RESEARCH ARTICLE OPEN ACCESS

Application of Representative-Based Clustering Using Hartigan & Wong Approach to Discern Electrical Switching Patterns

Tushar Gupta [1], Somnath Saha [2], Prerna Gaur [3]
Research Scholar [1] & [2], Associate Professor [3]
Department of Instrumentation & Control
Netaji Subhas Institute of Technology
India

ABSTRACT

The amalgamation of microelectronics and complex algorithms has proliferated smart devices which are enabling human evolution at a rate never seen before throughout the history of mankind. In tune with this modern trend of creating intelligent machines, a system is envisioned that can autonomously monitor and control power devices. Such intelligent power management systems do not require a manual trigger in order to create the response. By employing unsupervised learning in conjunction with a network of sensors and relays, our system observes the user over a set learning period and understands patterns in usage of various electrical devices it aims to control. It then applies this understanding to operate without user intervention thus imparting intelligence to the ever ubiquitous electrical ecosystem.

A smart and efficient switching pattern detection and interpretation algorithm is developed based on Representative based clustering using Hartigan and Wong approach that can serve as the brain of such an intelligent power management system.

The algorithm is tested using data from real life scenarios and its output is compared against the expectation to benchmark performance thus yielding promising results that reinforce the practicality of such a system.

***Keywords:* -** Hartigan and Wong Approach, Representative based clustering, Power Automation, Machine Learning

I. INTRODUCTION

The 21st century has brought upon technology a centennial moment. The amalgamation of microelectronics and complex algorithms has proliferated smart devices which are enabling human evolution at a rate never seen before throughout the history of mankind. Age old inventions like the telephone today have abilities much beyond what their creators initially envisioned.

In tune with this modern trend of creating intelligent machines we present a system that can autonomously monitor and control power devices. Such systems will be a boon to the elderly and specially-abled people. They can not only save time and effort from repetitive tasks of switching appliances on and off but can also be programmed to detect deviations from learned patterns of electrical usage while the user is away and raise alerts in order to prevent unauthorized access to the environment in which they operate.

In this paper we develop a smart and efficient electrical switching pattern detection and interpretation algorithm based on Representative based clustering using Hartigan and Wong approach that can create a decision matrix of an intelligent power management system.

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). Given a dataset with n points in a d-dimensional space, $\boldsymbol{D} = \{x_i\}_{i=1}^{n}$, and given the number of desired clusters k, the goal of representative-based clustering is to partition the dataset into k groups or clusters, which is called a *clustering* and is denoted as $C = \{C_1, C_2, \dots, C_k\}$ [1]. Further, for each cluster C_i there exists a representative point that summarizes the cluster, which we compute by taking the mean (also called the *centroid*) μ_i of all points in the cluster, that is,

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_i} x_j \qquad (1)$$

Here, $n_i = |C_i|$ is the number of points in cluster C_i.

A brute-force or exhaustive algorithm for clustering is simply to generate all possible partitions of n points into *k* clusters, evaluate some optimization score for each of them, and retain the clustering that yields the best score. The *exact* number of ways of partitioning n points into *k* nonempty and disjoint parts is given by the *Stirling numbers of the second kind*, given as

$$S(n,k) = \frac{1}{k!}\sum_{t=0}^{k}(-1)^t\binom{k}{t}(k-t)^n$$

Informally, each point can be assigned to any one of the *k* clusters, so at most k^n combinations are possible. However, any permutation of the k clusters within a given clustering yields an equivalent clustering; therefore, there are $O(k^n/k!)$ Distinct clustering of n points into *k* groups. It is clear that exhaustive enumeration and scoring of all possible clustering is not practically feasible.

