Clustering Study of Vehicle Behaviors Using License Plate Recognition

Daniel Bolaños-Martinez, Maria Bermudez-Edo, Jose Luis Garrido

Please cite as:
Bolaños-Martinez, D., Bermudez-Edo, M., Garrido, J.L. (2023). Clustering Study of Vehicle Behaviors Using License Plate Recognition. In: Bravo, J., Ochoa, S., Favela, J. (eds) Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022). UCAmI 2022. Lecture Notes in Networks and Systems, vol 594. Springer, Cham. https://doi.org/10.1007/978-3-031-21333-5_77

Clustering Study of Vehicle Behaviors Using License Plate Recognition

Daniel Bolaños-Martinez[0000−0003−0207−2908], Maria Bermudez-Edo[0000−0002−2028−4755], and Jose Luis Garrido[0000−0001−7004−1957]

University of Granada, C/Pdta. Saucedo Aranda s/n, 18014 Granada, Spain

Abstract. Ubiquitous computing and artificial intelligence contribute to deploying intelligent environments. Sensor networks in cities generate large amounts of data that can be analyzed to provide relevant information in different fields, such as traffic control. We propose an analysis of vehicular behavior based on license plate recognition (LPR) in a rural region of three small villages. The contribution is twofold. First, we extend an existing taxonomy of the most widely used clustering algorithms in machine learning with additional classes. Second, we compare the performance of algorithms from each class of the taxonomy, extracting behavioral patterns. Partitional and hierarchical algorithms obtain the best results, while density-based algorithms have poor results. The results show four differentiated patterns in vehicular behavior, distinguishing different patterns in both residents and tourists. Our work can help policymakers develop strategies to improve services in rural villages, and developers choose the correct algorithm for a similar study.

Keywords: IoE · AI · sensors · smart environments · clustering

1 Introduction

Nowadays, there are 35.82 million devices connected to the Internet and according to Statista, this number is expected to increase to 50 billion by 2030[1]. They are sometimes referred to as ubiquitous technologies [12]. These devices form an interconnected network that generates massive amounts of data in many different areas of social life. Access to a large amount of data collected by multiple sensors and heterogeneous sources allows researchers to apply this paradigm to numerous fields such as healthcare, smart environments or transportation [2,6,17].

The paradigms of ubiquitous computing and ambient intelligence play an important role in smart cities. By installing sensors in cities, it is possible to control aspects such as vehicle traffic control, evacuation systems, smart lighting control, or waste management [27,9,7,31]. In this field, terms such as Machine Learning (ML) or Big Data are becoming increasingly popular, allowing to study and extract precise conclusions. By performing a robust analysis of the data extracted by sensors, it is possible to obtain useful information that allows scientists to

[1] `https://findstack.com/internet-of-things-statistics/`

develop strategies to improve their market position with respect to their competitors (e.g. sustainability or super-exploitation of tourism) [19].

This paper is part of a project that aims at building a smart city platform in a rural area where individuals with different patterns of behavior coexist. Following the strategy of ubiquitous computing, in this work we have installed license plate recognition (LPR) devices in a rural region consisting of three touristic villages (Pampaneira, Bubión and Capileira) in The Alpujarra, Spain. From the data collected by the LPR cameras, we calculated the spatial visitation frequency and the period of stay of each vehicle in the different villages. We propose an unsupervised ML analysis of vehicle mobility patterns to classify each vehicle according to its behavior in the area. We compare several clustering algorithms and extend the taxonomy proposed in [18], with two additional classes, to cover all the algorithms studied in the literature. Our results are useful for policymakers to get insights about any potential problems or enhancements with the traffic. This work is also useful for developers and data scientists in choosing a clustering algorithm for their analysis.

The remainder of the paper is organized as follows. In Section 2 related work is summarized. Section 3 presents the unsupervised learning methodology, the sensor setup and the construction of the dataset. Section 4 includes the design of the experiments and Section 5 the analysis of the results. Finally, conclusions and future work are summarized in Section 6.

