Comparison of different clustering approaches on different databases of smart meter data

Martina Ferrando[1,2], Debora Nozza[3], Tianzhen Hong[2], Francesco Causone[1]
[1]Politecnico di Milano, Milano, Italy
[2]Lawrence Berkeley National Laboratory, Berkeley, California, United States of America
[3]Università Bocconi, Milano, Italy

Abstract

Various clustering methods have been applied to determine representative groups of buildings based on their energy use patterns. We reviewed and selected the most commonly used clustering methods, including k-means, k-medoids, Self-Organizing Map (SOM) coupled with k-means and hierarchical, and our proposed deep clustering algorithm for comparative performance assessment using datasets of smart meters. After the data preparation (data cleaning, segmentation, and normalization), the clustering is run, firstly, letting the number of clusters free to be chosen by the optimization process, and then forcing it to be equal to the number of primary functions of buildings. Depending on the purpose of clustering, e.g., to identify daily 24-hour load shape, to identify primary building use type (e.g., office, residential, school, retail), the optimal number of clustering can vary greatly. Thus, based on the final aim, forcing somehow the number of clusters is the most followed and suggested for engineering purposes. The k-means, the k-medoid, and the hierarchical algorithms show the best results, in all cases. While for the nature of the databases the additional step of adding a SOM to the k-means algorithms does not show improvements in terms of evaluation metrics. The direct comparison of the different algorithms gives a clear overview of the existing main clustering approaches and their performance in capturing typical use patterns in typical smart meter databases. The resulting cluster centroids could be used to better understand and characterize the energy use patterns of different buildings and building typologies with the final aims of benchmarking or customers segmentation.

Key Innovations

- The most commonly used clustering algorithms are selected from the literature and directly compared on three typical smart meter databases.
- A deep learning clustering is added as an algorithm that has never been applied for this kind of application before.
- The comparison is performed via three evaluation metrics and not limited to one.

Practical Implications

Overall, the hierarchical, the k-means, and the k-medoids algorithms show good and similar results. The hierarchical method could be better exploited because, via the dendrogram, it gives the possibility to visualize the clustering and chose the number of clusters easily, which is usually a complex step to solve but fundamental in engineering applications. The organization of the database and the number of clusters are fundamental steps that could bring very different results. Thus, these two aspects must be well-tuned, based on the final aim of the clustering.

Introduction

Smart meters differ from traditional ones because of their capability to read high-frequency data for different resources (e.g., electricity, gas, water) that eventually can be stored in repositories. Useful information and application may be derived from the raw smart meter data, including load shape benchmarking (Luo *et al.*, 2017), occupant behavior estimates (Causone *et al.*, 2019), costumers classification (Chicco *et al.*, 2004), anomalies detection (Devlin and Hayes, 2019), end-use disaggregation (Khalid *et al.*, 2018), detect or design buildings changes/retrofit (Ren, Heo and Sunikka-Blank, 2019). These applications mainly exploit clustering approaches to derive groups of buildings or customers with similar resource usage.

Clustering is a specific area of applications of machine learning that involves algorithms able to group a dataset into N number of clusters (C_i, i =1, 2, …, N). Usually, clustering implies that the partitioning is unsupervised, thus, there are not labeled examples from which the machine can learn, no target feature is expressed, and the aim is to find similarities and differences in the data. Unsupervised learning is a powerful technique that can be helpful when nothing can be set for sure in the dataset. The aim, in this case, is to find patterns in the input. The data are naturally structured such that certain patterns occur more often than others and the machine itself finds the way to group the similarities. Clustering processes divide the dataset into a given number of groups sharing similar features so that the data in different clusters have distinctive characteristics (Lucchi *et al.*, 2020). Thus, a cluster is a homogeneous subgroup existing within a population. The result will be the division of the data into some groups trying to minimize some criterion or error functions.

Currently, the main clustering techniques used on smart meter data in the building sector include k-means (Yilmaz *et al.*, 2019), k-medoids (Himpe and Janssens, 2019),

https://doi.org/10.26868/25222708.2021.30193

Self-Organizing Map (SOM) coupled with k-means (Ferrando *et al.*, 2019). The number of final clusters is assessed via metrics, among them: Davies–Bouldin index (DBI), Silhouette index (SHI), Calinski-Harabasz Index (CHI). However, the majority of research works make use of just one or a few of these methods and metrics and tune the algorithm on a specific database. This hinders a clear comparison of the performance of different algorithms. Moreover, the implementation of the algorithms on just a single database is helpful to find the best option for that specific case but not a general approach that shows fair results in most cases for pre-analysis of data.

This paper aims to compare the results from different clustering algorithms, and different performance metrics, applied on three databases, segmented from the Building Data Genome Project 2 based on three cities with different climate conditions (Miller *et al.*, 2020). Particularly, the goal is to find an algorithm that gives fair results on different databases for pre-analysis of data.