Table 1: List of symbols used and their meanings

Symbol	Meaning
μ_i^t	The centroid of ith cluster at 't' iteration of algorithm
C_j	The jth cluster detected by the algorithm
x_j	A input data point of algorithm
$SSE(C)$	Sum of squared errors for cluster C
A_{l^*}	Appliance denoted by l^*
N	Learning period
T	Tolerance factor
k	Optimal Number of Clusters

Given a clustering $C = \{C_1, C_2, \ldots, C_k\}$ we need some scoring function that evaluates its quality or goodness. This *sum of squared errors (SSE)* scoring function [1] is employed to measure the distortion for each cluster and it is defined as

$$SSE(C) = \sum_{i=1}^{k}\sum_{x_j \in C_i}\|x_j - \mu_i\|^2 \qquad (3)$$

The goal is to find the clustering that minimizes the SSE score:

$$C^* = \arg min\,\{SSE(C)\} \qquad (4)$$

k-means employs a greedy iterative approach to find clustering that minimizes the SSE objective. As such it can converge at local optima instead of a globally optimal clustering. It initializes the cluster means by randomly generating *k* points in the data space. This is typically done by generating a value uniformly at random within the range for each dimension. A single iteration of K-means consists of two steps: (1) cluster assignment, and (2) centroid update. Given the *k* cluster means, in the cluster assignment step, each point $x_j \in \boldsymbol{D}$ is assigned to the closest mean, which induces a clustering, with each cluster C_i comprising points that are closer to μ_i than any other cluster mean. That is, each point x_j is assigned to cluster C_{j^*} where, (2)

$$j^* = \arg \underset{i=1}{\overset{k}{min}}\left\{\|x_j - \mu_i\|^2\right\} \qquad (5)$$

In terms of the computational complexity of K-means, we can see that the cluster assignment step take $O(nkd)$ time because for each of the n points we have to compute its distance to each of the *k* clusters, which takes d operations in d dimensions [1]. The centroid re-computation step takes $O(nd)$ time because we have to consider a total of n d-dimensional points. Assuming that there are t iterations, the total time for *k*-means becomes $O(tnkd)$. In terms of the I/O cost, it requires $O(t)$ full database scans because we have to read the entire database in each iteration.

II. PRIOR WORK

Electrical Automation Systems have been a boon to mankind since its inception. It is a popular domain of research and is continually growing. However it has benefitted majorly the industrial domain, the developed countries and the affluent only. At the household level, the popularity of such systems is not equal in comparison but its growth is significant [2]. For the household applications, such devices majorly come as plug-in modules with a variety of add-on features. A lot of work has been done related to portability, feasibility, flexibility and inter-networking of such systems.

Numerous work related to home automation systems can be found, such as ZigBee-based systems [3], internet-based wireless automation systems [4], expandable home automation systems [5] etc. But the response of all such systems is based on user commands. Is it possible to have electrical systems that can work independently after observing user behavior? The paper strives to create such a system. A complete web based system for controlling and monitoring a network of electrical appliances is developed. User behavior is monitored on this system and the data is collected centrally. A simple yet efficient algorithm is developed based on representative based clustering to discern switching patterns from the collected data.

The clustering problem has also been worked on for a long time. It has been addressed in many contexts and by researchers in many disciplines. Cluster analysis has been used extensively since the 1970s by social scientists [6], market researchers [7], numerical taxonomists [8], and many others. Cluster analysis is also at the heart of many current approaches in machine learning, pattern recognition, and data mining. Clusters of related user switching data are recognized from collected user data and this is used by the systemto take decisions.

III. HARDWARE SYSTEM DESIGN

At the heart of the system, we have the microcontroller (ATMega 328P) that interacts with the relay unit and Ethernet module (Wiznet W5100). The relay unit is used to control the on/off states of the load appliances as instructed by the microcontroller unit. The Ethernet module is used to connect to the internet so that the on-off states of all the load appliances can be sent to the web. Furthermore, it can also help to take instructions from the web so as to change the state of a load appliance. Any device with connectivity to the internet can be used to monitor and alter the state of an appliance.

Thus, such a system provides the facility to remotely control and monitor all the devices connected within our network. All the data is saved on a centralized server on which representative based clustering is applied for recognizing switching patterns. The hardware system may be upgraded to behave according to the resultant patterns.

