



HAL
open science

A blind event-based learning algorithm for non-intrusive load disaggregation

Moomal Qureshi, Christian Ghiaus, Naveed Ahmad

► To cite this version:

Moomal Qureshi, Christian Ghiaus, Naveed Ahmad. A blind event-based learning algorithm for non-intrusive load disaggregation. *International Journal of Electrical Power & Energy Systems*, 2021, 129, pp.106834. 10.1016/j.ijepes.2021.106834 . hal-04411255

HAL Id: hal-04411255

<https://hal.science/hal-04411255v1>

Submitted on 23 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A blind event based online learning algorithm for non-intrusive load monitoring

Moomal Qureshi, Christian Ghiaus, Naveed Ahmad

Univ Lyon, CNRS, INSA-Lyon, Université Claude Bernard Lyon 1, CETHIL, UMR5008, F-69621, Villeurbanne, France;christian.ghiaus@insa-lyon.fr;naveed.ahmad@insa-lyon.fr

* Correspondence: moomal.qureshi@insa-lyon.fr

Abstract

Non-intrusive loading monitoring (NILM) provides a smart solution to the problem of electrical energy monitoring of households at appliance level. In blind disaggregation, the power level of each appliance is not known a priori. In this paper we propose an event based blind online disaggregation algorithm which uses Gaussian mixture models (GMM) for clustering to automatically detect two state appliances from the aggregate data. The benefit of using a Gaussian mixture models over other clustering methods is that they are able to learn automatically the statistical distributions present in the data. This is beneficial, especially when we have appliances that have similar power consumption. Since Gaussian mixture models do not determine the number of clusters automatically, we use Bayesian information criteria (BIC) to determine the number of clusters. The blind disaggregation method is tested with data from a real house collected by a smart meter which samples the aggregate consumption at 3.4 kHz and also from Reference Energy Disaggregation Dataset (REDD) public data, sampled at frequency of 1 Hz. We saw improved performance by using Gaussian mixture models instead of mean shift clustering in various accuracy measures. We were able to track the total power consumed 12 % better then by mean shift clustering; root mean square error is also considerably lower on our measured data. The performance of the algorithm on the public data is comparable to other methods.

Keywords: Non-intrusive load monitoring, real-time disaggregation, unsupervised, online disaggregation, Gaussian mixture models

1 Introduction

Load monitoring provides real time feedback by analyzing the aggregate power data captured by the smart meter and extracting the information about the load devices present at the consumer. The goal of load monitoring is to provide a real time feedback about the consumption of the energy in order to detect problems and to encourage energy saving behaviors. Annual saving of up to twelve percent is reported by providing a direct feedback to the consumer about the consumption of energy [1].

Load monitoring can be divided into intrusive and non-intrusive. In intrusive load monitoring, smart meters or plugs are attached to each appliance to monitor the individual consumption. In non-intrusive load monitoring (NILM) a single smart meter, which measures the aggregated consumption of the whole house, is used to extract information about the consumption of individual appliances present in a household. NILM does not require installing and maintaining additional equipment, such as sensors that monitor the individual consumption of the appliances. The earliest research on non-intrusive energy monitoring was done by G. W. Hart in late 1980 [2]. Non-intrusive load monitoring (NILM) can also enable the energy supplier to anticipate the demand and schedule the resources appropriately. Over the years, there has been an increasing interest in appliance load monitoring, especially on non-intrusive load monitoring.

40 2 State of the art

41 The main steps for non-intrusive load monitoring include data acquisition, event identification, feature
42 extraction and load identification [3, 4]. Data acquisition determines the rate at which the aggregate power
43 is measured, which can be low frequency (1 Hz or less) or high frequency (kHz). Most of the non-intrusive
44 load monitoring algorithms require a sampling frequency of 1 Hz [5]. Although higher frequency sampling
45 can improve the recognition capabilities of the disaggregation algorithm (as more detailed features can be
46 extracted from the data), high overhead cost is incurred [6].

47 Data acquisition is followed by event identification. An event is a change in the operational state of an
48 appliance. An appliance functions in a number of states depending on which category the appliance belongs
49 to. There are four categories of appliances present in a household [2]:

- 50 – Type 1: two state appliances that have an ON and an OFF state such a lamp.
- 51 – Type 2: multi state appliances that have a finite number of operating states like a washing
52 machine.
- 53 – Type 3: variable power appliances such as a light dimmer.
- 54 – Type 4: constant consumption appliances like an internet box [3, 5].

55 Event-based approaches use the recognition of the change of state of an appliance which results in state
56 transition edge. A pair of these transitions — positive and negative — makes an event [7, 8]. Each category
57 of appliance has a consumption pattern that is unique, called appliance electrical signature. Appliance
58 signatures have been studied widely for identification of individual appliance from aggregate load
59 measurement [5]. Non-intrusive load monitoring algorithms use two main categories of features: transient
60 and steady state features. Transient features are observed in the duration when the appliance is changing
61 state and exist for a short time, such as the duration of the change, harmonics, shape and amplitude of the
62 transient [5]. Steady state signatures are extracted when the appliance has made the transition and it is in a
63 specific state of operation. Steady state features include active and reactive power, current and voltage
64 waveform. The resulting extracted features are analyzed for making inference about the electrical
65 consumption of the individual appliances.

66 The final phase of non-intrusive load monitoring is learning. It falls into three categories: supervised,
67 unsupervised or semi-supervised. Supervised algorithms are built on learning the individual behavior of an
68 appliance by using training data which is acquired intrusively with the help of sub-metering. Un-supervised
69 algorithms, on the other hand, require no individual training for determining the behavior of an appliance
70 from the aggregate data.

71 Recently, efforts in the community concentrate on the development of a real time disaggregation algorithm
72 that can infer the state of the individual appliances and indicate the total energy consumed by the appliance
73 from the aggregate measurements in an unsupervised manner [5]. Previously the aim was to improve the
74 accuracy which requires a long computation time which is inappropriate for real time applications [6, 4].

75 A real time implementation of a disaggregation algorithm is tied to certain constrains [9]. Firstly, an
76 unsupervised approach should be completely data driven as the collection of data for individual appliances
77 and user feedback is intrusive and unfeasible. Secondly, it is more desirable to design a low frequency
78 disaggregation algorithm with a sampling period of 1 Hz as smart meters are able to log and process this
79 sampling period effectively. Most of the commercial installations of smart meters transmission of data
80 greater than 1 Hz create an overhead in terms of both processing power and storage, requiring additional

81 infrastructure. Also, data collected at higher frequency adds to the processing time for the disaggregation
82 algorithm.

83 In an effort to solve the disaggregation problem in unsupervised fashion, various approaches using hidden
84 Markov models (HMMs) [10, 11, 12, 13], online learning-based approaches [14, 15, 16, 17] and deep
85 learning based techniques [18, 19, 20] have been developed. These unsupervised techniques fall into two
86 main categories: transfer learning and online learning. In transfer learning, generic appliance models are
87 built and then mapped to the collected data [11, 21]. Both hidden Markov and deep learning-based
88 approaches have been explored in literature and they have been found to be comparable [22]. Although
89 results of these approaches are good, they show low success when applied to other countries or geographical
90 regions [21]. The model of an appliance may be different from country to country or even for the same
91 country, if the appliance has various operating modes. This could also change the performance of the
92 algorithm [23]. A lot of data is required for training in deep learning methods. Therefore, transferring
93 learning-based methods is possible, but performance estimation in the presence of unknown appliances
94 could be detrimental to the performance [13]. In online learning based techniques, the appliance models are
95 created from data, without any prior information about the type of appliances. The benefit of using an online
96 learning solution is that it can adapt and learn the operating state and the behavior of the appliance. This
97 approach has not received much attention but can provide a reasonable solution [17]. Some of these
98 techniques are discussed hereafter. A blind online disaggregation algorithm was proposed that used graph
99 signal processing techniques, such as adaptive thresholding, for selecting a candidate event. Clustering and
100 feature extraction were developed by down-sampling public datasets at 1 min and another at 8 sec resolution
101 [14]. A blind disaggregation technique proposed for Type 1 appliances by using subtractive clustering,
102 which is a density-based clustering technique, was demonstrated to be more robust in the presence of noise
103 than k-means or hierarchical clustering [15]. An online learning method based on hidden Markov model
104 (HMM) using Daubechies 9 (db9) discrete wavelet filter was used to filter the low frequency noise and the
105 apparent power in the transient state was employed for distinguishing between different appliance models
106 [16]. An event based strategy using mean shift clustering identified the power profiles in the data; then,
107 from the obtained power profiles, appliance models were learnt using additive factorial hidden Markov
108 models [24]. A filter pipeline was developed to clean the signal and to minimize the variation for event
109 detection by building appliance models using Gaussian probability distribution and a knapsack algorithm
110 for disaggregation [17].

