A Data-Driven BIRCH Clustering Method for Extracting Typical Load Profiles for Big Data

Anthony D. Fontanini and Joana Abreu

Abstract—**In typical load shape analysis, many different clustering methods have been used to segment customers, interpret behavior and inform marketing reach out strategies. Due to memory requirements and computational efficiency, many clustering algorithms do not have the capabilities to perform analysis at the urban-scale. In this paper, a scalable data-driven BIRCH clustering algorithm is used to extract the typical load shapes of a neighborhood. The BIRCH radius threshold is determined by solving an optimization problem. For global clustering, a metric is created that can rank the best possible options for the agglomerative phase of the BIRCH algorithm. The developed method allows large time series data at the urban-scale to be quickly analyzed**

Index Terms—**Clustering, BIRCH, smart meter data, typical load shapes, load profile.**

I. Introduction

THE current deployment of advanced metering infrastructures (AMI), and the consequent availability of large electricity consumption datasets, at hourly and sub-hourly resolution, allows for hidden trends in local electricity generation and consumption to be more intelligible. These readings can be analyzed and grouped together into typical load shapes, that can uncover hidden behavior, and segment customers. The load shapes can determine the timing of peak demand for a population, and support other forecasting tools for power distribution planning engineers that aim to model and forecast the and other parts of the distribution system [1], [2]. The results of the analysis have been used to inform better policy and tariff design in deregulated markets [3]. By understanding the typical behavior of the residences being metered [4], [5], each residence can be evaluated as potential candidates for demand response (DR) or energy efficiency (EE) programs [6], [7].

Many different methods have been used to cluster load shapes together and extract meaningful features in periods of peak demand [6] and energy use [8]. Some of these methods include, exclusive (k-means/k-medoids) [2], [6]–[8], overlapping (fuzzy C-means) [1], hierarchical (divisive or agglomerative) [6], neural networks [8], and others. However, in today’s rich databases of historical consumption timeseries, the current challenge is to find robust highly scalable data-driven methods for clustering load shapes at the urban-scale. The clustering methods should be both scalable in the number of samples and the number of clusters, be memory efficient, and naturally self-organize into a set of clusters. In the methods listed above, most of the algorithms do not scale very well to large number of samples [9] or the number of clusters needs to be known a priori.

To overcome the scalability challenges, the balanced iterative reducing and clustering using hierarchies (BIRCH) algorithm has been applied [10]. Advantages of the BIRCH algorithm includes that the method, a) clusters incrementally in a single sweep of the data; b) the whole data does not need to fit into memory; and c) the structure of the algorithm allows for merging and splitting of sub-clusters incrementally and consistently. The BIRCH algorithm creates a tree based on two inputs a radius threshold and a branching factor. The threshold parameter is often determined by heuristics, but other techniques exist: for example, some authors have recently use the gap-statistic to choose the threshold [11]. In their work, they mention that the gap-statistic is not easy and have developed an alternative approach to calculate the gap-statistic. If the branching factor is included in the optimization then the formulation moves away from a continuous optimization problem to a constrained integer optimization problem, which is NP-hard. Other authors have thus proposed to use a genetic algorithm to determine both input parameters [12]. However, the technique shows that for a small branching factor the number of clusters increases exponentially.

The purpose of this paper is to showcase the use of a scalable data-driven BIRCH clustering algorithm for automatic load shape extraction. The inputs are either chosen to the algorithm structure free or are formulated as an optimization problem. A set of cost functions for the optimization routine are examined for both load shape interpretability and contextualization for a set of quantities of interest (QOI). A salable clustering method is implemented and extracts load shapes for high, intermediate, and low consumption days. Using this clustering method allows for extremely large samples to be clustered efficiently and effectively and can be used for online learning and modification to the typical load shapes overtime.

This work was supported in part by the U.S. Department of Commerce under Grant BS123.

A. D. Fontanini is a Mechanical Engineer in the Building Enclosures Group at Fraunhofer CSE, Boston, MA 02210 USA (e-mail: afontanini@cse.fraunhofer.org).
J. Abreu is a Behavior Scientist, with the Building Energy Systems Group at Fraunhofer CSE, Boston, MA, 02210 USA (e-mail: jabreu@cse.fraunhofer.org).

