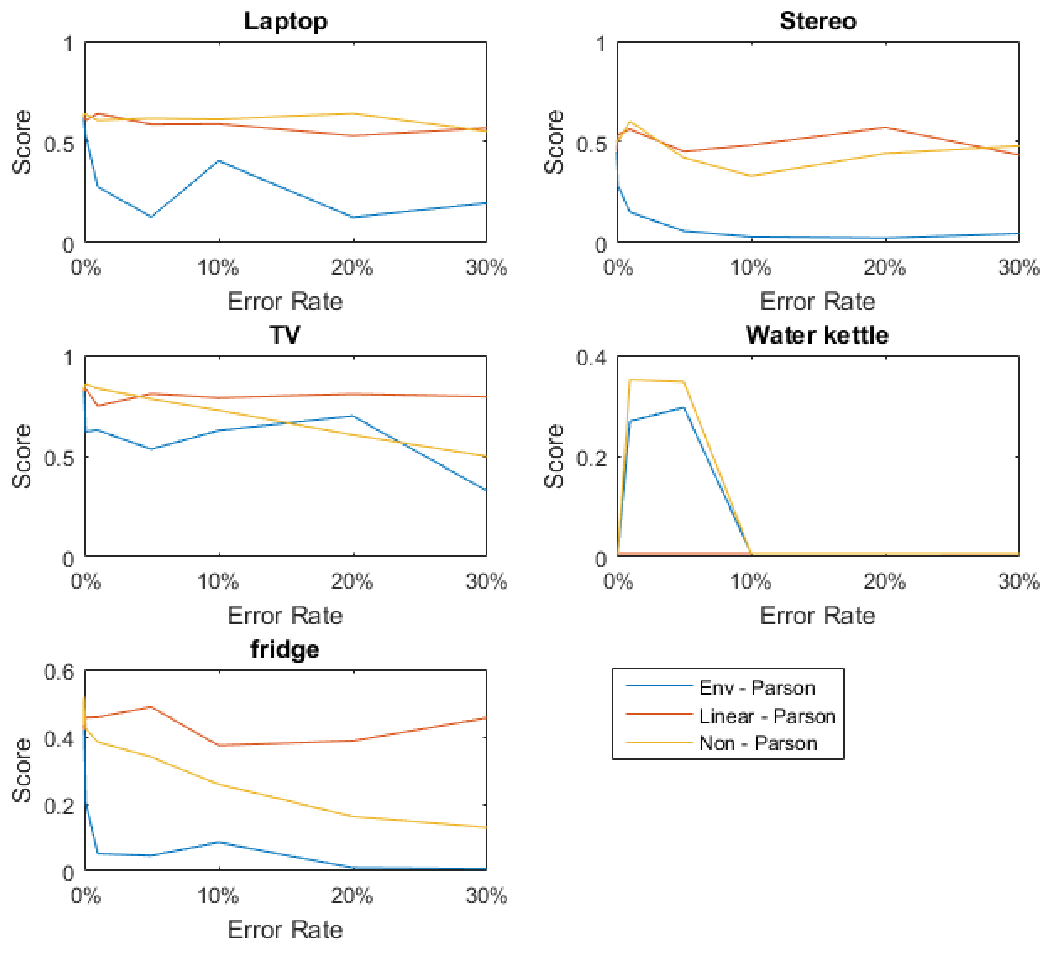

Figure 4.12: Recovery in Parson algorithm

On 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 does looks at jumps in the power, and usages a mean filter to only show the significant jumps. This behaviour can explains why a linear reconstruction gives almost no performance gain compared to no reconstruction for some of the devices.