111 Density-based clustering methods, such as mean-shift clustering and subtractive clustering techniques, are
112 robust in the presence of noise in the data [25, 15]. These clustering techniques have been adopted in
113 literature because they automatically determine the number of clusters.

114 Gaussian mixture models support mixed membership, i.e. data can belong to one or more clusters.
115 Therefore, using Gaussian mixture models could be an attractive technique when there are overlapping
116 cluster points. We propose an approach similar to Henao et al. (2017), but choosing Gaussian mixture model
117 as our clustering technique. Usually mean shift clustering is selected as the choice of clustering technique
118 in unsupervised algorithms because it automatically determines the number of clusters in the data. We
119 compare the results of Gaussian mixture models with mean shift clustering and found improved
120 performance using Gaussian mixture model.

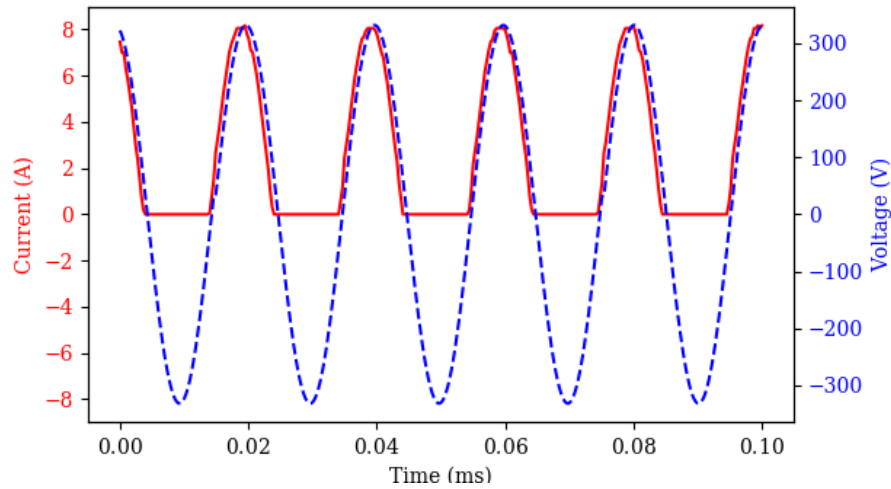
121 In this paper, we develop a framework for real time online disaggregation that can monitor appliance
122 consumption in real time and for which the learning process is completely data driven. The experimental
123 setup for the data collected is discussed in section 3. The methodology is discussed in section 4. The results
124 of the validation, by using experimental data collected in a real situation as well as the public data, are
125 presented in section 5.

126 **3 Experimental setup**

127 The aggregate data is measured by Eco-Touch[®] smart meter [26]. The meter samples the input current
128 through a current transformer at the frequency of 3.4 kHz. The meter measures only the positive cycles of
129 the current. There are around 3400 samples per second out of which half of the values for the current signal
130 are zero (Figure 1).

131 Due to electromagnetic interference (EMI), sometimes the number of samples is fewer than 3400 samples
132 per second. Usually, the number of samples per 20 milliseconds vary, being 67 or 68 samples. Therefore, a
133 synchronization mechanism is required to track the signal. Synchronization is achieved by using an Infra-
134 Red Emitting Diode (IRED) coupled with a phototransistor to track the phase of the voltage. The current

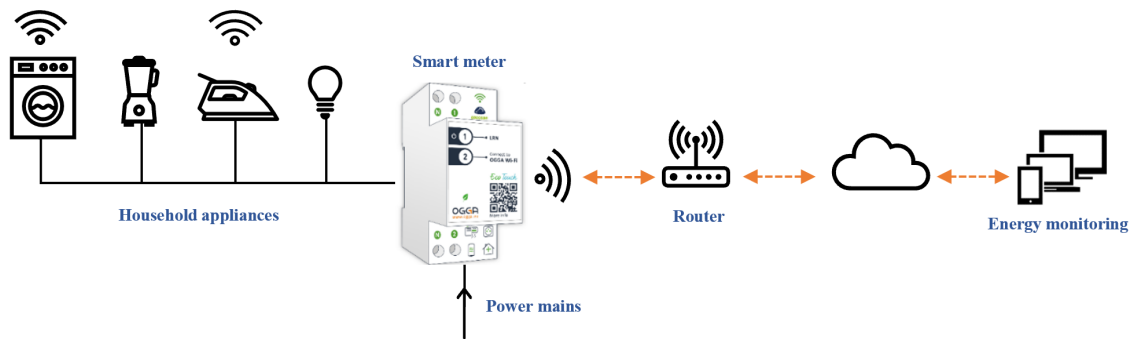
135



136

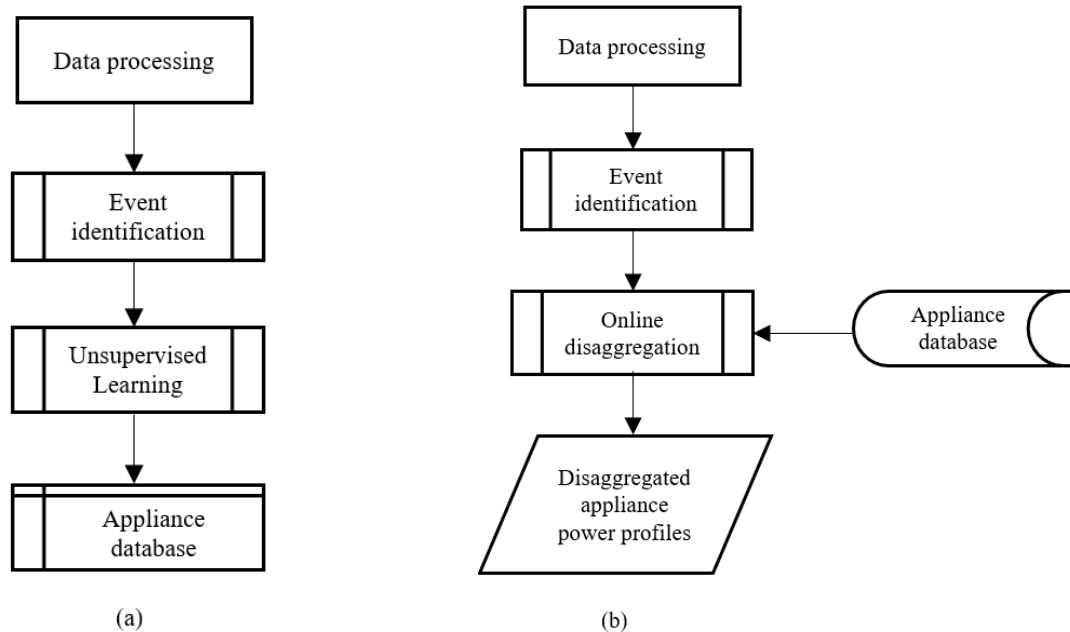
Figure 1. Current and voltage synchronization signal values.

137



138

Figure 2. Data acquisition setup using ECO Touch.



140 Figure 3. Framework of the disaggregation algorithm: a) offline appliance modeling; b) online disaggregation.

141 and the voltage values are shown in Figure 1. The current values are displayed in ampere while the voltage
 142 varies with the phase of the current between 332 to -332 volts.

143 The collected data is transmitted to a server on the internet which stores the aggregate consumption of the
 144 house. In addition, the smart meter is capable of collecting individual appliance consumption. It uses
 145 NodOn[®] smart plugs that communicate with Eco-Touch[®] by EnOcean protocol [27]. The data for the
 146 disaggregation algorithm is collected using the experimental setup shown in Figure 2. Eco-Touch is
 147 connected to the power mains of the house, where the meter samples the aggregate consumption of the
 148 house. NodOn[®] smart plugs are used to measure the individual consumption of the device [27]. Although
 149 the majority of the appliances are connected with smart plugs, there are some appliances present in the
 150 house whose consumption is not monitored. The information from the smart plugs is used as ground truth
 151 to validate the disaggregation algorithm. The meter is further connected to energy monitoring devices where
 152 the aggregate consumption and individual consumption can be viewed on a web portal or a mobile
 153 application.

154 4 Methodology

155 The aim of the disaggregation algorithm is to identify the group to which appliances belong as a function
 156 of the power changes. We will assume that:

- 157 1. Only two state appliances are present in the aggregate power data, with states {ON, OFF}.
- 158 2. At a given moment, only one appliance changes the state from ON to OFF or from OFF to ON.
- 159 3. Appliance that have the same power consumption are categorized as the same group (they are
 160 considered to be identical).
- 161 4. If there are multi-state appliances, then each state is considered as a different appliance.

162 5. The states of the devices change in first in – first out (FIFO) sequence: if an ON change is detected
 163 for a device from a group, the first OFF change for that group is considered to be of the same
 164 device.