2 Related Work

Technological advances have made detecting vehicles and other means of transport with sensors increasingly powerful. The massive data source represented by these devices provides the information needed to analyze vehicle behavior and apply it in almost any context. The analysis of mobility patterns with the aim of improving traffic problems [25], and the clustering of vehicles to extract useful conclusions for the management of cities [21], are the most common works in this area of interest.

To infer mobility patterns from the raw data, unsupervised ML is widely used. The most commonly used clustering algorithms are density-based, as they can adapt to problems where irregular behavior occurs within the population. For example, [28] presents a framework for vehicle data collection from Road Side Units (RSU) and data transmission to the cloud. To reduce the data's complexity, they perform a clustering analysis (using adapted versions of DBSCAN and OPTICS). In [3], a modified version of DBSCAN (iterative and multi-attribute) was used to cluster the different zones in the port, with the objective of improving organization and solving port congestion. In [37], they propose constructing the spatio-temporal similarity matrix using the dynamic time warping (DTW) algorithm from the data extracted by LPRs and applying DBSCAN to group different displacement patterns on the similarity matrix.

Other works have opted to use partitional clustering algorithms. In [38], they use KMeans from data obtained by LPR devices in a smart city to analyze the spatio-temporal travel patterns of citizens during the COVID-19 pandemic. Another study [39], uses ISODATA to cluster mobility patterns and a decision tree to create decision rules between the attributes and the labeling obtained from the clustering.

In our work, we use the LPR technology, used in most of the related works and perform a clustering analysis based on the spatio-temporal information of vehicles. Unlike previous works, we applied a study directly on individuals and their behavior based on their spatial frequencies of visitation. Additionally, we carry out a comparative study on the popularly used clustering algorithms.

3 Methodology

3.1 Clustering Algorithms and Metrics

Clustering is an unsupervised ML task that aims to find patterns in a given event's observations. According to the method adopted to define clusters, most taxonomies divide the algorithms into four categories [18]:

Partitional Clustering: Decomposes the dataset into disjoint clusters by an iterative process usually based on centroids. Examples include algorithms such as KMeans or MiniBatchKMeans, a more scalable version of KMeans using small random batches to update the clusters until convergence is reached [4]. Another example is ISODATA [24] which uses iterative self-organizing data analysis.

Hierarchical Clustering: Proceeds in an agglomerative or divisive way with the construction of clusters by adding or removing individuals respectively. This algorithm may vary its performance depending on the metrics and linkage criteria used. Most notably in this category is the BIRCH algorithm [40] which uses an unbalanced height tree to split the data points dynamically.

Density-based Clustering: Clusters are dense regions of objects in the data space separated by low-density regions. These are known to handle noise well and to adapt to arbitrary shapes in the data. The most popular algorithm in this category is DBSCAN [14] and some improved versions of it, such as OPTICS [1] or HDBSCAN [23], which compute a density function for each cluster found. Other examples such as Mean-shift [10] create the clusters based on regions of maximum density attraction, so it can be seen as a version of KMeans using density functions, which makes it adaptable to arbitrary shapes in clusters.

Distribution-based Clustering: Clusters are created based on the probability of each individual belonging to the same distribution. The most commonly used distribution is the Gaussian distribution, based on the expectation-maximization algorithm [36]. These algorithms result in Gaussian Mixture models, which are also classification algorithms. In some cases, they are a generalization of KMeans (each individual has a probability of belonging to each cluster).

Grid-based Clustering: First, the space is organized into a finite number of cells, and then clustering operations are defined in the quantified space. Some common examples of algorithms that follow this clustering strategy are: STING [35], WaveCluster [32] and CLIQUE [15].

To the above taxonomy, we add two new categories that we believe cover the algorithms found in the related works:

Message Passing-based Clustering: Creates the clusters by iteratively sending messages between the different data points until they converge. The Affinity Propagation (AP) algorithm is an example of this category [16]. Improved AP proposals include IWC-KAP [30] and ScaleAP [33].

