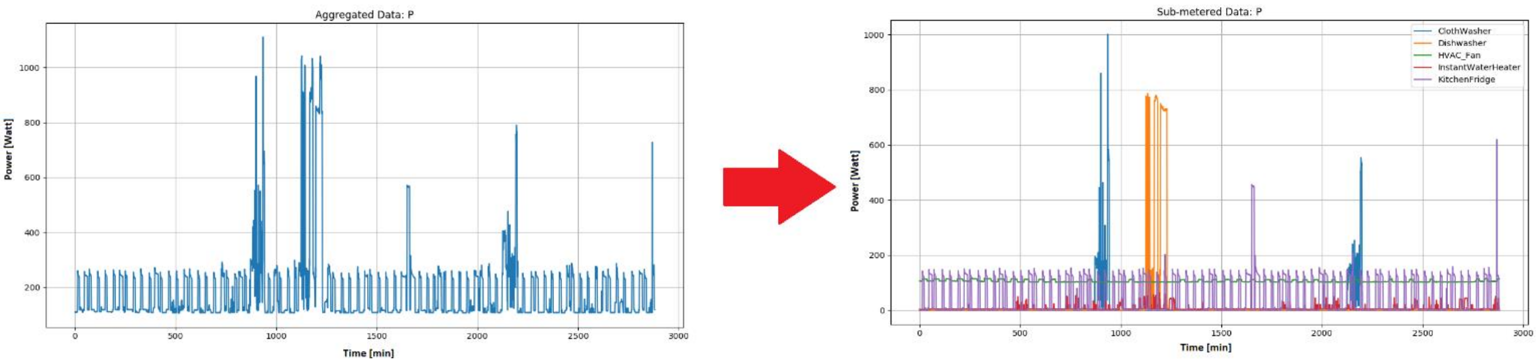
2 Month Progress Report
23 Feb 2018

Non-Intrusive Load Monitoring (NILM)

Non-Intrusive Load Monitoring (NILM) is an algorithm to disaggregate the power from the household meter into appliance energy information.



From the research, I understand that the NILM consist of two parts.

- Load Disaggregation
- Load Identification

In the load disaggregation process, first we have to disaggregate the aggregated into possible appliances present in the house. The disaggregation process is a difficult problem, because

- We don't know exactly which appliances are running in the house beforehand
- There are different types of appliances [ON/OFF, Finite State Machine, Continuous Variable Machine, Always ON]
- At a given time instant any number of appliances can be running simultaneously, contributing to the aggregated power
- The turn on/off cycles each appliances is not consistent. For example, for the washing machine the cycle is fixed and all its active operation period takes same amount of time. But for a microwave the active operation period is not fixed. It depends on how long the the timer is set on the microwave.

In the load identification process, once the aggregated data is disaggregated, then the identification of the disaggregated data happens. This is a typical classification problem.

Methods

Load Identification

The load identification is typically a supervised classification problem. Given a dataset with instances of each appliance, different features can be extracted from the data and can be used for the classification. Traditional classifiers like SVM, Naïve Bayes, KNN, Random Forest and others can be used.

Reference:

Gao, J., Kara, E., Giri, S., & Berges, M. (2015). *A feasibility study of automated plug-load identification from high-frequency measurements*. Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on, 220-224.

Load Disaggregation

Load disaggregation is a hard problem.
There are many approaches that have been used for the purpose of load disaggregation.
Most successful approaches are,

1. **Combinatorial Optimization**: Since the number of appliances in a house are fixed and discrete, the problem of disaggregation can be boiled down to finding the combination of features of different appliances, with multiple states [ON, OFF intermediate], that minimizes the sum of aggregated features. Let us assume that we use Real Power (watt) as a feature on which we will run disaggregation.

   K = Appliance power state. For example washing machine will have 4 power state, then $K \in \{1,2,3,4\}$.
   z^n = Appliance state sequence for n^{th} appliance. $z^n \in [1,2,\ldots,K]$
   $z^n_{t,k} \in 0,1$ = Weather n^{th} appliance is in k^{th}state at time t
   $\theta^n_{t,k}$ = Measured power for appliance n^{th} appliance at time t at state k
   μ^n_k = Power draw by n^{th} appliance at k^{th} state

   Now, at a given time an appliance can be only be in one given state,

   $$\sum_{k=1}^{K} z^n_{t,k} = 1$$

   The power consumption by n^{th} appliance in k^{th}state at time t is given by,

   $$\widehat{\theta^n_{t,k}} = \sum_{k=1}^{K} z^n_{t,k}\mu^n_k$$

   The overall power consumption of all appliances at a given time t is given by,

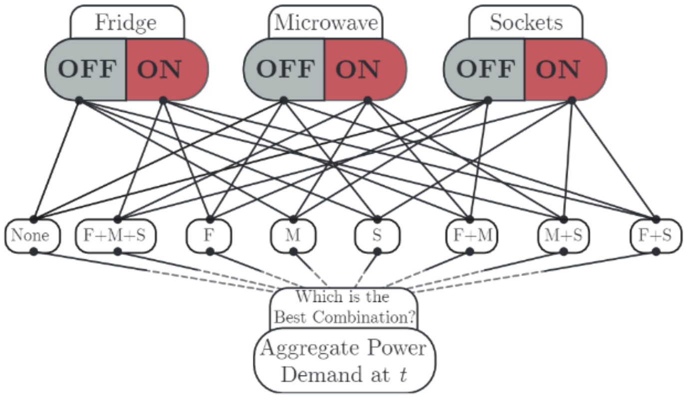
   $$\hat{x}_t = \sum_{n=1}^{N}\sum_{k=1}^{K} z^n_{t,k}\mu^n_k$$

