Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2024) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

AIIT 4th International Conference on Transport Infrastructure and Systems (TIS ROMA 2024), 19th - 20th September 2024, Rome Italy

Empirical analysis of traffic patterns: Leveraging machine learning techniques for in-Depth insights

Z. Lahijanian[a,*], N. Isaenko[a], G. Fusco[a], C. Colombaroni[a]

[a]*Sapienza University of Rome, Via Eudossiana 18, 0018,Rome, Italy.*

Abstract

Efficient traffic modeling and control depend on accurate travel condition data. Previously, limited data forced analysts to use single days or small averages, missing minor variations. Now, increased data improves accuracy but complicates traffic monitoring, as daily model calibration is impractical. This paper presents a systematic methodology that uses machine learning-based cluster analysis to categorize days into distinct groups representing specific traffic patterns providing more precise responses algin with temporal variation travel conditions. Instead of pre-selecting groups, clustering discovers natural groups that exist within the data. Initially, traffic time series data over eleven months from a motorway section in Italy were cleaned and processed against missing data and noises. Subsequently, three clustering approaches, K-Means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Affinity Propagation (AP) were used to identify groups with similar traffic patterns and the performance of different approaches are evaluated. Results confirm the compatibility of the K-Means and DBSCAN algorithm effectiveness. Conversely, AP suggests a significantly higher number of clusters, with negligible differences. As expected, notable distinction is noted by clusters between the school period, non-school period, weekdays, and weekends. The usage of clustering algorithms facilitated the identification of six distinct day types within the dataset. Furthermore, this approach can be applied daily to detect outliers and anomalies in both demand and supply. This is a noticeable feature to support mobility agencies in planning and applying special policies to face anomalies conditions.

Peer-review under responsibility of the scientific committee of the Transport Infrastructure and Systems (TIS ROMA 2024)

Keywords: Traffic patterns; Clustering algorithms; data analysis.

* Corresponding author. Tel.: +06-44-58-5145

E-mail address: zahra.lahijanian@uniroma1.it

1. Introduction

The huge amount of mobility data that is accessible and stored in databases enables an extensive awareness of network travel conditions, as well as the development of modelling tools and control strategies. However, realizing

Peer-review under responsibility of the scientific committee of the Transport Infrastructure and Systems (TIS ROMA 2024)

these benefits requires the implementation of an efficient and effective approach of data analysis and pattern identification, (Han et al., 2022).

Cluster analysis is a fundamental method in data analysis that helps identify natural groupings in a dataset (Han et al., 2022) with widely acknowledged as a valuable tool for uncovering latent patterns in various scientific domains including genetics (Jiang et al., 2004), marketing and sales (Chang et al., 2009), evaluation of underground water quality (Celestino et al., 2018), and analysis of traffic patterns in (Chung, 2014), (Weijermars Wendy, 2005), (Habtemichael and Cetin, 2016a), and (Rao and Reddy, 2012). Clustering divides items into groups with similar properties, while different clusters have distinct attributes. Data preparation, attribute selection, clustering approach, and stopping criteria significantly impact the outcome and performance (Steinbach et al., 2004). Among different applied clustering techniques addressing traffic pattern detection, three widely used approaches like partitioning, density-based, and exemplar-based category of methods stand out.

Partitioning clustering aims to divide data into distinct, non-overlapping groups, while an object can only be part of one cluster, and each cluster includes at least one object. Application of popular partitioning clustering algorithm addressing traffic pattern recognition are K-Means (Erman et al., 2006; Jiang et al., 2004), K-nearest neighborhood (Habtemichael and Cetin, 2016) and Hierarchical (Wang et al., 2018) clustering algorithms. The key strength of density-based clustering approach is its capacity to identify high-density zones in datasets, which makes it possible to identify abnormal behavior among typical patterns. Its durability and adaptability make it suitable for a wide range of applications, including pattern recognition and anomaly detection. Some examples of DBSCAN can be found in (Erman et al., 2006; Ester et al., 1996; Savvas et al., 2018). Finding a set of exemplars that comprehensively summarize the whole dataset and allocating each data point to an exemplar is the aim of exemplar-based clustering techniques such as AP (Frey and Dueck, 2007). To categorize days into distinct groups and indicate unique observed travel conditions, this study uses a range of clustering algorithms, each with unique properties, namely DBSCAN, AP, and K-Means followed by validation analysis of method's performances.

