1 Month Progress Report
15 April 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.

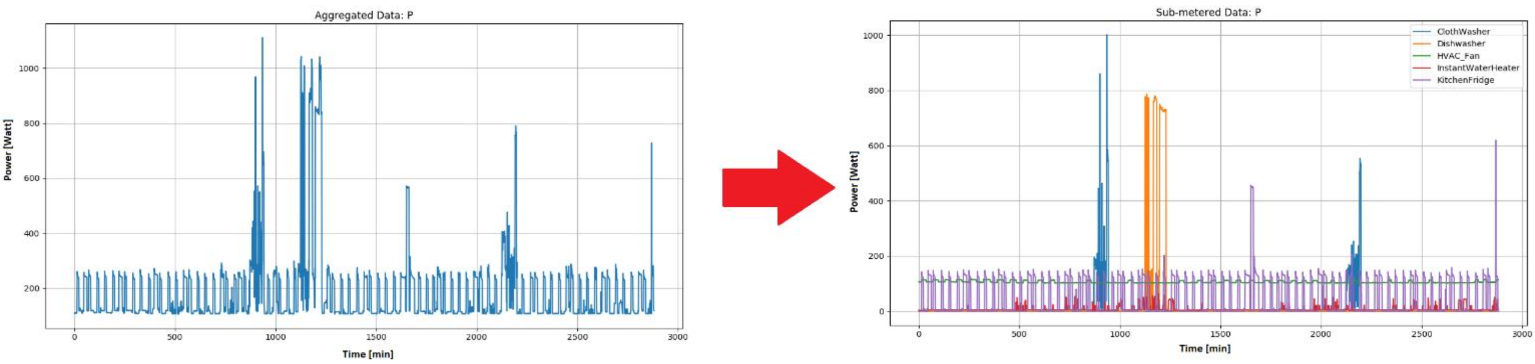


Fig 1 Non-Intrusive Load Monitoring (NILM)

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.

Proposed NILM Algorithm

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).

I started with the AMPDS data for the experiment, but found that the data was not in manner that I wanted to use. The data set had 2 different Main voltages and there were only 5 appliances on 120V. Also, the data was not appliance-specific rather location specific. For example, the data from bedroom had readings from all the bulbs, charging point and fans, Kitchen dataset had reading combined with kettle, coffee machine, and other smaller appliances.

I wanted to use appliance-specific dataset.

Recently, I found AC-F2 dataset from Switzerland that has dataset for each appliance in two period of 1hr each.

The characteristics of dataset –

a. Main voltage – All the appliances are at 230V
b. 10hz sampling frequency
c. Number of appliance category – 11
d. Number of appliance in each category - 30
e. Duration of the dataset – 2hr split across two session of 1hr each
f. Features in the Dataset – Voltage, Frequency, Current, Real Power, Reactive Power, Power Factor.

The dataset does not have any aggregated power, but that I will create manually.

In the dataset, I will use the session 1 (15 appliances in each category) for training and then randomly select appliances from session 2 to create an Aggregated data to run the inference.

Toy-Dataset

For the purpose of creating and testing the model for load disaggregation, and making sure that I know what is happening in the model, I created my own dataset.

Individual Signals

In the dataset, I created 3 different types of signal of length 150 sec mimicking the real appliances.
The different signals are -