II. Methodology

A. Smart Meter Data

The electrical consumption time series data was collected smart meters in Los Alamos, NM. The period of the study is between August 1st, 2013 and January 20th, 2016. During this period, there are 902 total days. There is a total of 1641 unique smart meters. Therefore, there are approximately 1.48 million daily consumption realizations to be clustered. The resolution of the consumption data used in this study is sampled hourly.

B. Mathematical Preliminaries

The electrical consumption of a smart meter, p, can be described by a discrete time series, $s_p(t): t = 1: n_p$, with n_p uniformly sampled realizations. In vector form the time series becomes Eq. 1.

$$S_p = \left[s_p(1), s_p(2), \dots, s_p(n_p)\right] \quad (1)$$

This time series can be collected into a matrix, $\boldsymbol{l}_p \in \mathbb{R}^{N_p \times 24}$ where $N_p = n_p/24$, of daily load shapes where the entries of the matrix are calculated in, Eq. 2.

$$l_p(i,j) = s(24[i-1]+j), i = 1: N_p\ , j = 1: 24 \quad (2)$$

The total number of load shapes is then $N_{tot} = \sum_{i=1}^{p} N_i$. Then the load shapes for all the smart meters can be collected into a complete load shape matrix $\boldsymbol{L} \in \mathbb{R}^{N_{tot} \times 24}$, Eq. 3. Where the rows indicate the load shape sample realization and the columns the hourly features of the daily load shape.

$$\boldsymbol{L} = \left[\boldsymbol{l}_1, \boldsymbol{l}_2, \dots, \boldsymbol{l}_p\right]^T \quad (3)$$

C. Preprocessing Data

From the raw data, the data is preprocessed in three steps before the clustering is performed Fig. 1. Following aspects of previous work of S. Xu et al. [7], the authors point out that many analysis of typical load shapes neglect overall consumption of the load shape, peak time and peak overlap. In their analysis, the load shapes were separated into three groups a high consumption days, intermediate (Inter.) consumption days, and low consumption days. In this study, these same groups are created by using the 25th percentile (9.53 kWh) and 75th percentile (25.18 kWh) daily consumption values. The authors also make a case for smoothing the consumption data to compensate for noise [7]. This study uses SSA [13] to filter the time series and the corresponding load shapes, $\hat{\boldsymbol{L}} = f(\boldsymbol{L})$, Fig. 2. The number of components to reconstruct the time series was determined by the Kaiser rule [4], [5]. After filtering, the load shapes are normalized [6], [7], Eq. 4.

$$\tilde{\boldsymbol{L}} = \tilde{L}(i,j) = \frac{\hat{L}(i,j)}{\sum_{j=1}^{24} \hat{L}(i,j)}\ , i = 1: N_{tot}\ , j = 1: 24 \quad (4)$$

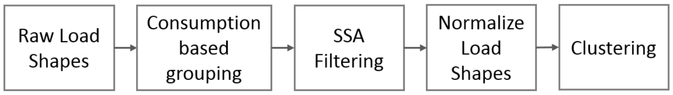

Fig. 1: The intermediate steps in the preprocessing procedure from the raw data load shapes to the clustering algorithm.

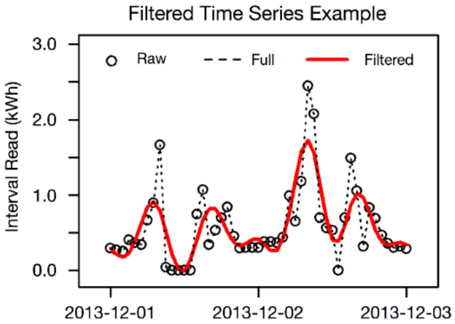

Fig. 2: An example of the result of the SSA filtering over a two-day period. The "Full" represents all components being reconstructed. The "Filtered" is based on the Kaiser rule to choose the number of reconstructed components.

D. Clustering Metrics

Before clustering the load shapes, a set of metrics are needed to be in place for determining the input parameters of the BIRCH algorithm and evaluating the clustering performance. The goal of the clustering algorithm is to create a set of k clusters, $\boldsymbol{C}_q$ with $q = 1: k$ and where points in the cluster are load shapes from $\tilde{\boldsymbol{L}}$. Each cluster has a centroid, c_q. The standard notion of a cluster is that the within cluster distance be small (dense cluster) while the between cluster dispersion be large (well separated clusters), Eq. 5 and Eq. 6.

