Driving Sequence Analysis for Intelligent Driver Assistance: Insights on Clustering Techniques

Gunasekaran Megala, *Member, IAENG,* and Rudhrakoti Venkatesan, *Member, IAENG*

***Abstract*—In the ever-evolving landscape of automotive safety and technology, driver assistance systems are pivotal in improving road safety and enhancing the driving experience. This paper explores the application of clustering techniques to driving sequences to gain insights into driver behavior and optimize driver assistance systems. The research delves into various clustering algorithms, including K-Means, DBSCAN, and hierarchical clustering, and discusses their applicability in identifying patterns within driving sequences. Features such as speed, acceleration, lane changes, and braking patterns are extracted to create a rich dataset for analysis. This study aims to categorize these patterns to facilitate the identification of prevalent errors during the act of driving. Furthermore, the study investigates the potential benefits of temporal clustering to capture dynamic driving behaviors over time. This research provides valuable insights into the clustering of driving sequences, enabling the development of more responsive and context-aware driver assistance systems. By recognizing common driving styles, anomalies, and critical safety scenarios, these systems can better adapt to the unique needs of individual drivers and contribute to overall road safety. The findings presented shows the importance of leveraging clustering techniques as a powerful tool in advancing driver assistance systems, ultimately leading to safer efficient journeys.**

***Index Terms*—Clustering techniques, Driving sequence, Assistance system, Silhouette Coefficient.**

I. Introduction

EACH year, over 1.35 million lives are lost globally due to road traffic deaths, with driver behavior being the primary causative cause in nearly 90% of these incidents. Driving behavior [6] refers to the choices made by drivers when operating automobiles in different driving circumstances. The decision-making behavior demonstrated by drivers is contingent upon a multitude of circumstances, including driving conditions and individual driver characteristics. Hence, drivers exhibit distinct patterns in executing different maneuvers, commonly referred to as driving styles. In the past thirty years, numerous investigations have endeavored to categorize driving styles by the utilization of various methodologies, such as self-reports or the observation of kinematic behaviors.

In the realm of automotive technology, the pursuit of safer and more efficient driving experiences has led to significant advancements in Driver Assistance Systems (DAS). These systems, ranging from adaptive cruise control to lane-keeping assist, are designed to augment the driver's capabilities and improve overall road safety. In recent years, the proliferation of sensors, cameras, and data-driven technologies has presented a unique opportunity to further enhance these systems by gaining deeper insights into driver behavior.

The current body of literature predominantly focuses on driving patterns among individuals operating cars, with a noticeable absence of research pertaining to drivers of heavy passenger vehicles, such as buses. Heavy vehicles for passengers (HVP) refer to buses that are utilized for public or private transportation services and are designed to carry passengers. These vehicles have a gross vehicle weight that exceeds 12,000 kg. HVPs contribute to around 6.6% of road traffic accidents in India, leading to a total of 43,000 individuals sustaining injuries. Furthermore, it is worth noting that non-collision injuries, namely those resulting from occupants stumbling or falling while standing in moving buses, hold considerable significance in terms of both quantity and importance, comparable to injuries sustained in actual wrecks. The abrupt increase or decrease in velocity may lead to pain among individuals who are standing, hence increasing the potential for harm, even in the absence of a vehicular collision. The driving behavior exhibited by bus operators is a crucial determinant that impacts the likelihood of balance loss or injury for passengers who are standing. In the present setting, the primary objective of this study was to examine the acceleration and braking patterns exhibited by drivers of high-performance vehicles and gain insights into the driving behaviors exhibited by individuals about these patterns.

This paper explores the application of clustering techniques [1], [2] to driving sequences as a means of unraveling the intricate tapestry of driver behavior patterns. Understanding these patterns is not only crucial for improving the accuracy and responsiveness of DAS but also for advancing our understanding of the diverse behaviors exhibited by drivers in various contexts. By segmenting driving sequences into meaningful clusters, we can unveil hidden trends, identify common driving styles, and detect potential safety hazards more effectively.

The motivation behind this research stems from the recognition that drivers exhibit a wide spectrum of behaviors, influenced by factors such as road conditions, traffic, weather, and individual preferences. To harness this wealth of information, we delve into a range of clustering algorithms [4], [23], each with its unique strengths and applications. Through feature extraction and dimensionality reduction, we create a comprehensive dataset that encapsulates crucial driving characteristics, such as speed profiles, acceleration patterns, lane-changing behaviors, and braking tendencies.