Section 2 of the following provides a detailed explanation of the applicable procedures, while Section 3 explains the development of the methodology. Section 4 ultimately presents the study's findings.

2. Methodology

In this study, we evaluate the effectiveness of clustering techniques in detecting underlying similarities of traffic profiles. To enhance this ability, we propose a systematic procedure of data analysis based on clustering algorithm along with adjusting clustering parameters and validated by real data. The step for this analytical study includes: 1) data preparation, 2) clustering algorithms and criteria identification, 3) result validation, and 4) clusters revision, thoroughly explained in following.

2.1. Data preparation

Data is typically collected from loop detectors which often becomes incomplete or imperfect. What concerns data preparation mostly be considered dealing with missing data, noises, and outliers. This involves filtering the dataset from unfeasible scenarios or in other words outliers, such as high-density regions with high-speed data points or high flow rate with low-speed measurements. Unlike outliers, noisy data are not easy to define, since natural fluctuation on flow profile caused by road topology could make local sharp increase in network density. It's advisable to complement count data with additional sources of data to assess the impact of external factors like accidents or adverse weather conditions and to verify network topology. Additionally, the level of data aggregation has a direct effect in diluting the data fluctuation and data absence.

2.2. Clustering algorithms

The selection of the clustering algorithm depends on the properties of the data, data massiveness, the goals of the work, and computing considerations. The following exposes and differs the mechanism of three applied method in this study while a comparison of each algorithm steps is illustrated in Table 1.

2.2.1. K-Means

The widely used K-Means algorithm operates under the assumption of spherical cluster shapes and requires a predetermined number of clusters. It is appropriate for datasets with distinct, well-defined cluster structures because of its emphasis on centroid initialization and iterative optimization. In this algorithm, 'K' is a user-defined parameter that specifies the number of clusters into which the data is segmented. Determining the right number of clusters for K-Means clustering poses a significant challenge. A low number of clusters could lead to incorrect data categorization, whereas a large number could result in many clusters with minimal differences. In this case, the algorithm groups the traffic profiles based on the Euclidean distance, equation (1).

$$d(pi, qi) = \sqrt{\sum(qi - pi)^2} \quad (1)$$

Where d is the Euclidean distance between two i-dimensional points, p and q. The algorithm starts by randomly selecting K data points as the initial centroids and assigning data points to the nearest centroid based on the distance between them and the centroid. The centroid of clusters after the first data point assignment changes to the average of participant data in that cluster and will be updated till the centroids no longer change or a maximum number of iterations is reached.

2.2.2. DBSCAN

Based on local density fluctuations, DBSCAN is highly effective in recognizing clusters of different sizes and shapes. It excels in analyzing complex and large datasets when the number of clusters is unknown, as unlike K-Mean, it does not need users to predetermine the number of clusters. The procedure begins by choosing an unexplored point and uses the Euclidean distance to find all the points within an Epsilon (Eps), or radius, around that point. A cluster is created if the radius's point count is higher than or equal to the given minimum point count (MinPts). Selecting the Eps radius presents a significant challenge when using the DBSCAN algorithm, since if Eps is set excessively high, numerous data points that belong to distinct clusters may be combined into a single, large cluster. However, certain data points that should be included in a cluster may be labeled as noise points if the Eps is set too low. Selecting an appropriate value for epsilon frequently requires some trial and error and optimization techniques, and it frequently depends on the dataset and the intended clustering outcomes.