$$W_k = \frac{1}{N_{tot}} \sum_{q=1}^{k} \sum_{l \in C_q} \sqrt{\sum_{i=1}^{24} \left[l(i) - c_q(i)\right]^2} \quad (5)$$

$$B_k = \frac{1}{k} \sum_{q=1}^{k} \sqrt{\sum_{i=1}^{24} \left[c_q(i) - c(i)\right]^2} \quad (6)$$

In Eq. 5, c is the centroid of $\tilde{\boldsymbol{L}}$. These two metrics have been used in a few previous studies [3], [14] and are used to determine the input parameters of the BIRCH algorithm in this work. To evaluate the clustering performance this study looked at using the silhouette coefficient, SMI indicator, entropy, the cluster size standard deviation, and the mean estimated threshold from Kwac. et al. [6]. Of these indicators, the measures of entropy, the cluster size standard deviation, and the estimated threshold showed some promise for determining the cluster performance, Eq. 7 – Eq. 9 respectively.

$$E_k = -\frac{1}{N_{tot}} \sum_{q=1}^{k} p(c_q) \log\left(p(c_q)\right) \quad (7)$$

$$\sigma_k = \sqrt{\frac{\sum_{q=1}^{k} \left[\mathrm{card}(\boldsymbol{C}_q) - \frac{1}{k} \sum_{i=1}^{k} \mathrm{card}(\boldsymbol{C}_i)\ \right]}{n-1}} \quad (8)$$

$$\hat{\theta}_k = \frac{1}{N_{tot}} \sum_{q=1}^{k} \frac{\sum_{l \in C_q} \sum_{i=1}^{24} \left(l(i) - c_q(i)\right)^2}{\sum_{i=1}^{24} c_q(i)^2} \quad (9)$$

The card$(\cdot)$ operator is the cardinality of the cluster set.

E. BIRCH Clustering Algorithm

The BIRCH algorithm has been developed specifically for large datasets, especially when the entire data cannot be loaded into memory. The BIRH algorithm has four phases.

1. Load data into memory by building a CF tree
2. Condense the data (optional)
3. Global clustering
4. Cluster refining (optional)

Given a set of k clusters of load shapes from $\tilde{\boldsymbol{L}}$, the BIRCH algorithm creates a set of k clustering features (CF)s and a dendrogram called a CF-tree, Fig. 3. The CF for cluster i is defined as $CF_i = \{m_i, LS_i, SS_i\}$, where m_i is the number of load shapes in the cluster, LS_i is the linear sum of the load shapes in the cluster, and SS_i is the squared sum of the load shapes in the cluster. For the CF-tree, there are two parameters: 1) a branching factor B and 2) a radius threshold T.

The tree is created serially by following the closest CF down to the leaf nodes. Every sample in the leaf nodes must satisfy T, otherwise create a new leaf node. A maximum size of the leaf node can be included, but is not necessary since the radius is bounded by T. If the sample can be inserted with the threshold and size requirements then update the CF. If the sample cannot be inserted due to the size requirement, split the leaf node and reassign the samples[1]. Then update all the CFs for the leaf nodes and non-leaf nodes traversed to the root.

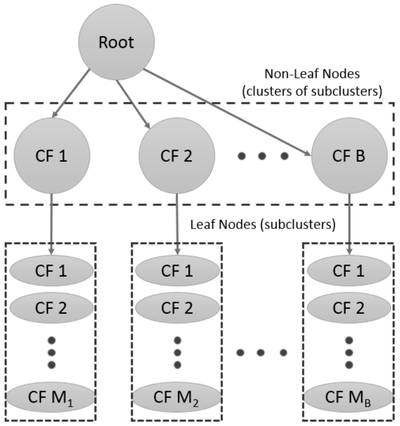

Fig. 3: Graphical depiction of the BIRCH CF tree, with each CF representing a subcluster or a cluster of subclusters. If the number of non-leaf nodes are larger than the branching factor, another layer of non-leaf nodes will be in the CF-tree.

