An Automated Clustering Process for Helping Practitioners to Identify Similar EV Charging Patterns across Multiple Temporal Granularities

René Richard[1][a], Hung Cao[1][b] and Monica Wachowicz[1,2][c]
[1]*People in Motion Lab, University of New Brunswick, Canada*
[2]*RMIT, Australia*

Keywords: Agglomerative Hierarchical Clustering, EV Adoption, Charging Infrastructure Usage Patterns, Clustering Process, Cluster Validity Indices.

Abstract: Electric vehicles (EVs) are part of the solution towards cleaner transport and cities. Clustering EV charging events has been useful for ensuring service consistency and increasing EV adoption. However, clustering presents challenges for practitioners when first selecting the appropriate hyperparameter combination for an algorithm and later when assessing the quality of clustering results. Ground truth information is usually not available for practitioners to validate the discovered patterns. As a result, it is harder to judge the effectiveness of different modelling decisions since there is no objective way to compare them. In this work, we propose a clustering process that allows for the creation of relative rankings of similar clustering results. The overall goal is to support practitioners by allowing them to compare a cluster of interest against other similar clusters over multiple temporal granularities. The efficacy of this analytical process is demonstrated with a case study using real-world Electric Vehicle (EV) charging event data from charging station operators in Atlantic Canada.

1 INTRODUCTION

Globally, national and local government commitments to electrify the transport sector will have a positive impact on smart cities. The vision for smart cities fosters advanced and modern urbanization, which results in a core infrastructure that enables a good quality of life for citizens and the sustainable management of natural resources. Supporting the usage of EVs contributes to improved air quality, sustainable mobility and therefore contributes to this vision.

The high capital costs of setting up public charging infrastructure and the usage of public funds to foster a shift to EVs necessitates informed decision making at all stages of the adoption life-cycle. Given early EV adoption challenges, some charging stations can be under-utilized, others will serve a disproportionate amount users. Clustering stations together based on utilization patterns is a useful planning tool for operators. Additionally, as vehicle electrification grows, so does the demand for electricity and the possible strain on power grids. Utilities and other power generators need to prepare for increased demand. Accurate load forecasting is one tool which can help operators ensure service consistency.

Clustering is an unsupervised learning method which assists practitioners in discovering hidden patterns from a data set. It has been utilized by practitioners in the energy domain to group similar consumers, predict future demand, and increase EV adoption. Statistical models, built with data from charging stations having similar charging patterns will reportedly have superior accuracy (Straka and Buzna, 2019). Therefore, energy load forecasting methods might preform better when applied to homogeneous clusters of stations as opposed to all stations. The patterns in energy usage behavior are core to improving services provided by utility companies, which are responsible for managing peaks and imbalances in charging infrastructure usage patterns (Iglesias and Kastner, 2013).

Although clustering is widely used in many knowledge domains, it remains arduous for practitioners to select the proper clustering algorithm with hyperparameter combination and later assess the quality of clustering results. The subjectivity found in the required expert knowledge that is needed for determining the level of "success" achieved during clustering, is likely to be one of the main reasons why

[a] https://orcid.org/0000-0002-1342-6225
[b] https://orcid.org/0000-0002-0788-4377
[c] https://orcid.org/0000-0002-4659-0101

Richard, R., Cao, H. and Wachowicz, M.
An Automated Clustering Process for Helping Practitioners to Identify Similar EV Charging Patterns across Multiple Temporal Granularities.
DOI: 10.5220/0010485000670077
In *Proceedings of the 10th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2021)*, pages 67-77
ISBN: 978-989-758-512-8

existing AutoML frameworks tend to focus on supervised learning tasks that require labeled data as input (Oliveira, 2019). One of the challenges is that the identification of the most similar clusters can be subjective and it usually requires multiple approaches to automate this process (Poulakis, 2020). The difficulty in clustering is finding the results that aligns with a practitioner's needs because in many complex data sets, there are several plausible clusters, and practitioners may have different priorities and preferences. An unsupervised clustering algorithm has no way to intrinsically infer which clusters embody desired priorities and preferences (Bae et al., 2020).

Additionally, in data with a temporal component such as EV charging events for example, assessing the structure consistency of discovered clusters over different temporal granularities, is often a lengthy manual undertaking. Metrics such as inter-cluster separation, inter-cluster homogeneity, density, and uniform cluster sizes can be computed to determine structure consistency. However, the question of how to select a particular clustering result that is more meaningful than another based on user priorities and preferences, still depends on the practitioner's capacity of distinguishing similar clusters. Towards this challenge, this research work explores whether given the prospect of a clustering result of interest, a process of objectively highlighting and recommending similar clustering results can be automated in order to support practitioners in evaluating how clustering patterns persist over multiple temporal granularities, allowing practitioners to find meaningful clusters according to their preferences and priorities. The overall motivation of this work is to assist the practitioner in navigating multiple clustering results for different temporal partitions of the same data. Providing the practitioner with an initial ranked list of clustering results and a mechanism to identify clustering similarities can assist practitioners in downstream analytical tasks such as improving regression or classification model performance.

Therefore, we propose a clustering process which uses internal cluster validity indices to enable the identification of similar clustering results across various temporal slices of data. Of primary concern in this work is the comparison of clustering results from a-priori selected temporal granularity (e.g weekly, monthly and seasonal) and how to support practitioners in identifying similar results using a reference result of interest. A case study using real-world charging event data from EV station operators in Atlantic Canada is used to evaluate the proposed clustering process in identifying similar clusters of charging stations according to their usage patterns (e.g high vs low usage).