1. Bulb

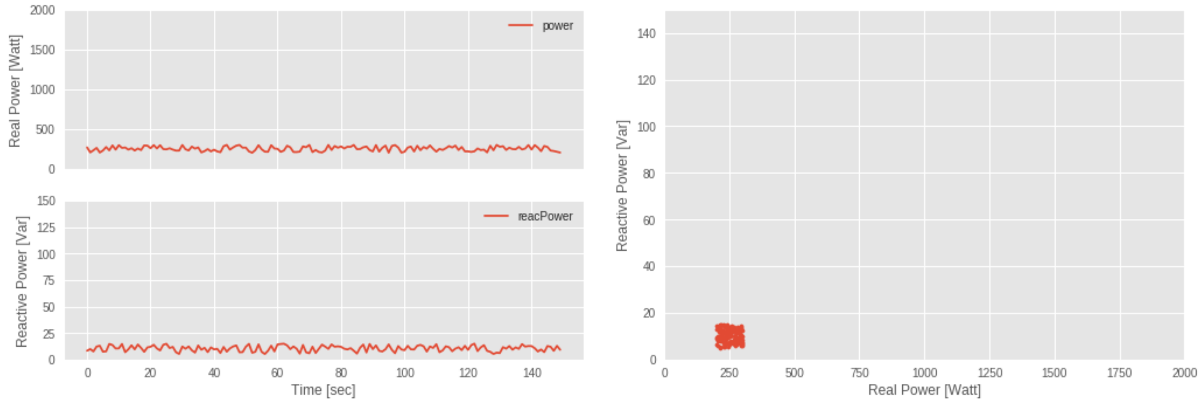


Fig 2 Toy Bulb Signal

Real Power Amplitude – 200 W + Random noise in range (0, 100 W)
Reactive Power Amplitude – 5 VAR + Random noise in range (0, 10 VAR)

This signal only contains the ON sequence data mimicking a 200 W bulb.

2. FSM

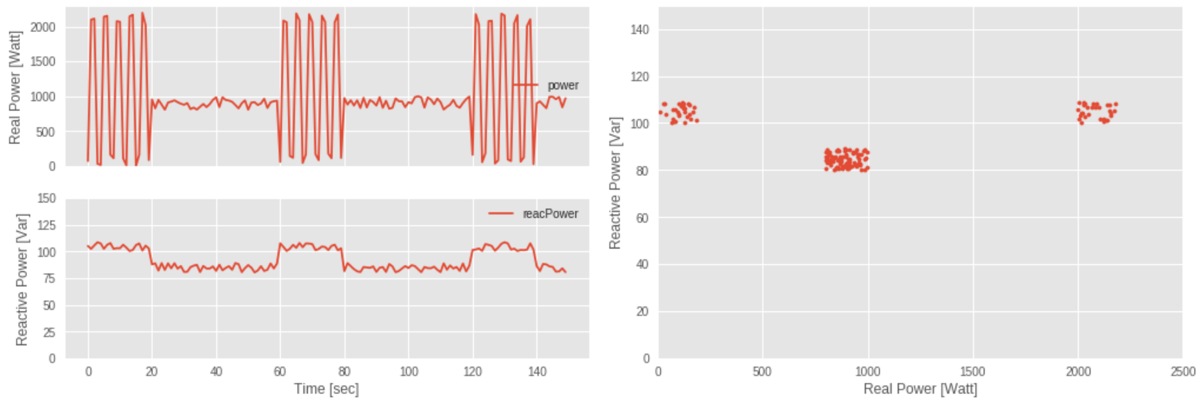


Fig 3 Toy Finite State Machine (FSM) Signal

Real Power Amplitude – 800 W + Random noise in range (0, 100 W)
Reactive Power Amplitude – 80 VAR + Random noise in range (0, 10 VAR)
Periodic Real Power – 2000 W with frequency 1Hz
Reactive power during periodic signal – 100 VAR + Random noise in range (0, 10 VAR)

The idea with the periodic signal was to mimic a 2 state machine with intermittent periodic signal similar to a dishwasher. This also represent only the active state of the appliance.