F. Optimization of the BIRCH input parameters

In the creating of the BIRCH CF-tree (Phase 1), two main inputs drives the tree structure 1) the branching factor and 2) the threshold. The smaller the branching factor, the larger the number of clusters. Since interpretability is normally a requirement, the value of B should be sufficiently large. In this work, a minimum of 2 clusters are required, so the value of the branching factor is $B = N_{tot}/2$. In this work, the problem of determining the threshold is posed as a simple optimization problem, Eq. 10.

$$\min_{T>0} g\big(W_k(T), B_k(T)\big): \tag{10}$$

The cost function, $g(\cdot)$, takes in the threshold input parameter and the resulting within and between distances from the BIRCH algorithm. The form of this cost function problem has some flexibility, and therefore a couple choices are tested. These functions are based on a linear combination (LC), Eq. 11, the Euclidean distance (EU), Eq. 12, the squared Euclidean distance (SQEU), Eq. 13, the Bray-Curtis dissimilarity (BC), Eq. 14, and the inverse of the cluster dispersion index (I-CDI), Eq. 15.

$$g(W_k, B_k) = [W_k + (1 - B_k)]/2 \tag{11}$$

$$g(W_k, B_k) = \sqrt{W_k^2 + (1 - B_k)^2} \tag{12}$$

$$g(W_k, B_k) = W_k^2 + (1 - B_k)^2 \tag{13}$$

$$g(W\_k, B\_k\ ) = |W_k - B_k\ |/(W_k + B_k) \tag{14}$$

$$g(W_k, B_k) = |B_k/W_k - 1| \tag{15}$$

The optimization problem is solved using Brent's method to a tolerance of 1e-5.

G. BIRCH: Global Clustering Phase

In this study, only phase 1 and phase 3 are used in the BIRCH algorithm. On a single pass, the BIRCH algorithm has been known to produce more clusters than necessary partially due to the algorithms sensitivity to input order. For this reason, a global agglomerative clustering of the BIRCH CF-tree is often performed. During the agglomerative processes if a small cluster is combined with either a large of or small cluster the evaluation measures $(E_k, \sigma_k, \hat{\theta}_k)$ will only change a small amount. However, if two large clusters are combined a large increase can be expected in σ_k and $\hat{\theta}_k$ and a large decrease can be expected in E_k. To determine when these large changes occur a combined metric equation has been developed to determine the number of clusters, Eq. 16.

$$\begin{aligned} h_i\big(E_i, \sigma_i, \hat{\theta}_i\big) = &\frac{|E_{i+1} - E_i|}{2\max(E)} + \frac{|\sigma_i - \sigma_{i+1}|}{4\max(\sigma)} \\ &+ \frac{\big|\hat{\theta}_i - \hat{\theta}_{i+1}\big|}{4\max\big(\hat{\theta}\big)}\ ,\ \ i = 1{:}\,k \end{aligned} \tag{16}$$

The maximum of E is when $i = k$. The maximum of both σ and $\hat{\theta}$ is when $i = 2$. The weighting is mainly due to entropy expecting to decrease while σ and $\hat{\theta}$ are expected to increase and the weights bounds $h_i(\cdot)$ from [0-1].

III. Results

A. Choosing a cost function

For each consumption group and for each cost function, the BIRCH threshold parameter was optimized. For each scenario, the resulting number of clusters, standard deviation of the cluster size, the normalized entropy, and estimated threshold were calculated, Table 1. It is clear for the high and low consumption groups that the BC and I-CDI cost function result in the smallest number of clusters. If interpretability of the load shapes is a goal of the analysis, BC and I-CDI are the better metrics. However, the normalized entropy score or the amount of information contained in the load shapes is much lower for the BC and I-CDI metrics. If a finer resolution of load shapes is needed for providing context behind the load shapes, then the LC, EU, and SQEU would be better choices. The number of clusters for these metrics could be reduced further using the

[1] This may result in further splitting at the parent level based on the number of entries in the non-leaf nodes.

global clustering phase of the algorithm. A possible explanation for the small number of clusters produced by the BC and I-CDI cost functions is that for densely populated datasets W_k is bounded by B_k. Therefore, by minimizing Eq. 14 and Eq. 15 would support large equally spaced clusters.

As for the low consumption group, the cost functions perform in a similar manner based on the evaluation metrics chosen in this analysis. This may be due to very small consumption days having very irregular normalized load shapes. Global clustering could be used to further reduce the number of clusters. Based on the results of Table 1, the BC cost function is used in the rest of the paper. The BC cost function showed to have the smallest number of clusters for the high and low consumption groups. With the small number of clusters, this cost function (at least for this dataset) allows for the cluster load shapes to be easily interpreted.

TABLE I

Performance of the cost functions for the different consumption groups.