Spectral Clustering: Use the spectral radius of a similarity matrix of the data in a multidimensional complex problem, applying methods (e.g. PCA) to reduce its dimensions in order to obtain a linearly separable problem. There are different versions of Spectral Clustering algorithms depending on how the eigenvectors are selected from the Laplacian of the similarity matrix [34]. New versions have also emerged such as Attributed Spectral Clustering (ASC) which improves the degree of affinity of nodes in the same density region [5].

To evaluate the performance of the different algorithms, we will use the four most popular internal evaluation metrics in the literature: silhouette coefficient, calinski-harabasz score, davies-bouldin index and Density-Based Clustering Validation (DBCV, specific for density-based algorithms).

- **Silhouette Coefficient:** measures the similarity of an individual to its own cluster compared to other clusters [29]. The value of the coefficient is defined in the interval $[-1, 1]$, where a value close to 1 indicates good clustering and a value close to -1 indicates poor individual aggregation.
- **Calinski-Harabasz Score:** like the silhouette coefficient, measures how similar an individual is to its group relative to other groups [8]. A higher value minimizes the intracluster covariance of individuals and maximizes the intercluster covariance.
- **Davies-Bouldin Index:** small values indicate compact clusters, whose centers are well separated from each other [11].
- **Density-Based Clustering Validity:** this validation measure takes values in the interval $[-1, 1]$, values close to 1 indicate a better density-based cluster solution [26].

3.2 Setup

Four devices incorporating vehicle detection sensors collect the data. These devices are Hikvision LPR IP cameras with ANPR system based on Deep Learning, supporting a higher degree of deviation and increasing the hit rate. The devices feature 2MP resolution, 2.8-12mm varifocal optics, and IR LEDs with 50m range.

We placed the four cameras in strategic positions to cover the entrances and exits to each village in the target area: (i) entrance to Pampaneira from the western part of the Alpujarra, (ii) entrance to Pampaneira from the eastern part of the Alpujarra, (iii) entrance to Bubión (single road), and (iv) entrance to Capileira (single road). The structure of the roads allows us to control the mobility of all vehicles circulating in the Poqueira area using only four LPRs.

3.3 Dataset

The LPR cameras defined in Section 3.2 return the license plate and time stamp of each vehicle passing through its coverage area. Based on this data, we perform a preprocessing to calculate the number of visits, the average time, and the total time spent in each village. In case of missing data, i.e., we cannot calculate the entry and exit time of a vehicle, we discard it from the dataset.

We have produced a dataset from the above information, which contains information on 17.924 vehicles. The attributes used are total visits, total time spent in the area, average visit time, and standard deviation of the average visit time. The above attributes take into account visits to any of the three villages.

The dataset contains four months of data (from February to May 2022). However, we have eliminated the festive periods relating to a long weekend in February (a holiday in Andalusia, Spain) and Holy Week. This first approximation aims at establishing a vehicular pattern restricted to periods of normal daily mobility (without considering big holidays).

4 Experiments

We opt for unsupervised ML because we need to group individuals that are not labeled into different classes. In this paper, we test different clustering solutions to analyze which one best suits the pattern recognition problem and whether it can find a realistic solution to the problem. We tried to use at least one algorithm from each category described in Section 3.1. In particular, we used the following algorithms: KMeans and MiniBatchKMeans from the category partitional clustering, Agglomerative Clustering, Ward (Agglomerative Clustering with ward linkage criteria) and BIRCH from hierarchical clustering, Gaussian Mixture from distribution-based clustering, a version of Spectral Clustering applied to a projection of the normalized Laplacian and finally, from the density-based algorithms, we used DBSCAN, HDBSCAN, and MeanShift. In our study, we discard the Affinity Propagation algorithm as it is not scalable [33] and OPTICS for being HDBSCAN an improved version of the latter [22]. In addition, we have not used any grid-based clustering algorithms because: (i) there are no well-tested implementations, and (ii) it is not a clustering category widely used in related work.

The existence of attributes at different scales and measured in different units (number of visits vs. time) makes some variables more influential than others in the clustering process. To avoid this bias, we apply two types of normalization to all dataset attributes.