The scientific contributions of this paper are as follows.

- Our work is unique in proposing a combination of eight internal cluster validity indices to characterize clusters at different granularities (e.g.weekly, monthly or seasonally). Previous research work has usually focused on using these indices apart from each other.
- These internal validity indices are then used to compute a proximity measure (i.e. Euclidean distance) for helping practitioners to identify similar clusters. To the best of our knowledge, this clustering procedure has never been used as an objective measure to reduce the cognitive load of practitioners in understanding clustering results.
- The use of real-world data from EV charging stations advances the understanding of charging behavior. To the best of our knowledge, no previous work has implemented an end-to-end automated clustering process that facilitates the comparison of clustering results by practitioners with different priorities and preferences.

The rest of the paper is organized as follows. In Section 2, previous research work is described. Section 3 describes the proposed clustering process underpinning our work. Section 4 provides a detailed description of the real-world EV charging event data and the end-to-end automated implementation of our proposed clustering process. In Section 5, we discuss the results. Finally, Section 6 concludes and indicates future research work.

2 RELATED WORK

In clustering, various steps must be taken by a practitioner such as the selection of an appropriate algorithm and its hyperparameters, the choice of an adequate proximity measure, and how to validate the modeling results. Fig. 1 outlines a typical cluster analysis process.

Additionally, the temporal granularity of an algorithm's input data can generate different clusters over time. A common problem in clustering is how to objectively and quantitatively evaluate the results. Cluster validation is an important task in the clustering process because it aims to compare clustering results and solve the question of optimal cluster count. Many internal validity indices have been proposed to assess the level of "success" that a clustering algorithm achieves in finding the natural clusters in data without any class label information (Rendón et al., 2011), (Liu et al., 2010).

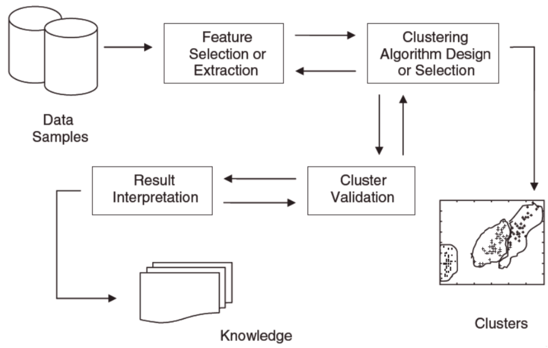


Figure 1: The Main Tasks of a Clustering Process as Described in (Messina, nd).

The preponderance of studies validating cluster results have been focused on the computation of individual cluster validity indices (CVI), which are usually selected to determine the relative performance of clustering results. In (Arbelaitz et al., 2013), Arbelaitz et al. perform an extensive comparative study of 30 CVI that are evaluated by using an experimental setup which recommends the "best" partitioning in multiple data sets where ground truth information exists. The optimal suggested number of partitions is defined as the one that is the most similar to the correct one measured by partition similarity measures. The authors found that noise and cluster overlap had the greatest impact on CVI performance. Some indices performed well with high dimensionality data sets and in cases where homogeneity of the cluster densities disappeared. The conclusion in this work suggests using several CVI to obtain robust results.

In the energy domain, clustering has played an important role in revealing new insights in energy usage behavior, in particular, the EV charging demand (Al-Ogaili et al., 2019) . For example, in (Straka and Buzna, 2019), the authors demonstrated the potential of clustering to understand the usage patterns related to segments of charging stations by comparing k-means, hierarchical, and DBScan algorithms. The clustering algorithms have successfully identified four groups of EV charging stations characterized by distinct usage patterns.

In contrast, very few attempts have been found in exploring CVI for evaluating the clustering results. In (Xydas et al., 2016), the Davies-Bouldin index is used to determine the best value for the cluster count parameter using the k-means algorithm. Sun et al. (Sun et al., 2020) proposed a time series clustering method using a modified Euclidean distance to group the similar charging tails from ACN-Data collected from smart EV charging stations. In this work, they evaluated their clustering results with Dynamic Time Warping distance (DTW) and Euclidean distance method using the silhouette coefficient.

In summary, the traditional usage of CVI has been for validation purposes. However, utilizing multiple CVI together in combination with a proximity measure such as Euclidean distance has a strong potential to offer a new pairwise similarity measure that can enhance the comparison of clustering results by practitioners. Certainly, this is not a common practice in Data Science as well as in the energy domain.

3 THE PROPOSED CLUSTERING PROCESS

Our proposed clustering process extends the well-known process introduced in the previous section. Fig. 2 provides a conceptual overview of the main tasks of our proposed clustering process. The numbered items in the figure link back to individual Python scripts described in detail in the implementation section. At the end of the process, a database is used to persist all clustering results and a RESTful Application Programming Interface (API) facilitates querying these results by different practitioners.