165 We define the state of the system as the combination of states of individual appliances. For example, if we
 166 consider three appliances in the aggregate data, then all possible states of the system are shown in Table 1.
 167 We define an event as the change of the state of a device from OFF to ON or from ON to OFF. An event
 168 changes the state of the system. The assumption 2° implies that only states in which one bit is changed are
 169 possible. For example, if the system is in state 0, then transitions to states 1, 2 and 4 are possible; if the
 170 system is in state 3, then transitions to state 1, 2 and 7 are possible.

171 Table 1. State combinations for three two-state appliances.

State	Appliance 1	Appliance 2	Appliance 3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

172 The disaggregation algorithm is divided into two parts (a) to identify group of events in the aggregated data,
 173 which we call offline appliance modeling (b) to track the power consumption by pairing the events and
 174 calculated the total consumption of the appliances which are in the cluster of that pair of events, which we
 175 call online disaggregation as shown in Figure 3.

176 We are using a low frequency event-based approach, but the data sampled at high frequency by the meter;
 177 therefore, in data processing we down sample the data measured at 3.4 kHz to 1 Hz. The processing of the
 178 data acquired by the meter is discussed in section 4.1.1. After processing, the input is passed to an event
 179 identification module.

180 An event detector identifies significant changes in power level during a training window. A training window
 181 corresponds to a duration of time in which we want to extract information from the data. This duration can
 182 be any length of data ranging from a few hours to a few minutes. To validate the algorithm, we have selected
 183 three windows, one for the public dataset REDD and two for our private dataset.

184 For the REDD dataset we have selected a training window of 24 hours. For our private data, we have tested
 185 the algorithm on two training time windows 1) a window of a few minutes to validate the algorithm for
 186 Type 1 appliances 2) a window of four hours consisting of aggregate data collected from a real house in
 187 which all four categories of appliances are present.

188 After all events are identified in the training window, the next step is to determine the number of clusters
 189 of events by unsupervised clustering using Gaussian mixture models (section 4.1.3). Both positive power
 190 changes (positive event) and negative power changes (negative events) are observed during event
 191 identification and it is assumed that all appliances are of Type 1. Instead of clustering positive and negative
 192 events separately, the clustering algorithm is trained using the absolute value of the event. This assumption
 193 eliminates the requirement of matching positive and negative event pairs.

194 After clustering the events, a Gaussian Bayes classifier is trained with the cluster labels to create an
 195 appliance database (section 4.1.4). The identified clusters are then used during online disaggregation to

196 determine which appliance is working. Once the data has been trained and the appliance database has been
 197 created, the online disaggregation module can be used to match the identified event with the appliance
 198 database. A classifier is used to determine the association of the events. Disaggregation is done once both
 199 ON and OFF events for an appliance are found, as explained in section 4.2.

200 4.1 Offline appliance modeling

201 4.1.1 Data processing

202 The data is collected in a residential building in France. The input signal to the smart meter is single phase
 203 power at frequency of 50 Hz. Eco-Touch[®] samples current at 3.4 kHz. As a low frequency event-based
 204 approach is used, the data is processed in order to calculate the power at 1 Hz sampling frequency. For
 205 voltage, the instantaneous value is,

$$v(t) = V_o \cos(\omega t). \quad (1)$$

206 and the instantaneous value of the current is

$$i(t) = I_o \cos(\omega t + \varphi). \quad (2)$$

207 where

- 208 t - time instant,
- 209 ω - angular frequency,
- 210 φ - phase angle;
- 211 I_o - amplitude of the current,
- 212 V_o - maximum power of the voltage.

213
 214 The energy in one cycle $[0, 0.02 \text{ s}]$ is calculated as using equation (2) and (1),

$$E = 2 \int_{t_1}^{t_2} v(t)i(t)dt. \quad (3)$$

215 The factor of 2 in equation (3) is to accommodate the fact that we have only positive cycles for the current
 216 signal. The average power is calculated as:

$$\bar{P} = \frac{1}{T} \int_0^T P dt \quad (4)$$

217 \bar{P} represents the total energy consumed in the duration T , which equals one second.

218 4.1.2 Event identification

219 An event is specified as a change of state of the power signal greater than a specified threshold. The change
 220 of state is characterized by the difference of the power signal,

$$\Delta P = \bar{P}[t] - \bar{P}[t - 1] \quad (5)$$

221 where ΔP is the difference between two successive values of the power signal at time t and $t - 1$, separated
 222 by a time step of 1 second.

223 ΔP represents an event when the difference between two successive samples is greater than a threshold
 224 value γ . If $\Delta P > \gamma$, then an ‘‘ON’’ event is observed. If ΔP is less than $-\gamma$, then an off event is observed.
 225 Type 1 appliances have only two states, ON and OFF. An ON state corresponds to a positive event and an
 226 OFF state corresponds to a negative event. A positive event is defined as [28],

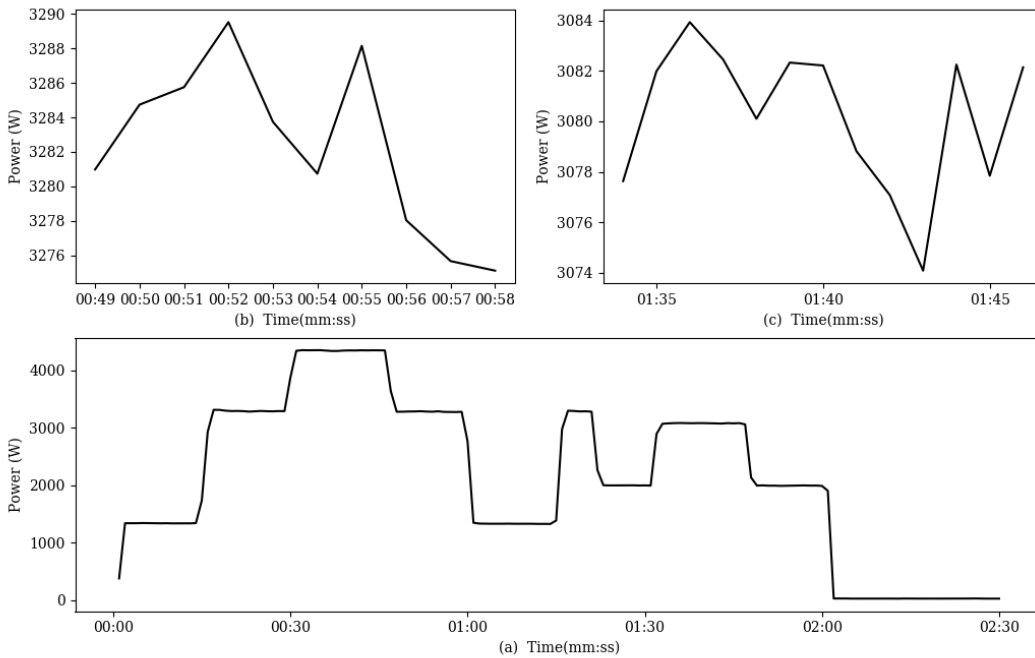
$$\Delta P^+ = \begin{cases} 1, & \text{if } \Delta P \geq \gamma \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

227 and a negative event is defined as

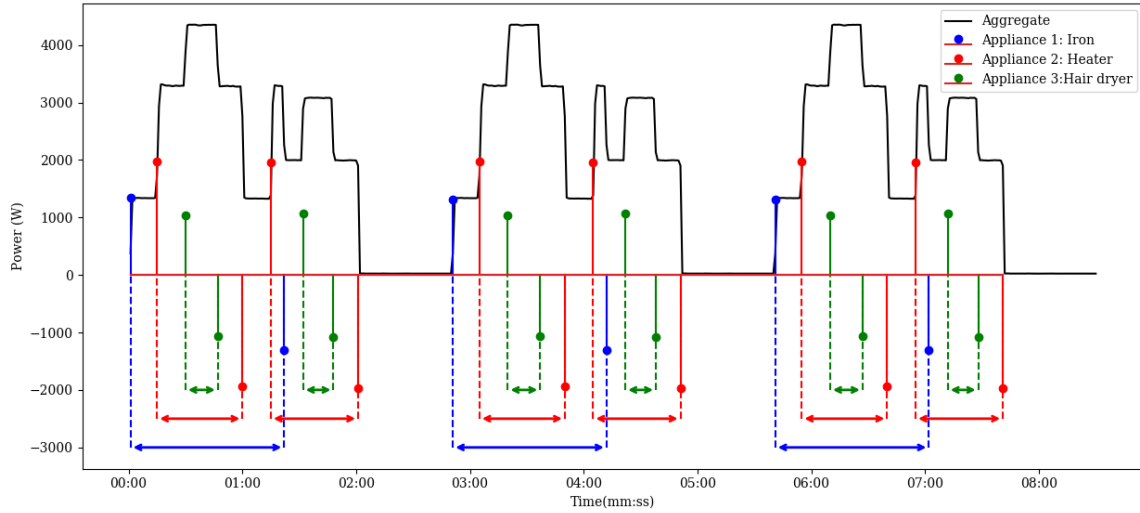
$$\Delta P^- = \begin{cases} 1, & \text{if } \Delta P \leq -\gamma \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

228 The observed events ΔP^+ and ΔP^- are related to the selected value of γ . Therefore, only changes greater
 229 than the specified γ are categorized as an event. It is assumed that a specific moment of time, only one
 230 appliance changes its state [28, 29].