The database

The Data Genome Project 2 database for electricity uses (Miller *et al.*, 2020) is exploited. In particular, it has been filtered to create three different sub-databases to be used separately and compared. The three databases are related to three different universities and U.S. time zones. *Table 1* is a summary of the main characteristics of the databases. The three universities are identified via a code name (i.e., Bear, Rat, and Fox). They are located close to three large cities in the US (i.e., Berkeley (CA), Washington D.C., and Phoenix (AZ)), with different climate condition with respect to the Köppen climate classification (Köppen, 1884) (i.e., warm-summer Mediterranean climate (Csb), humid subtropical climate (Cfa), and hot desert climates (Bwh)). In addition to these location data, the total number of buildings is provided with other basics characteristics of the buildings such as the primary usage, the area, the year of construction, and the number of floors. *Figure 1* summarizes the percentage distribution of the primary usage in the three databases.

Table 1: Summary of the main characteristics of the three databases.

University code name	**Bear**	**Rat**	**Fox**
Latitude, Longitude	37.9, -122.3	38.9, -77.0	33.4, -111.9
Köppen climate classificatio n	Csb	Cfa	Bwh
Time zone	US/ Pacific	US/ Eastern	US/ Mountain
Number of buildings	91	304	134
Primary building use	7 types, *Figure 1*	11 types, *Figure 1*	12 types, *Figure 1*
Building floor area (square meter)*	M = 8874.0, S = 7969.2, Mi = 121.2, Mx = 39101.5	M = 6808.7, S = 9773.9, Mi = 74.4, Mx = 76141.2	M = 9196.8, S = 10511.6, Mi = 94.9, Mx = 75207.8
Year built*	M = 1954.2, S = 32.1, Mi = 1903, Mx = 2016	M = 1961.8, S = 35.3, Mi = 1900, Mx = 2017	M = 1975.2, S = 25.8, Mi = 1907, Mx = 2014
Number of Floors*	M = 5.3, S = 2.7, Mi = 1, Mx = 14	N.A.	N.A.

* M = Mean, S = Standard Deviation, Mi = Minimum, Ma = Maximum

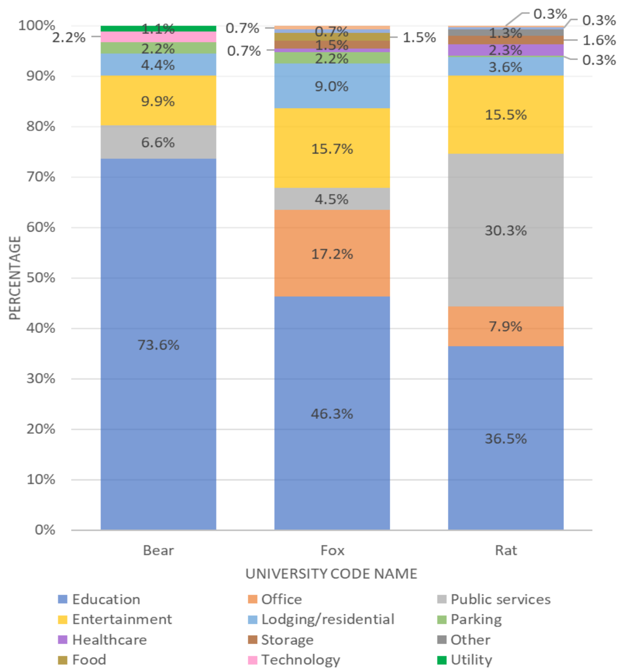

Figure 1: Building uses distribution in the databases.

Methods

The most used clustering methods have been selected from the literature (i.e., k-means, k-medoids, SOM coupled with k-means and hierarchical), moreover, a deep clustering algorithm is added. Looking at the metrics, the paper aims to understand how different algorithms change the results within the same database and understand which of the option perform better among databases.

The workflow is subdivided into three main steps (*Figure 2*). Firstly, the data are cleaned, segmented, and normalized for the clustering process (Step 1). Then, the characteristics of the single algorithm are optimized, and the best parameters are used to run a final clustering (Step 2). Finally, the clusters are compared via metrics among them and databases (Step 3). Steps 2 and 3 are performed twice, once with the possibility for the algorithm to freely choose the best number of clusters (A-steps), secondly, imposing a number of clusters equal to the number of primary usages (B-steps). This number is chosen because different building usages bring to different average electricity usage and also different hourly patterns (Carnieletto *et al.*, 2021).

In particular, the data are investigated with traditional statistical approaches and a few outliers were eliminated (e.g., data cleaning). Moreover, the dataset covers two years with hourly values (from 1st of January 2016 to 31st

https://doi.org/10.26868/25222708.2021.30193

of December 2017). However, the year 2016 has numerous gaps, thus, only the year 2017 has been used for the research (e.g., data segmentation). The final aim is to have a set of 205 values for each building, thus, the electricity uses are divided by the building area (i.e., the values are in kWh/m^2) and then organized as follow: 1 annual value, 12 monthly values, the average 24 hourly values for 8 typical days (i.e., workday and holiday for the four seasons). The national holidays, Saturdays, and Sundays are set as holidays. Finally, normalization of these data is needed since a few clustering algorithms cannot treat non-normalized data or they could bring to large optimization errors. The normalization is achieved by dividing each of the 10 groups of data (i.e., annual values, monthly values, and 8 days by four seasons hourly values) by the maximum within the group. At this point, the database is ready to be used in the clustering process. The A-steps (*Figure 2*) are run resulting in the optimal number of clusters to be used. While in the B-steps the number of clusters is imposed to be equal to the number of primary usage of buildings.