   The error in the power signal after the load assignment is given by,

   $$e_t = \left|\hat{x}_t - \sum_{n=1}^{N}\sum_{k=1}^{K} z^n_{t,k}\mu^n_k\right|$$

   In the combinatorial optimization we want to find the states that will minimize this error,

   $$argmin_{z_t}\left|\hat{x}_t - \sum_{n=1}^{N}\sum_{k=1}^{K} z^n_{t,k}\mu^n_k\right|$$



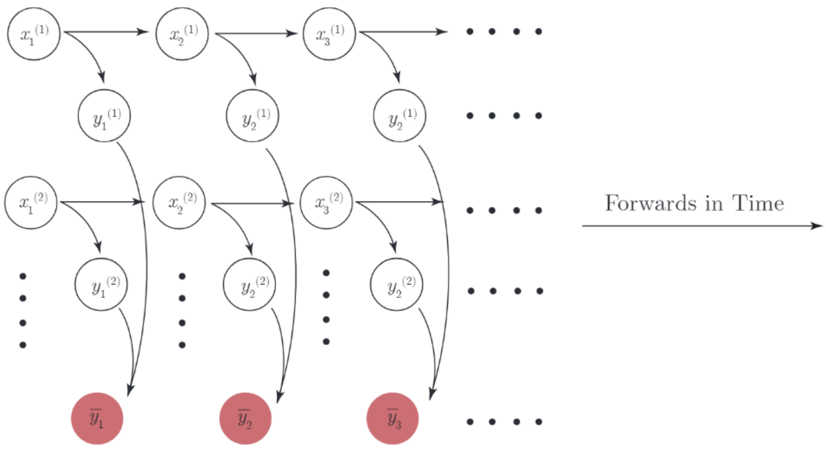
Issues with the approach:

a. As the number of appliances increases, the number of possible combinations increases. Also, most of appliances have more than one state (washing machine have 4 states). The algorithm have no means to find the state of appliances.
b. Also, the algorithm cannot detect low power appliances.
c. The appliances with frequent fluctuations are not detected
d. The appliance detection is hugely affected by the noise in the signal.

Reference:
http://blog.oliverparson.co.uk/2011/04/nialm-as-combinatorial-optimisation_15.html
N. Batra, H. Dutta and A. Singh, "INDiC: Improved Non-intrusive Load Monitoring Using Load Division and Calibration," *2013 12th International Conference on Machine Learning and Applications*, Miami, FL, 2013, pp. 79-84.
doi: 10.1109/ICMLA.2013.21

2. <u>Hidden Markov Model (HMM)</u>

As the electrical data is usually time series data, HMM provides a natural approach for this purpose. Factorial HMMs (FHMM) are an extension of the HMM where hidden states are factored into multiple state variables.



In the above diagram, the $\overline{y}_t$, is the aggregated power consumption at time t. The $y_t^{(i)}$ is the power consumption of appliance i at time t and $x_t^{(i)}$ are the hidden states of the appliance. Since the observed state is continuous, the emission probabilities are modelled with Gaussian distribution:

$$P\left(y_t \middle| x_t^{(1:N)}\right) = \mathcal{N}(\sum_{i=1}^{N} \mu^{(i)}, \Sigma)$$

Issues with FHMM:

a. The algorithm is computationally heavy
b. As it is computationally heavy, it usually only works with slow sampling data

Reference:
Kim, H.; Marwah, M.; Arlitt, M.; Lyon, G.; Han, J. Unsupervised Disaggregation of Low Frequency Power Measurements. In Proceedings of the 11th SIAM International Conference
on Data Mining, Mesa, AZ, USA, 28–30 April 2011

3. Deep Learning

The deep learning techniques have been used for the purpose of load disaggregation. J. Kelly used CNN, Autoencoders and RNN networks for the purpose of NILM. The Neural NILM performs better than the FHMM and Combinatorial Optimization techniques and can also work on high sampling data.

Issues:

a. Need lot of data for the purpose of training
b. The algorithm creates one network for each appliances. So with 10 appliances, in house, the algorithm will have 10 different networks. Scaling is an issue here.
c. Networks need lot of space for storage, hence cannot be used in embedded systems

Reference:
Kelly, Jack, and William Knottenbelt. "Neural NILM: Deep neural networks applied to energy disaggregation." Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. ACM, 2015.
arXiv:1507.06594v3

4. Techniques from Signal Processing

The load disaggregation can be treated as a Blind Source Separation (BSS), a process of separating a set of source signal from the mixture of signals.