231 For demonstration, we use a small dataset which consists of aggregate signal collected by EcoTouch[®] meter
 232 when three appliances: an iron, a heater and a hairdryer, were turned ON and OFF. The data was collected
 233 for a duration of around 8 minutes and the ground truth about the appliances operation during the time were
 234 manually collected. Since for this data set, the power of the devices is on the order of hundreds of watts and
 235 the average signal variance (noise) is 19 W, the threshold of $\gamma = 50$ W is selected to avoid the interference
 236 of noise. Figure 4 shows a plot of the aggregate power, as it is visible from the snapshots of the signal that
 237 selecting a value of threshold fewer then the noise of the aggregate power data will impact the results of
 238 the event detection.



239
 240 Figure 4. (a) Plot of aggregate power data from the toy dataset (b) snapshot of the aggregate power data in a time interval (c)
 241 another snapshot of the aggregate power data in a interval.



242

243 Figure 5. Event identification on the toy dataset. The solid lines show the positive and negative events, while the color distinguishes
 244 the cluster they belong to.

245

246 The aggregate power collected by the meter and the results of the event identification are shown in Figure
 247 5. First, the iron, the heater and the hairdryer are turned ON, which is seen as a positive power transition.
 248 Next, the hairdryer is turned off, which results in negative power transitions and so on.

249 4.1.3 Unsupervised learning of event clustering

250 In blind disaggregation, the number of appliances and their states are not known in advance. We make
 251 assumptions on how many appliances are present in the aggregate data. Some assumption made are that in
 252 the aggregate power data only two state appliances are present. Each appliance exists in only one state at a
 253 given time. Appliances that have similar ΔP^+ or ΔP^- are assumed to be the same appliance, meaning
 254 appliances that have similar power consumption are considered as the same appliance, according to
 255 assumption 2 in section 4. After event identification, we then proceed to categorize the events into clusters
 256 i.e., each two-state appliance has equal ΔP^+ and ΔP^- value so for clustering we use the absolute value of
 257 ΔP ,

$$x = |\Delta P| \quad (8)$$

258 If ΔP is used as such to identify the clusters, then we get two clusters, one for positive events ΔP^+ , for
 259 appliances turning ON, and one for negative events ΔP^- , for appliances turning OFF. By using equation
 260 (8), we can avoid an additional matching to group together ON events and OFF events.

261 4.1.3.1 Gaussian mixture model

262 K-mean algorithm is not flexible as it relies on the distance between the cluster centers making hard cluster
 263 boundaries as each value belongs to one of the distinct clusters [30]. Gaussian mixture models (GMM) is a
 264 probabilistic machine learning techniques that can provide better approximation when the clusters are
 265 overlapping as compared to k-mean clustering. In Gaussian mixture models, each cluster is modeled as a
 266 Gaussian distribution with the probability density function with a mean μ and variance σ defined as

$$N(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (9)$$

267 If there are K appliances in the data and each cluster represents one of the K appliances, then for a mixture
 268 of Gaussians we have k Gaussian distributions

$$p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \sigma_k), \quad (10)$$

269 where K presents the number of Gaussian mixtures and π_k is the probability of the k^{th} Gaussian in the
 270 mixture with the constraint that their sum equal to one

$$\sum_{k=1}^K \pi_k = 1, \quad 0 \leq \pi_k \leq 1. \quad (11)$$

271 Since we have a mixture of Gaussians, the parameters $\{\mu_k, \sigma_k$ and $\pi_k\}$ have to be determined. These
 272 parameters are estimated by using Expectation Maximization (EM) method [31]. EM is used to calculate
 273 the maximum likelihood estimate for the parameters. EM consists of an expectation step and a
 274 maximization step. In the expectation step the initial estimates for the parameters are calculated using k-
 275 means algorithm and the likelihood of the data is calculated. In the maximization step the parameters are
 276 modified to maximize the likelihood of the data with the initial assignments. These steps are repeated till a
 277 local optimum is achieved [32].

278 4.1.3.2 Identifying the total number of clusters.

279 Supposing that there are n observed events, we have to determine how many clusters k are present in our
 280 data. The number of clusters k are identified using Bayesian information criterion (BIC):

$$BIC = \log(n) K - 2 \log(L) \quad (12)$$

281 where

282 n - number of events;

283 K - number of parameters required for building the Gaussian mixture model;

284 L - maximum of the likelihood function of the Gaussian mixture model.

285

286 “ $\log(n) K$ ” is a penalty term added to overcome the effects of over-fitting as more variable are added to a
 287 model for maximizing likelihood. The minimum value of BIC is desirable for model selection. For a
 288 small dataset, BIC is not a good model selection technique [33]. Therefore, for our small experimental
 289 dataset, we do not use BIC to find the number of clusters; instead, we specified them manually. Plot of BIC
 290 versus varying number of cluster k is shown for the REDD public dataset Figure 6.

291

292 Once the total number of clusters are identified and the parameters for the Gaussian mixture model are
 293 determined, each event is assigned to a cluster using Bayes theorem which determines the membership of
 294 event x to a cluster G_i

$$p(G_i|x) = \frac{p(G_i)p(x|G_i)}{\sum_{j=1}^K p(G_j)p(x|G_j)}. \quad (13)$$

295 Alternatively, since each event belongs to one of the clusters, as described in equation (9) to (11), the
 296 membership of an event x to a cluster could be written as,

$$p(G_i | x) = \frac{\pi_i N(x | \mu_i, \sigma_i)}{\sum_{j=1}^K \pi_j N(x | \mu_j, \sigma_j)}$$

297

298 The details of the event clustering is identified in Algorithm 1.

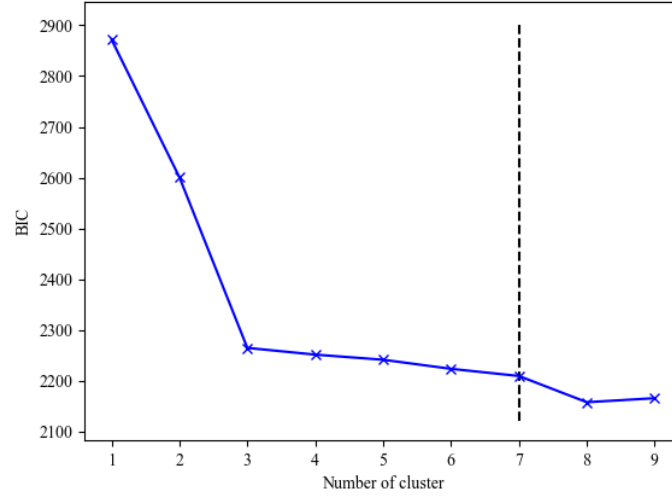


Figure 6. Plot of BIC versus the number of clusters k

299
300

Algorithm 1: Clustering the events using Gaussian mixture model

Input: *Events* and *Num_events_max***Output:** *Cluster Labels*

```

1  Data ← abs(Events)
2  Init_params ← "kmeans"
3  Covariance ← "full"
   Num_clusters ← 0
4  Labels ← [ ]
5  for event ← 1 to Num_event_max?
6     Cluster_model ← GaussianMixtureModel(number_components ← event, Init_params, Covariance)
7     Cluster_model ← Cluster_model.fit(Data)
8     Model_bic ← Cluster_model.bic(Data)
9     if Model_bic < Min_bic:
10        Min_bic ← Model_bic
11        Num_clusters ← event
12    end if
13 end for
14 cluster_model ← GaussianMixtureModel(Num_clusters, init_params, covariance)
15 Labels ← Cluster_model.fit_Predict(Data)
16 return Labels

```

301

302 4.1.4 Generation of appliance database

303 Each event ΔP^+ and ΔP^- in the training window is assigned to a cluster group, by assigning a cluster label
304 to each event. A cluster represents a grouping of appliances with the same power. For our toy dataset, there
305 are three clusters. The events belonging to the same cluster are highlighted using the same color as shown

306 on Figure 5. As the absolute values of the events are considered in determining the clusters, all appliances
307 that have similar power consumption are considered to belong to the same cluster.

308 The labels identified in 4.1.3 are assigned to the original identified events. The appliance database is created
309 by training a classifier with the events and their labels. During online disaggregation, events are assigned a
310 label by classification. For this implementation, a simple Gaussian Bayes classifier is used for classification
311 as described in equation (13).