2.2.3. Affinity Propagation

Exemplar-based method, called affinity propagation, offers flexibility in cluster form delineation by automatically determining cluster centroids using message passing interactions among data points. The communication exchanged between points includes evaluation data indicating the likelihood of each point serving as an exemplar for another point, as well as how well other points can represent that point. This communication occurs between every pair of points, contributing to the complexity of AP expensive calculation. The algorithm initializes by introducing three matrices, similarity matrix (s) based on pairwise Euclidean distance, the responsibility matrix (r), and the availability matrix (a) quantifying the influence of data points on each other. In each iteration, the algorithm computes, responsibility, and availability matrices indicating how well-suited each data point is to serve as an exemplar (cluster centre), equation (2) and equation (3) respectively. Through the procedure a damping factor in availability matrix is considered to ensure stability and prevent oscillations of applied updates, equation (4).

$$r(i,k) \leftarrow s(i,k) - \max[a(i,k') + s(i,k')]\ \forall k' \neq k \quad (2)$$
$$a(i,k) \leftarrow min\ [0, r(k,k) + \sum_{i \nexists \{i,k\}}(\max\ (0, r(k,k') \quad (3)$$
$$a(i,k) \leftarrow (1-\gamma)\ .r(i,k) + \gamma.a(i,k) \quad (4)$$

The selection of parameters, such as the damping factor, which regulates the pace of message passing convergence, have an important impact on how well AP performs.

Table 1. Flow chart of performed clustering algorithms.

Algorithm 1: k-Means clustering algorithm	**Algorithm 2**: DBSCAN clustering algorithm	**Algorithm 3**: AP clustering algorithm

Input:
D = {d_1, d_2, …, d_n} // set of n data items.
K // Number of desired clusters
 	
Output:
A set of k clusters.
 	
Steps:
1. Arbitrarily choose k data-items from D as initial centroids.
2. **Repeat**
2.1 Calculating the distance of all data with centroids.
2.2 Assigning data item di to the cluster which has the closest centroid.
2.3 Update the centroids (mean of data in cluster till now)
Until Convergence criterion met.
	Input:
	D = {d_1, d_2, …, d_n} // set of n data items.
	Eps // Radius of each d_i
	MinPts // Minimum number of samples for creating a cluster.
	
	Output:
	A set of clusters.
	
	Steps:
	1. Defining core-point and boarder points
	2. **Repeat**
	2.1 defining core-point, boarder point.
	2.2 Randomly choose a core-point.
	2.3 Assigning other core points in the circle of each core point to the cluster.
	2.4 Assigning the boarder points to the cluster.
	Until No boarder point left
	Assign all remained points to the last cluster.
		Input:
		D = {d_1, d_2, …, d_n} // set of n data items.
		γ // Damping factor
		
		Output:
		A set of clusters.
		
		Steps:
		1. Initialize performance matrix P, responsibility matrix R, and availability matrix A to zero.
		2. Compute similarity matrix S.
		3. **Repeat**
		3.1 Compute the sum of similarity and availability for each di and dj
		3.2 Update the responsibility and availability matrix with damping.
		3.3 Compute Exemplars and clusters.
		3.4 Assign data to its nearest exemplar to form a cluster.
		Until Convergence criterion met

2.3. Clustering Cross-Validation

Since the proposed methods function as unsupervised learning models, evaluating the cluster quality is necessary to determine how well the clusters that are produced work. Numerous statistical techniques, such as Silhouette Analysis, Dunn Index, Gap Statistics and Rand Index, can be used to accomplish this evaluation. Silhouettes Analysis is employed to express how similar the point is to its own cluster in relation to other clusters. Every data point has a silhouette score, $S(i)$ running from -1 to 1, with a value near +1 relating to an ideal fit to the data point's own cluster and an inadequate match to clusters nearby. A score close to -1 indicates that the data point may have been assigned to the wrong cluster. Mathematically, the silhouette score S(i) for a data point i can be calculated as follows:

$$a(i) = \frac{\sum_{j \in Ci} d(i,j)}{|C_i|-1} \tag{4}$$

$$b(i) = min_{k \neq i} \frac{\sum_{j \in C_k} d(i,j)}{|C_i|-1} \tag{5}$$

$$S(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{6}$$

Where d(i,j) represents the distance between data points i and j (typically Euclidean distance), and |Ci| is the number of points in cluster Ci, a(i) the average distance of point i to all points in other clusters, and b(i) the minimum value in a(i) set of distances for point i.