A set of source signals $s(t) = \{s_1(t), s_2(t), \dots , s_n(t)\}^T$ is mixed with a mixing matrix $A = [a_{ij}] \in R^{m x n}$ to produce a mix signal $x(t) = \{x_1(t), x_2(t,), \dots , x_m(t)\}^T$,such that

$$x(t) = A * s(t)$$

In case of load disaggregation, the system is undetermined, as $n > m$.

There are different algorithm that can be used for BSS, like,

1. Independent Component Analysis
2. Principal Component Analysis

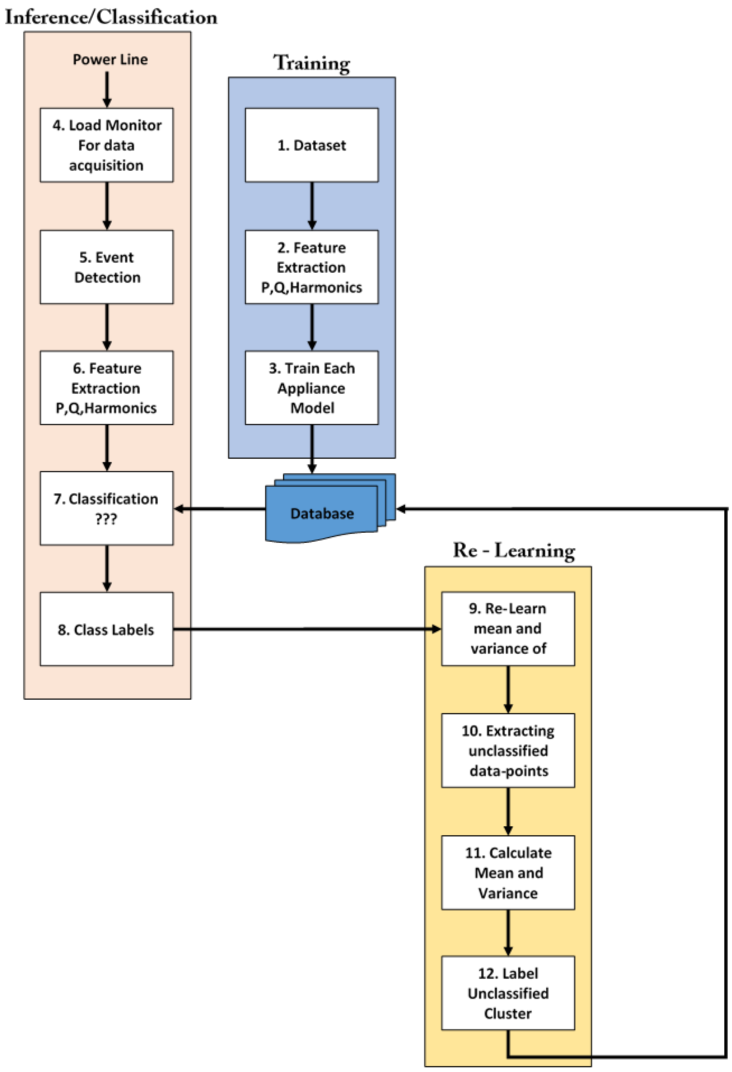
For undetermined blind source Orthogonal Matching Pursuit (OMP) algorithm can be used.

Requirement of the Thesis

The goals of the thesis are listed below -:

1. Investigate on techniques to create an electrical feature model for each appliance that can be generalized across various brands and models of appliances.
2. Analyse different electrical features that can be used for the inference/classification of each appliance in the fast-sampling (> 1 Hz) load data.
3. Derive an algorithm for load inference/classification from an aggregated data that has low storage requirement and is computationally cheap, so that it can be easily run on small embedded systems like Raspberry-Pi.
4. Research the re-learning process to tune the general appliance model to represent the appliances with the specific house and improve on the classification accuracy when compared the classification done on general model.
5. Perform experimentation across the dataset and show that the proposed solution performs better that the state-of-the-art supervised and un-supervised techniques.

Proposed NILM Algorithm



In the proposed algorithm, we want to use Gaussian Clusters for each appliance and corresponding states. Then each instance in the aggregated data can be classified into one of these clusters.

With the help of Gaussian Mixture Models (GMM), we can identify the different states of the appliances automatically and much accurately than Combinatorial Optimization techniques.

Also, these clusters can be updated easily after the inference step to make it more accurate for the house the algorithm is running.

Dataset

For the purpose of creating appliance model (Gaussian Clusters) we need sub metered training data for each appliance.

The dataset hence should be,

1. Main Voltage – 240V (Sweden main is 240V/ 50Hz)
2. The aggregated data from the Single phase line with different features - Active Power, Reactive Power, Apparent power, Current and Voltage.
3. The sub-metered data from different appliances (fridge, dishwasher, coffee, HVAC, lightning, etc)
4. Sub-metered data of an appliance from different brands to create a generalizable model for the appliance. More appliance the better.
5. Sub-metered data with different features - Active Power, Reactive Power, Apparent power, Current and Voltage.
6. The sampling frequency for both aggregated and sub-metered data should be smaller (smaller than 1 minute, preferred 1 sec or smaller).