Furthermore, we explore the temporal aspect of driving behavior by considering the dynamic nature of sequences over time. By incorporating techniques like Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW), we

Manuscript received March 10, 2025; revised August 07, 2025.

G. Megala is an Assistant Professor in School of Computer Science and Engineering, Presidency University, Bangalore, India, 560064 (corresponding author to provide e-mail: megala.g39@gmail.com).

R. Venkatesan is an Assistant Professor in School of Computing, SASTRA University, India (corresponding author to provide e-mail:venkatesaneng@gmail.com).

aim to capture evolving driver behaviors, transitions between driving states, and potential abrupt deviations from typical patterns.

The objectives of this research are twofold: firstly, to shed light on the practical implementation of clustering techniques for driving sequences, and secondly, to highlight the tangible benefits of Driver Assistance Systems. By understanding the nuances of driver behavior at a granular level, DAS can adapt more intelligently to individual drivers, offer personalized recommendations, and respond proactively to critical safety situations.

In the subsequent sections of this paper, we will delve into the methodologies employed, present the results of our clustering analyses, and discuss the implications for the future of driver assistance technology. Ultimately, this research underscores the significance of leveraging clustering techniques [18], [22] as a powerful tool in shaping the future of road safety and driver experience enhancement.

A. Major Contribution

- The use of a Driver Assistance System (DAS) aims to provide support to both truck drivers and traffic managers. Several functionalities are included.
- This work aims to elucidate driver behaviors in alignment with the Hours of Service (HOS) rule.
- The task involves the identification of infringements and the underlying causes that contribute to their occurrence.
- Proposing novel legal modes of transportation to meet delivery demands in accordance with driver preferences.
- The clustering and summarizing of driving sequences enables simplified monitoring of driver behavior.

II. Related works

The classification of driving styles was initially developed using self-report instruments, which involve the use of questionnaires that assess several dimensions of driver behavior [5], [15]. The driver behavior survey, style of driving questionnaire, and multi-dimensional driving pattern inventories are often employed self-report measures designed to categorize drivers or driving styles. Several studies have also examined the correlation between self-reported activities and the likelihood of being involved in a car crash. Nevertheless, the subjective evaluation was found to be susceptible to reporting bias due to the tendency of drivers to gradually forget earlier experiences. Furthermore, persons who had previous knowledge regarding the goal of the survey have a tendency to behave in a manner that aligns with the desired responses of the experiment being conducted. This might potentially introduce bias into the data obtained. Furthermore, it is crucial to note that the self-reports exhibit a deficiency in real-time driver information on performance, hence strengthening the development of continuous monitoring methodologies [9], [19], [24].

Traditional driver assistance systems often rely on rule-based approaches. These systems follow predefined rules and thresholds for actions like collision avoidance, lane-keeping, and adaptive cruise control. While effective, they may lack adaptability to complex real-world scenarios. Machine learning techniques, such as supervised learning and regression, have been used to predict driver behavior based on historical data. These models can provide valuable insights but might struggle with capturing nuanced behaviors. Deep learning models [29], [30], incorporating convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [31], have shown promising results in analyzing driving sequences behaviors. CNNs can process visual data from cameras, while RNNs can model sequential behaviors. Advanced Driver Assistance Systems platforms integrate multiple sensors like radar, lidar, cameras, and ultrasonic sensors [13]. They use sensor fusion techniques to enhance situational awareness and make decisions based on the combined sensor data. In autonomous driving [7] research, behavioral cloning involves training a neural network to mimic the behavior of a human driver using a dataset of human driving actions. This approach enables autonomous vehicles to learn from human drivers.

Reinforcement learning algorithms can optimize driving policies by learning through trial and error. They receive rewards or penalties based on driving actions and adapt their behavior [16], [17] accordingly. Naturalistic Driving Studies states that researchers conduct naturalistic driving studies by equipping vehicles with data-recording instruments and studying real-world driving behaviors. This approach provides valuable insights into how drivers behave in uncontrolled environments.

Fleet Telematics management systems often incorporate driver behavior analysis to improve safety and efficiency. They track parameters like speeding, harsh braking, and sharp turns to encourage safe driving habits. Many modern Sensor Fusion and Perception systems use sensor fusion techniques to combine data from various sensors, such as cameras, radar, and lidar, to create a more comprehensive perception of the driving environment.