3. Application results

3.1. Case study

The dataset utilized in this study is derived from an advanced data collection system, which provided detailed traffic counts aggregated per minute over the span of eleven months in the year 2021 related to a critical carriageway including a section of expressway and a highway portion connecting the cities of Padua and Mestre in Italy. The lengths of the examined expressway and highway portions are about 45 km and 53 km, respectively, offering substantial data for robust analysis. This segment is strategically important as it comprises two carriageways, each with three traffic lanes, monitored by 48 count sections in total and includes a total of 20 access points, split evenly between on-ramps and off-ramps.

3.2. Preprocessing and data cleaning

A major component of clustering analysis is data preprocessing. The initial step to address this involves cleansing the field data based on accuracy attributes provided by the data collection systems. The data then undergoes a feasibility analysis to discard any instances of unrealistic traffic conditions. Recorded counts illustrating average speed of three lanes, above 120 km/hr and hourly flow more than 4000 vehicle per km for light vehicle and hourly flow for heavy vehicle more than 1000 are excluded. By joining external data sources related to event recorded within this network, days with lane closure, accidents, maintenance operations and adverse weather conditions which was got last more than 2 hours and had occurred in morning or evening peak hour has eliminated from data set. Additionally, this process identifies areas with abnormally high or low flow rates. Subsequently, normalization is performed to detect abrupt changes in flow for each segment. This is followed by hourly aggregation, which helps to manage the swings seen at shorter periods. Traffic profiles from one count location before applying clustering algorithms are depicted in Fig. 1 to illustrate the initial flow profiles state for a selected count section.

3.3. Performing clustering and algorithm parameters identification

The algorithm's parameters play an essential role in functionality and performance of clustering approach and should be carefully adjusted to match the features of the dataset. Every clustering algorithm has its own set of specifications. For instance, the number of clusters in K-Means, epsilon value and minimum number of points in DBSCAN and damping factor in AP are the parameters required to be calibrated before applying algorithm. To address these considerations, in analytical studies the statistical Elbow method is often employed. This method calculates the effective changes on outcome of a procedure by changing a single input value. In Fig. 1- a, the graph shows the distortion inside clusters value by increasing the number of clusters (K) for K-Mean clustering algorithm. There is a decrease in variance within clusters as the number of clusters grows since it is more likely that comparable data will be clustered together. This decrease in variance eventually reaches a plateau, suggesting that adding more clusters will not increase the effectiveness of clustering.