Currently, I am running experiments with AMPDs dataset from Canada.

The characteristics of dataset –

a. Main voltage – Two voltage supplies, one of 120/50Hz and another of 240/50Hz
b. 1 minute sampling frequency
c. Number of Houses – 1
d. Duration of the dataset – 2 years from April 2012 – March 2014
e. Features in the Dataset – Voltage, Frequency, Current, Real Power, Reactive Power, Apparent Power, Apparent Power Factor, Displacement Power Factor, Real Energy, Reactive Energy, Apparent Energy

From the current dataset I took the appliances that are 120V/50Hz and sub-metered data have only one appliance. (There are some measurement of sub-metered for entire room like bed-room, garage. The data I took are simply 1 single appliance).

For the purpose of analysis, currently I took only Real Power and Reactive Power for clustering each appliance model.

Figure 1 and Figure 2, shows the sub-metered data and aggregated data for period of 2 days.

Within this 2 days data, there were some specific event are not present. For example, a peak in Kitchen Fridge is only seen after 3 day.

Fig 1 Sub-metered Data for 2- days

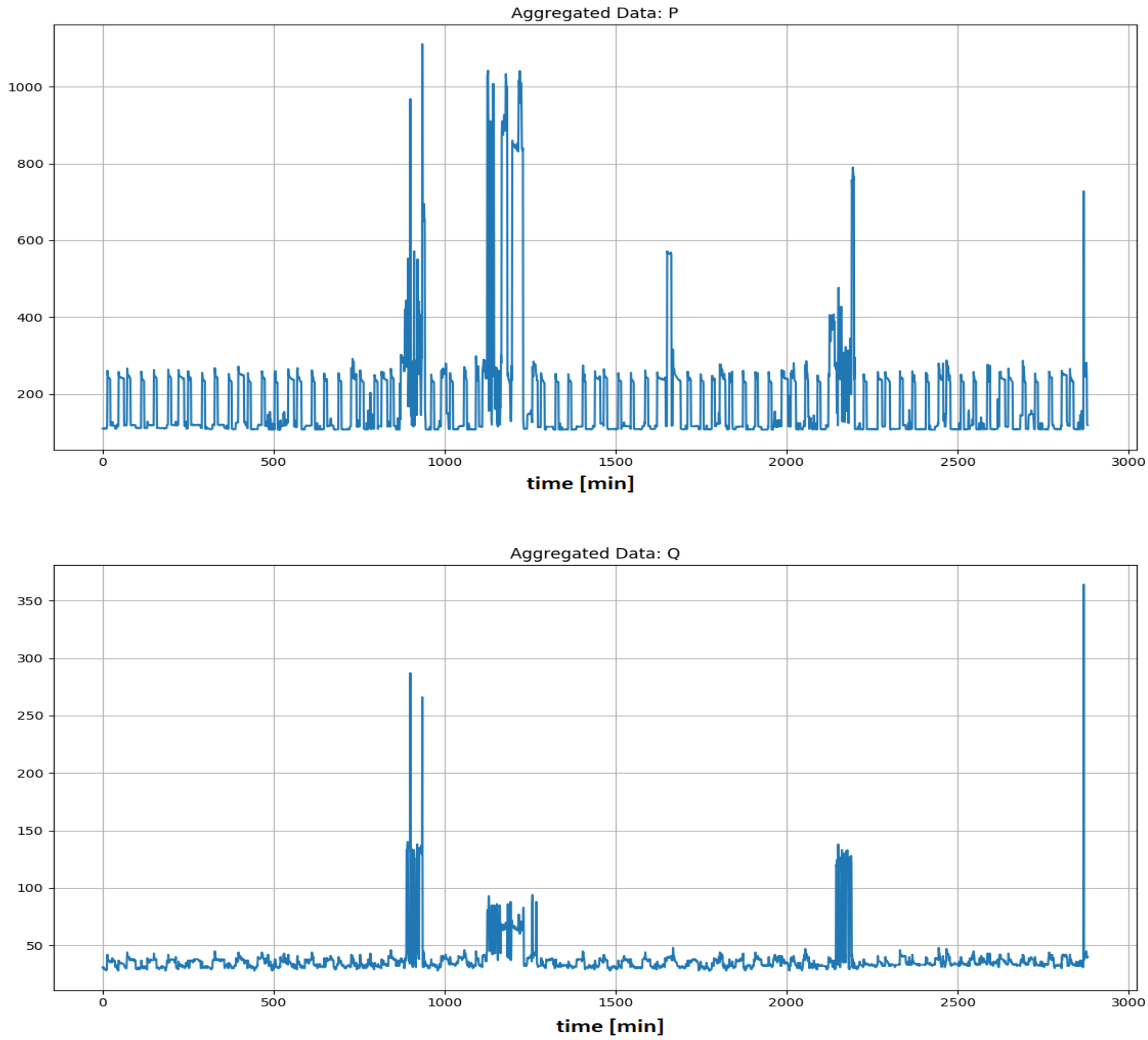