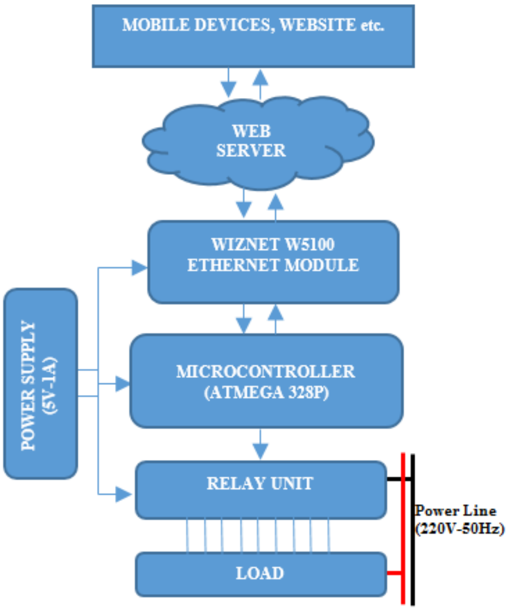
MOBILE DEVICES, WEBSITE etc.
WEB SERVER
WIZNET W5100 ETHERNET MODULE
POWER SUPPLY (5V-1A)
MICROCONTROLLER (ATMEGA 328P)
RELAY UNIT
Power Line (220V-50Hz)
LOAD

Figure 1: Hardware setup for data collection

IV. PSEUDO-CODE FOR ELECTRICAL SWITCHING PATTERN DETECTION ALGORITHM

```
K-Means (D, k):
  t = 0
  Randomly initialize k centroids: $\mu_1^t, \mu_2^t, \dots\dots, \mu_k^t \in \mathbb{R}^d$
  Repeat
        $t \leftarrow t + 1$
        //Array to track contents of cluster
        $C_j \leftarrow \emptyset \ for\ all\ j = 1, \dots, k$
        // Cluster Assignment Step
        For each $x_j \in D\ do$
        $j^* \leftarrow argmin_i \left\{ \| x_j - \mu_i^t \|^2 \right\}$ // Assign $x_j$ to closest centroid
        $C_{j^*} \leftarrow C_{j^*} \cup \{x_j\}$
        // Centroid Update Step
        For each i = 1 to k do
        $\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$

  Until $SSE(C^t) - SSE(C^{t-1})) \leq 0$

Cluster-Analysis (C, k, T):
  //Array to track pattern for '$a$' appliances
  $A_l \leftarrow \emptyset \ for\ all\ l = 1, \dots., a$
  For each $C_j \in C$ do
            For each $x_i \in C_j$ do
            $l^* \leftarrow \lceil (x_{ix}/N) \rceil$
            $A_{l^*}[\|\mu_{jy}\|] \leftarrow A_{l^*}[\|\mu_{jy}\|] + 1$
    For each $A_l \in A$ do
        For each n = 1 to 1440 do
            If $A_{l^*}[n] \geq T$ then
                $hours \leftarrow \left\lfloor \frac{MID}{60} \right\rfloor$
                $minutes \leftarrow MID - hours$
```

V. ELECTRICAL SWITCHING PATTERN DETECTION ALGORITHM

The algorithm developed aims at discerning electrical switching patterns which are defined as the recurrence in time at which an appliance was switched ON or OFF over a learning period during which usage data was recorded. It entails the following steps to complete its learning objectives.

1. Data translation
2. Computation of optimal number of centroids
3. Random initialization of centroids
4. Cluster assignment
5. Centroid reassignment
6. Optimization metric check
7. Cluster analysis
8. Data interpretation and application

These steps are elaborated in the following sections.

A. Data Translation

The algorithm requires usage data of each appliance being controlled by the system over a learning period of N days. This is recorded by the microcontroller and translated into a format that is appropriate for the algorithm to correctly visualize and identify clusters which later become part of the learned usage pattern that is applied to automate functioning of the appliance. We develop a visualization scheme for the data on a 2D planar graph (Figure 3) as follows:

1. We represent on the Y-axis a 24 hour day in 1440 intervals each of 1 minute.

2. We assign N consecutive points to each appliance on the X-axis, 1 for each learning day.

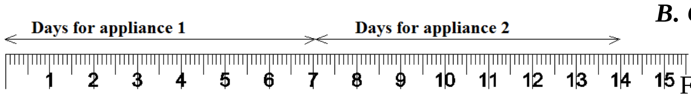

Figure 2: X axis division for allotting days to different appliances

For e.g. the points 1 to 7 on X-axis represent the days of observation for appliance 1, points 8 to 14 represent days for appliance and so on.