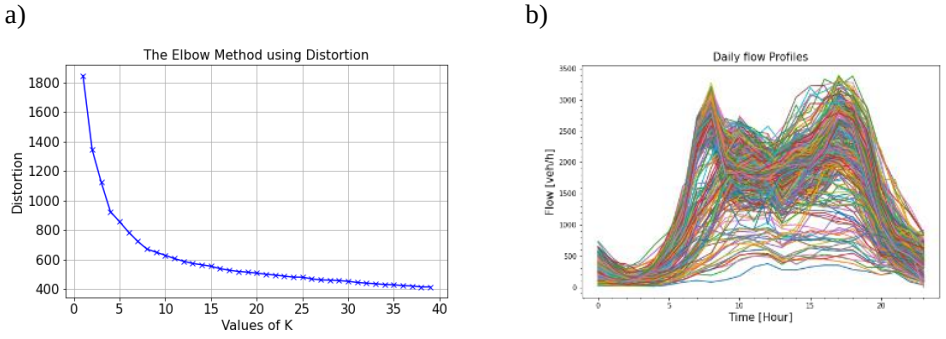

Fig. 1. a) Elbow method, b) Daily profiles of a selected count section

Following a careful review of the data, the decision was made to adopt six clusters as the optimal structure for analysis. In parallel, a series of trials utilizing DBSCAN were performed to examine the impact of different epsilon values and the minimum required points for creating a cluster. These trials demonstrated that the count of clusters generally persisted within a stable bracket of 1 to 10, so in this study, five points are chosen to serve as the MinPts. According to the findings, the range of 400 to 600 was filled by the epsilon value that corresponded to a significant increase in the derivative of clustered data for many count locations data, thus 500 veh/km was employed as epsilon value. Different damping factors were tried for AP using the same methods. It was found that the maximum damping and best outcomes were obtained when the damping factor was set to 0.5.

Fig. 2 displays the centroids of the clusters formed for a certain count section via three clustering algorithms. As can be seen, different numbers of unique clusters are identified by each technique. While DBSCAN recognizes two

flow regions, one with a high peak and one with a low peak, K-Means only recognizes regions with rising flow peaks. On the other hand, AP recognizes several groups that represent various flow patterns.

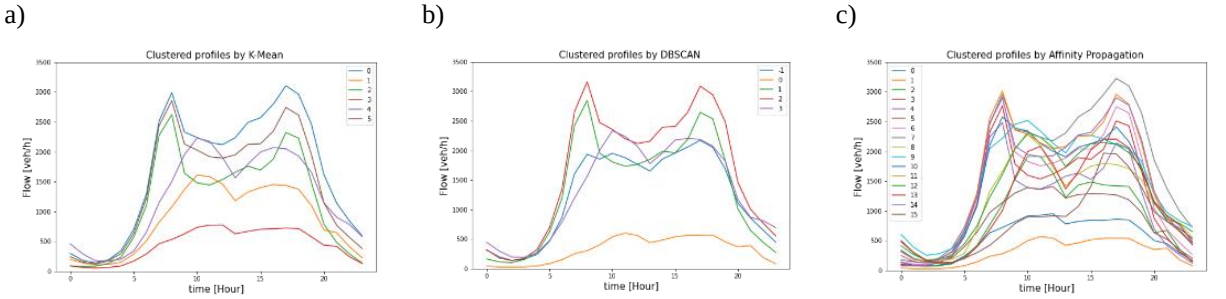

Fig. 2. Clusters' centroids for same count location by, a) K-Means algorithm, b) DBSCAN algorithm, and c) Affinity Propagation algorithm

The Table 2 shows the distribution of days in each cluster as determined by various algorithms, corresponding to the selected count section. Following assessment, three of the five clusters generated by DBSCAN have much less days than the other clusters, but K-Means shows a more consistent distribution across all clusters except for cluster number 5. In contrast, AP tends to produce a higher number of clusters. A comparison between DBSCAN and AP reveals that certain profiles deemed as outliers by DBSCAN are distinctly different from the profiles AP categorizes into clusters with high likelihood.

Table 2. Distribution of days in clusters by applied algorithms.

Algorithm\Cluster	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
K-Mean		37	23	60	21	42	125										
DBSCAN	139	8	152	4	5												
AP		12	9	11	33	30	8	54	16	13	10	12	37	14	12	34	3