In the following sections, each of the implemented algorithms, their most influencing parameters, and the used metrics will be briefly explained.

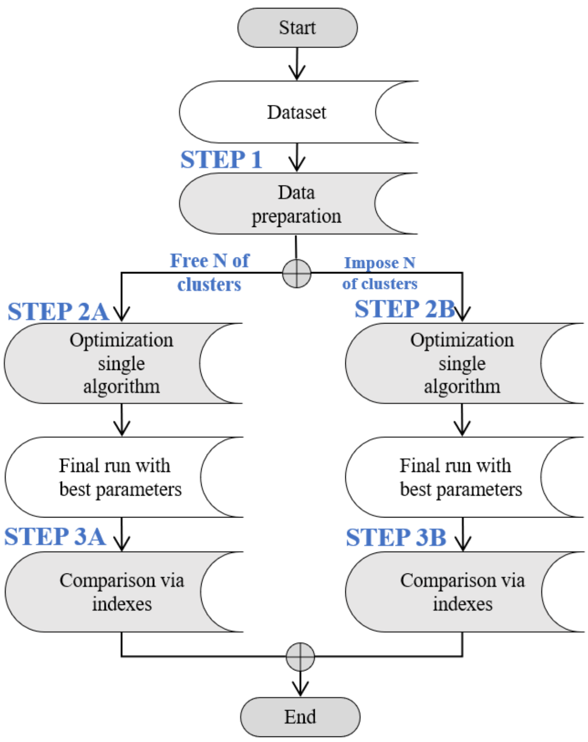

Figure 2: Schematic of the workflow.

k-means and k-medoids

The k-means and k-medoids algorithms are the simplest and most commonly used unsupervised learning algorithms (Aha, Kibler and Albert, 1991). They solve the problem of clustering given a fixed number (k) of centroids that can be intended as a multi-dimensional average, representing the center of a cluster. Firstly, the algorithm takes each input of the dataset and associates it to the nearest centroid. Next, it recalculates the k centroids as the barycenter of the clusters resulting from the previous step. The inputs are again associated with the centroids and a new complete iteration is computed. The calculation ends when the centroids change their location of a meaningful distance and the inputs associated with a specific centroid become the cluster. For this clustering technique, the input can move from cluster to cluster at each timestep, during the analysis. While the k-means tries to reduce the total squared error, k-medoids minimizes the sum of differences between the points assigned to a cluster and a point chosen as the center of that cluster, selecting data points as cluster centers (called medoids).

The main parameters for the k-means and k-medoids algorithms are the method for initialization (i.e., how the initial clusters are selected), the number of initializations (i.e., the number of times the algorithm will be run with different centroid seeds), the maximum number of iterations of the algorithm for a single run, and the actual algorithm use to find the maximization in the iterative process.

SOM coupled with k-means

In some researches, coupling the SOM (Kohonen, 1990) with the k-means is proved to improve the clustering results (Vesanto and Alhoniemi, 2000; Hernández *et al.*, 2012). The SOM is part of Artificial Neural Networks (ANN), which consists of computational algorithms that try to simulate the behavior of a biologic brain and its neurons. The neurons are usually organized in layers, and their organization creates a structure called architecture or neural network. The neurons are interconnected by neighborhood relationships, called topology. In particular, the SOM is an unsupervised neural network method, able to classify data into clusters and it is exploited to display multidimensional data in a low-dimensional grid. The SOM uses a competitive learning approach: when an input vector is presented to the network, the similarity with each neuron's synaptic weight is computed and the weight of the neuron more similar to the input vector is the winner. In this research, a two level-approach clustering method was used, combining the SOM with the more classic unsupervised k-means algorithm. The SOM algorithm is used to create proto-clusters that are further grouped with a k-means algorithm to find the final clusters. The two main benefits are the minimization of the computational cost and the noise reduction since the proto-clusters are local averages of the original samples and, for this reason, less sensitive to single high or low cases in the data sample. The size of the initial SOM is given by the heuristic formula (Vesanto *et al.*, 2000):

$$m = 5\sqrt{n} \quad (1)$$

In which m is the final number of proto-clusters, n is the number of data samples given as input. Moreover, the ratio of the side-lengths of the lattice would be the ratio between the two biggest eigenvalues of the covariance matrix of the given data, and the actual side-lengths are then set in such a way that their product is as close as possible to the desired m.

https://doi.org/10.26868/25222708.2021.30193

Besides the SOM dimension, other important parameters to be tested are the learning rate (i.e., the step size at each iteration as it moves towards the minimum), the radius of the different neighbors in the SOM, the activation distance (i.e., the distance used to activate the map), the topology of the map (i.e., the shape of the node also regulating how many neighbors each node has, could be rectangular or hexagonal), and the neighborhood function (i.e., the function that weights the neighborhood of a position in the map).

Hierarchical

A different approach is used in the hierarchical clustering algorithm (Nielsen, 2016), in which, in the beginning, each observation is seen as a separate single cluster. Lately, repeatedly the two closest clusters are merged until there is only one cluster counting all the original observations. The distance between two clusters can be computed using different metrics. Finally, based on the initial number of cluster (k) given to the algorithm, the process stops. An interesting aspect of the use of hierarchical clustering is the visualization of the relationships among clusters in the dendrogram, which is a tree-like graph able to show the similarities within groups.