Group	Metric	k	σ_k	E_k	$\hat{\theta}_k$	T
High	LC	3271	799	0.430	0.0071	0.032
High	EU	1963	1118	0.400	0.0085	0.037
High	SQEU	2004	1093	0.403	0.0084	0.036
High	BC	9	57870	0.110	0.0446	0.087
High	I-CDI	9	45310	0.137	0.0379	0.079
Inter.	LC	5156	1058	0.444	0.0082	0.034
Inter.	EU	2860	1655	0.410	0.0097	0.039
Inter.	SQEU	3463	1343	0.421	0.0093	0.037
Inter.	BC	4	90100	0.092	0.0624	0.087
Inter	I-CDI	4	90100	0.092	0.0624	0.087
Low	LC	660	5947	0.257	0.0182	0.066
Low	EU	605	6115	0.258	0.0186	0.067
Low	SQEU	543	6518	0.251	0.0194	0.068
Low	BC	519	6402	0.255	0.0188	0.068
Low	I-CDI	496	6670	0.252	0.0196	0.070

B. Global Clustering for large number of clusters

Global clustering is used to reduce the number of clusters, by agglomerating the smaller clusters into larger clusters. The metric h, Eq. 16, is used to determine the optimal number of clusters for the low consumption group. Each term of the combined metric can be seen in Fig. 4. When two large clusters are combined, these metrics see large increase or decrease. This results in very large spikes in the combined metric, Fig. 5. The largest peaks in order occur at 8, 3, 319, 68, and 10. Since 8 has the largest peak, 8 clusters will be used for the rest of the paper for the low consumption group. For visual display only two smallest clusters considered only have a single day in the cluster and are not shown or discussed.

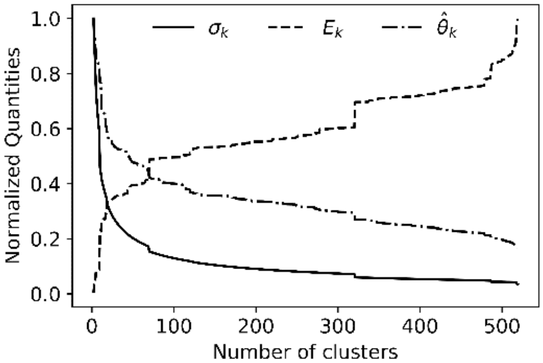

Fig. 4: The individual metrics from Eq. 7 – Eq. 9 during the agglomerative step of the BIRCH algorithm for the low consumption group.

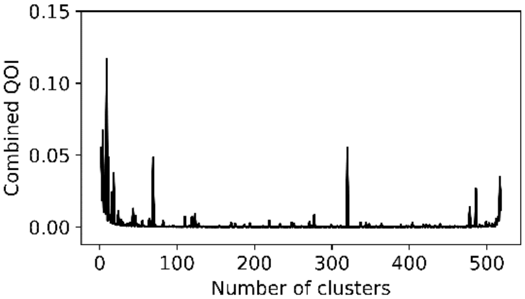

Fig. 5: The combined metric from Eq. 16 showing which possible number of clusters are good candidates (large peaks) during the agglomerative step of the BUIRCH algorithm for the low consumption group.

C. The typical load shapes from the clustering

From the previous section 9 clusters emerged from the high consumption group, 4 clusters from the intermediate group, and 6 clusters in the low consumption group. Each of these typical load shapes, average daily consumption, and the number of days as a percentage for their respective group are shown in Fig. 6. From inner quartile range and the inner 90% range, the high consumption group has the smallest range, followed by the intermediate consumption group and then the low consumption group. Each group has a baseline (flat) curve as its largest cluster (cluster 0, 9, and 13). For the high, intermediate, and low consumption groups the cluster with the largest average consumption (cluster 7, 11, and 13, respectively) seems to be representative of large night, afternoon, and baseline consumption, respectively. The low consumption group load shapes seem to be mostly large peaks during the early morning and late evening, while the high and intermediate groups have a mix of single and double peak load shapes.

D. Scalability of the implemented algorithm

To ensure that the algorithm can perform well for large data sets, the rate at which the algorithm scales with the number of samples is important. The BIRCH algorithm scales $O(N_{tot})$, Fig. 7a. For the smaller thresholds, the computational time BIRCH algorithm starts to increase for large number of samples. This is mainly due to the number of clusters increasing, Fig. 7b, and the larger CF-tree needs to be traversed. If the number of load shapes is sufficiently compressed into relatively small number of clusters the algorithm scales very well.