3. TSM

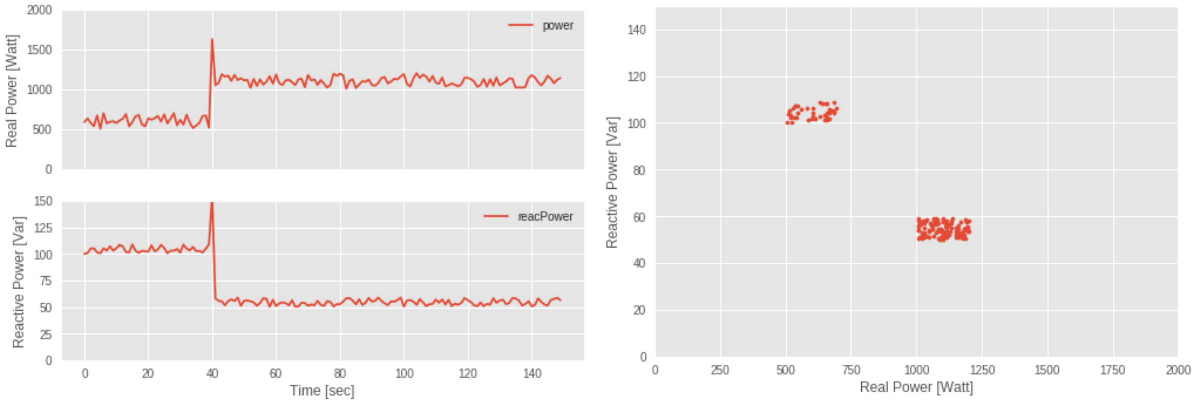


Fig 4 Toy Two State Machine (TSM) Signal

Real Power Amplitude (State 1) – 500 W + Random noise in range (0, 100 W)
Real Power Amplitude (State 2) – 1000 W + Random noise in range (0, 100 W)
Reactive Power Amplitude (State 1) – 100 W + Random noise in range (0, 10 VAR)
Reactive Power Amplitude (State 2) – 50 W + Random noise in range (0, 10 VAR)

Here the signal represent a 2 state machine with a peak at the change-over similar to a refrigerator. This represent only the active state of the appliance.

Aggregated Signal

The aggregated data is just a linear combination of all the 3 signal in different combination. In order to check that the model works for all different combination, total 7 intervals of active data is present in the aggregated data. The following present the sequence of combination present the aggregated data. Anything in between the active data or between the intervals is considered GND. The signals in the data are

- Only bulb,
- Only FSM,
- Only TSM,
- Bulb + FSM,
- Bulb + TSM,
- FSM + TSM,
- Bulb + FSM + TSM