Algorithm 2: Creating appliance database

```
1  Event[Labels] ← Labels
   Classifier_positive = GaussianNB()
   Classifier_negative = GaussianNB()
2  if Event > 0:
3     | Classifier_positive.fit(Event)
2  if Event < 0:
3     | Classifier_negative.fit(Event)
```

312

313 We train two classifiers: one for positive events and one for negative events. The steps of the training and
314 generation of appliance database are described in Algorithm 2. The appliance database is stored and used
315 during online disaggregation. The appliance database saves the time required to learn the appliance models
316 in the data.

317 4.2 Online disaggregation

318 The appliance database created during offline appliance modeling is used during online disaggregation for
319 tracking the consumption of the appliances in a disaggregation window. A disaggregation window is the
320 length of time for which we want to find the individual consumption. The online disaggregation follows
321 the method as shown in Figure 7. The identified events are matched with the appliances database created
322 in the training phase of the algorithm as described in section 4.1.4. Using the labels from the appliance
323 database, each event is matched with a label.

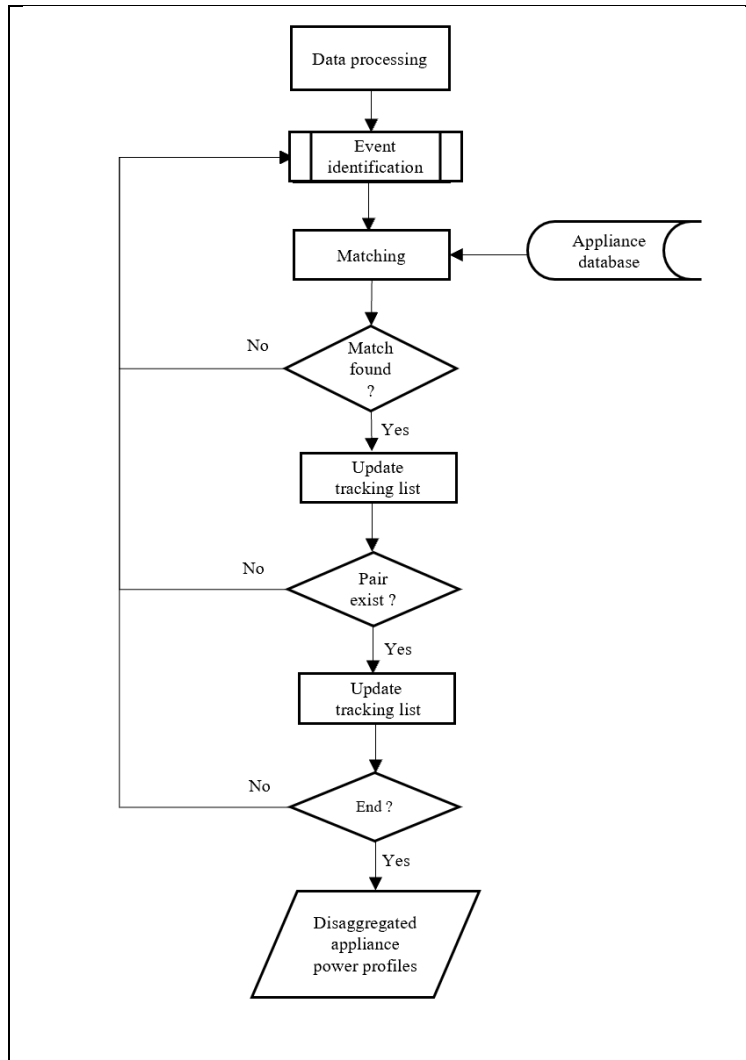


Figure 7. Detailed breakdown of online disaggregation.

324

325 Each positive and negative event is matched separately. If ΔP^+ is observed, then the event is matched with
 326 a classifier trained with positive events. And if ΔP^- is observed, then the event is matched with a classifier
 327 trained with the negative events. If a matching is found in the appliance database, then the event tracking
 328 list is updated. The event tracking list is then checked for an event pair. Each ON and OFF event constitutes
 329 a pair. If ΔP^+ and ΔP^- belong to the same cluster G_i and the time of transition of ΔP^+ is earlier than that of
 330 ΔP^- , then a pair is generated [2]. In short, if multiple matches are available for ΔP^+ , then ΔP^- closest in
 331 time is selected. The tracking list keeps an inventory on the identified events. The tracking lists contains a
 332 list of event pairs ΔP^+ and ΔP^- along with the label and the time stamp. Once the grouping has been done,
 333 these pairs are stored in the tracking list as shown in Table 2.

334

Table 2: Tracking list

Time on (epoch)	ΔP^+	Time off (epoch)	ΔP^-	Label
1593554137	1300	1593554223	-1289	1

335

336 Matching is continued till the end of the disaggregation window. For calculating the individual contribution
337 of appliances with the same label, the power consumed in each pair and duration gives the individual
338 consumption of the appliance.

339 Table 3. Results of disaggregation using data from the small dataset

Appliance	Mean power (W)	Duration (s)	Total energy consumed (kWh)
Heater	1958.18	273	0.148
Hair dryer	1059.84	99	0.029
Iron	1309.94	243	0.088

340

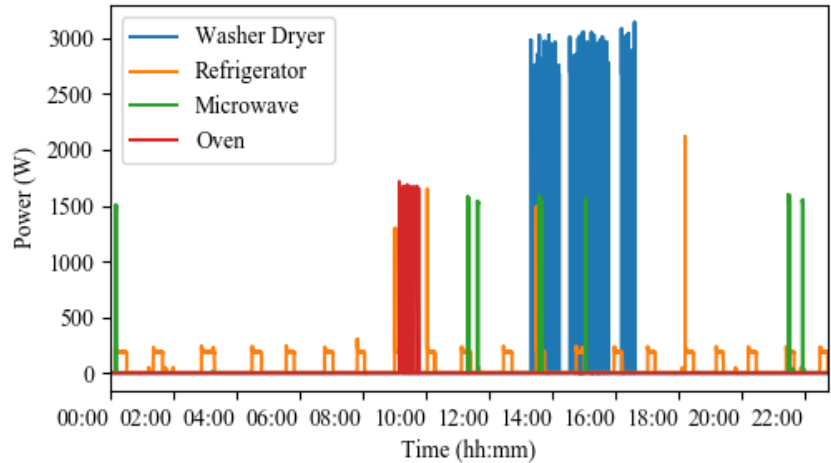
341 The results of the disaggregation for the small dataset are shown in Table 3. The mean power consumed by
342 each of the appliance along with the estimated duration and the amount of energy consumed is shown. Each
343 appliance corresponds to a cluster identified by the disaggregation algorithm. For the data in Table 3 the
344 cluster label is replaced with the appliance name.

345 5 Experimental validation of the proposed method

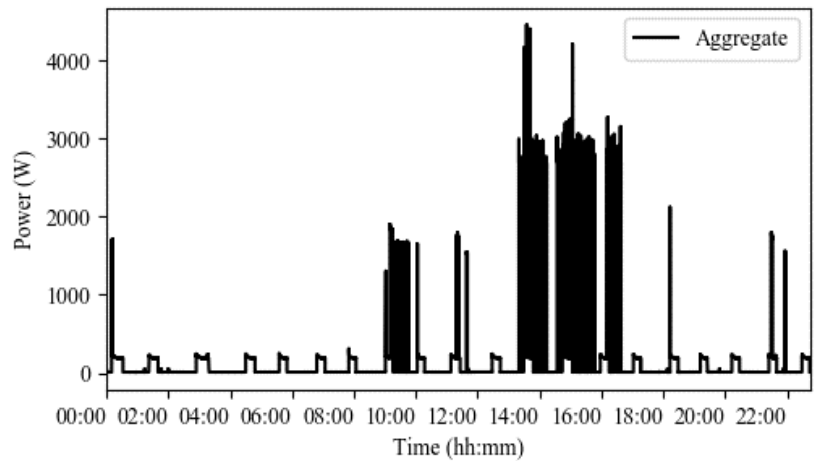
346 In this section, the results are validated in two different scenarios. In the first scenario, the algorithm is
347 validated on Reference Energy Disaggregation Dataset (REDD) for testing the performance of two state
348 appliances [34] In the second scenario, we report the disaggregation results of data collected by EcoTouch[®]
349 smart meter in a real house containing aggregate power data from various appliances in normal use.
350 Although we begin with the assumption that only two state appliances are present in the data, we further
351 inspect how well we can track the power levels in the aggregate data in the presence of multi-state
352 appliances.

353 5.1 Disaggregation results on REDD dataset

354 The Reference Energy Disaggregation Dataset (REDD) contains the aggregate power and individual plug
355 level power data for five houses in the US [34]. We use the low frequency power data to test the results of
356 the algorithm. The low frequency data consists of the average consumption recorded by the two power
357 mains and the individual plug data consumptions at 1 Hz. For the experiment, we selected plug level data
358 for four common appliances present in the dataset for a 24-hour window. The selected appliances are dryer,
359 refrigerator, microwave and oven. The consumption of these four appliances were extracted for a 24-hour
360 duration, as shown in Figure 8 (a). The power profiles of the individual appliance were then grouped
361 together to form the aggregate consumption as shown in Figure 8 (b).