The main parameters in the hierarchical algorithm are the affinity (i.e., the metric used to compute the linkage between samples), and the linkage criterion (i.e., which distance criterion is used to merge the pairs of clusters that minimize the criterion itself).

Deep learning

Deep learning, is part of ANN as SOM but makes use of a more complex architecture in which the neurons are organized in far more layers. In the used deep learning algorithm (Xie, Girshick and Farhadi, 2016) a parameterized non-linear mapping from the data space to a lower-dimensional feature space is defined, optimizing a clustering objective. Unlike previous work, which operates on the data space or a shallow linear embedded space, a stochastic gradient descent via backpropagation on a clustering objective to learn the mapping is used, which is parameterized by a deep neural network. The aim is to simultaneously solve the cluster assignment and the underlying feature representation. However, being an unsupervised learning method, an iteratively refining of clusters is run. This process gradually improves the clustering as well as the feature representation.

The main parameters of the deep learning algorithm are the learning rate, the dropout fraction (i.e., the fraction of neurons which should be set to 0 randomly for regularization purposes), the encoders dimension (i.e., the size of the underlying feature representation), and the iterations for the fine-tuning.

The evaluation metrics

The Calinski-Harabasz (CHI) (Caliński and Harabasz, 1974), the Davies Bouldin (DBI) (Davies and Bouldin, 1979), and the Silhouette (SHI) (Rousseeuw, 1987) are the three metrics used to compare the results among the databases. CHI is a heuristic metric, defined as the ratio between the within-cluster dispersion and the between-cluster dispersion. There are no limits to its value, but it can be used to compare the results of clustering algorithms. The higher this metric is, the better is the clustering result, meaning that clusters are dense and well-separated. The DBI is used as the CHI, to compare the similarity (or dissimilarity) between each cluster and the similar one, comparing the distance between clusters with the size of the clusters themselves. In this case, the lower is this metric, the better is the clustering result and zero is the lowest possible score. Finally, the SHI is a value indicating how similar a sample is to its cluster with respect to other clusters. It is computed for each sample and is a combination of two scores, the mean distance between a sample and all other points in the same class and the mean distance between a sample and all other points in the next nearest cluster. The higher is this metric, the better is the clustering and it is bounded between -1 for incorrect clustering and +1 for dense and well-separated clusters.

Results and Discussion

A-steps (free number of clusters)

Firstly, the main parameters and the number of clusters are optimized. The algorithms' parameters among the three different databases after the optimization are very similar. For example, for k-means and k-medoids, the number of iterations is set to 10 in practically all cases. The SOM is always chosen to be rectangular and the Mexican hat (i.e., one of the typical continuous wavelets used in machine learning as neighborhood function (Kohonen, 1990)) is chosen as a neighborhood function. In the hierarchical algorithm, the affinity is always Euclidean, and the linkage criterion is Ward's method or the average in Rat's database. Also for deep learning, the optimized parameters are very similar among the databases, especially the same encoders dimension is chosen. These similar results could be explained by the fact that the databases are organized and normalized in the same way. Always a low number of clusters is chosen (ranging from 2 to 5). The deep learning algorithm is the one selecting the highest number of clusters.

For the Bear's database (*Figure 3*), the k-means and the hierarchical algorithms show the best results with respect to the three metrics used. Also, the deep learning and k-medoids algorithms show relatively good results, comparable with k-means and hierarchical. The SOM coupled with the k-means, on the other hand, shows the worst results. For Fox's database (*Figure 4*), there is no distinction between k-means, k-medoids, and hierarchical because they bring the same result (with only 2 clusters). Overall, also the deep learning shows good results among the three metrics, while the addition of the SOM to the k-means also in this case corresponds to a worsening of the results. Finally, in Rat's databases (*Figure 5*), the evaluation metrics show the highest differences. The CHI assigns the best clustering to the k-means, while the DBI and the SHI to the hierarchical. In general, the k-means and the k-medoids are the two algorithms showing good results in all three metrics. Also, in this case, the SOM

https://doi.org/10.26868/25222708.2021.30193

coupled with k-means does not show an improvement in the clustering results.

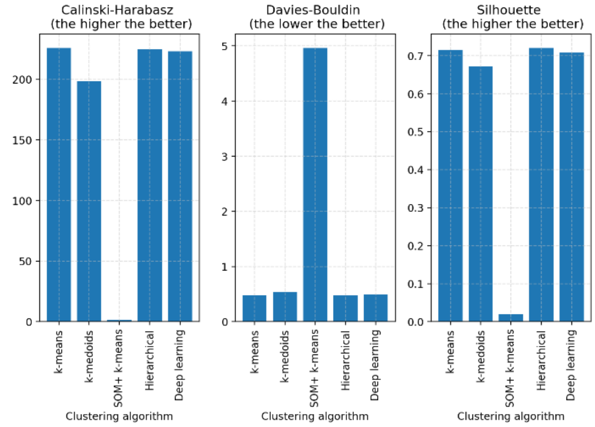

Figure 3: Metrics results for Bear database with the free number of clusters.