IV. Conclusions

BIRCH clustering algorithm can quickly and efficiently extract load shapes from large databases that cannot fit into memory A set of cost functions based on cluster compactness and separation were used to cluster high, intermediate, and low consumption load shapes. The cost functions were evaluated for their suitability to producing larger and smaller clusters for contextualization and interpretability. A combined metric based on the estimated threshold, entropy, and average cluster size standard was used to determine appropriate number of clusters during the global clustering phase of the BIRCH algorithm. The implementation was shown to scale with $O(N_{tot})$ samples. Using this algorithm, the typical load analysis can be performed at the urban-scale and potentially for continual real-time online learning and classification by utilities companies.

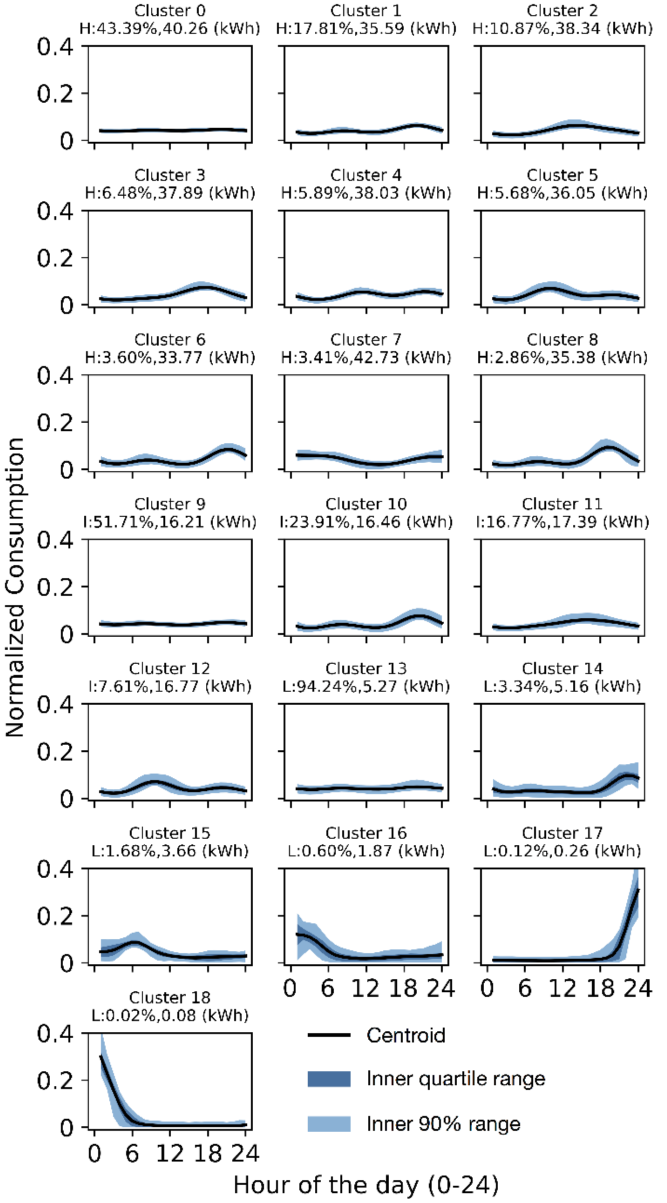

Fig. 6: The clusters extracted from the clustering algorithm with their centroid, inner quartile range, and inner 90th percent range. Each cluster corresponds to a group (H-High, I-Intermediate, and L-Low consumption), the percentage of the days that the cluster represents in their respective group, and the average daily consumption.

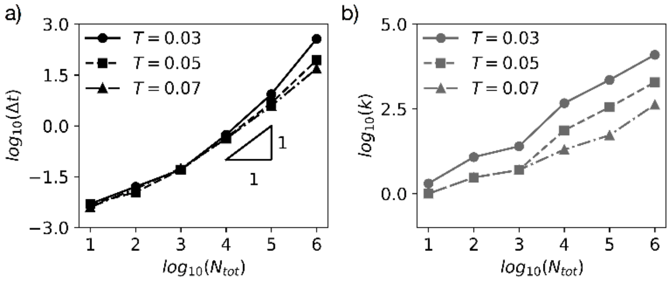

Fig 7: a) Computational time in seconds of the BIRCH algorithm for different thresholds on an Intel i5-2540M 2.60GHz processor. b) The corresponding number of clusters for the BIRCH algorithm for different thresholds.