(a)



(b)

Figure 8. REDD dataset (a) Ground truth power consumption of individual appliances (b) Aggregate consumption for a 24-hour window

362
363

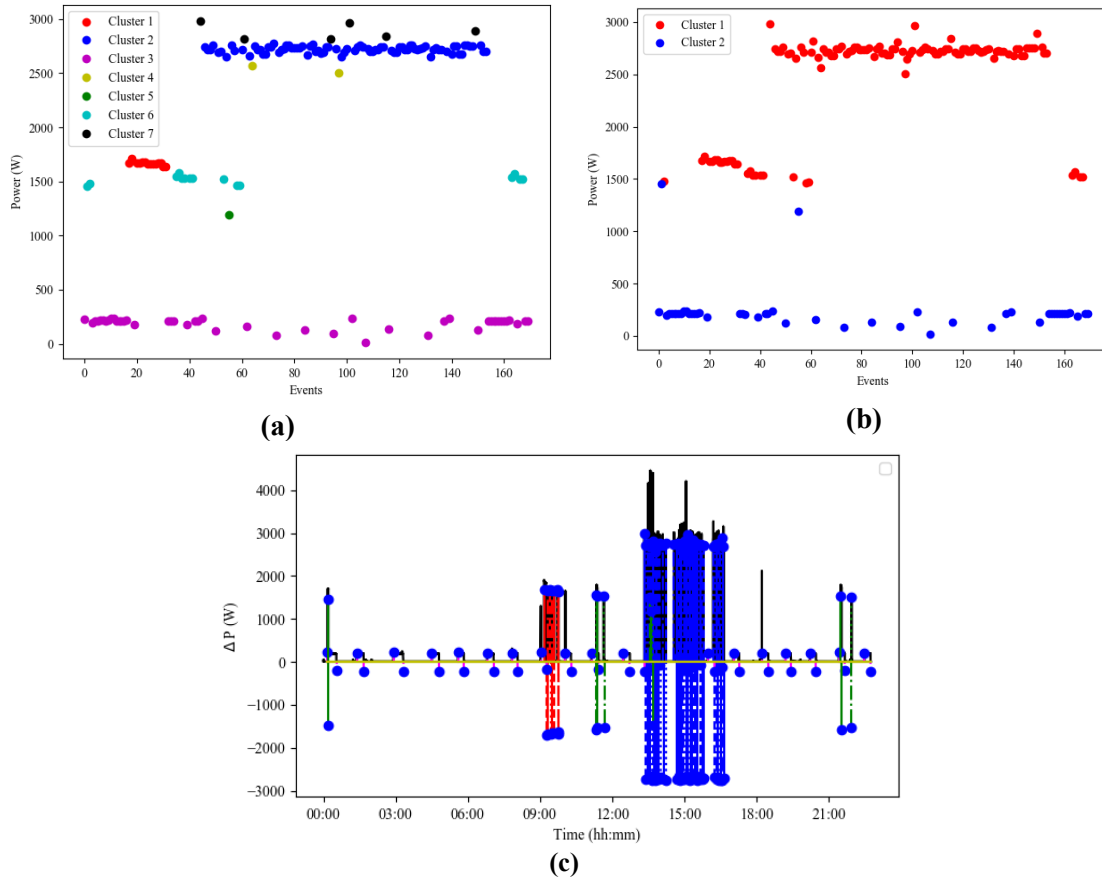
364 For event identification, threshold of $\gamma = 50 W$ is selected. The clusters identified by using unsupervised
 365 learning with Gaussian mixture models for clustering are shown in Figure 9 (a). Although there are four
 366 appliances, the unsupervised learning identifies seven cluster. There were only a few samples corresponding
 367 to three clusters, so they were pruned during pairing, so effectively we had only four clusters to work with.
 368 The identified clusters in relation to the observed events are shown in Figure 9 (c). The identified clusters
 369 are then saved as an appliance model during offline appliance modeling as discussed in section 4.1. Figure
 370 9(b) shows the identified clusters using mean shift clustering, using absolute value for the identified events.
 371 Mean shift clustering is a non-parametric based algorithm that automatically determines the number clusters
 372 based on the mean values of the data. Even if events were clustered without using the absolute value, we
 373 still had two clusters, one containing positive values and one containing negative values.

374

375

376

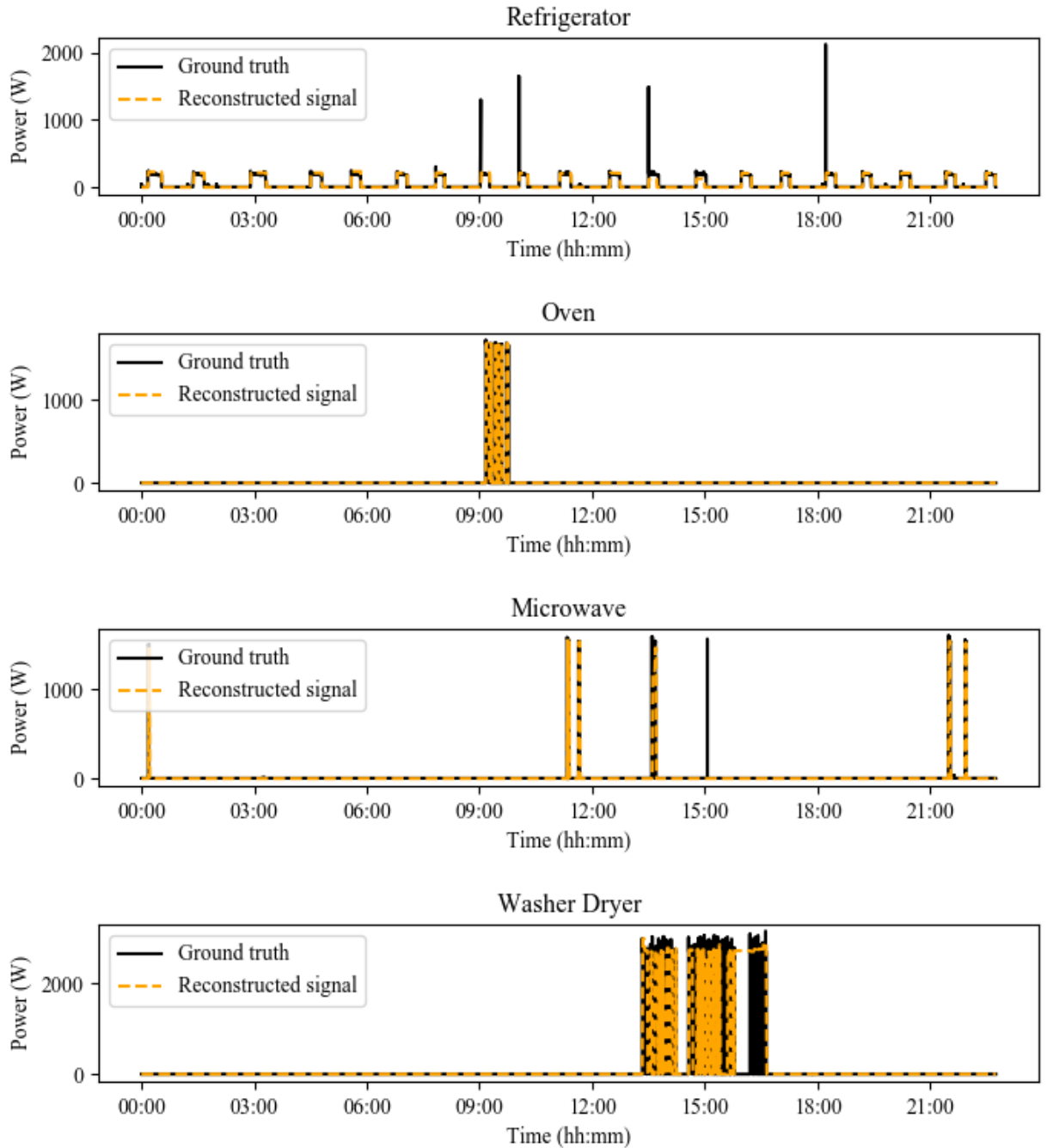
377



378 Figure 9 Results of the disaggregation (a) Identified clusters using unsupervised clustering with Gaussian mixture models.
 379 (b) clusters with mean shift clustering. (c) Identified clusters on the aggregate consumption using Gaussian mixture model.

380 We demonstrate the results of the online disaggregation using the REDD dataset. The ground truth signal
 381 and the reconstructed signal for the appliances is obtained by the online disaggregation is shown in Figure
 382 10. The reconstructed signals are obtained from the aggregate signal using the online disaggregation
 383 method, as described in section 4.2. For the refrigerator, it is visible that the majority of the ON and OFF
 384 power cycles are identified by the disaggregation algorithm from the aggregate signal. The results show
 385 that the disaggregation procedure is effective for the refrigerator, with the exception of the power
 386 consumption between 14:00 and 15:00. This is because multiple loads were active at the specific time in
 387 the aggregate data and some of the the power transitions were inaccurately identified.