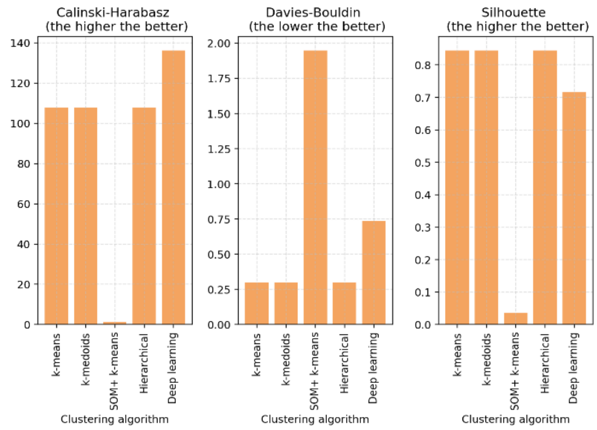

Figure 4: Metrics results for Fox database with the free number of clusters.

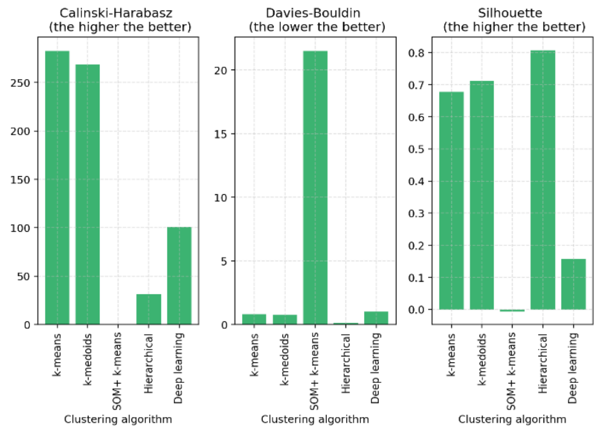

Figure 5: Metrics results for Rat database with the free number of clusters.

Fixed number of clusters

A small number of clusters (i.e., 2) is usually not an interesting result for building engineering applications. Thus, the number of clusters is imposed to be equal to the number of primary usages present in the database (i.e., 7 for Bear's database and 11 for Fox's and Rat's ones). Also in this case, the main parameters of each algorithm are optimized and they show similar results among the databases and compared to the run with a free number of clusters. However, in all cases the number of iterations for the k-means algorithms increases (average around 400). The SOM topology is always chosen to be rectangular and the neighborhood function is chosen to be again the Mexican hat. In the deep learning algorithms, the same encoders dimension is chosen as with the free number of clusters. However, for the hierarchical clustering, the taxicab geometry affinity is chosen (i.e., the distance is the summation of the absolute differences of their coordinates on a Cartesian plane). The same three metrics (i.e., CHI, DBI, and SHI) are used to compare the clustering results. *Figures 6, 9, and 12* show the plots of the metric values.

In Bear's database (*Figure 6*), higher variability with respect to the case with the free number of clusters is registered. Looking at the DBI and the SHI, hierarchical clustering is the best option. However, for the CHI, the k-medoids algorithm shows the highest value. Generally, k-means, k-medoids, and hierarchical result in good clustering. While, the SOM, also in this case, does not improve the results. The hierarchical algorithm gives the possibility to visualize the clustering via a dendrogram (*Figure 7*), from which is visible that there are two families of samples very different from one another (this is the reason why keeping the optimization free to select the number of clusters, 2 was the final choice). In *Figure 7*, one family of samples (the red one) is larger than the other (green), and inside them, two other big sub-families are visible. The clustering is stopped when the samples are divided into 7 clusters (black line). Thus, the green family will be divided into three main groups, while the red one in four main groups. In *Figure 8*, the centroids of the k-medoids algorithm of the 7 clusters are plotted. Keeping it in mind that the first value of the cluster is the annual sum, the next 12 values are monthly sums, and then the last 192 are the 8 average hourly patterns of the typical days, the differences between the clusters are visible. The algorithm is dividing the samples based on their values in terms of kWh/m^2 (e.g., the difference between clusters 7 and 5), and also on their daily pattern (e.g., the difference between clusters 7 and 4).

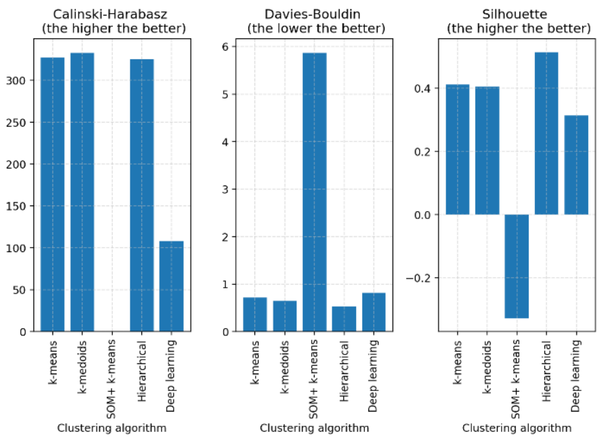

Figure 6: Metrics results for Bear database with a fixed number of clusters (i.e., 7)

https://doi.org/10.26868/25222708.2021.30193

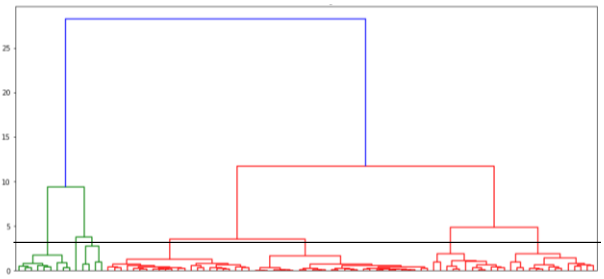

Figure 7: Dendrogram resulting from the hierarchical clustering for Bear database