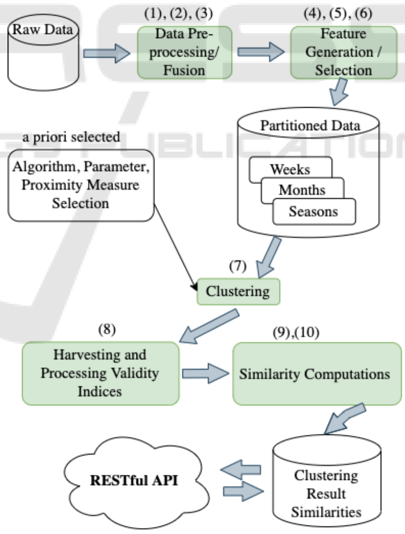


Figure 2: Our Proposed Clustering Process.

3.1 Data Preprocessing and Fusion

The data preprocessing and fusion task uses raw data from the public EV charging stations. Preprocessing consists of data cleaning and consolidation steps. Data cleaning, ensures good data quality and produces a set of cleaned files by eliminating errors, inconsistencies, duplicated and redundant data rows, and handling missing data. Data consolidation combines data from various data files into a single data set. A variety of files from the cleaned data set are used as the input for this operation. The output of these steps is a unique file that merges all attributes into one big table.

Moreover, data fusion consists of combining multiple data sources followed by a reduction or replacement for the purpose of better inference. In our proposed clustering process, consolidated station location information and charging event data files are combined to produce more consistent, accurate, and useful data files.

3.2 Feature Generation and Selection

The aim of the feature generation and selection task is to enrich pre-processed and fused data files by adding new attributes to each data row according to a specific context. This task is defined by a contextualization function that can produce a set of new data rows using contextualization parameters to add new attributes to the fused data rows. Transformed data is then partitioned using multiple temporal granularities (e.g. e.g.weekly, monthly or seasonally).

3.3 Clustering

The aim of the clustering task is to find the patterns from transformed input data using a hierarchical agglomerative clustering algorithm. The algorithm seeks to build a hierarchy of clusters by merging current pairs of mutual closest input data points until all the data points have been used in the computation. The measure of inter-cluster similarity is updated after each step using complete Ward linkage. This a priori selected algorithm is utilized to fit the various temporal granularities of input data, producing multiple clustering results. Internal clustering validity indices are recorded during each application of the clustering algorithm.

3.4 Harvesting and Processing Validity Indices

Each application of the clustering algorithm generates a record consisting of the cluster count parameter value, the various cluster validity index values and the input data used to generate the clusters. Processing the validity indices involves selecting and normalizing the index values in preparation for Euclidean distance computations. This task utilizes the combination of eight cluster validity indices which are described as follows:

3.4.1 Silhouette Index

The silhouette width of a data point measures how similar the data point is to its own cluster compared to other clusters. For clusters $X_j = (j = 1,..c)$, the silhouette width of the i_{th} data point in cluster X_j is defined as (Rendón et al., 2011):

$$S(i) = \frac{(b(i) - a(i))}{max\{a(i), b(i)\}} \tag{1}$$

Where $a(i)$ is the average distance between the i_{th} data point and all data points included in X_j ; $b(i)$ is the minimum average distance between the i_{th} data point and all of the data points clustered in $X_k = (k = 1,..c, k \neq j)$.

From individual silhouette width calculations, an aggregated global silhouette index is obtained (Petrovic, 2006). The silhouette index values range from -1 to 1 where a value closer to 1 indicates clusters are well separated and clearly distinguished. A value closer to -1 indicates data points are not properly clustered.

3.4.2 Caliński-Harabasz Index

The Caliński-Harabasz (CH) index is expressed as a ratio of between-cluster variance and the overall within-cluster variance. A recent comparative study of available clustering indices demonstrated this index as one of the best cluster validity indices (Arbelaitz et al., 2013). Well defined clusters yield high values of this index. Therefore, the maximum value of the index is used to select the best partition. For n data points, k clusters where B and W are the between within cluster scatter matrices, the index is computed as (Gurrutxaga et al., 2011):

$$CH = \frac{traceB/(k-1)}{traceW/(n-k)} \tag{2}$$

3.4.3 Davies-Bouldin Index

The Davies-Bouldin (DB) index is defined as follows (Gurrutxaga et al., 2011):

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1,..,k;\, j\neq i}(d_{ij}) \tag{3}$$

where

$$d_{ij} = \frac{s_i/s_j}{d(c_i, c_j)} \tag{4}$$

In this formula, k is the number of clusters, s_i is the average distance of all data points in cluster i to their cluster centroid and $d(c_i, c_j)$ is the distance between the centroids of clusters i and j. With this index, a minimum value denotes the best partitioning of the data.

3.4.4 Cohesion

Cohesion is measured by the sum of squared distances from each data point to its respective centroid. Also referred to as the within sum of squares (WSS), it measures how closely related data points are in a cluster. The WSS is defined as (López et al., 2017) :

$$WSS = \sum_{i=1}^{N_c} \sum_{x \in C_i} d(X, \bar{X}_{C_i})^2 \tag{5}$$

Where C_i is the cluster N_c is the number of clusters $\bar{X}_{C_i}$ is the cluster centroid, and $\bar{X}$ is he sample mean. The goal in clustering is to minimize the value of WSS.

3.4.5 Separation

Measure how distinct or well separated a cluster is from other clusters. Calculated as the sum of the squared deviations between the groups, it is defined as (López et al., 2017):

$$BSS = \sum_{i=1}^{N_c} |C_i| \bullet d(\bar{X}_{C_i}, \bar{X})^2 \tag{6}$$

In this formula, $|C_i|$ is the size of the cluster N_c is the number of clusters $\bar{X}_{C_i}$ is the cluster centroid, and $\bar{X}$ is he sample mean. An optimal clustering will have a higher value of BSS.