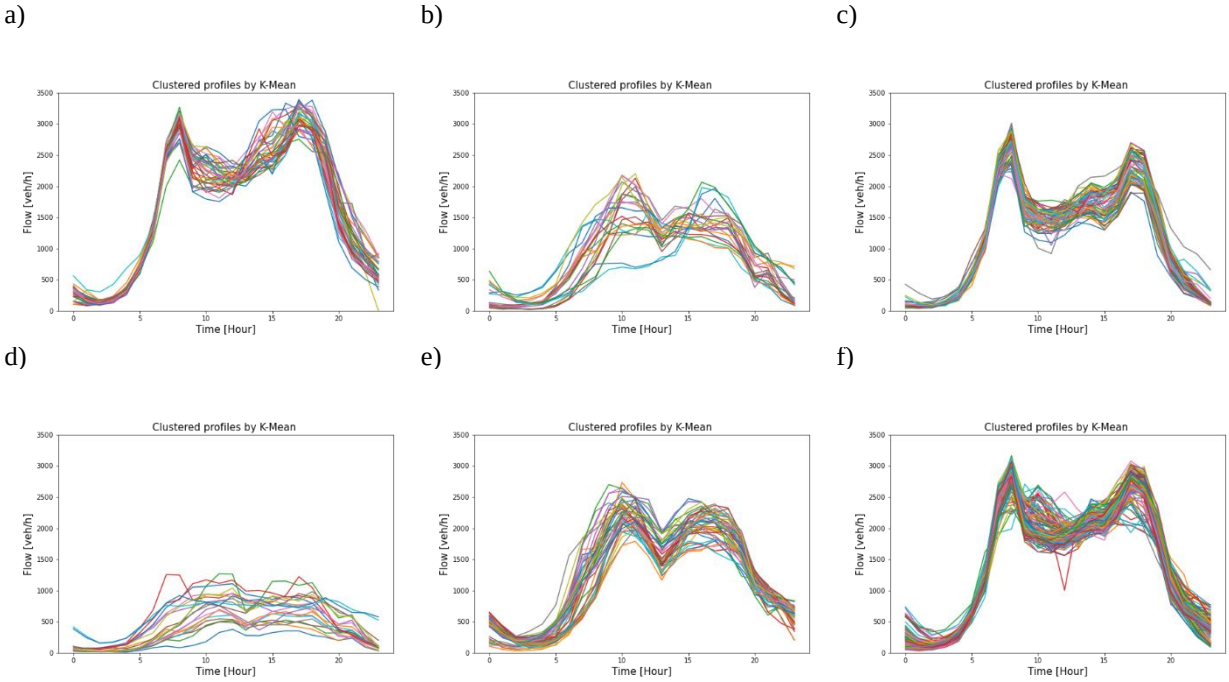

Fig. 3. Clustered data by K-Mean algorithm, a) Cluster 0, b) Cluster 1, c) Cluster 2, d) Cluster 3, e) Cluster 4, f) Cluster 5.

Fig. 3 illustrates the uniqueness of the profiles in each cluster and reveals the temporal variation of traffic in daily basis. These differences extend beyond just the maximum capacity reached in a day but also encompass the timing and placement of peaks. The standard deviation observed across the data for each hour before clustering and within each clustered dataset illustrates the reduction in deviation within the clustered data particularly during morning peak hours, Fig. 4. Looking at the contribution of day types in forming the clusters by K-Mean, Table 3, reveals a more significant disparity than anticipated in the traffic behaviour between weekdays and weekends as well as noticeable distinction between the days in June, July and August compared to the rest of the year.

Table 3 Cluster's combination by K-Means clustering algorithm for selected count location.

Cluster	No. of days	Major types of days
Cluster 0	37	Saturday and Friday of months June, July, and August.
Cluster 1	23	Saturday and Sunday of months February, March, April, and May.
Cluster 2	60	Weekdays of months January, February, March, and April.
Cluster 3	21	Sunday of months January, March, and April.
Cluster 4	42	Sunday and Saturday of months, June, July, and August.
Cluster 5	125	weekdays of months May, June, July, and August.