1. **Min-max normalization:** normalizes the values to the interval $[0, 1]$ maintaining the distances for each data point by using the minimum and maximum in the attribute domain.
2. **Standard normalization:** scales the values so that the mean of the data domain is 0 and the standard deviation is equal to 1.

We use the scikit-learn library, which contains implementations for all the algorithms[2] defined in Section 3.1 (except HDBSCAN, which has its own library with the same name). We use the implementation of the Silhouette metrics, CH and DB, from the metrics module of scikit-learn, while we adopt the DBCV metric from a github repository [26].

We calculate the Silhouette and DB metrics and apply the elbow method to choose the optimal number of clusters generated by each algorithm [13]. These metrics are widely used to select the optimal number of clusters using KMeans [20], so we will test their performance with all the selected algorithms and with both normalizations. The next step will be the selection of the most frequent number of clusters (for each normalization) based on each algorithm's metric (DB and Silhouette) results. Once we have the most frequent cluster numbers, we will calculate the four metrics defined in Section 3.1 for each algorithm by imposing parameters that generate the selected cluster numbers. We do this to compare each algorithm with the best solution for each metric that has generated a given number of clusters. Finally, we compare the cluster segmentation performed by each algorithm and describe the individuals in each cluster generated by the best solution obtained for each type of data normalization.

5 Results and Discussion

After calculating the Silhouette and DB metrics for the algorithms, for the min-max normalization, the most frequent number of clusters is 2 (selected as the optimal number of clusters 14 times). We discarded the options of 4 clusters (3 times), 3 clusters (2 times) and 6 clusters (1 time), as they account for less than 20% of the cases. The standard normalization produces 2 (9 times), 3 (7 times) and 4 (4 times), so we take all these values (2, 3 and 4 clusters) for our experiment.

After calculating all the metrics of Section 3.1 for all algorithms by imposing for min-max normalisation 2 clusters and for standard normalisation 2,3 and 4 clusters, we obtain the following results:

- Agglomerative Clustering obtains the best results for silhouette and DB metrics for any number of clusters.
- Spectral Clustering obtains similar results to Agglomerative Clustering for standard normalization.
- CH metric obtains the best values for KMeans algorithm, followed by MiniBatchKMeans. For the standard normalization, Gaussian Mixtures, obtains good results for this metric.
- The DBCV metric generally performs better for density-based algorithms. However, the DBCV results are not significant, as they do not obtain values higher than 0 [26]. In addition, the execution time is 200 times higher than the other metrics, which implies scalability issues on massive datasets. For these reasons, we discard this metric from further analysis and will study a proper implementation of the metric in the future.

[2] `https://scikit-learn.org/stable/modules/clustering.html`

Table 1. Performance of the 4 best algorithms for the Calinski-Hasbaraz metric (fixed cluster number) for each normalization.