3. We introduce a new quantity called MID (minutes into the day) which depicts how many minutes into the day a particular appliance was switched on or off.

4. The data recorded by the microcontroller in a 24 hour format for each appliance over the learning period is approximated to within a minute and changed to MID.

Hence all the points on the graph can be quantified into the following set which becomes the input dataset '**P**' to the algorithm.

$$\boldsymbol{P} = \{(x, y), x \in X, y \in Y\}$$

Where,

X: Set of positive integers from 1 to learning period N times the number of appliances.

Y: MID at which the appliance changed state to on or off.

The algorithm also takes as input a tolerance factor '**T**' which defines the minimum representation an appliance should have in a cluster to correctly qualify for state switch at the MID defined by the cluster. This input ensures that the algorithm can be tuned to perform according to end user expectation.

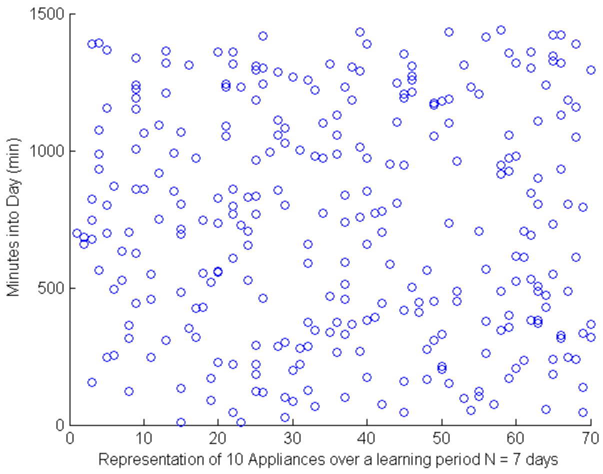

Figure 3: Graph showing visualization of randomly generated data for 10 appliances and N=7

B. Computation of Optimal Number of Centroids

For effectively deciding the optimal number of centroids *k* required to initialize the algorithm we analyze the worst case scenario of data distribution and the maximum number of clusters required in that case to correctly discern the input dataset. This case arises when 2 or more appliances represented consecutively on the X-axis by our visualization technique were not switched simultaneously at any point of time during the day over the learning period. In such a case the number of required clusters is given by Eq. 6 and keeping in mind this worst case we choose this value as the optimal number of clusters k.

$$k = \frac{|P|}{N} \qquad (6)$$

Where,

|P|: Cardinality of set P

N: Learning Period.

C. Random Initialization of Centroids

We employ a pseudo random generator to create the initial values of the centroids. This serves as the starting point for the algorithm and the centroids shift from these initial values to the center of each cluster as identified by the algorithm.

D. Cluster Assignment

Assign each $x_j \in D$ to the cluster whose centroid has the minimum Euclidean distance from the point. The Euclidean distance between x_j & μ_i^t is given by$\left\{\left\|x_j - \mu_i^t\right\|^2\right\}$. This is computed for k centroids and the minimum distance j^* is used to identify the cluster to which the point x_j needs to be assigned. The assignment is represented by Eq. 7.

$$C_{j^*} \leftarrow C_{j^*} \cup \{x_j\} \qquad (7)$$

Where C_{j^*} represents the cluster corresponding to the minimum distance j^*.