Driver Monitoring Systems utilize cameras and sensors to monitor the driver's attention and alertness. They can detect drowsiness, distraction, and fatigue, providing warnings or interventions as needed. These existing approaches span a spectrum from rule-based systems to sophisticated AI and machine learning methods. The choice of approach often depends on the specific goals of the driver assistance system, available sensor data, and computational resources. Researchers and engineers continue to innovate in this field to improve road safety and enhance the driving experience.

The dataset used in this study comprises a collection of acceleration and braking maneuvers, which are defined by a set of multi-dimensional kinematic properties. As evidenced by past research, the process of clustering multivariate data often leads to the formation of groups that are challenging to analyze and assign a distinct driving trend. The clusters are characterized by a composite of all the attributes, with each feature exhibiting a range of values throughout the clusters. Furthermore, the absence of predetermined references or thresholds for feature levels distinguishes driving behaviors [3], [20], [21]. Hence, principal component analysis (PCA) was employed in order to decrease the dataset's dimensionality before clustering, resulting in more easily understandable outcomes. Principal Component Analysis (PCA) was conducted at two distinct stages. a) Initially, the task involves identifying the collection of features that significantly distinguish one cluster from another. Therefore, Principal Component Analysis (PCA) was conducted prior to clustering in order to decrease the dimensionality. Secondly, the driving performance is classified based on the levels of characteristics. This classification is conducted after clustering in order to interpret the results.

Principal Component Analysis (PCA) is a widely used technique in the field of data analysis and machine learning. It serves as a strategy for reducing the dimensionality of a dataset while simultaneously retaining the highest possible amount of variation. The initial characteristics inside the dataset undergo a linear transformation to generate a collection of new variables known as principle components (PCs), which are uncorrelated with each other. Principal Component Analysis (PCA) was conducted on both the datasets pertaining to acceleration and braking. Based on the study conducted by Constantinescu et al. [26], we have selected four principal components (PCs) for each dataset, ensuring a minimum variance of 80%. These selected PCs account for 86% and 85% of the variance in the respective datasets. The relationship between the modified variables or principal components (PCs) and the original characteristics is determined by the PC loading.

III. Methodology

This section outlines the systematic approach adopted to analyze driving behavior using clustering techniques. The objective of this study is to identify distinct patterns of acceleration and deceleration exhibited by drivers of human-powered vehicles (HPVs). The methodology consists of data acquisition, preprocessing, feature extraction, dimensionality reduction, and the application of unsupervised learning algorithms to identify patterns in driving sequences. The weekly driving pattern of HPVs drivers is depicted in Figure 1. Driving behaviors were recorded for a broad group of drivers, and instances of significant acceleration and braking were isolated. Due to the absence of labeled data, unsupervised learning techniques [10] were employed to group similar behavior patterns. Each event was characterized using multiple kinematic features including speed, acceleration/deceleration rate, heading, and duration.

The developed system features a user-friendly graphical interface, accessible via a local web application built using Streamlit. The application continuously receives data from a tachograph and provides real-time metrics such as:

- Current driving sequence compliance with Hours of Service (HOS) regulations
- Detected violations and their possible causes
- Recommended next activities based on driving history
- Remaining allowable driving time before an infringement occurs

Additionally, the Traffic Manager module [12], [25] analyzes historical tachograph data to identify risky or non-compliant driving patterns. The clustering results and their distribution across drivers are further discussed in the results section.

A. Data Collection and Preprocessing

Driving data was collected from tachograph logs and on-board diagnostics (OBD) systems installed in heavy passenger vehicles (HPVs). The dataset includes timestamped records of key kinematic features such as Vehicle speed, Acceleration and deceleration values, Braking instances, Lane changes, Engine status and rest periods. The raw data was cleaned by removing incomplete entries, filtering out noise, and standardizing time intervals. Non-numeric categorical features such as activity labels and day types were encoded using ordinal encoding. Time series sequences were aligned and segmented by driving days to ensure consistency in sequence length and activity context.