3.4.6 Root Mean Square Standard Deviation

The Root Mean Square Standard Deviation (RMSSD) measures homogeneity within clusters. A lower RMSSTD value means a better separation of clusters. Large values of RMSSTD indicates that clusters are not homogeneous. The metric is computed as (Rujasiri and Chomtee, 2009):

$$RMSSTD = \sqrt{\frac{\sum_{i=1..p}^{j=1..k} \sum_{a=1}^{n_{ij}} (x_a - \bar{x}_{ij})^2}{\sum_{i=1..p}^{j=1..k} (n_{ij} - 1)}} \tag{7}$$

Where k is the number of clusters, p is the number of independent variables in the data set, $\bar{x}_{ij}$ is the mean of values in variable j and cluster i, and n_{ij} is the number of data points which are in variable p and cluster k.

3.4.7 R-squared

The R-square (RS) value captures whether there is a significant difference among data points in different clusters and that data points in the same cluster have high similarity. RS values range from 0 to 1 where a value closer to 0 indicates there is no difference between clusters. If the R-squared value is zero, there is no difference between clusters. On the other hand, if the value is closer to 1, then the partitioning of clusters is closer to an optimal allotment. The metric is computed as (Rujasiri and Chomtee, 2009):

$$RS = \frac{SS_t - SS_w}{SS_t} \tag{8}$$

$$SS_t = \sum_{j=1}^{p} \sum_{a=1}^{n_j} (x_a - \bar{x}_j)^2 \tag{9}$$

$$SS_w = \sum_{j=1}^{i=1..k} \sum_{a=1}^{n_{ij}} (x_a - \bar{x}_{ij})^2 \tag{10}$$

In these equations, SS_t is the sum of squared distances among all variables, SS_w is the sum of square distances among all data points in the same cluster, k is the number of clusters, p is the number of independent variables in the data set, $\bar{x}_j$ is the mean of data in variable j, $\bar{x}_{ij}$ is the mean of the data in variable j and cluster i and n_{ij} is the number of data which are in variable p and cluster k.

3.4.8 Xie-Beni Index

The Xie-Beni (XB) index is applicable to fuzzy and crisp clustering results. It is defined as the quotient between the mean quadratic error and the minimum of the minimal squared distances between the points in the clusters. The index is defined as (Chakrabarty, 2010) :

$$XB(K) = \frac{\sum_{k=1}^{K} \sum_{j=1}^{n} (\mu_{kj})^m \parallel x_j - z_k \parallel^2}{n \times \underset{\leq i \leq K, 1 \leq j \leq K}{min} \parallel x_j - z_k \parallel^2} \tag{11}$$

Where the nominator measures cluster compactness and the denominator measures the separation between different cluster centers. The value of the XB index should be minimum for the optimum number of clusters in the data. The parameter m is called the fuzzifier and is usually set between 1 and 2.

3.5 Similarity Computations

Our work uses a proximity measure in the clustering task and in the computation of the results similarity matrix. Selecting a this measure to determine how similar or dissimilar two data points is an important step in any clustering process. Proximity measures affect the shape of clusters as some data points may be close to one another according to one measure and far from each other according to another. Euclidean distance is a preferred distance measure by researchers in the field of clustering and is defined as (Chakrabarty, 2010):

$$D(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \tag{12}$$

In addition to the clustering task, the similarity computation task uses Euclidean distance as the proximity measure between clustering results. All index values (e.g. multidimensional points in Euclidean space) of each clustering results are used in the distance computations. The pair-wise similarity comparisons (e.g. the similarity matrix) are then persisted in a database for down-stream results exploration via a RESTful API.

The similarity matrix is stored in the database using two tables. The first table summarizes clustering results with rows consisting of a unique clustering result ID (*result_id*) and meta-data about running the algorithm (e.g. input file name, clustering execution time, all validity index values, etc.). The second table, which is linked to the first table, contains rows consisting of a source result ID (*from_result_id*), a target result ID (*to_result_id*) and a Euclidean distance. Links between result IDs are not duplicated as directionality is not considered.

4 IMPLEMENTATION

This work makes use of real operational data from public EV charging stations provided by the New Brunswick Power Corporation. 9,505 EV charging events that occurred between the dates of April 2019 and April 2020 at Level-2 (L2) and Level-3 (L3) public charging stations were included in the analysis. Table 1 describes the raw EV charging data set features. Our practitioners are managers and planners of an utility company who are responsible for coordinating various projects including EV charging station condition assessments, operating and capital budget forecasting, and maintenance and operation practices development. Fig. 3 describes the overall end-to-end implementation of our EV use case.

Custom-written Python code and a scientific Python stack were leveraged to implement the proposed clustering process. Task elements were executed in sequence from a centralized management script (Richard et al., 2020). The software programs used in this work were packaged using a Docker (Boettiger, 2015) container in order to ensure a reproducible and consistent computational environment.

Table 1: Raw Data.