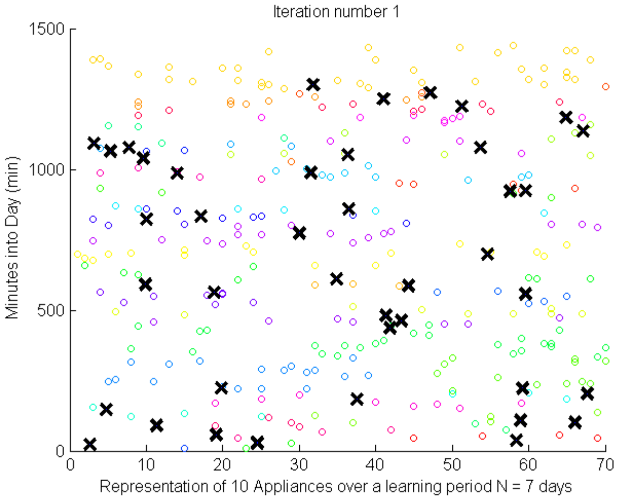

Figure 4: Plot demarcating the initial position of the centroids by 'X'

E. Centroid Re-Assignment

For each cluster k, the coordinates of the centroids are recomputed by taking the mean of all the points currently assigned to the cluster using Eq. 8. The centroid coordinates are then updated with these new values due to which the centroid shifts in the direction which minimizes distortion (Eq. 3) in each cluster.

$$\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \qquad (8)$$

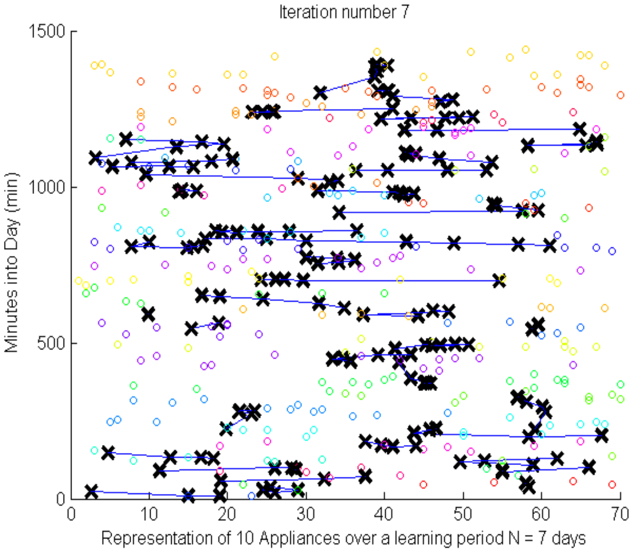

Figure 5: Centroid shift to minimize cluster distortion

F. Optimization Metric Check

The convergence of the algorithm is evaluated by minimizing the sum of squared errors score (Eq. 3) which indicates cluster wise distortion over a number of iterations of steps 5.4 & 5.5.

G. Cluster Analysis

The algorithm in its native design gives the centroids of all the clusters it detects as the output. We employ memorization over the native design to store the constituent points of each cluster in hash map to efficiently perform cluster analysis with the following two objectives:

1. Determine the switching time of appliance
2. Determine the appliances that comprise the cluster

To utilize the output of the algorithm for switching purposes, we establish the MID at which each appliance changed state from on to off or vice versa through the Y coordinate of the cluster centroid. Next to figure out all the appliances that represent a particular cluster, we evaluate the length of each cluster.

The length of the cluster is the range of the X coordinate of the constituent points of the cluster. We also estimate the representation each appliance has in the cluster to tune switching decisions according to the tolerance factor defined by the end user.

H. Data Interpretation and Application

The learning objective of the algorithm is analyzed under the below mentioned cases:

Case 1: State of the appliance depicted by the appliance

The clusters detected by the algorithm can be used to analyze whether the appliance was turned ON or OFF at the given MID.

This is set by the user and reflects the user preference to turn appliances ON or OFF at the learned MID. The control signals generated for the microcontroller are in accordance with the change of state required.

The user can also be provided with an option to record and analyze raw data for both ON and OFF state changes over the same learning period. The results of these two analysis can then be interleaved to completely automate when appliance turns ON and when it turns OFF.

Case 2: Appliance specific clusters

This scenario occurs when a cluster detected by the algorithm corresponds only to a single appliance i.e. the X coordinates of all constituent points represent one appliance over the learning period.

In this case the decision of state switching is taken according to the tolerance factor as follows:

1. $\left(\frac{\text{Length of cluster}}{N}\right) \geq T \rightarrow$ State of appliance switched at the corresponding MID.