TABLE I
Notations Used

Notations	Description
CDD	Continuous Driving day
NDD	Normal Driving Day (≤ 9hr)
EDD	Extended Driving Day (9hr to 10hr)
BT1	Uninterrupted Break Type1 (>45m)
BT2	First Split of Break(¿15m)
BT3	Second split of Break (¿30m)
DR1	Daily Rest
DR2	Reduced Daily Rest [9hr, 11 hr]
DR3	First split of Daily Rest(¿3hr)
DR4	Second split of Daily Rest (¿9hr)
WR1	Normal Weekly Rest (¿45hr)
WR2	Reduced Weekly Rest [24hr, 45hr]

B. Feature Engineering

Each driving sequence was transformed into a numerical representation by computing statistical features such as Mean and variance of speed and acceleration, Number of lane changes, Duration and frequency of rest periods, and Compliance flags with Hours of Service (HOS) rules. Additionally, semantic features such as break types (full break, split break) and legality status (compliant, non-compliant) were included to contextualize driving behavior.

1) Hours of Service (HOS) Compliance: According to Regulation (EC) No 561/2006, a driver may not drive for more than 4.5 hours without taking a break of at least 45 minutes. This break can be split into two parts: a minimum of 15 minutes followed by at least 30 minutes. Following 9 hours of cumulative driving, an uninterrupted rest of 11 hours is mandated. This rest can also be split into two parts: the first for at least 3 hours and the second for at least 9 hours. Drivers may reduce rest periods to 9 hours three times per week, and extend daily driving time to 10 hours twice per week. The regulation imposes the following constraints:

- Weekly driving limit: 56 hours
- Weekly working limit: 60 hours
- Minimum weekly rest: 45 hours (or reduced to 24 hours with compensation in a future week)
- Biweekly driving limit: 90 hours
- Four-month average working time: ≤ 48 hours/week

Figure 2 visualizes different driving and break patterns. Table I outlines the notations used for clustering analysis. Notably, some sequences lacked a second split-break (BT3), which rendered the entire driving sequence invalid due to non-compliance.