Column Name	Description
Connection ID	Unique identifier for a connection
Recharge start time (local)	Timestamp denoting start of charging event
Recharge end time (local)	Timestamp denoting end of charging event
Account name	Unused (all null)
Card identifier	Unique identifier for a charging plan member
Recharge duration (hours:minutes)	Duration of charge event
Connector used	Connection used during charge event
Start state of charge (%)	State of charge % at beginning of charging event
End state of charge (%)	State of charge % after charging event is complete
End reason	Charge event end reason
Total amount	Unused (all null)
Currency	Unused (all null)
Total kWh	Energy transferred to vehicle during charging event
Station	Unique identifier for charging station

Fig. 4 highlights noteworthy aspects of the implementation. The numbered boxes that represent individual parameterized Python scripts. The data flow is such that the output of one script is the input for the next script. Input and output file names contain parameter values that were used when calling the workflow's scripts. The grey elements represent a job's input file(s). The blue elements represent a job's output file(s). The detailed implementation of each script is described as follows:

- **Script (1):** The ***one_way_hash.py*** script imports raw event data and casts column elements to appropriate types. Additionally, a one-way hash function is applied to the *Card identifier* column.
- **Script (2):** The ***locations_to_parquet.py*** script imports raw station location data and integrates multiple input files into one.
- **Script (3):** The ***fuse_location_w_events.py*** script

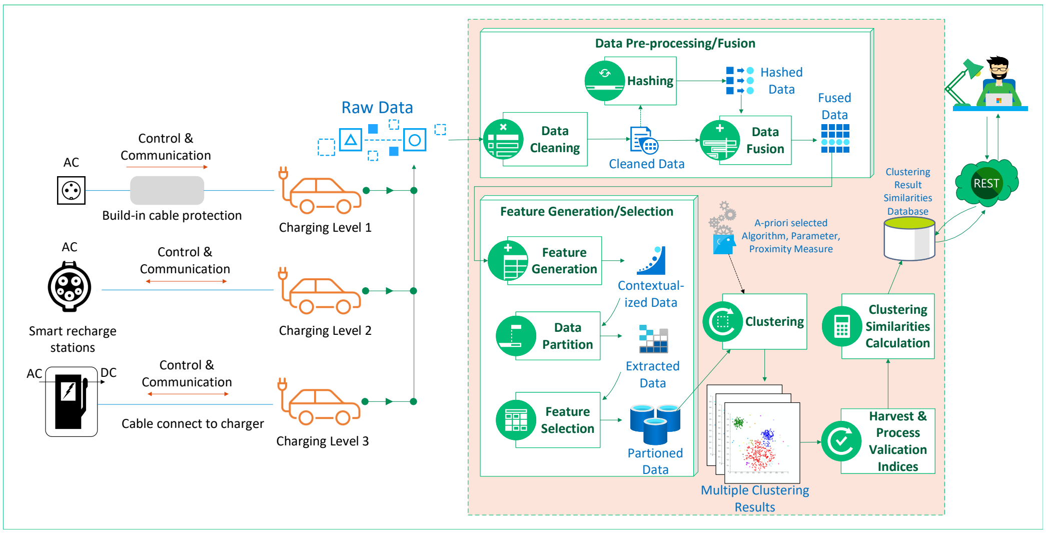


Figure 3: Overview of Our Implemented EV Use Case.

fuses event data with charging station location information.

- **Script (4):** This work focuses on recharge report event data in the downstream analysis. The ***feat_eng_rech_report.py*** script creates new features (contextualized) based on calculations involving existing data attributes and removes events with a duration of 5 minutes or less (eliminating 11% of the raw records).
- **Script (5):** The ***create_batch_ranges.py*** script creates temporal partitions of the data. These partitions facilitate the cluster analysis based on charging events occurring during a particular week, month or season of the year.
- **Script (6):** The ***generate_ev_station_features.py*** prepares the input data for clustering by calculating, for each charging station, station type and temporal granularity, the proportion of total charging events and the proportion of total power used to charge vehicles relative to all stations.
- **Script (7):** The ***cluster_data.py*** script applies the agglomerative clustering algorithm to all temporal slices of the data produced in the previous task. This is done for a cluster count hyperparameter that varies from 2 to 7. Other hyperparameter settings are kept constant to simplify the experimental setup. Internal clustering validity indices are recorded during each application of the clustering algorithm (See Table 2 for the list of indices).
- **Script (8):** The ***scale_indices.py*** script normalizes the internal clustering validity indices in preparation for the downstream Euclidean distance computations.
- **Script (9:)** The ***similarity_matrix.py*** script performs pairwise Euclidean distance computations for each clustering result. All index values (e.g. multidimensional points in Euclidean space) of each clustering results are used in the distance computations.
- **Script (10):** The ***load_data.py*** script persists the similarity matrix data produced in the previous task in a relational database to enable querying of clustering results and corresponding similarities across months, weeks and seasons. The database query functionality is made available via a RESTful API.

After results are generated and persisted (e.g. Script (10) in Fig. 4 is complete), the practitioner can navigate these results via a RESTful interface. Fig. 5 illustrates how the practitioner interacts with the results system. First, the practitioner requests ranked station clustering results for either L2 or L3 station types (Step 1). The system then returns a sorted list of clustering results ordered by silhouette score (Step 2). From this list, the practitioner selects one result as the reference result for which comparable results are desired and then request these comparable results from the system (Step 3). Finally, the system returns a sorted list of comparable clustering results that is ordered by Euclidean distance (Step 4). This sorted list contains result specific artefacts such as scatter plots, mapped station cluster memberships and silhouette plots.