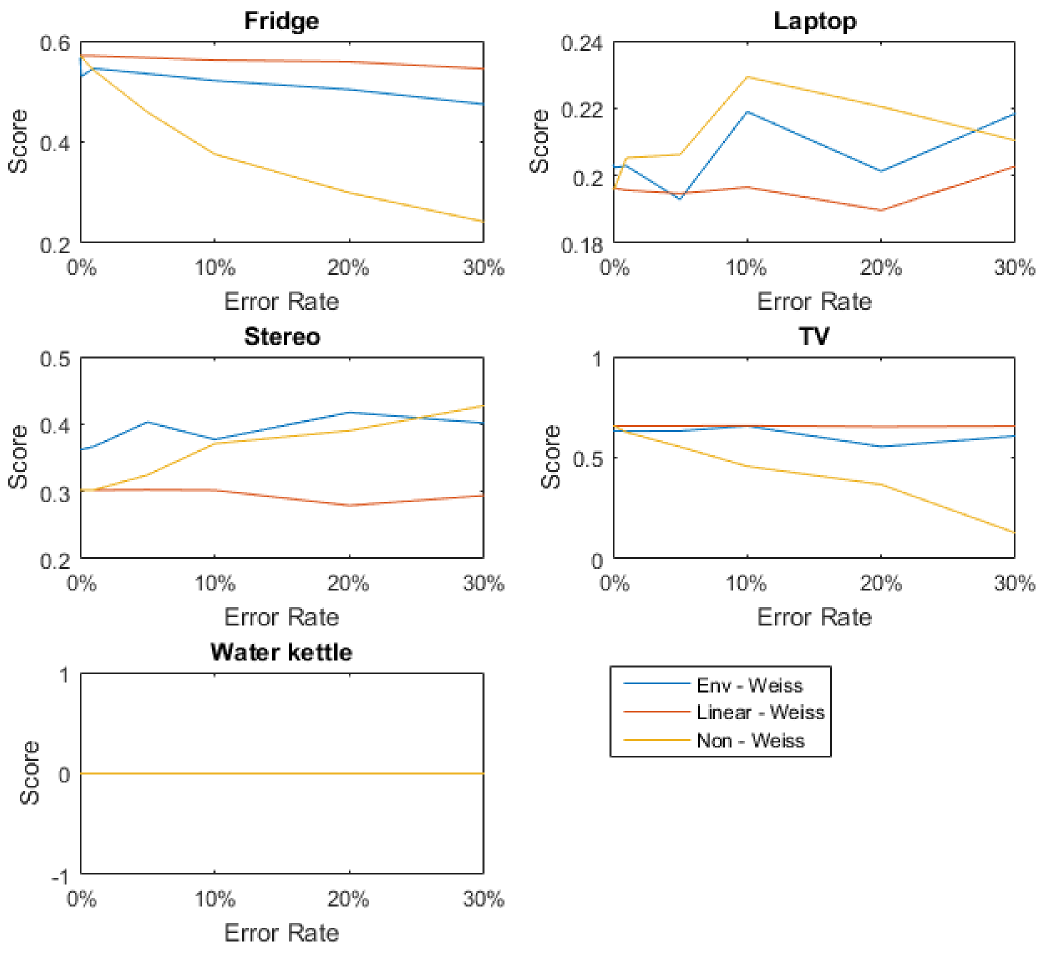

Figure 4.13: Recovery in Weiss algorithm

On figure 4.13 is the result from the reconstruction with the Weiss algorithm shown. For the Wiess 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.

Environment Influence 5

Not just the sample rate and error is 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 is deployed in. An example of an environment parameter can be number of known devices, number of unknown devices, number of simultaneously active devices or number of training days. In order to investigate some of the environment parameters effect the SmartHG dataset is used.

5.1 Challenges In The SmartHG Dataset

The Electricity Consumption and Occupancy (ECO) dataset, 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 6 month. Each house have only a small number of sub-meters. The sub-meters is mainly placed on little consumers such as televisions and stereos, which presents an interesting challenge for load disaggregation. The SmartHG dataset contains both the aggregated data, and the instantaneous power usages of the different households.

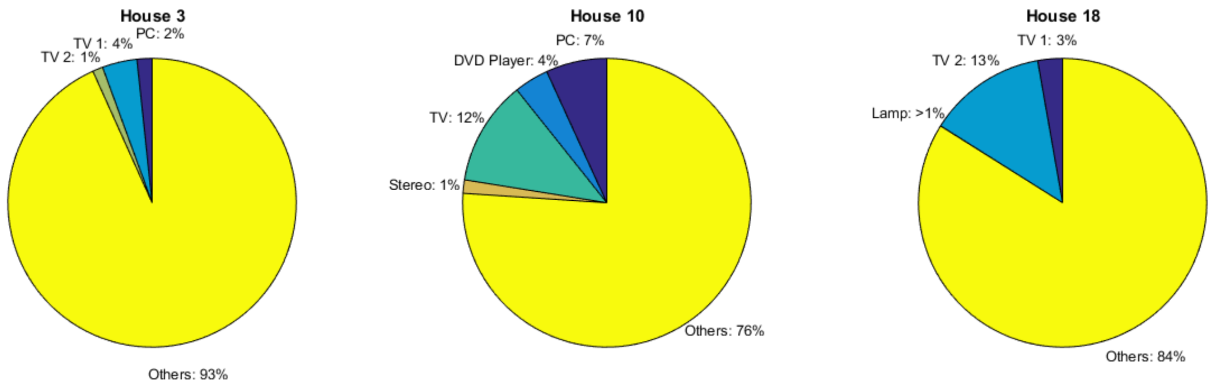

Figure 5.1: Frequency comparison of the reconstruction methods

In figure 5.1 is the power distribution of three households shown. 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 is 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.