Fig 3 Aggregated Data for 2-days

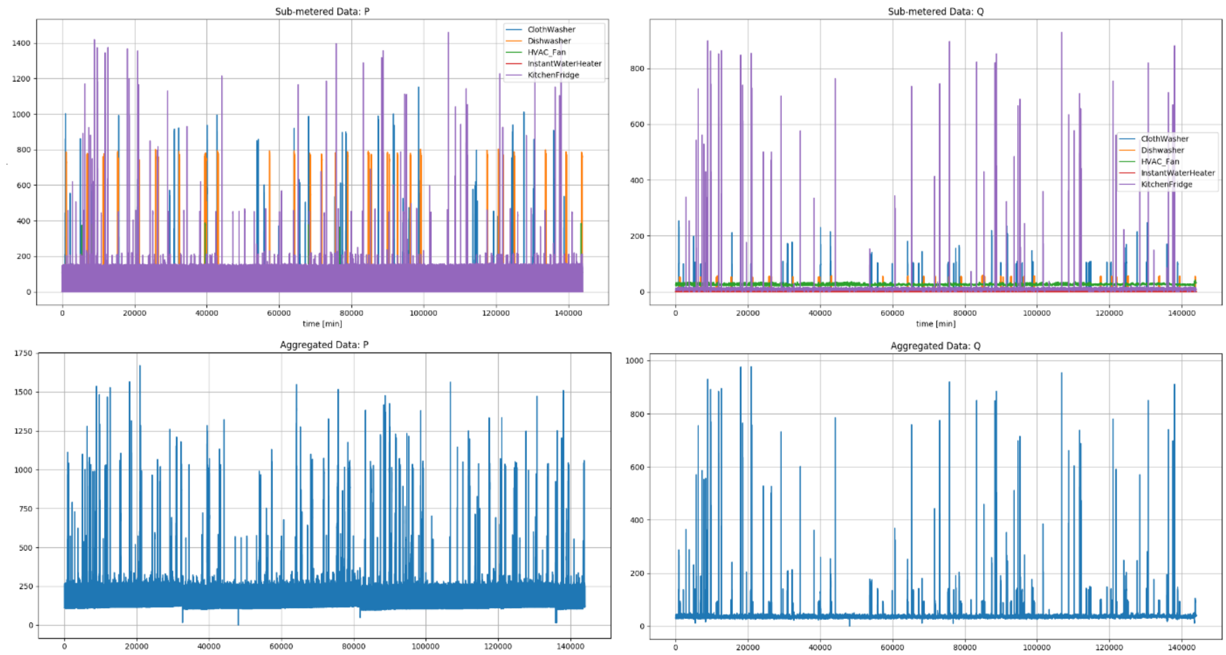


Fig 2 Data for 100 days

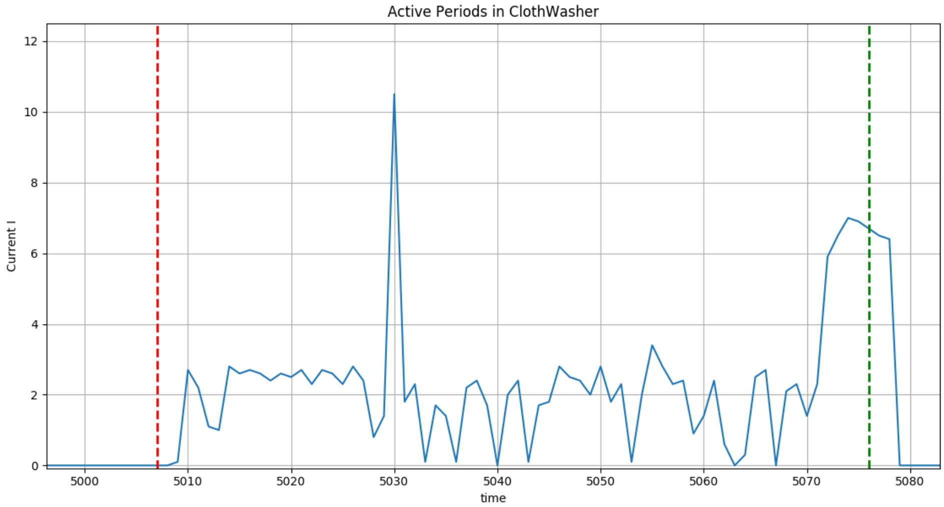
In the dataset,

- Kitchen Fridge has the most number of instances and have a much more variability, compared to any other dataset. The aggregated power is mostly the Kitchen Fridge. Also, it is always on, with infrequent peaks in power.
- Cloth washer and Dishwasher are less frequent than Kitchen Fridge and have intermittent period of activation. They also have different states of operation, when compared to Kitchen Fridge or HVAC Fan
- HVAC Fan power always ON with infrequent peaks
- Instant Water Heater has very small power draw compared to any other appliances and have lot if infrequent peaks

Currently, I am running the analysis only for Cloth Washer and Dish Washer.

Algorithm

- **Activation Period extraction**: In order to create a model for the appliances, first we need to extract the active period of each appliance from their sub-metered data. I use current data to find the active period in each appliance.
  The Figure 4 shows the one instance of active current for both dishwasher and cloth washer.