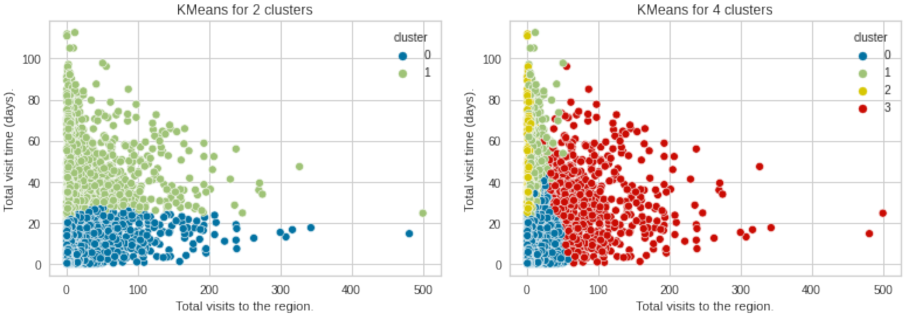
Min-max normalization (2 clusters)				**Standard normalization (4 clusters)**			
Algorithm	**CH score**	**clusters size**	**time**	**Algorithm**	**CH score**	**clusters size**	**time**
KMeans	25.405,190	0: 16660 (92.95%) 1: 1264 (7.05%)	1,82253	KMeans	14.306,313	0: 16542 (92.29%) 3: 601 (3.35%) 1: 536 (2.99%) 2: 245 (1.37%)	2,02979
MiniBatchKMeans	25.339,993	0: 16499 (92.05%) 1: 1425 (7.95%)	0,76922	MiniBatchKMeans	12.435,697	0: 14874 (82.98%) 2: 1776 (9.91%) 3: 699 (3.90%) 1: 575 (3.21%)	0,12010
Spectral Clustering	22.956,758	0: 17090 (95.35%) 1: 834 (4.65%)	35,13015	Ward	11.462,843	1: 15814 (88.23%) 0: 1095 (6.11%) 2: 864 (4.82%) 3: 151 (0.84%)	10,03337
Gaussian Mixture	24.086,366	0: 16803 (93.75%) 1: 1121 (6.25%)	0,29580	Gaussian Mixture	12.279,663	0: 16335 (91.13%) 2: 962 (5.37%) 1: 338 (1.89%) 3: 289 (1.61%)	0,37849

However, Agglomerative Clustering and Spectral Clustering that maximize or minimize the silhouette and DB metrics respectively obtain a segmentation with 2 clusters classifying all the data in one cluster except 2 records. Therefore, it is not a suitable partition. We believe that this result is because the metrics only consider the inter-cluster and intra-cluster distances, so they promote the creation of clusters formed by outliers. However, algorithms that maximize the CH give more confidence in creating clusters. This is because CH not only considers the distance between individuals in the same and different clusters but also minimizes/maximizes the covariance. Hence, we select the algorithms with the best CH results for each normalization.

Table 1 shows that the segmentation performed for algorithms that obtain comparable values for the CH metric is similar. To conclude the analysis, we characterized each cluster that has obtained the best CH, using scatter and box plots. KMeans has obtained the best results for the two types of normalization performed on the data. Hence, we study the segmentation performed for 2 clusters (min-max normalization) and 4 clusters (standard normalization).

Figure 1 **(a)** shows a segmentation across individuals through an imaginary horizontal axis located at 20 days. Performing a study for the attributes of the dataset (see Figure 2), the following characteristics define each cluster:

1. Individuals in cluster 0 have a length of stay in the order of hours. This cluster also has fewer visits on average, except for several outliers that can be seen in the initial scatter plot (Figure 1 **(a)**).
2. Cluster 1 contains individuals who have spent between 30 and 60 days in the Poqueira area and have a higher number of visits to the area and a higher average length of stay than cluster 0.



(a) Min-max normalization.