5.2 Appliance Noise Influence On Detection

The households in the SmartHG dataset have a relative big consumption created by other appliances than the ones with a sub-meter. The consumption can be thought of as "appliance noise", since it is a signal that is not included in the disaggregation model.

5.2.1 Detection In A Noisy Environment

To investigate the influence appliance noise have on a NILM application, disaggregation of the known appliances of house 3, 10 and 18 is done using the Parson and factorial hidden Markov models (FHMM) algorithm.

	FHMM		Parson	
	F1	Accuracy	F1	Accuracy
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
Average House 3	**0.14**	**0.81**	**0.14**	**0.32**
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
Average House 10	**0.15**	**0.94**	**0.10**	**0.81**
TV 1	0.36	0.65	0.24	0.14
Lamp	-	0.99	-	-
TV 2	0.73	0.58	-	0.00
Average House 18	**0.36**	**0.74**	**0.08**	**0.04**

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

On table 5.1 is the results from the disaggregation shown. 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. On figure 5.2 is a sample from the validation period shown. The blue line shows the data from the main meter. The other lines indicates the appliances. On the left side of the figure is the true consumption shown for all the meters. This consumption is known from the sub-meters on the appliances. On the right side is the true main meter signal shown, and the inferred usages of the different appliances created by the FHMM algorithm.

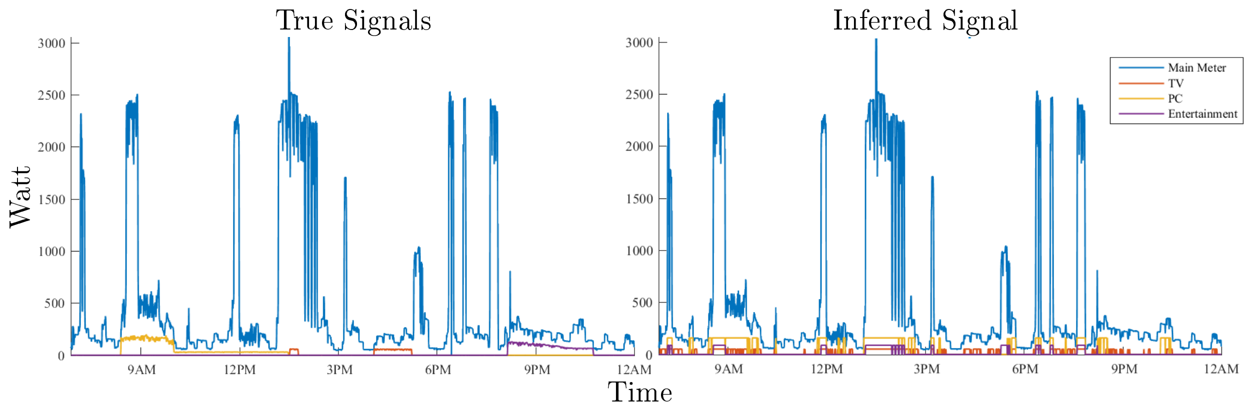

Figure 5.2: FHMM disaggregation snippet

On the figure it is shown how the main meter have many fluctuations. The algorithm tries to map each fluctuation to a change in a appliance, or noise. There are many of the noise fluctuations that are similar to the appliance changes, and this creates errors in the disaggregation.

5.2.2 Detection In Noise Free Environment

The assumption that the noise is what troubles the disaggregation, dictates that for a house where all appliances are known, and therefore there are no noise, will preform better.