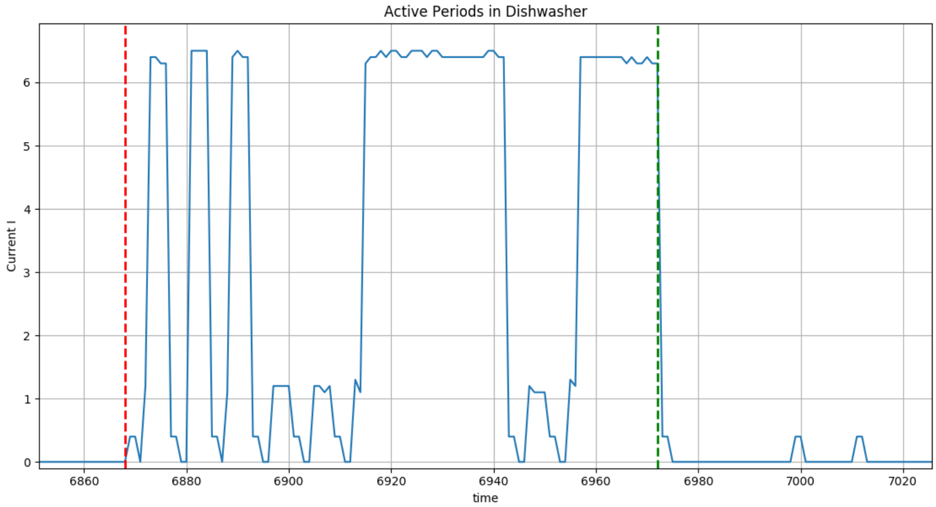


Fig 4 Active Period Extraction

- **Cumulative Feature**: Extracting all the active instances for both the appliances and then calculating the mean of the features. By this way we will get the most prominent real and reactive power curve for both the appliances.
The Figure 5 shows the all the instances of active period for both appliances.

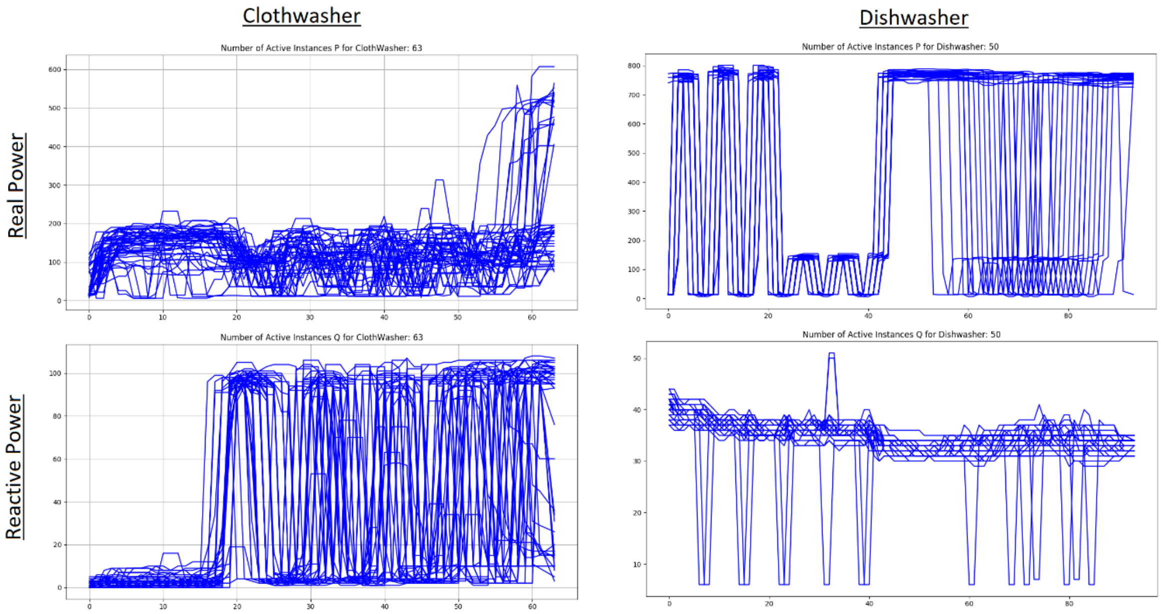


Fig 5 All Active Instance for appliances

- **Mean Calculation**: From all the active instances of Real and Reactive power for cloth washer and dishwasher, mean was calculated.
The Figure 6 shows the mean for both the appliances