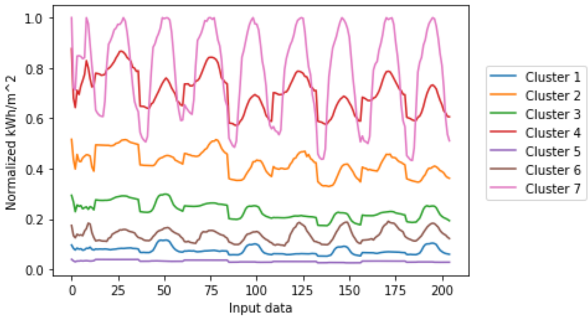

Figure 8: k-medoids centroids for Bear database

In Fox's database, the metrics (*Figure 9*) show similar results compared to Bear's one. Looking at the DBI and SHI hierarchical algorithm is the best option, while the CHI assigned to the k-means the highest value. The dendrogram (*Figure 10*), in this case, shows three distinct families (i.e., the green, the red, and the light blue). From this plot, we can see why the result of just two databases could not represent a good result since the two clusters could be only the distinction of the green family from the other two. The black line represents where the algorithm stops at 11 clusters. Thus, the green family and the red one are divided into two clusters each, while the light blue family is divided into 7 clusters. *Figure 11* shows the centroids of the k-means clustering for Fox's database, from which is again clear that the clusters represent differences in terms of kWh/m^2 (e.g., the difference between cluster 7 and 10), but also in terms of the daily pattern (e.g., the difference between cluster 7 and 2). It is also visible how the patterns are more different in terms of the daily pattern than in Bear's database, explaining the fact that the hierarchical clustering found three distinct families.

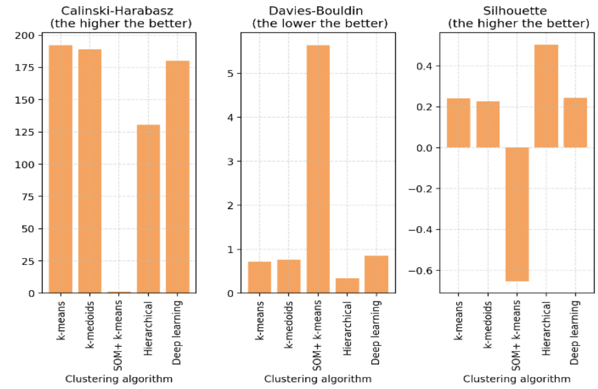

Figure 9: Metrics results for Fox database with a fixed number of clusters (i.e., 11)

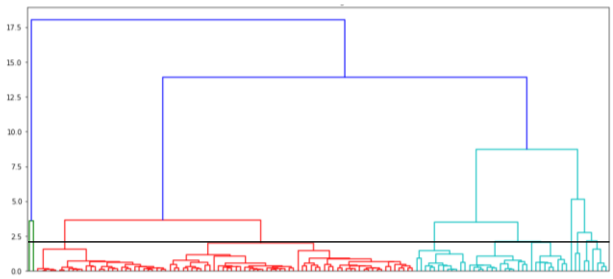

Figure 10: Dendrogram resulting from the hierarchical clustering for Fox database

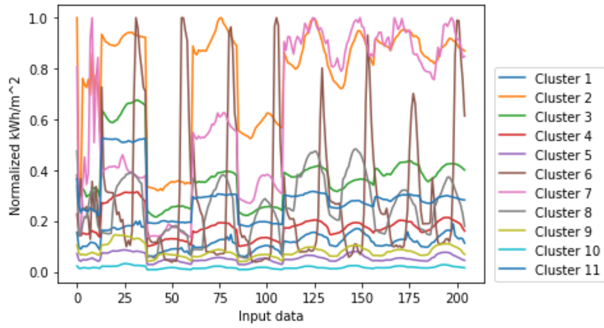

Figure 11: k-means centroids for Fox database

Finally, also for the Rat's database (*Figure 12*) DBI and SHI assign to the hierarchical algorithm the best results, shole the CHI to the k-means as in Fox's database. The dendrogram (*Figure 13*) shows that, in this case, there are two distinct families of inputs (i.e., red and green). The green one is more numerous than the red one. The black line, that indicates where the algorithm stopped at 11 cluster, shows that the green family is divided into three main clusters, while the red one has a higher variability, and it is divided into 8 clusters. The k-means algorithm centroids (*Figure 14*) show that for this database the values in terms of average kWh/m^2 are more similar and the clustering divided the samples more in terms of hourly patterns. In particular, clusters 3 and 4 have similar averages in terms of kWh/m^2, but they are characterized by very different daily patterns.