388 Comparing the reconstructed signal in Figure 10 for microwave and oven, the results are quite good,
 389 although the power consumption of both appliances is similar. The algorithm was able to identify individual
 390 power consumption quite accurately. For the washer and dryer, the reconstruction consumption shows that
 391 some of the events were not identified, which propagated the error in power consumption.



392

Figure 10. Ground truth for refrigerator vs. the reconstructed power signal from REDD dataset.

393

We report the performance of the disaggregation in terms of estimated consumption accuracy, total energy
 394 disaggregated, and root means square error. The estimated consumption accuracy measure is a means of
 395 tracking the consumption of the appliances as [35, 10],

396

$$EC = 1 - \frac{\sum_{i=1}^N |y_i - g_i|}{2 \sum_{i=1}^N g_i} \tag{15}$$

397 where y_i is the predicted power and g_i is the ground truth for the respective appliance. N represents the
 398 length of the data. The factor of two in the denominator is to accommodate the double counting of errors
 399 due to absolute value in the numerator.

400 The total energy disaggregated is calculated as:

$$ED = 1 - \frac{|y_t - g_t|}{g_t} \tag{16}$$

401 where y_t is the total predicted consumption and g_t is the ground truth consumption of the respective
 402 appliance.

403 Root mean square is calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - g_i)^2}{N}} \tag{17}$$

404 The results of the metrics described for each of the identified clusters and the appliance they refer are
 405 shown Table 4. The disaggregation results are comparable to the disaggregation results reported in literature
 406 [14, 15].

407

Table 4. Appliance disaggregation results with REDD dataset

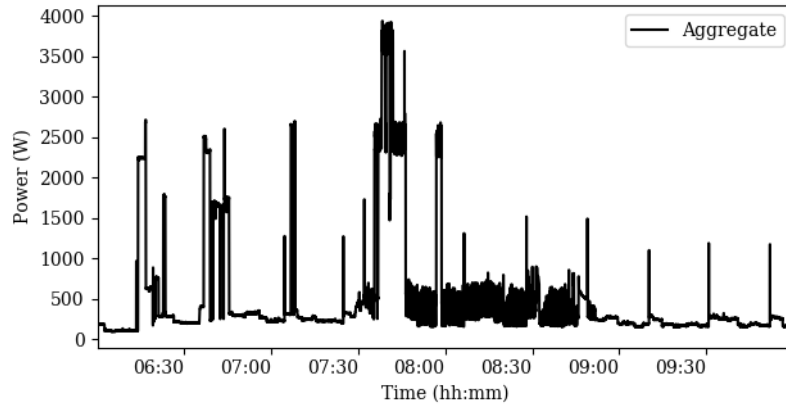
Appliance	EC	ED	RMSE
Washer dryer	0.51	0.84	708.60
Refrigerator	0.84	0.96	39.89
Oven	0.72	0.95	112.87
Microwave	0.49	0.97	128.71

408 5.2 Disaggregation results on data using Eco-Touch[®]

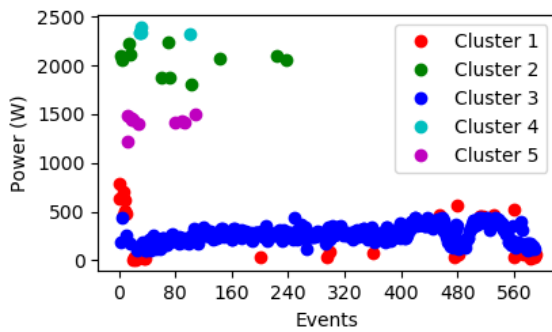
409 For validating the results of the disaggregation on the data collected by Eco-Touch[®], we have selected a
 410 window of four hour from a 24 hour window as shown in Figure 11(a). Although the intended use of the
 411 algorithm is for two state appliances, it is interesting to see how many power states could be tracked using
 412 data from a real house. In a typical household, usually there is a duration of time when people are active
 413 and power consumption is high. We have selected a window corresponding to high activity and power
 414 consumption. This is to inquire how successful we are in identifying the major power consuming appliances
 415 in the aggregate data.

416 Data acquired at the consumer site is processed using the method described in section 4.1.1. The aggregate
 417 data collected by the smart meter for the selected duration is shown in Figure 11(a). The results of section
 418 0-4.1.4 are shown in Figure 11(b). The threshold value γ is selected at 50 W for the experiment. We have
 419 a variety of appliances contributing to the aggregate consumption. The goal being the identification of the
 420 major power consuming appliances in the aggregate data. There is a direct relationship between the number
 421 of events identified and the time to process the data. If a smaller threshold is selected, then the processing
 422 time is increased.

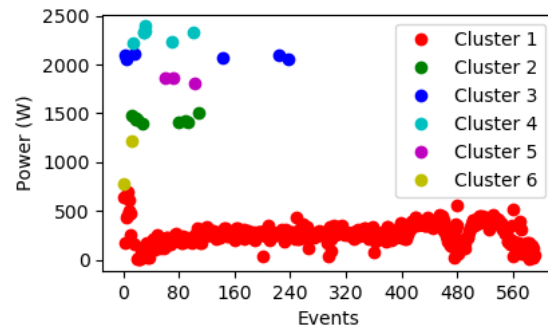
423 The results of the identified clusters on the aggregate data are shown in Figure 11(b). Each identified event
 424 is assigned to a cluster using the unsupervised method explained in 4.1.3. There are five clusters identified
 425 in the data, each cluster marked by a different color.



(a)



(b)

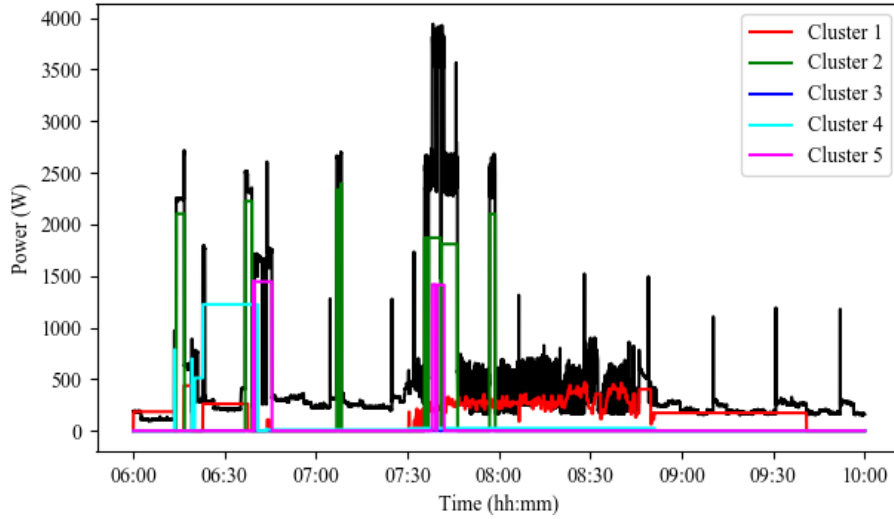


(c)

426 Figure 11. Data collected by the smart meter (a) Aggregate data over a 4-hour window, recorded by the smart meter. (b) Clusters
 427 identified with Gaussian mixture models (c) Clusters identified by mean shift clustering

428 We also compared the results of disaggregation by using mean shift clustering in unsupervised learning
 429 instead of Gaussian mixture model. There were six clusters identified by mean shift clustering shown in
 430 Figure 11(c).

431 The reconstructed power profiles by the online disaggregation and the aggregate consumption is shown in
 432 Figure 12. For calculating the accuracy for the disaggregation in this scenario, we use the equations (15),
 433 (16) and (17). We use the aggregate reconstructed power profile to compare with the aggregate power
 434 data instead of the ground truth from individual appliances.



435
436 Figure 12. Reconstructed power profile of individual states learnt from the data and the aggregate consumption.

437 The results of the disaggregation are presented in Table 5. The results show that the overall the performance
438 of GMM is better than mean shift clustering.

439 Table 5. Comparison of GMM vs mean shift clustering

Clustering algorithm	EC	ED	RMSE
GMM	0.67	0.93	219
Mean shift clustering	0.43	0.81	477

440 6 Conclusion

441 A blind unsupervised approach for non-intrusive load monitoring is presented using Gaussian mixture
442 model for two-state appliances. The benefits of using Gaussian mixture model are especially evident in the
443 situation where the data has mixed membership. There are many techniques that can be applied for
444 determining the number of clusters. We found that Bayesian information criteria provides a good reliable
445 measure to determine the number of clusters. It is further demonstrated that the algorithm can learn the
446 number of clusters effectively, thus enabling an unsupervised approach to the identification of appliance in
447 the aggregate data.