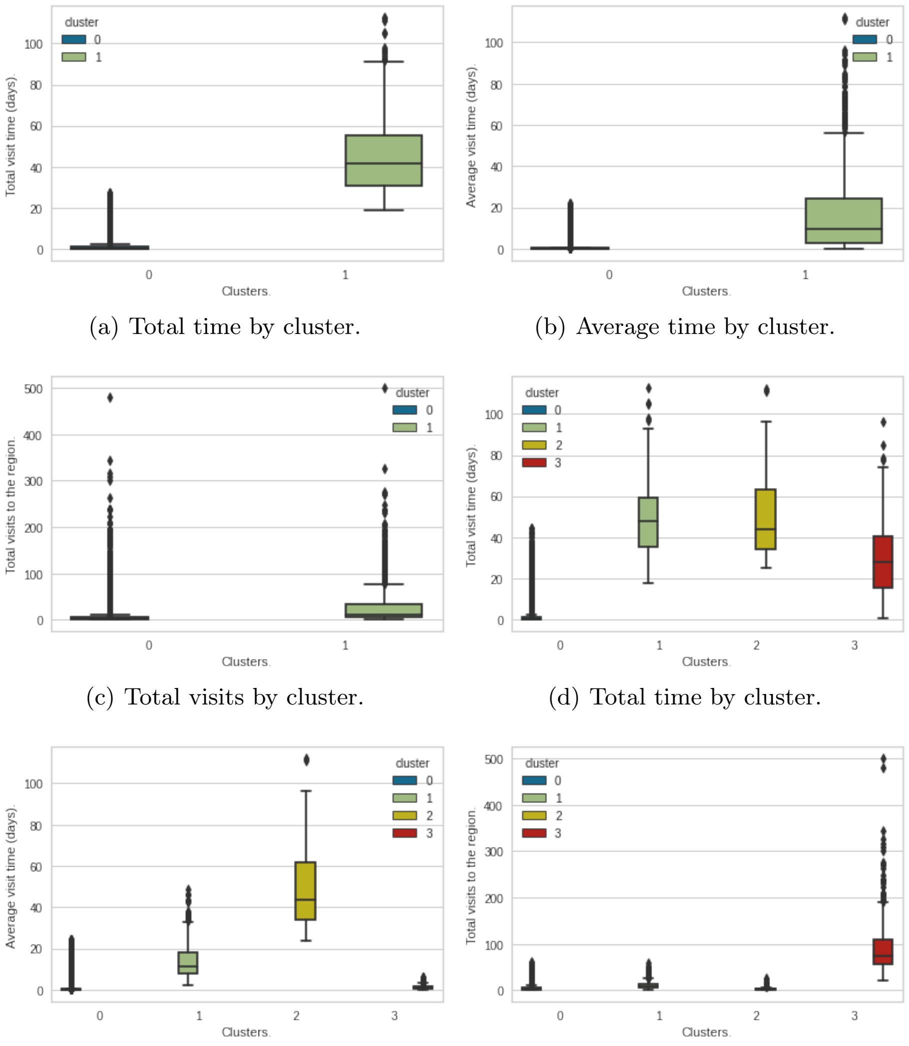
(b) Standard normalization.

Fig. 1. Scatter Plots generated for KMeans (total time vs. total visits). We have selected the number of clusters that maximises the CH metric for each normalisation.

Figure 1 **(b)**, contrary to Figure 1 **(a)**, shows a segmentation of the individuals through a vertical axis. Performing an analogous study for the attributes (see Figure 2), we can describe the following characteristics for each cluster:

1. For cluster 0, its average visit time for 50% of the individuals is between 1 and 3 hours. In addition, the average time of stay and total visits to the area are significantly lower compared to the other clusters (less than 10 visits to the area lasting less than 4 hours on average).
2. Clusters 1 and 2 have a similar distribution of total visit length, 50% of the individuals are in the area for more than 1 month. The difference between cluster 1 and 2 is in the average stay time and total visits. While the average stay time of cluster 2 is greater than cluster 1, the total number of visits of cluster 1 is bigger than cluster 2.
3. Cluster 3 includes individuals with similar average visit times to those in cluster 0, but with longer total lengths of stay. Its total number of visits to the area is higher than any defined clustering (more than 50), resulting in a higher total length of stay in the area (50% of the distribution has a total time in the area between 15 and 40 days).

From the previous results, we can conclude that the clustering performed by the min-max normalization describes individuals in cluster 1 as residents and those in cluster 0 as visitors. This is because the segmentation separates the two clusters parallel to the abscissa (the total time of stay in the area makes the difference). However, it is not very tolerant to outliers because our dataset has more visitors than residents. The min-max normalization benefits the minority class, obtaining a higher metric value for fewer clusters among which there are outliers.



(a) Total time by cluster.

(b) Average time by cluster.

(c) Total visits by cluster.

(d) Total time by cluster.

(e) Average time by cluster.

(f) Total visits by cluster.

Fig. 2. Boxplots for min-max **(a,b,c)** and standard **(d,e,f)** normalization.