2. $\left(\frac{\text{Length of cluster}}{N}\right) < T \rightarrow$ State of appliance NOT switched at the corresponding MID.

Case 3: Multiple appliances in single cluster

This scenario occurs when a cluster detected by the algorithm corresponds to multiple appliances i.e. the X coordinates of the constituent points represent multiple appliances over the learning period.

In this case the algorithm utilizes per appliance representation recorded in a particular cluster by step 7 and decides state of each of the constituent appliance in the cluster as follows:

1. $\left(\frac{\text{Length of cluster}}{N}\right) \geq T \rightarrow$ State of appliance switched at the corresponding MID.

2. $\left(\frac{\text{Length of cluster}}{N}\right) < T \rightarrow$ State of appliance NOT switched at the corresponding MID

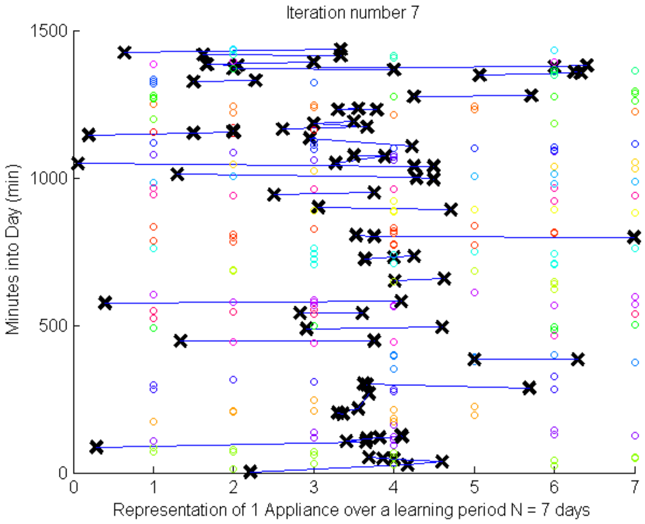

Figure 6: Clustering for a single appliance over a learning period of 7 days

Case 4: Single appliance learning over a longer period of time, say 30 days

In this scenario we can supply the usage data of a single appliance over a longer learning period and perform analysis around the usage frequency of the appliance over the period. Here the length of each cluster depicts the set of contiguous days over which the appliance changed state. The number of clusters detected by the algorithm is directly proportional to the frequency of usage of the appliance.

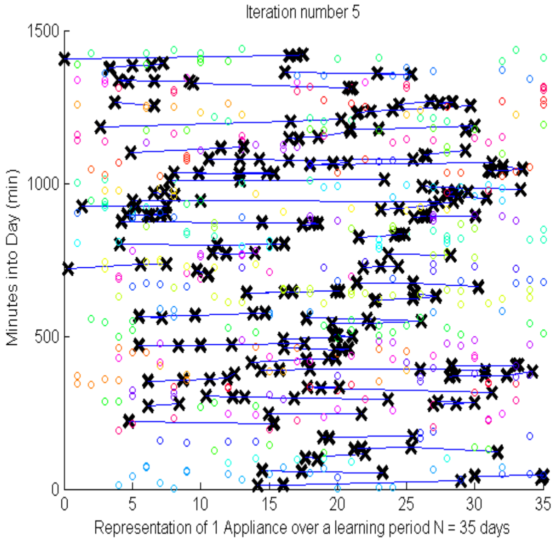

Figure 7: Clusters showing frequency of application usage over a period of 35 days.

I. Application of the Learned Pattern to Electrical Switching

On termination of the algorithm the output contains the set of appliances along with the MID values at which the appliances changed state to ON. The MID value is changed into the equivalent time of day and each time of day value is stored corresponding to the appliance whose state change it represents in a relational database management system. The control signals for the microcontroller are generated by processing the raw data from the database through a web service which then transfers these signals over the internet to the microcontroller. The microcontroller interprets these signals and controls electrical relays to accomplish the switching action.

VI. TESTING AND RESULTS

The testing of the algorithm is accomplished by beginning with a test case corresponding to the actual usage data of an electrical ecosystem. This scenario defines the following two things required for benchmarking the algorithm.