The clustering process implementation and RESTful API facilitate the comparison of clustering result similarities across various temporal granularities. This process is useful in identifying avenues for further analysis. One Level 3 station clustering result for the

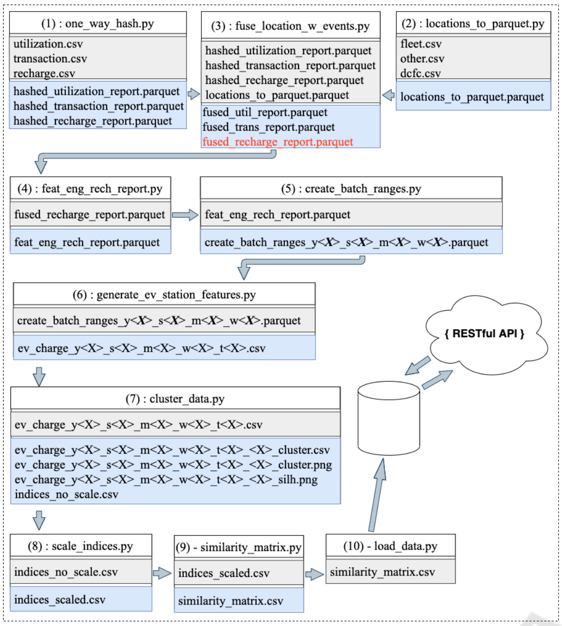


Figure 4: Data Flow Between Python Scripts of the Clustering Process Implementation.

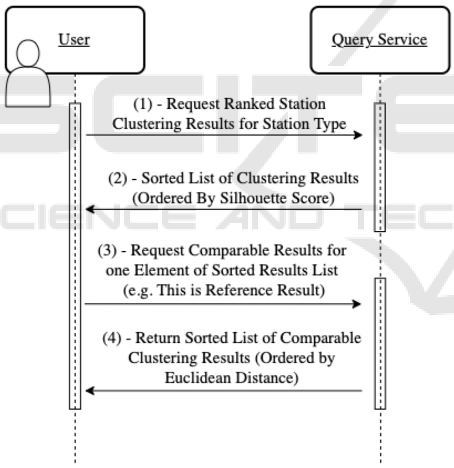


Figure 5: Results Query Sequence.

week of May 27th, 2019 has been selected as a case study to demonstrate our approach. The case study is presented in the next section.

5 RESULTS AND DISCUSSION

This section highlights the results of our proposed approach in identifying similar station clusterings over multiple weeks with a case study. Table 3 highlights similar clustering results relative to station clusterings for a target week starting on May 27th, 2019. In all results, the number of clusters is 2 and the station type is L3. The table is sorted in ascending order by Euclidean distance relative to the target week. According to the multi-dimensional pairwise distance calculations obtained using the features described in Table 2, the most similar clustering result to the week starting on May 27th, 2019 is the result for the week starting on February the 17th 2020. The least similar clustering result is the result for the week starting on December 2nd, 2019.

Table 2: Clustering Validity Index Data.

Column Name	Description
file_name	File name for clustering results for station type and time granularity
n_cluster	K parameter value used in applying the clustering algorithm
silhouette_score	Silhouette index value for clustering result
calinski_harabasz	Caliński-Harabasz index for clustering result
davies_bouldin	Davies-Bouldin index for clustering result
cohesion	Cohesion index for clustering result
separation	Separation index for clustering result
RMSSTD	Root mean square standard deviation index for clustering result
RS	R-squared index for clustering result
XB	Xie-Beni index for clustering results

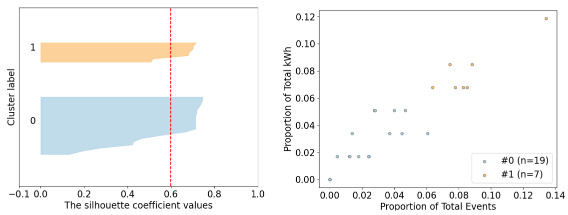
A corresponding visual presentation of the clustering results found in Table 3 can be seen in Figures 6 through 10. Each figure contains a silhouette and scatter plot describing the clustered data. In the silhouette plots, an observation with a silhouette width near 1, means that the data point is well placed in its cluster; an observation with a silhouette width closer to negative 1 indicates the likelihood that this observation might really belong in some other cluster.

Table 3: Clustering Similarities - L3 - May 27th, 2019.

WEEK	Sil	CH	DB	C	S	RMS	RS	XB	*Dist*
27/05/19	**0.60**	**51.37**	**0.51**	**1.12**	**2.40**	**0.15**	**0.68**	**0.09**	*N/A*
17/02/20	0.60	49.35	0.57	0.19	2.44	0.16	0.67	0.10	*0.081*
02/03/20	0.65	55.51	0.52	1.14	2.63	0.15	0.70	0.07	*0.101*
29/07/19	0.60	55.82	0.53	0.99	2.30	0.14	0.70	0.11	*0.105*
02/12/19	0.63	56.55	0.58	1.26	2.97	0.16	0.70	0.09	*0.177*

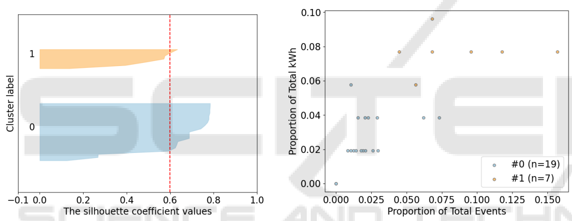
Column Name Abbreviations for Table 3

Sil :	Silhouette index
CH :	Caliński-Harabasz index
DB :	Davies-Bouldin index
C :	Cohesion
S :	Separation
RMS :	Root mean square standard deviation
RS :	R-squared
XB :	Xie-Beni index
Dist :	Euclidean distance between current and previous row



(a) (b)

Figure 6: L3 Station Clusters - MAY-27-2019.