However, the clustering performed by the standard normalization makes a detailed separation between residents and visitors. On the one hand, clusters 1 and 2 define residents with specific features distinguishing 2 different behaviors between the residents of the area. Cluster 1 contains residents who leave the village a few times a week, while cluster 2 contains residents who occasionally leave once or twice a month. On the other hand, cluster 0 and cluster 3 characterize visitors. While cluster 0 characterizes tourists who are occasional visitors with a short stay in the area, cluster 3 describes visitors who come to the area frequently. Furthermore, cluster 3 probably contains different behaviors that could be subdivided in turn into other groups (workers from other Alpujarra villages, shippers, etc).

For our use case, partitional and hierarchical algorithms obtain the best interpretable segmentations. However, density-based algorithms, used in many mobility-related works, have problems in performing segmentations for a small number of clusters. This may be because our problem's low number of attributes conditions the clustering.

6 Conclusions

This paper presents an extension of the common taxonomy used in clustering, adding two classes to cover the algorithms studied in the literature and performs a comparative analysis of the clustering algorithms in a specific use case. In particular, we characterized the vehicle's behavior in a rural touristic area, obtaining four different behaviours. Partitional and hierarchical algorithms achieve the best results, and density based algorithms performed poorly due to the low number of attributes in our dataset. From these results, policymakers can know the existing mobility patterns in the area and, on this basis, make strategic decisions. In the future, we will increase the number of attributes to enhance the results. For example, we will add information related to the division of the data by village (Pampaneira, Bubión, and Capileira), separation of data by holidays or working days, and time slots.

Acknowledgements The research is funded by the AEI. ref: PID2019-109644RB-I00/AEI/10.13039/501100011033, and the Spanish Ministry of Science and Innovation, LifeWatch-ERIC action (LifeWatch-2019-10-UGR-01).

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. ACM Sigmod record **28**(2), 49–60 (1999)
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer networks **54**(15), 2787–2805 (2010)
3. Bai, X., Ma, Z., Hou, Y., Yang, D.: A data-driven iterative multi-attribute clustering algorithm and its application in port congestion estimation. Available at SSRN 4086627 (2022)

4. Béjar Alonso, J.: K-means vs mini batch k-means: a comparison (2013)
5. Berahmand, K., Mohammadi, M., Faroughi, A., Mohammadiani, R.P.: A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix. Cluster Computing **25**(2), 869–888 (2022)
6. Bermudez-Edo, M., Barnaghi, P., Moessner, K.: Analysing real world data streams with spatio-temporal correlations: Entropy vs. pearson correlation. Automation in Construction **88**, 87–100 (2018)
7. Bhavadeesh, R., Kumar, P.T.C., Srinivas, D., Krishnaveni, R.: Iot based smart street lighting system for smart city. In: 2021 5th International Conference on Information Systems and Computer Networks (ISCON). pp. 1–3. IEEE (2021)
8. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)
9. Centelles, R.P., Freitag, F., Meseguer, R., Navarro, L., Ochoa, S.F., Santos, R.M.: A lora-based communication system for coordinated response in an earthquake aftermath. Multidisciplinary Digital Publishing Institute Proceedings **31**(1), 73 (2019)
10. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence **24**(5), 603–619 (2002)
11. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence (2), 224–227 (1979)
12. Dhyani, K., Bhachawat, S., Prabhu, J., Kumar, M.S.: A novel survey on ubiquitous computing. In: Data Intelligence and Cognitive Informatics, pp. 109–123. Springer (2022)
13. Drakos, G.: Silhouette analysis vs elbow method vs davies-bouldin index: Selecting the optimal number of clusters for kmeans clustering. GDCoder,[Online]. Available: https://gdcoder. com/silhouetteanalysis-vs-elbow-method-vs-davies-bouldin-index-selectingthe-optimal-number-of-clusters-for-kmeans-clustering/(visited on 11/01/2022) (2020)
14. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)
15. Forster, A., Murphy, A.L.: Clique: Role-free clustering with q-learning for wireless sensor networks. In: 2009 29th IEEE International Conference on Distributed Computing Systems. pp. 441–449. IEEE (2009)
16. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. science **315**(5814), 972–976 (2007)
17. Garcia-Moreno, F.M., Bermudez-Edo, M., Rodríguez-García, E., Pérez-Mármol, J.M., Garrido, J.L., Rodríguez-Fórtiz, M.J.: A machine learning approach for semi-automatic assessment of iadl dependence in older adults with wearable sensors. International Journal of Medical Informatics **157**, 104625 (2022)
18. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering algorithms and validity measures. In: Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001. pp. 3–22. IEEE (2001)
19. Haughton, G., Hunter, C.: Sustainable cities. Routledge (2004)
20. Humaira, H., Rasyidah, R.: Determining the appropiate cluster number using elbow method for k-means algorithm. In: Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA) (2020)
21. Lin, M., Zhao, X.: Application research of neural network in vehicle target recognition and classification. In: 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS). pp. 5–8. IEEE (2019)