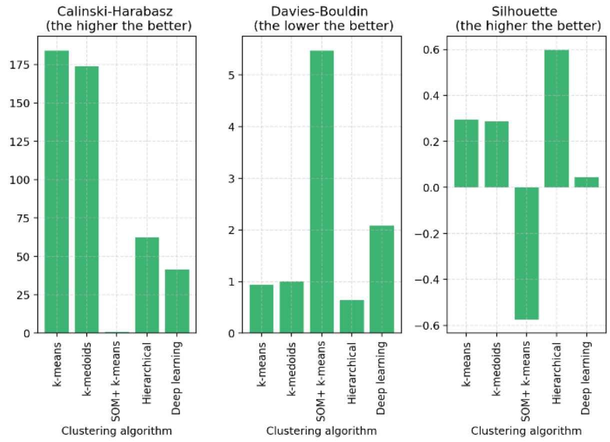

Figure 12: Metrics results for Rat database with a fixed number of clusters (i.e., 11)

https://doi.org/10.26868/25222708.2021.30193

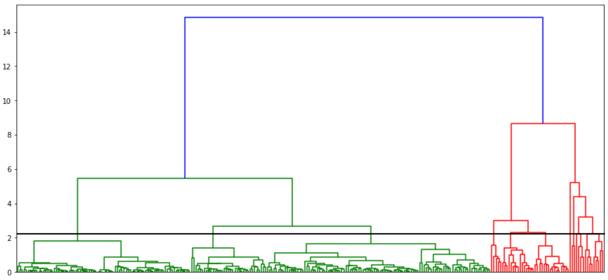

Figure 13: Dendrogram resulting from the hierarchical clustering for Rat database

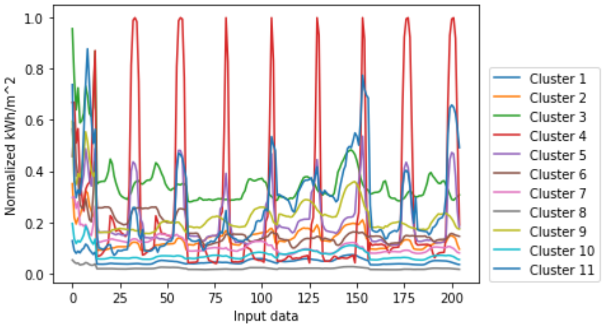

Figure 14: k-means centroids for Rat database

Overall, the hierarchical clustering shows the best results, followed by the k-means, k-medoid, and deep learning ones. The SOM, for the nature of the database, does not show an improvement in the results. In fact, the coupling of SOM with k-means is proved to improve the results in large and homogenous databases (Wu and Chow, 2004; Causone *et al.*, 2019). Thus, we can assume that this result is strictly related to the organization of the database, which, in this case, is not large and show large differences among samples.

Conclusion

Different clustering methodologies are implemented to determine a few representative groups of buildings with similar energy use patterns. When the number of clusters is left free to be chosen by the optimization process, the number of final clusters does not show great variability and a low number is always chosen. Some clustering methods, however, bring to very similar results (i.e., k-means, k-medoids, and hierarchical), both in terms of the sample grouping and evaluation metrics. When the number of clusters is forced to be equal to the number of primary uses of buildings, the hierarchical algorithm shows the best results, followed by the k-means and k-medoid. For the nature of the databases, the additional step of adding a SOM to the k-means algorithms does not show improvements in terms of evaluation metrics. The deep learning clustering shows good results. However, the computing time for its optimization is larger than all other algorithms. Thus, in this application, the use of this kind of algorithm could not be the best option. SOM Deep Learning involves a higher level of complexity that seems to overestimate the complexity of the problem resulting in lower performance. Hierarchical clustering performs better than k-means and k-medoids when the data is not well-separated into sphere-like clusters, which may be this case.

The shape of the input data (205 values) could greatly change the results, thus, in further investigations, different data organization and normalization could be use and compared. In this case, the three kinds of values are all fundamental in the division of the patterns. In some cases (e.g., *Figure 8*) the annual and monthly values could seem unnecessary. However, for the other cases (Figure 11 ad Figure 14) the differences between clusters are dictated also from the annual and monthly values. This could be related to the fact that the Bear database shows fewer differences in terms of monthly values due to the homogenous climate throughout the year. Moreover, in this case, the clustering was performed on the electricity use of the Building Data Genome Project 2 database was exploited, however, this same analysis could also be performed on other usages (e.g., gas, water, etc.) separately or integrated into the same database. A further investigation could also aim to increase the number of clusters. As a matter of fact, more and more differences in the daily pattern could emerge, and this could be interesting based on the final aim of the clustering.

In conclusion:

- The most used clustering algorithms (adding a deep learning clustering never been applied for this kind of application before) are selected from the literature and directly compared on three typical smart meter databases.
- The comparison is performed via three evaluation metrics and not limited to one.
- This comparison of the different algorithms gives a clear overview of the main existing clustering approaches and their strength and limitations.
- The resulting final cluster centroids could be exploited to better understand the energy patterns of different buildings and building typologies, especially, but not limited, to the final aims of benchmarking.
- It must keep in mind that the final aim of the clustering cannot be overlooked, and especially the number of clusters must be chosen based on this final aim.

References

Aha, D. W., Kibler, D. and Albert, M. K. (1991) 'Instance-based learning algorithms', *Machine Learning*, 6(1), pp. 37–66. doi: 10.1007/BF00153759.