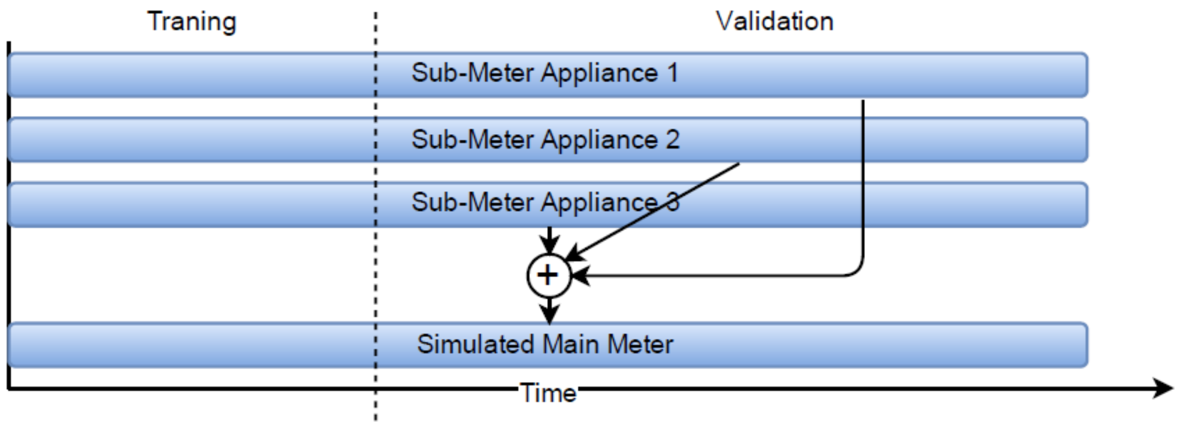

Figure 5.3: Artificially constructed main meter

In order to test this hypothesis was an main meter signal artificially constructed by summing the values of all the sub-meters in a house as illustrated on figure 5.3. In this manner was a new set of artificial houses created where there where no appliance noise. By applying the same disaggregation algorithms as for the noisy environment on the noise free environment it is seen that the disaggregation is much higher as shown in table 5.2.

	FHMM		Parson	
	F1	Accuracy	F1	Accuracy
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
Average House 3	**0.77**	**0.96**	**0.26**	**0.62**
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
Average House 10	**0.24**	**0.99**	**0.10**	**0.81**
TV 1	0.95	0.98	0.24	0.14
Lamp	-	0.99	-	-
TV 2	0.73	0.58	0.73	0.58
Average House 18	**0.56**	**0.85**	**0.32**	**0.24**

Table 5.2: Disaggregation of appliances on artificially constructed main meters

As shown on table 5.2 the F1 and accuracy score have both improved. Even though the artificial house only contains three or four appliances, and they are all accounted for by the model, 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 combine 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, to get the most accurate consumption disaggregation. This can make the model incapable of detecting change between the on/off states.

One other aspect to note is that even though the F1 scores is higher, only a few is higher than 0.9. This is due to the similarity effect. 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 what happens 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 can be a problem if the goal is to determine exactly witch appliance there is responsible for the power consumption. This is one of the key aspects of the little consumers problem, discussed in chapter 4.

This has a profound effect on the parson algorithm that are designed to spilt up appliances based on appliance category's, by utilizing a set of general models. Since a lot of the appliances are similar in their consumption changes from state to state the parson models are almost identical which lead to the events being wrongly categorised.

5.2.3 Noise Effect On Training And Validation

The experiments the two previous sections 5.2.1 and 5.2.2 suggest that appliance noise have 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 find the correct models in the noise, or are the noise just interfering with the correct models?

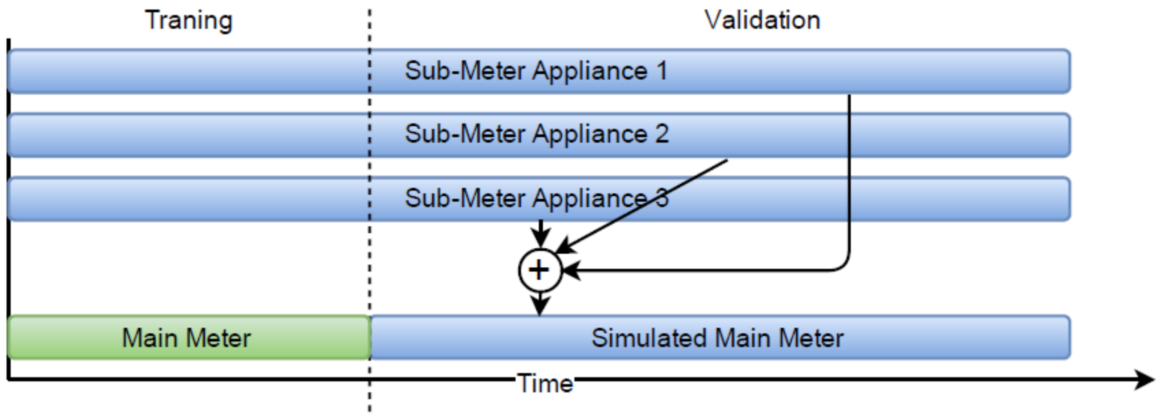

Figure 5.4: Illustration of dataset creation by combining real and constructed data

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 using the constructed main meter in the validation, as illustrated on figure 5.4.

This creates a very noisy environment in the training phase and a noise free environment in the validation phase. If the noise did not influence the creation of the statistical models that represent each appliance, then the results was expected to be just as good as for the noise free experiments conducted in section 5.2.2.