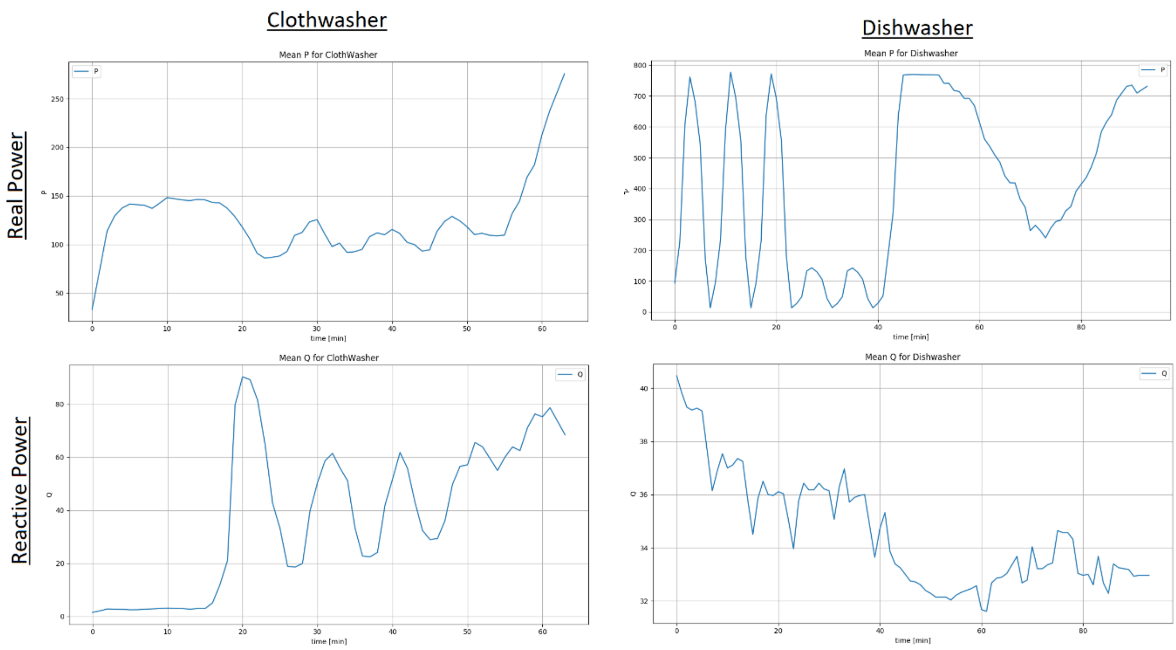


Fig 6 Mean of Features

- **Gaussian Cluster of appliances**: The mean values of real and reactive power for both the appliances were used to create Gaussian cluster.
  First we run a Gaussian Mixture Model to find if there are possible more clusters in each of the appliance. For the mean of dishwasher and cloth washer, 1 cluster gave use the best AIC and BIC values. Hence model both of them in one single cluster.

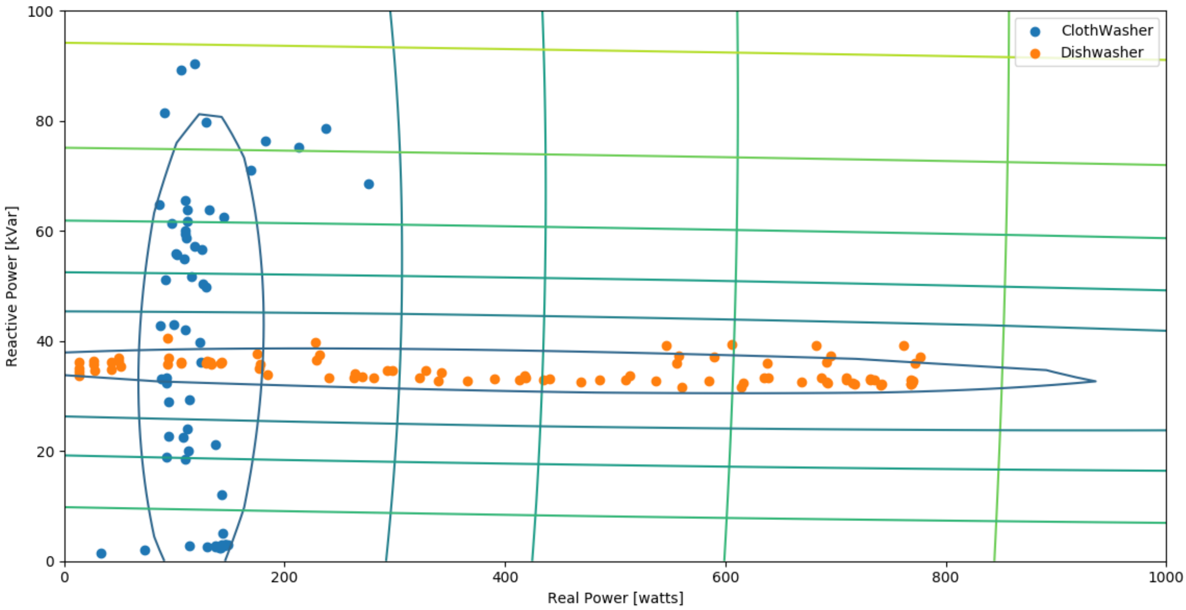


Fig 7 Gaussian Cluster for appliance mean values

- **Inference**: Using these clusters we then ran an inference for the aggregated data for cloth washer and dishwasher. We took first 2 days of aggregated data. Then calculated the probability density function across both these clusters. The maximum pdf for the point will be label for the point. Apart from these clusters, we also created a cluster for GND. Hence, the inference if currently running for 3 Gaussian clusters.

  In the figure 8, we can see that the inference is not very accurate. It gives very good classification for cloth washer, but has several misclassification for dishwasher, especially for smaller values of real power. Also, the part where both the dishwasher and cloth washer are working have been point at instances are classified wrong a lot.