448 The blind online disaggregation algorithm is able to identify two-state appliance quite efficiently.
449 Experimental results show improvement in performance as compared to using other unsupervised methods.
450 On REDD public dataset, we showed successful identification of different power consuming appliances
451 operating during a day. The algorithm was even able to distinguish between appliances with almost similar
452 power consumption level.

453 We found that the approach also works in the situation where we have multi-state appliances in addition to
454 two-state appliances. We were able to track 93 % of the total energy consumed as compared to 81 %, which
455 was achieved using mean-shift clustering on the data from a real house where multiple appliances, including
456 multi state appliances, were working.

457 There are a few things that could be done to improve the disaggregation algorithm. Some of the
458 recommendation presented in the literature are the use of high frequency data and use of more features can
459 improve the performance of the disaggregation algorithm, a change point detection algorithm to
460 automatically categorize the data in real time, leveraging the power of deep learning algorithm [23, 22, 36].

461 The limitation of the work is that we are able to identify the grouping of power levels. No discrimination is
462 done between the appliances that have the same power. The disaggregation algorithm uses a threshold based
463 event detector. In the future, we would like to improve the event detection as well as incorporate prediction
464 of appliances of various categories.

465

466 Funding

467 The research was partly funded by Higher Education Commission (HEC), Pakistan and Campus France
468 (80%), through a doctoral scholarship awarded to Moomal Qureshi and partly (20%) by OGGA France,
469 through a research contract.

470 Acknowledgement

471 The authors would like to thank Patrice Bouchand, Embedded Linux Software Engineer at OGGA for his
472 help and support for data collection.

473 References

474

- [1] K. Ehrhardt-Martinez, K. A. Donnelly and J. A. Laitner, "Advanced metering initiatives and residential feedback programs: a meta-review for household electricity saving opportunities," American council for an energy efficient economy, Washington D.C, 2019.
- [2] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, 80, pp. 1870-1891, 1992.
- [3] I. Abubakar, S. N. Khalid, M. W. Mustafa, H. Shareef and M. Mustapha, "Application of load monitoring in appliances' energy management–A review.," *Renewable and Sustainable Energy Reviews*, 67, pp. 235-245, 2017.
- [4] A. Ruano, A. Hernandez, J. Ureña, M. Ruano and J. Garcia, "NILM techniques for intelligent home energy management and ambient assisted living: A review," *Energies*, 12(11), p. 2203, 2019.
- [5] A. Zoha, A. Gluhak, M. A. Imran and S. Rajasegarar, "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey," *Sensor* 2012, pp. 16838-16866, 2012.
- [6] S. Barker, S. Kalra, D. Irwin and P. Shenoy, "'NILM redux: The case for emphasizing application over accuracy'," in *In NILM-2014 workshop*. , 2014.

- [7] R. S. a. B. Y. K. S. Barsim, "An approach for unsupervised non-intrusive load monitoring of residential appliances," in *Proc. of the 2nd Int.l Workshop on Non-Intrusive Load Monitoring*, Austin, TX, USA, 2014.
- [8] K. D. Anderson, "Non-intrusive load monitoring: Disaggregation of energy by unsupervised power consumption clustering.," Carnegie Mellon University, 2014.
- [9] C. Laughman, L. Kwangduk, R. Cox, S. Shaw, S. Leeb, L. Norford and P. Armstrong, "Power signature analysis," *Power and Energy Magazine*, 2003.
- [10] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proc. International conference on artificail intelligence and statistic*, La Palma, Canary islands, 2012.
- [11] O. Parson, S. Ghosh, M. Weal and A. Rogers, "'An unsupervised training method for non-intrusive appliance load monitoring,'" *Artificial Intelligence*, 217, pp. 1-19, 2014.
- [12] R. Jia, Y. Gao and C. J. Spanos, "A fully unsupervised nonintrusive load monitoring framework," in *Proc. of the IEEE Int. Conf. on Smart Grid Comm*, Miami, FL, USA, 2015.
- [13] S. Makonin, F. Popowich, I. V. Bajic and L. B. B. Gill, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on smart grid*, pp. pp. 2575-2585, 2016.
- [14] B. Zhao, L. Stankovic and V. Stankovic, "On a Training-Less Solution for Non-Intrusive Appliance Load Monitoring Using Graph Signal Processing," *IEEE Access*, pp. 1784-1799, 2016.
- [15] N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé and M. Fournier, "Approach in Nonintrusive Type I Load Monitoring Using Subtractive Clustering," *IEEE Transactions on Smart Grid*, pp. 812-821, 2015.
- [16] X. Huang, B. Yin, Z. Wei, X. Wei and R. Zhang, "An online non-intrusive load monitoring method based on Hidden Markov model," *Journal of Physiscs: Conference series*, (Vol. 1176, No. 4, p. 042036). IOP Publishing., 2019.
- [17] A. Rodriguez-Silva and S. Makonin, "Universal Non-Intrusive Load Monitoring (UNILM) Using Filter Pipelines, Probabilistic Knapsack, and Labelled Partition Maps," *arXiv preprint arXiv:1907.06299*, 2019.
- [18] J. Kelly and W. Knottenbelt, "Neural nilm: deep neural networks applied," in *Proc. of the 2nd ACM Int.l Conf. on Embedded Syst. for Energy-Efficient Built Environments - BuildSys '15.*, Seoul, Republic of Korea., 2015.
- [19] M. Z. C. Zhang, Z. Wang, N. Goddard and C. Sutton, "Sequenceto-point learning with neural networks for nonintrusive load monitoring," in *Proc. of the Thirty-Second AAAI Conf. on Artificial Intelligence.*, New Orleans, LA, USA., 2018.

- [20] L. Mauch and B. Yang, "A novel dnn-hmm-based approach for extracting single loads from aggregate power signals," in *Proc. of IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Shanghai, China, 2016.
- [21] D. Murray, V. Stankovic, S. Stankovic and a. S. S. Lulic, "Transferability of neural network approaches for low-rate energy disaggregation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [22] C. Klemenjak, S. Makonin and W. Elmenreich, "Towards Comparability in Non-Intrusive Load Monitoring: On Data and Performance Evaluation.," in *Proceedings of the 11th Conference on Innovative Smart Grid Technologies (ISGT), 2020*, 2020.
- [23] C. Nalmpantis and D. Vrakas, "Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparison," *Artificial Intelligence Review*, 52(1), pp. 217-243, 2019.
- [24] M. A. Mengistu, A. A. Girmay, C. Camarda, A. Acquaviva and E. Patti, "Cloud-Based On-Line Disaggregation Algorithm for Home Appliance Loads," *IEEE Transactions on Smart Grid*, pp. 3430-3439, May 2019.
- [25] A. Denton, "Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, 2005.
- [26] OGGA, "ECO Touch," 10 March 2016. [Online]. Available: <https://www.ogga.fr/eco-touch/>. [Accessed May 2020].
- [27] Nodon, "The EnOcean Metering Smart Plug," 5 January 2018. [Online]. Available: <https://nodon.fr/en/nodon/the-z-wave-smart-plug/>.
- [28] G. Hart, "Prototype Nonintrusive Appliance Load Monitor," MIT Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report, 1985.
- [29] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh and M. Srivastava, "Nilmk: an open source toolkit for non-intrusive load monitoring.," in *Proceedings of the 5th international conference on Future energy systems.*, 2014.
- [30] D. J. Bora, D. Gupta and K. Anil, "A comparative study between fuzzy clustering algorithm and hard clustering algorithm," *arXiv preprint arXiv:1404.6059*, 2014.
- [31] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm.," *Journal of the Royal Statistical Society: Series B (Methodological)*, pp. 39(1), 1-22, 1977.
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, (2), pp. 2825--2830, 2011.

- [33] K. P. & A. D. R. Burnham, "Multimodel inference: understanding AIC and BIC in model selection," *Sociological methods & research*, pp. 33(2), 261-304., 2004.
- [34] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *In proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, 2011.
- [35] S. Makonin and F. Popowich, "Nonintrusive load monitoring (NILM) performance evaluation," *Energy Efficiency*, p. pp. 809–814, 2014.
- [36] T. K. D. E. J. G. N. B. Z. O. & N. V. C. Nguyen, "Development of a real-time non-intrusive appliance load monitoring system: An application level model. International Journal of Electrical Power & Energy Systems," *International Journal of Electrical Power & Energy Systems*,, pp. 90, 168-180., 2017.
- [37] S. Giri and M. Bergés, "An energy estimation framework for event-based methods in Non-Intrusive Load Monitoring," *Energy Conversion and Management*,90, pp. pp.488-498, 2015.
- [38] T. M. Kodinariya and P. R. Makwana, "Review on determining number of Cluster in K-Means Clustering," *International Journal of advanced research in computer science and mangement science*, 1(6), pp. 90-95, 2013.

475

476

477