	FHMM		Parson	
	F1	Accuracy	F1	Accuracy
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
Average House 3	**0.11**	**0.92**	**0.25**	**0.63**
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
Average House 10	**0.00**	**0.93**	**0.10**	**0.81**
TV 1	0.00	0.86	0.24	0.14
Lamp	-	0.99	-	-
TV 2	0.73	0.58	0.73	0.58
Average House 18	**0.24**	**0.81**	**0.32**	**0.24**

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 only artificially constructed data. This indicates that the models obtained in the real data is influenced by the appliance noise, and is therefore not fitting accurately enough to the true appliance model.

This could indicate 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 enough 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 true data from the main meter as illustrated in figure 5.5.

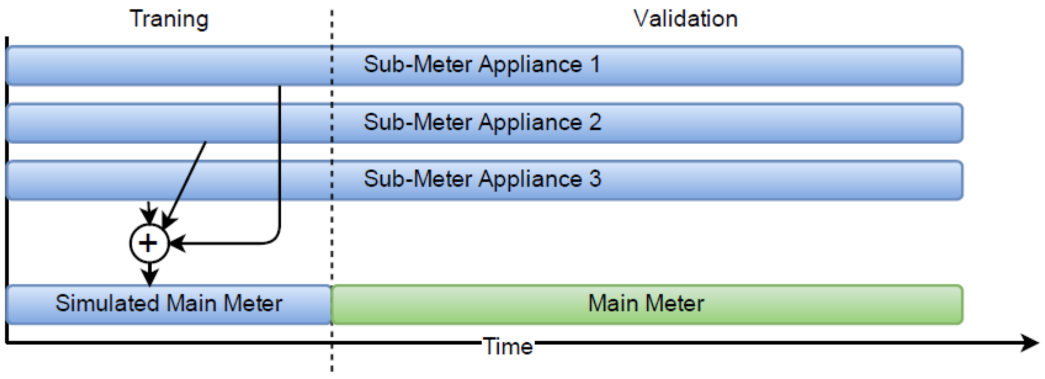

Figure 5.5: Illustration of dataset creation by combining constructed and real data

By training the model on the constructed data, is the true model for the appliances found. If the appliance noise in the real noisy environment is non-interfering can a performance like for the noise free environment described in section 5.2.2 be expected.

	FHMM		Parson	
	F1	Accuracy	F1	Accuracy
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
Average House 3	**0.15**	**0.39**	**0.14**	**0.33**
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
Average House 10	**0.10**	**0.81**	**0.10**	**0.81**
TV 1	0.27	0.26	0.24	0.14
Lamp	-	0.99	-	-
TV 2	0.73	0.58	0.73	0.58
Average House 18	**0.33**	**0.61**	**0.32**	**0.24**

Table 5.4: Disaggregation in constructed and real main meters combined

The results are shown in table 5.4. Here we see that the performance of this system is lower than the one from the noise free experiment. This lead to the conclusion that the noise 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 is compared with the results from the the experiments, where the models was extracted from constructed, data validated on the real data, is shown in table 5.5.

	FHMM Real		FHMM Constructed	
	F1	Accuracy	F1	Accuracy
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
Average House 3	**0.14**	**0.81**	**0.15**	**0.39**
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
Average House 10	**0.15**	**0.94**	**0.10**	**0.81**
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
Average House 18	**0.36**	**0.74**	**0.33**	**0.61**

Table 5.5: Trained in real vs. constructed data

The tables 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 noise free environment and than deployed in a noisy environment. This is in contrast to what one might think, since the noise free environment should have supplied more correct models. But when the appliance noise is removed from a house in the training process is a bias error introduced.

Appliances such as refrigerators and freezers that are always on will create a local house bias. 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 is modelled in each house, as the smartHG case, is the bias almost exclusivity a part of the appliance noise. When the models are fitted by training in the real data, is the bias 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 is 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 bias, 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.

5.3 Model Size And Completeness Influence

Some of the parameters that often differ in the many different environments NILM applications are deployed in, is the number of appliances in the environment, and the number of appliances that are known in the environment.

The model size or complexity is a parameter that tells how many devices that are in a given environment. As the complexity in a NILM application is increasing the detection rate is deceasing[**?**]. 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 model completeness is a metric describing how many appliances is known by the application in relation to the total amount of appliances. This can also be seen as the amount of appliance noise as discussed in section 5.2.

5.3.1 Test Set Creation

In order to experiment with the completeness and complexity in the smartHG dataset, a more controllable series of datasets was artificially constructed from the smartHG dataset.