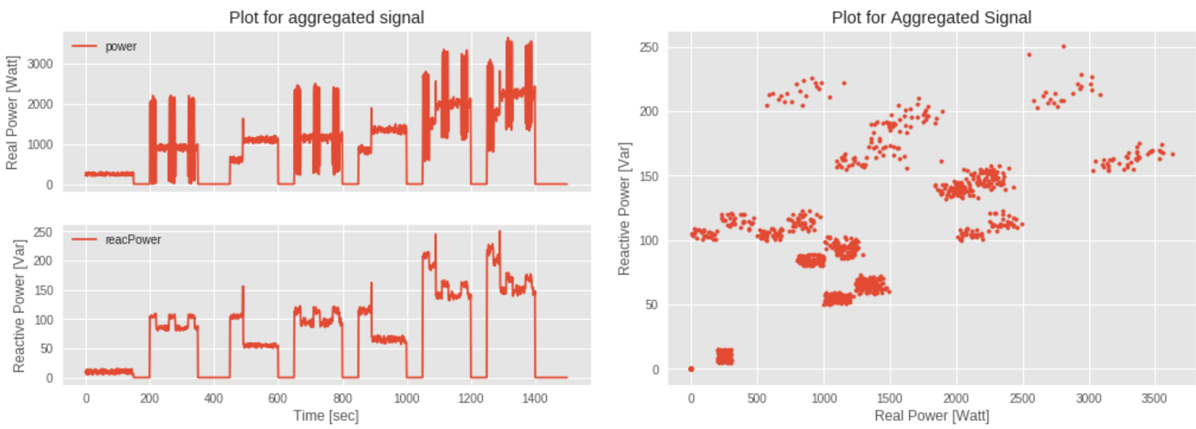


Fig 5 Aggregated Signal with 8 different states

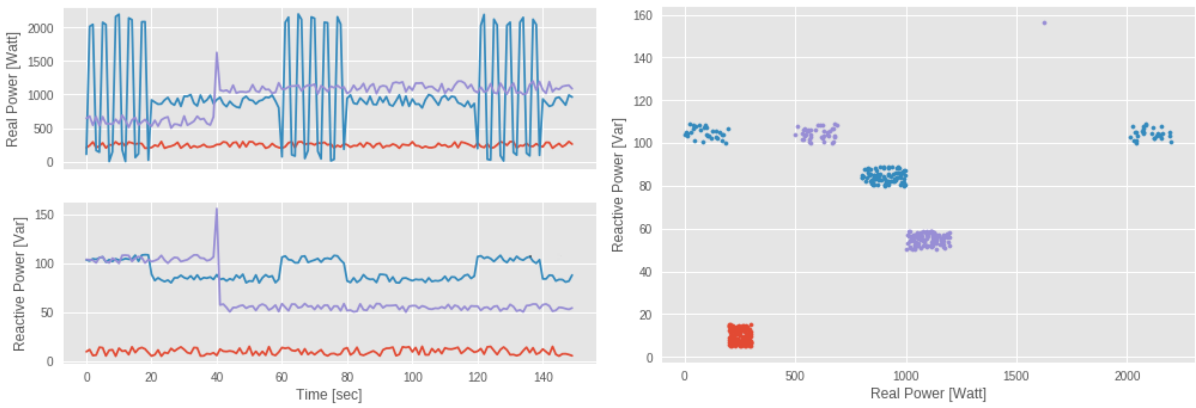


Fig 6 Individual Signals

Algorithm

Step 1: Creating the Gaussian Cluster for each Appliances

We take appliance data separately and then create Gaussian Cluster for each appliance. As the appliances can have different active states, we use Gaussian Mixture Model (GMM) to automatically find out how many states are there in each appliance and then create Gaussian cluster for each state separately.

The Clustering step is sub-divided into 2 steps:

1. **Model Estimation**: We use Bayesian Information Criterion (BIC) to find the optimum number of active states in each appliance, and then use this information to create Gaussian Clusters. This helps us to determine the different states in the appliance automatically, and break down a multi-state appliance into multiple 1x state appliance. This helps us to even infer the operating state of the appliance from the aggregated data.

2. **Gaussian Clustering**: From model estimation step, the optimum number of clusters is given as an input to the GMM algorithm, and then it outputs the mean, variance, and weights of the cluster created for each appliance. Each cluster here represents a state in the appliance.

The following represents the different clusters for each appliance in the toy dataset.

Bulb

Clusters	[1]
Mean	$[253.05240384 \quad 9.91468583]$
Variance	$\begin{bmatrix} 759.531901 & -2.91335183 \\ -2.91335183 & 7.80815853 \end{bmatrix}$
Weights	[1.0]

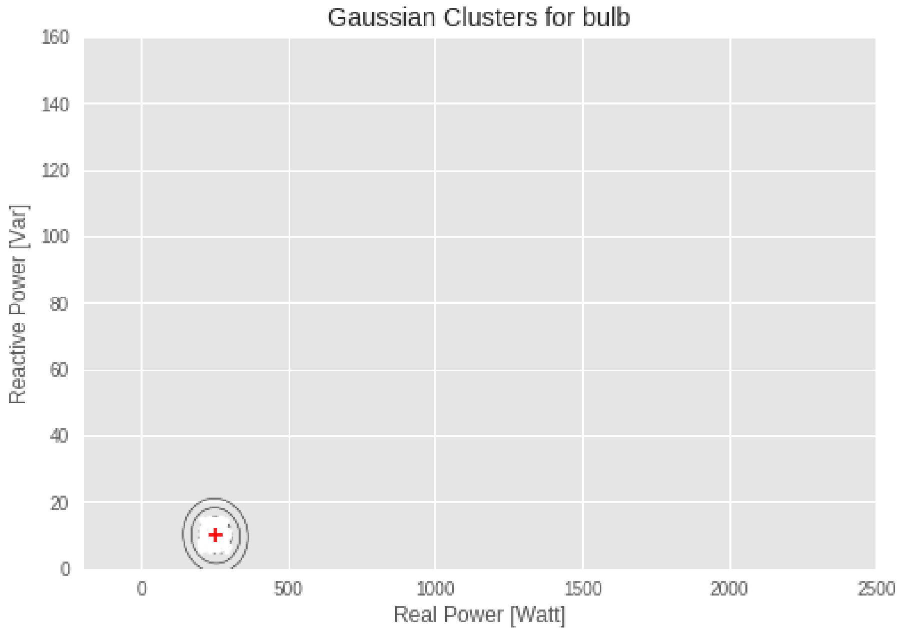


Fig 7 Clustering for Bulb (1 state)