The output as desired by the user after the algorithm terminates.

The raw electrical usage data as recorded by the sensor network which is to be fed as input to the algorithm.

Table 2 shows the raw usage data for an electrical ecosystem of 4 appliances over a learning period N = 7 days. Figure 8 shows this raw data as plotted by the visualization scheme developed in section 5.1.

Table 2: Raw usage data as visualized by algorithm (Section 5.1)

Appliance Number	Day number	Minutes into the Day for ON cluster				
1	1	310	501	833	1250	1410
	2	312	500	839	1251	1411
	3	311	502	840	1248	1409
	4	314	505	835	1252	1412
	5	309	504	0	1253	1413
	6	0	499	837	1249	1415
	7	315	506	0	1254	1416
2	1	250	0	747	1193	1333
	2	255	0	752	1192	1334
	3	254	410	751	1190	1331
	4	253	411	748	1189	1329
	5	251	412	753	1191	1324
	6	250	415	0	1194	0
	7	250	414	749	0	1336
3	1	363	593	1011	1213	1430
	2	360	595	1013	1215	1435
	3	361	596	1015	1216	1434
	4	362	594	1017	1217	1431
	5	366	591	1018	1214	1429
	6	365	590	1014	0	1432
	7	367	0	1011	0	1431
4	1	150	630	943	1110	1353
	2	148	633	940	1112	1356
	3	147	639	945	1113	1357
	4	145	637	946	1115	1352
	5	0	635	941	1111	1351
	6	149	634	942	1110	1355
	7	150	631	0	1116	1355

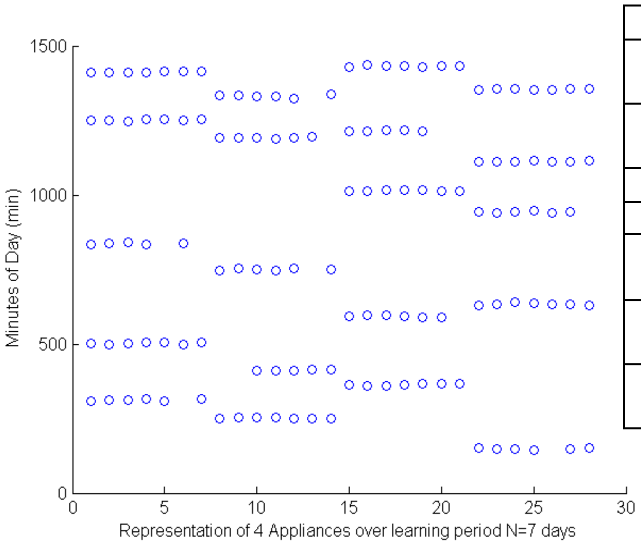

Figure 8: Raw usage data of 4 Appliances over a learning period N = 7 days

Table 3 shows the time of day at which the user expects switching to happen and the actual time according to the understanding of the algorithm at which the state is actually switched. The desired tolerance factor is set as $\frac{4}{7}$ for the purpose of this test case and clusters are interpreted as only ON clusters i.e. Case 1 in section 5.8.

Table 3: The switching pattern as understood by the algorithm

Appliance, ON Time – S. No.	MID as Expected by user	ON Time expected by user	Actual Learned MID	Actual ON Time learned by algorithm
1,1	311	05:11 am	312	05:12 am
1,2	501	08:21 am	502	08:22 am
1,3	836	01:56 pm	837	01:57 pm
1,4	1250	08:50 pm	1251	08:51 pm
1,5	1411	11:31 pm	1422	11:42 pm
2,1	250	04:10 am	252	04:12 am
2,2	411	06:51 am	412	06:52 am
2,3	750	12:30 pm	750	12:30 pm
2,4	1191	07:51 pm	1202	08:02 pm
2,5	1330	10:10 pm	1344	10:24 pm
3,1	362	06:02 am	363	06:03 am
3,2	592	9:52 am	615	10:15 am
3,3	1012	04:52 pm	981	04:21 pm
3,4	1214	08:14 pm	1202	08:02 pm
3,5	1431	11:51 pm	1422	11:42 pm
4,1	148	02:28 am	148	02:28 am
4,2	633	10:33 am	615	10:15 am
4,3	942	03:41 pm	981	04:21 pm
4,4	1112	06:32 pm	1112	06:32 pm
4,5	1355	10:35 pm	1344	10:24 pm