22. Malzer, C., Baum, M.: Hdbscan (ϵ): An alternative cluster extraction method for hdbscan. CoRR, abs/1911.02282 (2019)
23. McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. J. Open Source Softw. **2**(11), 205 (2017)
24. Memarsadeghi, N., Mount, D.M., Netanyahu, N.S., Le Moigne, J.: A fast implementation of the isodata clustering algorithm. International Journal of Computational Geometry & Applications **17**(01), 71–103 (2007)
25. Mondal, M.A., Rehena, Z.: Identifying traffic congestion pattern using k-means clustering technique. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). pp. 1–5. IEEE (2019)
26. Moulavi, D., Jaskowiak, P.A., Campello, R.J., Zimek, A., Sander, J.: Density-based clustering validation. In: Proceedings of the 2014 SIAM international conference on data mining. pp. 839–847. SIAM (2014)
27. Ning, Z., Huang, J., Wang, X.: Vehicular fog computing: Enabling real-time traffic management for smart cities. IEEE Wireless Communications **26**(1), 87–93 (2019)
28. Peixoto, M.L.M., Maia, A.H., Mota, E., Rangel, E., Costa, D.G., Turgut, D., Villas, L.A.: A traffic data clustering framework based on fog computing for vanets. Vehicular Communications **31**, 100370 (2021)
29. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
30. Serdah, A.M., Ashour, W.M.: Clustering large-scale data based on modified affinity propagation algorithm. Journal of Artificial Intelligence and Soft Computing Research **6**(1), 23–33 (2016)
31. Sharma, M., Joshi, S., Kannan, D., Govindan, K., Singh, R., Purohit, H.: Internet of things (iot) adoption barriers of smart cities' waste management: An indian context. Journal of Cleaner Production **270**, 122047 (2020)
32. Sheikholeslami, G., Chatterjee, S., Zhang, A.: Wavecluster: A multi-resolution clustering approach for very large spatial databases. In: VLDB. vol. 98, pp. 428–439 (1998)
33. Shiokawa, H.: Scalable affinity propagation for massive datasets. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 9639–9646 (2021)
34. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
35. Wang, W., Yang, J., Muntz, R., et al.: Sting: A statistical information grid approach to spatial data mining. In: Vldb. vol. 97, pp. 186–195. Citeseer (1997)
36. Yang, M.S., Lai, C.Y., Lin, C.Y.: A robust em clustering algorithm for gaussian mixture models. Pattern Recognition **45**(11), 3950–3961 (2012)
37. Yao, W., Chen, C., Su, H., Chen, N., Jin, S., Bai, C.: Analysis of key commuting routes based on spatiotemporal trip chain. Journal of Advanced Transportation **2022** (2022)
38. Yao, W., Yu, J., Yang, Y., Chen, N., Jin, S., Hu, Y., Bai, C.: Understanding travel behavior adjustment under covid-19. Communications in Transportation Research p. 100068 (2022)
39. Yao, W., Zhang, M., Jin, S., Ma, D.: Understanding vehicles commuting pattern based on license plate recognition data. Transportation Research Part C: Emerging Technologies **128**, 103142 (2021)
40. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. ACM sigmod record **25**(2), 103–114 (1996)