FSM

Clusters	[3]
Mean	$[913.22214415 \quad 84.36302293]$ $[2088.64029322 \quad 105.00131204]$ $[92.30516159 \quad 103.33264773]$
Variance	$\begin{bmatrix} 3783.74628725 & 25.04697587 \\ 25.04697587 & 6.36370384 \end{bmatrix}$ $\begin{bmatrix} 2778.01425541 & -16.19696661 \\ -16.19696661 & 7.46487097 \end{bmatrix}$ $\begin{bmatrix} 4188.36823236 & -41.07530429 \\ -41.07530429 & 4.97288739 \end{bmatrix}$
Weights	$[0.6 \quad 0.2 \quad 0.2]$

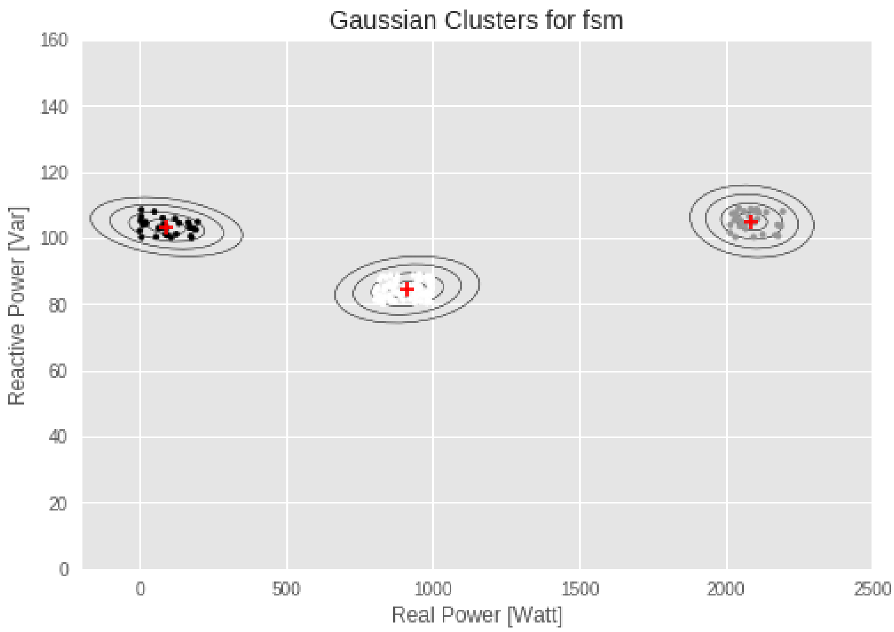


Fig 8 Clustering for FSM (3 states)

TSM

Clusters	[2]
Mean	[1108.18131771 55.94172721] [587.50238082 105.03044446]
Variance	[6102.57803965 482.08319271 482.08319271 88.33923515] [3149.18982934 9.21631013 9.21631013 6.59700206]
Weights	[0.73333333 0.26666667]