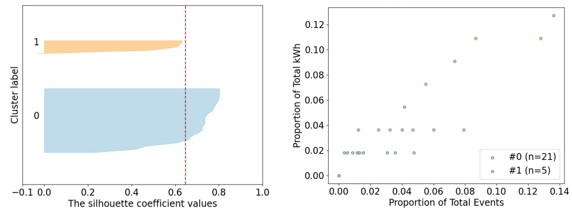
(a) (b)

Figure 7: L3 Station Clusters - FEB-17-2020.

We can see from Figures 6 and 7, reasonable structures in the data have been found. Stations are grouped in terms of relatively higher or lower utilization rates. The average silhouette score is 0.60 in both clustering results. The number of observations in each cluster for both results are also the same. Cluster 0 in both situations have more stations with relatively lower utilization rates. Results for the week of May 27th, 2019 are slightly better when considering all cluster validation indices. This can also be observed visually. Data points seem to be closer together in the scatter plot of Fig. 6b than in Fig. 7b. The in-between cluster separation in both results are similar.

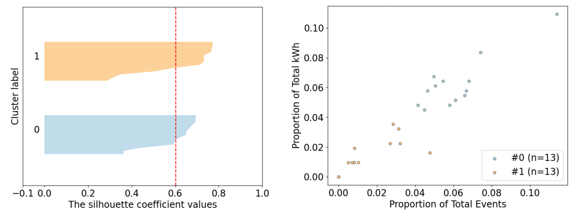
The silhouette plot in Fig. 8a suggests a less optimal clustering. This plot indicates that some observations would seemingly belong to clusters other than the one they are in; these observations have a negative silhouette width value.

The silhouette plot in Fig. 9a and the average silhouette score of 0.60 suggest a reasonable structure in



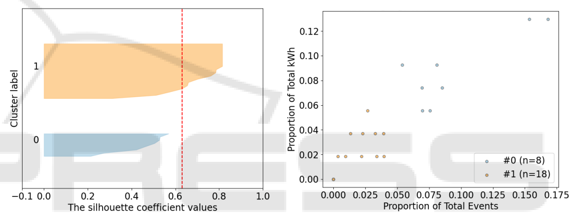
(a) (b)

Figure 8: L3 Station Clusters - MAR-02-2020.



(a) (b)

Figure 9: L3 Station Clusters - JUL-29-2019.



(a) (b)

Figure 10: L3 Station Clusters - DEC-02-2019.

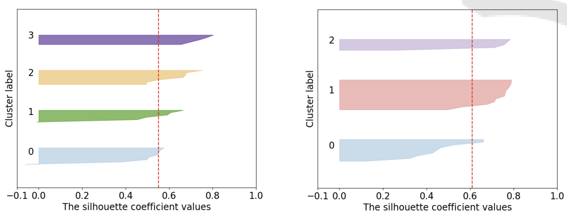
the data has also been found in this week. The number of observations in each cluster for both clustering results are different. Based on the various indices, clustering results for July 29th, 2020 are better in some aspects and inferior in others to results for the week of May 27th, 2019. This result was identified as being the 3rd most similar result for our target week.

The decreasing relative similarity of results is especially visible when comparing the results for the week of May 27th, 2019 with results having the least similarity (i.e, results for the week of December 2nd, 2019). In Fig. 10a we can see that all cluster 0's members have below average silhouette scores and the clustering of stations is much less similar than the other clusterings.

Individual index calculations embed implicit trade-offs on what is prioritized when expressing inter-cluster separation, inter-cluster homogeneity, density, and compactness as one numeric value. One can view the various indices as averages where a certain precision is lost in the summary. This can lead to situations where one index will suggest a better clustering relative to another grouping and another index will inverse this assessment. This is illustrated in Table 3 where for example, the silhouette and Caliński-Harabasz index values for December 2th suggest a better clustering than on the week starting on May 27th. However, the Davies-Bouldin and R-squared index values inverse this assessment.

Capital investments in public charging infrastructure involves the use of public funds and necessitates robust informed decision making. Identifying similar station utilization patterns over multiple weeks can be useful planning information for station operators. The cluster analysis presented in our case study provides useful insights by identifying similar groupings of EV charging stations according to their usage patterns in time.

The results highlighted in the case study provided in this section demonstrate that given a clustering result of interest, a process of objectively highlighting and recommending similar clustering results can indeed be automated in order to support the practitioner in evaluating how structure in data persists over multiple time slices in a data set with temporal properties. The relative ranking of similar clustering results that our approach affords makes it easy to objectively identify similar station groupings over multiple weeks based on a reference week. Not highlighted in the case study, are the clustering results for other a-priori selected temporal partitions in the data, which are also available as reference points for exploring monthly or seasonal clustering similarities. For example silhouette plots representing a reference month (where K=4) and season (where K=3), see Fig. 11.