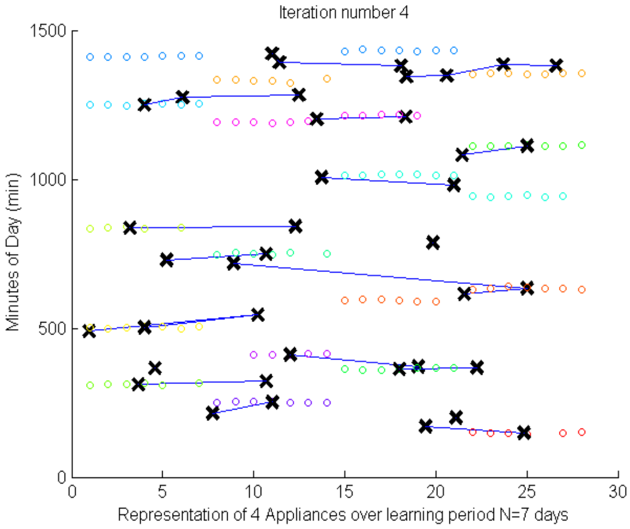

Figure 9: Clusters as detected by the algorithm

Figure 9 shows the centroid positions of the detected clusters and the path of traversal of the centroid from their initial randomly initialized position to their final position at which the sum of squared errors within each cluster is minimum.

The actual learned data when compared against expected values shows the alacrity of the algorithm in discerning the electrical switching patterns.

VII. CONCLUSION

We addressed the challenge of discerning electrical switching patterns on the basis of usage data collected by the sensor and relay network of the system. A smart and efficient switching pattern detection and interpretation algorithm using representative-based clustering has been developed and rigorously tested for its ability to correctly clone user activity after observing the user over a set learning period.

We are currently working towards incorporating continuous learning into the algorithm to capture and dynamically account for deviations from learned pattern that occur as user behavior changes over time.

REFERENCES

[1] M. Zaki and W. Meira Jr., "*Fundamentals of Data Mining Algorithms*", Cambridge, 2010.

[2] Krell Mike, "*What's Driving All The Home Automation Growth?*", Forbes, 2015. [Online]. Available: Forbes-http://www.forbes.com/sites/moorinsights/2015/07/15/whats-driving-home-automation/.

[3] Gill K., Shuang-Hua Yang, Fang Yao and Xin Lu, "*A zigbee-based home automation system*", Consumer Electronics, IEEE Transactions, Vol. 55, No. 2, pp. 422-430, 2009.

[4] Alkar A.Z. and Buhur U., "*An Internet based wireless home automation system for multifunctional devices*", Consumer Electronics, IEEE Transactions, Vol. 51, No. 4, pp. 1169-1174, 2005.

[5] Reuel O. Launey, Peter A. Grendler, Donald L. Packham, James M. Battaglia and Howard E. Levine, "*Expandable home automation system*", US 5086385 A, 1992.

[6] Ahlquist S. John and Christian Breunig, "*Model-based Clustering and Typologies in the Social Sciences*", Political Analysis (Winter 2012), Oxford University Press, Vol. 20, No. 1, pp. 92-112, 2012.

[7] G. Punj and D.W. Stewart, "*Cluster analysis in marketing research: review and suggestions for application*", Journal of Marketing Research, Vol. 20, No. 2 (May 1983), pp. 134-148, 1983.

[8] PHA Sneath and RR Sokal, "*Numerical taxonomy. The principles and practice of numerical classification*", Principles of Numerical Taxonomy, 1973.

[9] A. K. Jain, M.N. Murthy and P.J. Flynn, "*Data Clustering: A Review*", ACM Computing Surveys, Vol. 31, No. 3, 1999.