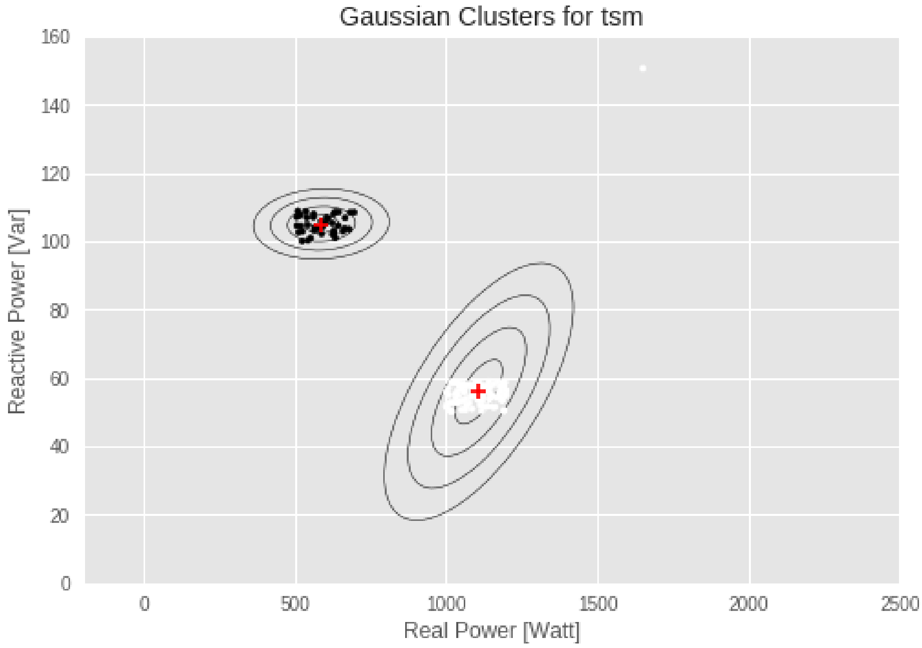


Fig 9 Clustering for TSM (2 states)

Step 2: Merging the clusters

In order to find all the possible combination of appliances that are running at a particular instance, we have to create all the possible combination of clusters from which the aggregated data could have been generated.
For the above 3 appliance with total 6 clusters, there are total 24 different combinations of clusters, called super clusters.

Cluster 1	Cluster 2	Cluster 3
0	0	0
Bulb	0	0
FSM1	0	0
FSM2	0	0
FSM3	0	0
TSM1	0	0
TSM2	0	0
Bulb	FSM1	0
Bulb	FMS2	0
Bulb	FMS3	0
Bulb	TSM1	0
Bulb	TSM2	0
FSM1	TSM1	0
FSM1	TSM2	0
FSM2	TSM1	0
FSM2	TSM2	0
FSM3	TSM1	0
FSM3	TSM2	0
Bulb	FSM1	TSM1
Bulb	FSM1	TSM2
Bulb	FMS2	TSM1
Bulb	FMS2	TSM2
Bulb	FSM3	TSM1
Bulb	FSM3	TSM2

We take the advantage of the additive property of Gaussians -: the sum of two normally distributed random variable will also be normally distributed.

Let say X and Y be two random variable given by,

$$\begin{cases} X\ N(\mu_x, \Sigma_x) \\ Y\ N(\mu_y, \Sigma_y) \\ Z = X + Y \end{cases}$$