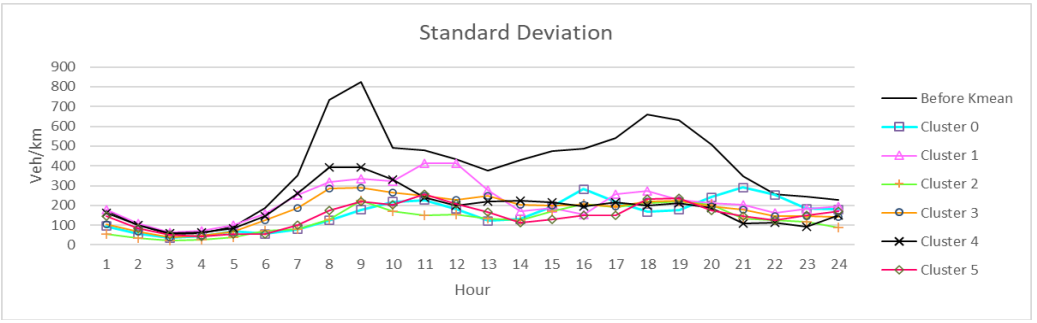

Fig. 4. Standard deviation of data for a selected count section

3.4. Clustering validation

The results of silhouette analysis for 48 count places across three methods are presented in the Fig. 5. With an average score of about 0.5, the K-Means algorithm performs better than the others in many count locations.

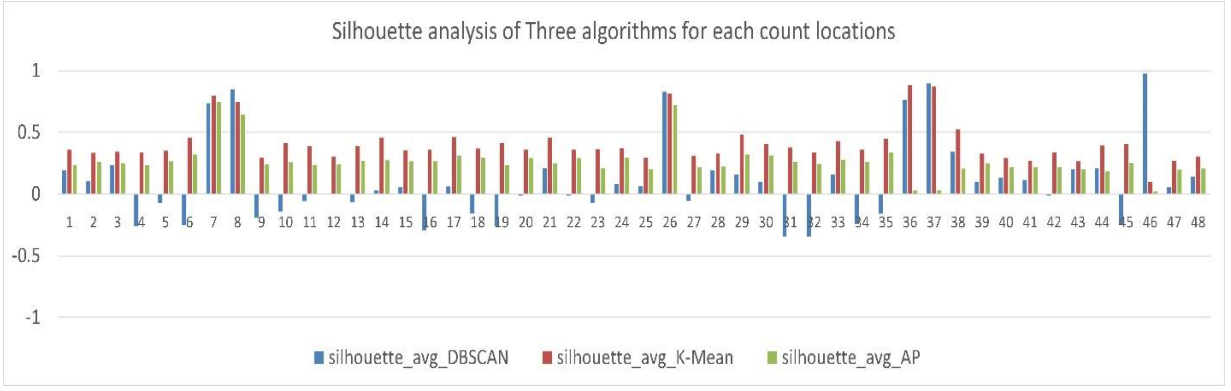

Fig. 5. Silhouette analysis results.

Interestingly, K-Means consistently produces greater scores whenever it is used compared to the AP algorithm. On the other hand, DBSCAN shows negative scores, with only a few segments showing comparatively higher values (sections n:7, 8, 26, 36, 37, and 38); even in these cases, the DBSCAN's scores are still lower than K-Means. When comparing the results of silhouette analysis with the number of clusters generated by each method, as shown in Fig.

6, K-Means outperforms the other two methods at most count locations. DBSCAN shows good performance in instances where its number of clusters is close to that determined by K-Means.

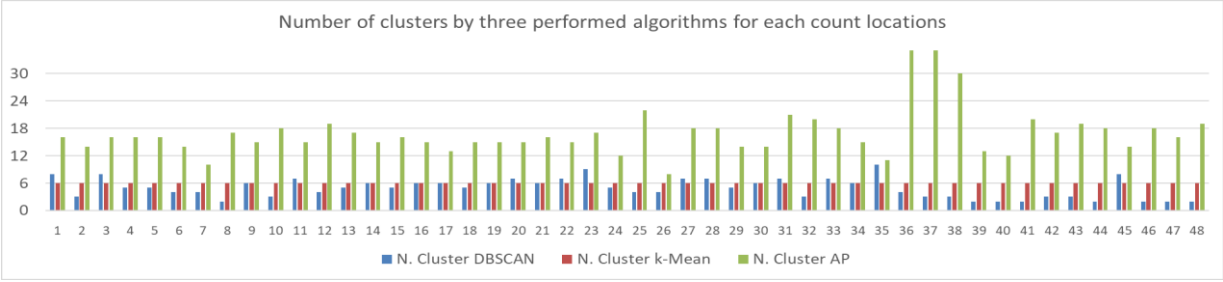

Fig. 6 Number of clusters performed by clustering algorithms.