First an artificial house was 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 and combining them two one artificial house with 7 TV's.

This dataset contains of dominant appliances, and there is no appliance noise from unknown devices. It is still worth noticing that all the 7 appliances is 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 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.

$$H_{full} = \{TV_1, TV_2, ..., TV_7\} \tag{5.1}$$

Since there is no appliance noise the main meter data can be found as the sum of sub-meters data as shown in equation 5.2.

$$M_{main} = \sum_{X \in H_{full}} X \tag{5.2}$$

Using this information it is possible to create a set with a specified complexity, by taking a subset of the H_{full} set with the cardinality of the specified complexity. The main meter can now be found on 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.

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

This can be graphically shown as in figure 5.6.

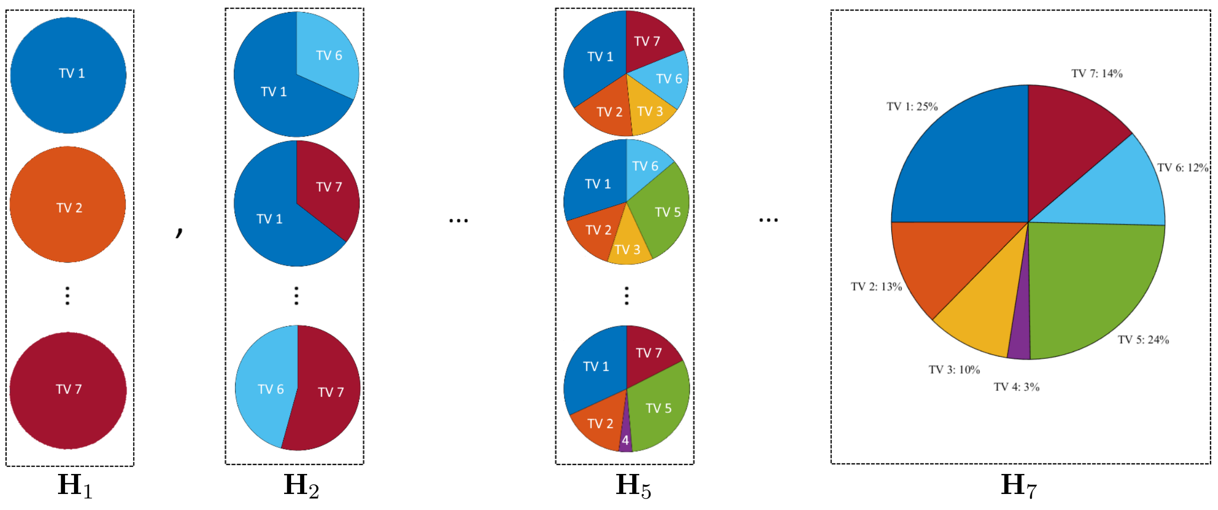

Figure 5.6

Where the $\mathbf{H}_7$ only contains the H_{full} set. In this manner it is possible to artificially create one or more datasets in a desired complexity equal to or less than the H_{full} complexity. The amount of artificiality created houses for a given complexity can be found by using the simple combinatorial equation shown in equation 5.4.

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

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

As illustrated on figure 5.6 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 A_c can be calculated as in equation 5.5. In order to ensure that the combination of appliances does not have a effect on the experiments, is the experiments conducted on all combinations in $\mathbf{H}_c$. The results for each experiment is a average of the performance for the appliance in the experiment.

	$c = 1$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$
$\|\mathbf{H}_c\|$	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 $|\mathbf{H}_c|$ and A_c values for the different complexity be seen in table 5.6.

5.3.2 Model Complexity Test

To investigate the effect the number of appliances in a house, have on the performance of a NILM system, is disaggregation done on every artificial house in $\mathbf{H}$.