Caliński, T. and Harabasz, J. (1974) 'A dendrite method for cluster analysis', *Communications in Statistics*. Taylor & Francis, 3(1), pp. 1–27. doi: 10.1080/03610927408827101.

Carnieletto, L. *et al.* (2021) 'Italian prototype building models for urban scale building performance simulation', *Building and Environment*, 192. doi: 10.1016/j.buildenv.2021.107590.

Causone, F. *et al.* (2019) 'A data-driven procedure to model occupancy and occupant-related electric load profiles in residential buildings for energy simulation',

https://doi.org/10.26868/25222708.2021.30193

Energy and Buildings. Elsevier, 202, p. 109342. doi: 10.1016/J.ENBUILD.2019.109342.

Chicco, G. *et al.* (2004) 'Load pattern-based classification of electricity customers', *IEEE Transactions on Power Systems*, 19(2), pp. 1232–1239. doi: 10.1109/TPWRS.2004.826810.

Davies, D. L. and Bouldin, D. W. (1979) 'A Cluster Separation Measure', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), pp. 224–227. doi: 10.1109/TPAMI.1979.4766909.

Devlin, M. and Hayes, B. P. (2019) 'Non-Intrusive Load Monitoring using Electricity Smart Meter Data: A Deep Learning Approach', in *IEEE Power and Energy Society General Meeting*. IEEE Computer Society. doi: 10.1109/PESGM40551.2019.8973732.

Ferrando, M. *et al.* (2019) 'Pattern Recognition and Classification for Electrical Energy Use in Residential Buildings', in *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*. Rome, Italy, pp. 2246–2253. doi: 10.26868/25222708.2019.210750.

Hernández, L. *et al.* (2012) 'Classification and clustering of electricity demand patterns in industrial parks', *Energies*, 5(12), pp. 5215–5228. doi: 10.3390/en5125215.

Himpe, E. and Janssens, A. (2019) 'Identification of energy use time patterns of occupied dwellings using smart meter data', *E3S Web of Conferences*, 111(201 9). doi: 10.1051/e3sconf/201911105011.

Khalid, K. *et al.* (2018) 'Performance comparison of artificial intelligence techniques for non-intrusive electrical load monitoring', *Bulletin of Electrical Engineering and Informatics*. Institute of Advanced Engineering and Science, 7(2), pp. 143–152. doi: 10.11591/eei.v7i2.1190.

Kohonen, T. (1990) 'The self-organizing map', *Proceedings of the IEEE*, 78(9), pp. 1464–1480. doi: 10.1109/5.58325.

Köppen, W. (1884) 'Die Wärmezonen der Erde, nach der Dauer der heissen, gemässigten und kalten Zeit und nach der Wirkung der Wärme auf die organische Welt betrachtet (The thermal zones of the earth according to the duration of hot, moderate and cold periods and to the impac', *Meteorologische Zeitschrift (2011)*, pp. 351–360. doi: 10.1127/0941-2948/2011/105.

Lucchi, E. *et al.* (2020) 'A Density-Based Spatial Cluster Analysis Supporting The Building Stock Analysis In Historical Towns', *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*, 16(v), pp. 3831–3838. doi: 10.26868/25222708.2019.210346.

Luo, X. *et al.* (2017) 'Electric load shape benchmarking for small- and medium-sized commercial buildings', *Applied Energy*. Elsevier, 204(April), pp. 715–725. doi: 10.1016/j.apenergy.2017.07.108.

Miller, C. *et al.* (2020) 'The Building Data Genome Project 2: Hourly energy meter data from the ASHRAE Great Energy Predictor III competition'. Available at: http://arxiv.org/abs/2006.02273.

Nielsen, F. (2016) 'Chapter 8: Hierarchical Clustering', in *Introduction to HPC with MPI for Data Science*. 1st edn. Springer International Publishing, p. 282. doi: 10.1007/978-3-319-21903-5.

Ren, G., Heo, Y. and Sunikka-Blank, M. (2019) 'Investigating an adequate level of modelling for retrofit decision-making: A case study of a British semi-detached house', *Journal of Building Engineering*. Elsevier Ltd, 26(June), p. 100837. doi: 10.1016/j.jobe.2019.100837.

Rousseeuw, P. J. (1987) 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, 20, pp. 53–65. doi: https://doi.org/10.1016/0377-0427(87)90125-7.

Vesanto, J. *et al.* (2000) *SOM Toolbox for Matlab 5*. Edited by SOM Toolbox Team. Espoo: Libella Oy.

Vesanto, J. and Alhoniemi, E. (2000) 'Clustering of the self-organizing map', *IEEE Transactions on Neural Networks*, 11(3), pp. 586–600. doi: 10.1109/72.846731.

Wu, S. and Chow, T. W. S. (2004) 'Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density', *Pattern Recognition*, 37(2), pp. 175–188. doi: 10.1016/S0031-3203(03)00237-1.

Xie, J., Girshick, R. and Farhadi, A. (2016) 'Unsupervised Deep Embedding for Clustering Analysis', in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. JMLR.org (ICML'16), pp. 478–487.

Yilmaz, S. *et al.* (2019) 'Exploratory study on clustering methods to identify electricity use patterns in building sector', *Journal of Physics: Conference Series*, 1343(1), pp. 0–6. doi: 10.1088/1742-6596/1343/1/012044.