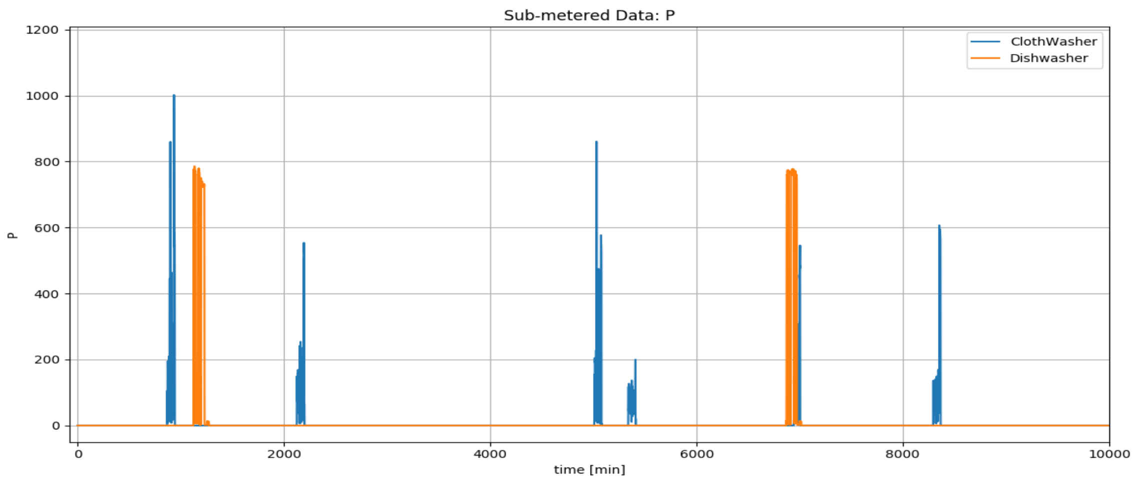
Conclusion

It is clear that the inference is not very accurate. Even when only dishwasher is running, the algorithm classify the smaller value of dishwasher as cloth washer. The reason maybe because the clusters for both the dishwasher and cloth washer are merged around the smaller values. Also, when both the appliances are running, there are lot of misclassification. One possible solution to avoid this is to create a super cluster, with aggregated values for both the cloth washer and dishwasher. Hence we will have total 4 cluster (in case of 2 appliances – CW, DW, CW+DW, GND) to run inference.

Also, the active extraction part of the algorithm depends on the spike in the data. But then for always ON appliances (fridge, HVAC) we have to spike as such to denote the active region. Hence, have to figure out how to take care of that.

Clustering is not so good right now. We can see that there are clearly 2 states in both dishwasher and cloth washer, but the GMM only could find on clusters for each. Hence, have to think more about what kind of features to use to create clusters such that we can identify different states better.

For detection of multiple appliances running together, we have to create super clusters. Hence with increase of appliances and there states, the total number of cluster are exponential. This will take a lot of processing power with more number of appliances.



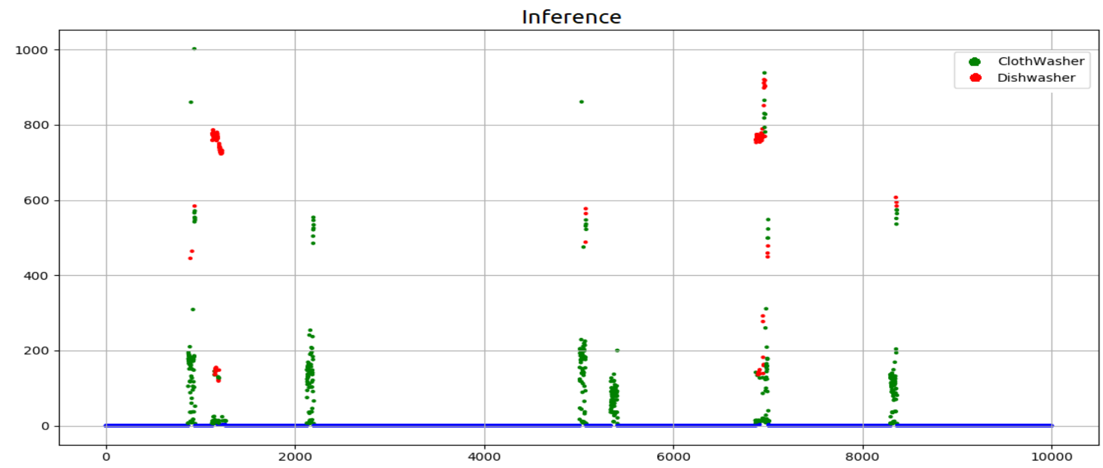


Fig 8 Inference for 3 days data

Further steps

1. Analyze the inference with metric defined in the thesis
2. Figure out how to use always ON appliances (HVAC, Kitchen Fridge) using this technique.
3. Check other features that can improve the clustering
4. Check the algorithm from speech signal processing, Orthogonal Matching Pursuit (OMP) and Independent Component Analysis (ICA) for the purpose of disaggregation.