(a) (b)

Figure 11: L3 Station Clustering References - August and Spring.

6 CONCLUSIONS

Although clustering has become a routine analytical task in many research domains, it remains arduous for practitioners to select a good algorithm with adequate hyperparameters and to assess the quality of clustering and the consistency of identified structures over various temporal slices of data. The process of clustering data is often an iterative, lengthy, manual and cognitively demanding task. The subjectivity in determining the level of "success" that unsupervised learning approaches are able to achieve and the required expert knowledge during the modeling phase suggest that a human-in-the-loop process of supporting the practitioner during this activity would be beneficial. Ascertaining whether a particular clustering of data is meaningful or not requires expertise and effort. Doing this for multiple results on data that has been sliced by weekly, monthly or seasonal partitions prior to applying the clustering algorithm would be very time consuming. Manually identifying one meaningful result of interest and then having an automated mechanism to select similar results is extremely useful in reducing the amount of effort required to identify avenues that merit further analysis and assist in downstream analytical tasks such as improving regression or classification model performance.

A case study using real-world charging event data from EV station operators in Atlantic Canada was used to validate the approach and identify similar groupings of charging stations according to their usage patterns. Our work demonstrates that given a clustering result of interest, the process of objectively highlighting and recommending similar clustering results can be automated in order to support the practitioner in evaluating how structure in data persists over multiple time slices and reduce the cognitive load of identifying multiple meaningful clustering results from a large number of modeling artifacts.

Presenting the practitioner with an initial ranked list of clustering results leveraging all index values simultaneously instead of just using the silhouette scores (as described in Step 1 of Fig. 5) may improve the initial results exploration process. Framing the creation of the initial ranked list of results as a Multiple Criteria Decision Making (MCDM) problem will be included in future work. Additionally, we will explore if an expert can label a portion of the modeling artifacts as meaningful or not and whether a semi-supervised or other algorithm can automatically label the rest of the unseen modeling results from the labels provided by the practitioner. Finally, other avenues will explore whether this work can be adapted to implement a novel change point detection approach in identifying significant changes in station groupings in temporal slices of the data.

ACKNOWLEDGEMENTS

The authors of this paper like to thank the New Brunswick Power Corporation for providing access to station operator users and the EV charging data referenced in this research. This work was partially supported by the NSERC/Cisco Industrial Research Chair, Grant IRCPJ 488403-1.

REFERENCES

Al-Ogaili, A. S., Hashim, T. J. T., Rahmat, N. A., Ramasamy, A. K., Marsadek, M. B., Faisal, M., and Hannan, M. A. (2019). Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: challenges and recommendations. *Ieee Access*, 7:128353–128371.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., PéRez, J. M., and Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

Bae, J., Helldin, T., Riveiro, M., Nowaczyk, S., Bouguelia, M.-R., and Falkman, G. (2020). Interactive Clustering: A Comprehensive Review. *ACM Computing Surveys*, 53(1):1–39.

Boettiger, C. (2015). An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79.

Chakrabarty, A. (2010). *An investigation of clustering algorithms and soft computing approaches for pattern recognition*. PhD thesis, Assam University.

Gurrutxaga, I., Muguerza, J., Arbelaitz, O., Pérez, J. M., and Martín, J. I. (2011). Towards a standard methodology to evaluate internal cluster validity indices. *Pattern Recognition Letters*, 32(3):505–515.

Iglesias, F. and Kastner, W. (2013). Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies*, 6(2):579–597.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916. IEEE.

López, S. L. S., Redondo, R. P. D., and Vilas, A. F. (2017). Discovering knowledge from student interactions: clustering vs classification. In *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 1–8.

Messina, E. (n.d.). Cluster analysis, powerpoint slides. (Accessed 2020-11-07).

Oliveira, M. (2019). 3 Reasons Why AutoML Won't Replace Data Scientists Yet. (March 2019).

Petrovic, S. (2006). A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. In *Proceedings of the 11th Nordic Workshop of Secure IT Systems*, pages 53–64. Citeseer.

Poulakis, G. (2020). Unsupervised automl: a study on automated machine learning in the context of clustering. Master's thesis, Πανεπιστήμιο Πειραιώς.

Rendón, E., Abundez, I., Arizmendi, A., and Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of computers and communications*, 5(1):27–34.

Richard, R., Cao, H., and Wachowicz, M. (2020). Discovering ev recharging patterns through an automated analytical workflow. In *2020 IEEE International Smart Cities Conference (ISC2)*, pages 1–8.

Rujasiri, P. and Chomtee, B. (2009). Comparison of clustering techniques for cluster analysis. *Nat. Sci*, 43:378–388.

Straka, M. and Buzna, L. (2019). Clustering algorithms applied to usage related segments of electric vehicle charging stations. *Transportation Research Procedia*, 40:1576–1582.

Sun, C., Li, T., Low, S. H., and Li, V. O. (2020). Classification of electric vehicle charging time series with selective clustering. *Electric Power Systems Research*, 189:106695.

Xydas, E., Marmaras, C., Cipcigan, L. M., Jenkins, N., Carroll, S., and Barker, M. (2016). A data-driven approach for characterising the charging demand of electric vehicles: A uk case study. *Applied energy*, 162:763–771.