4. Conclusion

In this paper, we evaluated the capability of three different clustering algorithms, namely K-Means, DBSCAN, and AP, for constructing optimum number of clusters containing identical traffic profiles and clusters with high predictive power of traffic classes. In general, the effectiveness of the employed methodologies in revealing previously unseen similarities in the dataset under a careful cluster's parameter tunning has been confirmed. Among examined methods, the K-Means algorithm has the highest overall accuracy even though AP algorithm performs compatibly with K-Mean, no meaningful relationship could not be found in large number of negligible differences in clusters made by AP. On the other hand, the overall accurateness of the DBSCAN method is not as high as that of K-Means in many count sections, it proves its ability in outlier detection. But the results of the AP algorithm are noticeably different. The usage of clustering algorithms facilitated the identification of six distinct day types within the dataset, school period, non-school period, weekdays, and weekends. The information gathered from the resulting clusters not only aids in projecting future travel demand, but also lends itself to traffic modeling and serves as the foundation for developing traffic management strategies. The suggested method's simplicity and fast calculation time provide useful insights that increase our understanding of network design, allowing us to identify irregularities using a uniform technique that can be applied to a variety of data sets.

References

Jiawei Han M. K, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, An Imprint of Elsevier, 2006.

Daxin Jiang, Chun Tang and Aidong Zhang, "Cluster analysis for gene expression data: a survey," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370-1386, Nov. 2004, doi: 10.1109/TKDE.2004.68.

Chang, Pei-Chann, Chen-Hao Liu, and Chin-Yuan Fan. "Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry." Knowledge-Based Systems 22.5 (2009): 344-355.

Marín Celestino, Ana Elizabeth, et al. "Groundwater quality assessment: An improved approach to K-means clustering, principal component analysis and spatial analysis: A case study." *Water* 10.4 (2018): 437.

Chung, Edward. "Classification of traffic pattern." *Proc. of the 11th World Congress on ITS*. 2003.

Weijermars, Wendy, and Eric Van Berkum. "Analyzing highway flow patterns using cluster analysis." *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*. IEEE, 2005.

Habtemichael, Filmon G., and Mecit Cetin. "Short-term traffic flow rate forecasting based on identifying similar traffic patterns." Transportation research Part C: emerging technologies 66 (2016): 61-78.

Rao, N.K., and G. S. Reddy. "Discovery of Preliminary Centroids Using Improved K-Means Clustering Algorithm", International Journal of Computer Science and Information Technologies, Vol. 3 (3), 2012, 4558-4561.

Steinbach, M., Ertöz, L., Kumar, V. (2004). The Challenges of Clustering High Dimensional Data. In: Wille, L.T. (eds) New Directions in Statistical Physics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-08968-2_16

Erman, Jeffrey, Martin Arlitt, and Anirban Mahanti. "Traffic classification using clustering algorithms." Proceedings of the 2006 SIGCOMM workshop on Mining network data. 2006.

Wang, Yulong, et al. "Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data." ISPRS International Journal of Geo-Information 7.1 (2018): 25.

Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." kdd. Vol. 96. No. 34. 1996.

I. K. Savvas, A. V. Chernov, M. A. Butakova and C. Chaikalis, "Increasing the Quality and Performance of N-Dimensional Point Anomaly Detection in Traffic Using PCA and DBSCAN," 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2018, pp. 1-4, doi: 10.1109/TELFOR.2018.8611947.

paper Brendam J. frey and Delbert Dueck, "Clustering by passing messages between data points", 2017.