then,

$$Z\ N(\mu_x + \mu_y, \Sigma_x + \Sigma_y)$$

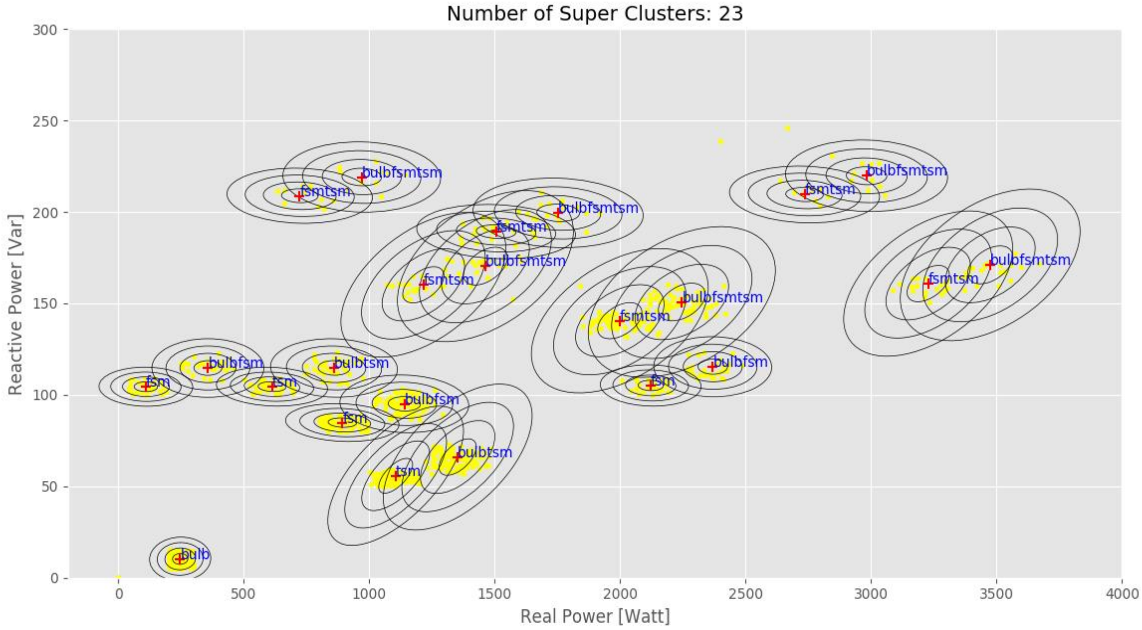
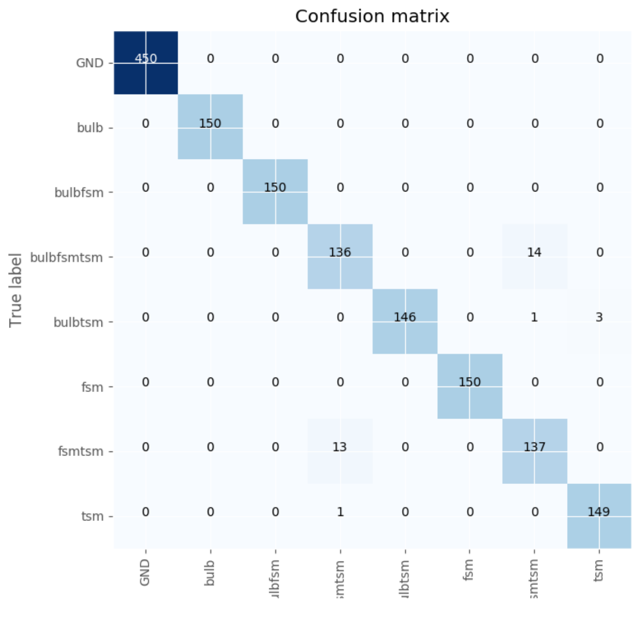


Fig 10 Super clusters after merging

Step 3: Inference

On the aggregated data, we then run maximum likelihood algorithm at each data point from the aggregated data to check, across all 24 super clusters, from which cluster the data point would have been generated. There are totally 8 different cluster from which the aggregated data belongs. Out of 24 merged cluster, the data is predicted correctly from those 8 clusters.



True label	GND	bulb	bulbfsm	bulbfsmtsm	bulbtsm	fsm	fsmtsm	tsm
GND	450	0	0	0	0	0	0	0
bulb	0	150	0	0	0	0	0	0
bulbfsm	0	0	150	0	0	0	0	0
bulbfsmtsm	0	0	0	136	0	0	14	0
bulbtsm	0	0	0	0	146	0	1	3
fsm	0	0	0	0	0	150	0	0
fsmtsm	0	0	0	13	0	0	137	0
tsm	0	0	0	1	0	0	0	149

The F-score is he harmonic mean of precision and recall. Precision is the time in which an appliance was correctly predicted to ON that it was OFF and Recall is the time which the appliance was correctly predicted ON that it was actually ON. They are calculated using metrics True Positive (TP), False Positive (FP), True Negative (TN) and False Negative FN).

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

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

$$F - score = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

The F-score for the toy dataset:

1. Bulb: 0.9832775919732442
2. FMS: 0.9983361064891847
3. TSM: 1.0