V. References

[1] J. Nazarko and Z. A. Styczynski, "Applications of statistical and neural approaches to the daily load profiles modelling in power distribution systems," in *IEEE Transmission and Distribution Conference*, 1999, vol. 1, pp. 320–325.

[2] H. L. Willis, A. E. Schauer, J. E. D. Northcote-Green, and T. D. Vismor, "Forecasting Distribution System Loads Using Curve Shape Clustering," *IEEE Power Eng. Rev.*, vol. PER-3, no. 4, pp. 891–901, 1983.

[3] I. P. Panapakidis, M. C. Alexiadis, and G. K. Papagiannis, "Load profiling in the deregulated electricity markets: A review of the applications," *2012 9th Int. Conf. Eur. Energy Mark.*, pp. 1–8, 2012.

[4] J. Abreu, F. C. Pereira, J. Vasconcelos, and P. Ferrão, "An approach to discover the potential for demand response in the domestic sector," in *IEEE CITRES 2010*, 2010, pp. 240–245.

[5] J. M. Abreu, F. Câmara Pereira, and P. Ferrão, "Using pattern recognition to identify habitual behavior in residential electricity consumption," *Energy Build.*, vol. 49, no. October, pp. 479–487, 2012.

[6] J. Kwac, J. Flora, and R. Rajagopal, "Household energy consumption segmentation using hourly data," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 420–430, 2014.

[7] S. Xu, E. Barbour, and M. C. Gonzalez, "Household Segmentation by Load Shape and Daily Consumption," *Proc. ACM SigKDD 2017 Conf. Halifax, Nov. Scotia, Canada, August 2017*, 2017.

[8] F. McLoughlin, A. Duffy, and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data," *Appl. Energy*, vol. 141, pp. 190–199, 2015.

[9] F. Farnstrom, J. Lewis, and C. Elkan, "Scalability for clustering algorithms revisited," *ACM SIGKDD Explor. Newsl.*, vol. 2, no. 1, pp. 51–57, 2000.

[10] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Databases Method for Very Large," *ACM SIGMOD Int. Conf. Manag. Data*, vol. 1, pp. 103–114, 1996.

[11] B. Lorbeer, A. Kosareva, B. Deva, and A. Küpper, "A-BIRCH: Automatic Threshold Estimation for the BIRCH Clustering Algorithm," in *Advances in Intelligent Systems and Computing*, 2017, pp. 169–178.

[12] J. Vahidi and S. Mirpour, "Introduce a New Algorithm for Data Clustering by Genetic Algorithm," *J. Math. Comput. Sci.*, vol. 10, pp. 144–156, 2014.

[13] N. Golyandina and A. Zhigljavsky, *Singular Spectrum Analysis for Time Series*. New York, NY, USA: Springer, 2013.

[14] S. Ramos, J. M. Duarte, F. J. Duarte, Z. Vale, and P. Faria, "A data mining framework for electric load profiling," *IEEE PES Conf. Innov. Smart Grid Technol. Lat. Am. (ISGT LA)*, pp. 1–6, 2013.

VI. Biographies

Anthony D. Fontanini (Ph.D.) received his B.S. degree in mechanical engineering from the University of Wisconsin-Platteville, Platteville, WI, USA in 2009 and his Ph.D. degree in mechanical engineering at Iowa State University, Ames, IA, USA in 2016.

He is a Member of Technical Staff at the Fraunhofer Center for Sustainable Energy Systems (CSE) in the Building Enclosures and Materials Group. His research spans many areas of numerical analysis, machine learning and software development for building energy efficiency, performance of air distribution systems, and indoor air quality. These areas include uncertainty quantification, analyzing characteristics of contaminant transport in indoor environments, placement of sensors, developing and implementing energy models and algorithms for building energy analysis. The different software platforms developed for these applications are implemented for high performance computing (HPC) and high throughput computing (HTC) environments.

Joana Abreu (Ph.D.) leads behavioral research for Fraunhofer CSE's Building Energy Management Group. At Fraunhofer CSE, she works with a team comprising engineers, building scientists and psychologists to apply experimental psychology methods to complex systems-level research in both field and laboratory research projects. She developed an externally-funded interdisciplinary research program that includes such diverse subjects as energy efficiency, social science, data analytics, environmental engineering and geographical information systems. She holds a Doctoral Degree in Sustainable Energy Systems from the MIT Portugal Program studying habitual behavior and feedback in the residential sector.