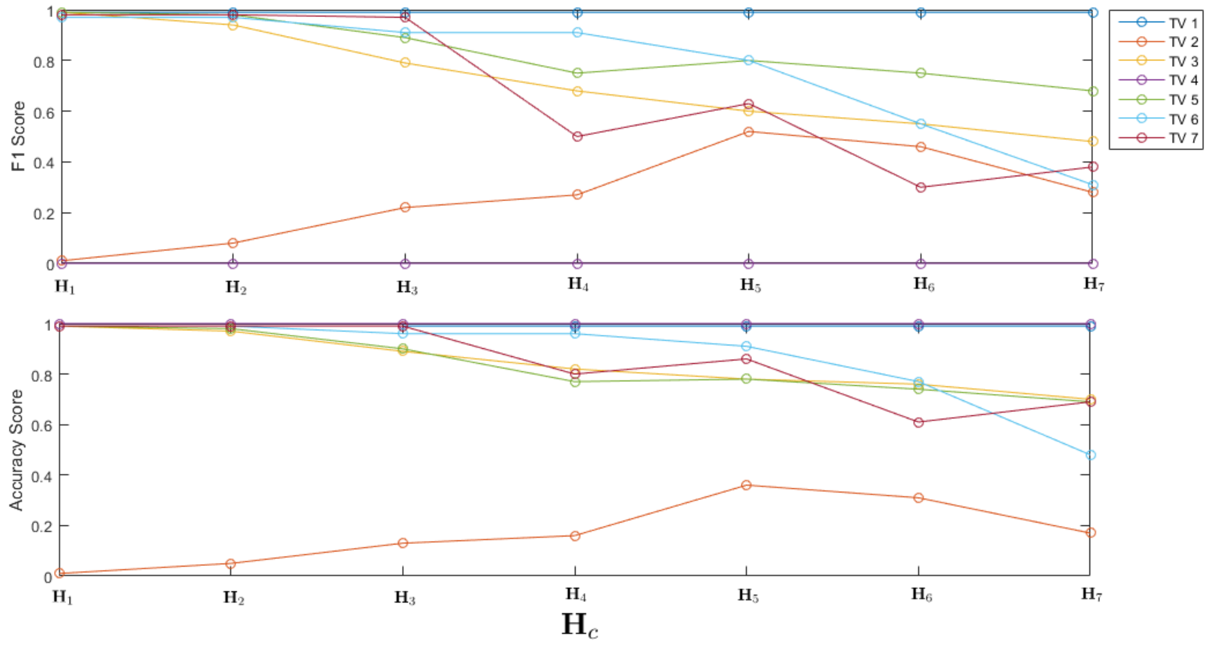

Figure 5.7: Average score at different complexity

The test is preformed on every combination, to ensure that there is not some combination that are more favourable than others. The results shown in figure 5.7 is 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 decrees as the complexity of the house increase. This is due to the application spill over effect. The spill over 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 a 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 the spill effect from the other TV's. Since all the appliances are TV's they have a lot of overlap in their usage, since a lot of people watches TV between 18-22. This makes the algorithm encounter signal's that are similar in structure for the different TV's, and it have a hard time deciding which appliance is responsible. Therefore can some of the events generated by the other TV's spill over in 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 spill over effect decrease the F1 and accuracy score as seen on the figure.

5.3.3 Model Completeness Influence

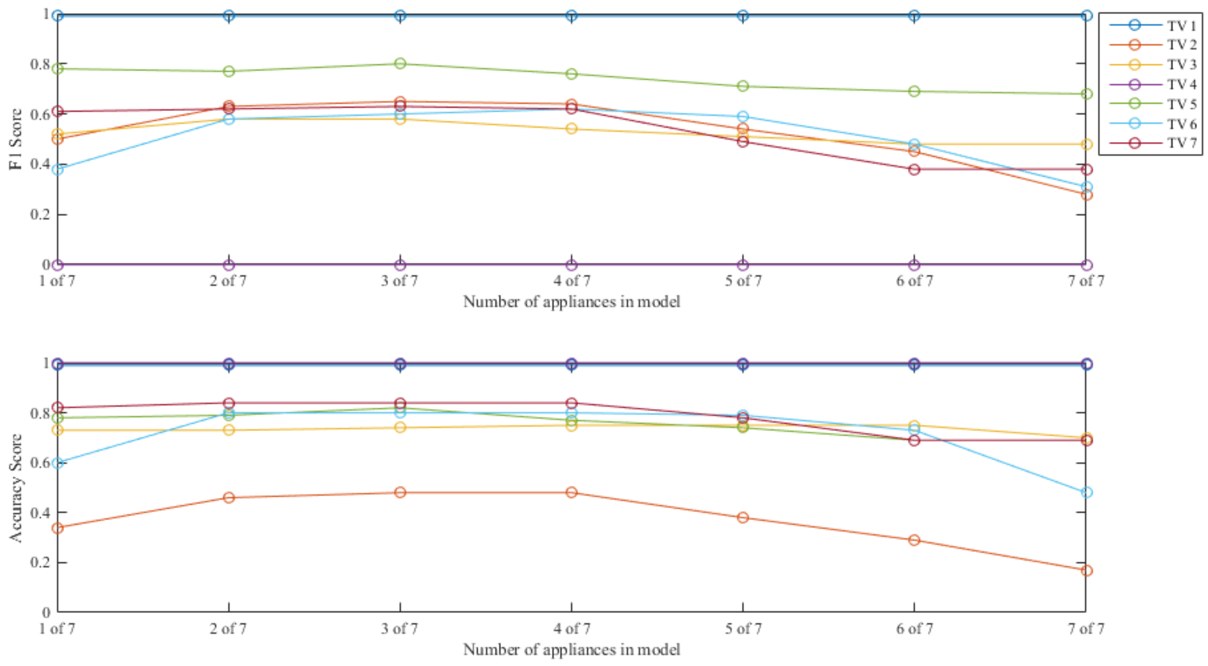

Figure 5.8

NILM As an Application 6

Discussion 7

7.1 measurements how to measure

7.2 Data collection suggestions

7.3 Capability

7.4 Electric companies as data-brokers

7.5 Privacy

Conclusion 8

Bibliography

[1] 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. 16838–16866, 2012. Appliance Load Monitoring (ALM) is essential for energy management solutions, allowing them to obtain appliance-specific energy consumption statistics that... Appliance Load Monitoring (ALM) is essential for energy management solutions, allowing them to obtain appliance-specific energy consumption statistics that can...

[2] Z. Ghahramani and M. I. Jordan, “Factorial hidden markov models,” *Machine Learning*, vol. 29, no. 2, pp. 245–273, 1997. Hidden Markov models (HMMs) have proven to be one of the most widely used tools for learning probabilistic models of time series data. In an HMM, information... Hidden Markov models (HMMs) have proven to be one of the most widely used tools for learning probabilistic models of time series data. In an HMM, information...

[3] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “Non-intrusive load monitoring using prior models of general appliance types,” in *Proceedings of theTwenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, pp. 356–362, July 2012.

[4] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, “The eco data set and the performance of non-intrusive load monitoring algorithms,” in *Proceedings of the 1st ACM International Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys 2014). Memphis, TN, USA*, pp. 80–89, ACM, nov 2014.

[5] N. Regnauld, “Generalisation and data quality,” 2015-08-01T00:00:00Z. Type: article; CC BY.

[6] I. 8402:1994, “Quality management and quality assurance – vocabulary,” 1994-03-24.

[7] 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. Type: Text; © The Author(s) 2015; Open AccessThis article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

[8] I. 19157:2013, “Geographic information – data quality,” 2013-12-15.

[9] 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,” in *e-Science (e-Science), 2014 IEEE 10th International Conference on*, vol. 1, pp. 271–278, 2014. ID: 1.

[10] 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. Type: article; CC BY.

[11] 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. Type: Software; Copyright 2015 Simader et al.

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

[13] 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. Analyze the performance of an iterative algorithm, similar to the discrete Papoulis-Gerchberg algorithm, and which can be used to recover missing samples in... An analysis of the performance of an iterative algorithm that can be used to recover missing samples in finite-length records of band-limited data.No...

[14] M. Marques, A. Neves, J. Marques, and J. Sanches, “The papoulis-gerchberg algorithm with unknown signal bandwidth,” vol. 4141, pp. 436–445, 2006.

[15] 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. 15941–15962, 2001.

[16] 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. The majority of data sets in the geosciences are obtained from observations and measurements of natural systems, rather than in the laboratory. These data... The majority of data sets in the geosciences are obtained from observations and measurements of natural systems, rather than in the laboratory. These data sets...

[17] A. Moghtaderi, P. Borgnat, and P. Flandrin, “Gap-filling by the empirical mode decomposition,” 2012. We propose a novel gap-filling technique, based on the empirical mode decomposition (EMD). The idea is that a signal with missing data can be decomposed into a...

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

[19] 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. This paper proposes a neural-network (NN)-based approach to nonintrusive harmonic source identification. In this approach, NNs are trained to extract important... Several NN-based classification models including multilayer perceptron (MLP), radial basis function (RBF) network, and support vector machines (SVM) with...

[20] 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,” in *the Residential Power Line,” Proc. 9th Int’l Conf. Ubiquitous Computing (Ubicomp 07), ACM*, pp. 271–288, Press, 2007.

[21] 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. Type: text.

[22] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," 2015-07-23; 2015-09-28. Type: text.

[23] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, "Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview," 2015. Research on Smart Grids has recently focused on the energy monitoring issue, with the objective to maximize the user consumption awareness in building contexts...

[24] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)* (N. D. Lawrence and M. A. Girolami, eds.), 2012.

[25] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, "Leveraging smart meter data to recognize home appliances," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pp. 190–197, 2012. ID: 1.

[26] W. Kleiminger, C. Beckel, and S. Santini, "Household occupancy monitoring using electricity meters," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015)*, (Osaka, Japan), sep 2015.