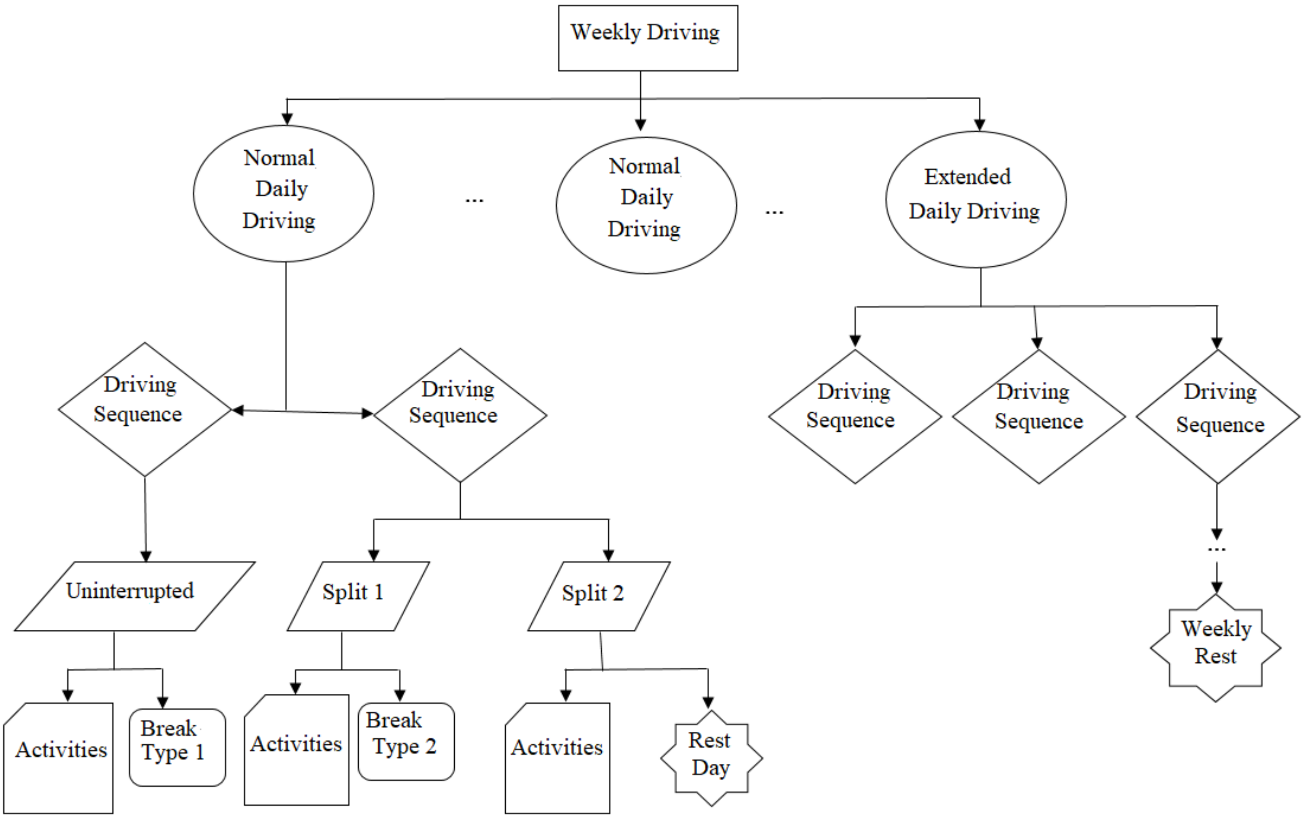


Fig. 1. Weekly Driving Pattern

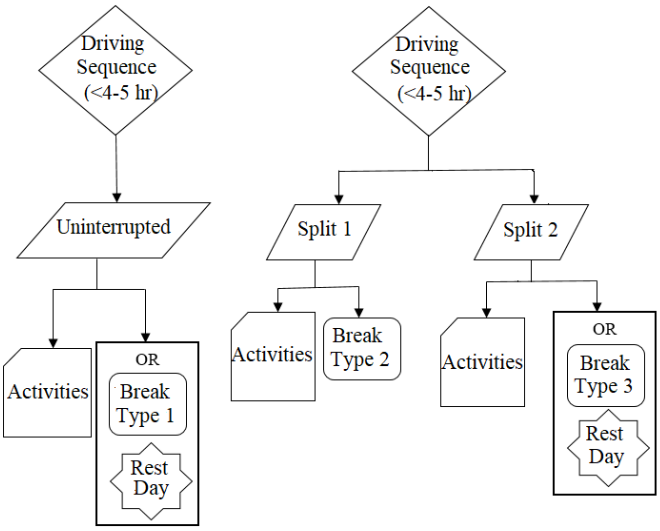


Fig. 2. Driving sequence with different break types

C. Sequence Embedding

Two approaches were used to convert variable-length driving sequences into fixed-length feature vectors of daily driving activity:

- Bag-of-Words (BoW): Applied to encoded categorical sequences for a frequency-based representation.
- Doc2Vec: Captured semantic structure and temporal order in the sequences by training a distributed memory model with vector size = 300 and epochs = 70.

Each sequence includes six attributes: activity, day type, sequence number, break type, token, and legality (e.g., [2, 0, 1, 0, 1]). Duration and normalization were debated for inclusion. Redundancy due to token retention was also assessed. Categorical representations includes,

- Enumerated daily actions
- Single activity per time unit
- Single consolidated sequence per day

The resulting embeddings provided a rich feature space for clustering while preserving the underlying structure of driver behavior.

D. Dimensionality Reduction

To enhance cluster separability and visualization, dimensionality reduction techniques were applied. Principal Component Analysis (PCA) is used to reduce high-dimensional vectors to 2D and 3D for cluster visualization. t-Distributed Stochastic Neighbor Embedding (t-SNE) is used to emphasize local structure and evaluate cluster compactness.

E. Clustering Pipeline

The embedded and normalized feature vectors were input into the following clustering algorithms:

- K-Means: For its efficiency in partitioning data into spherical clusters.
- DBSCAN: To detect arbitrarily shaped clusters and outliers based on density.
- Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN): For uncovering nested behavioral patterns in data with variable density.

Each algorithm was configured with hyperparameters tuned through grid search. For instance, K-Means was iteratively run with different values of k (number of clusters), and DBSCAN parameters eps and minPts were adjusted based on nearest-neighbor distance plots.

F. Evaluation Metrics

Clustering performance was evaluated using three internal validation metrics:

- Silhouette Coefficient (SC): Measures how similar an object is to its own cluster compared to others. Higher values indicate better-defined clusters.
- Calinski-Harabasz Index (CHI): Evaluates between-cluster dispersion versus within-cluster dispersion. Higher scores indicate well-separated clusters.
- Davies-Bouldin Index (DBI): Quantifies average similarity between clusters; lower values imply better clustering

The clustering analysis and corresponding metric evaluations are presented in the subsequent section, allowing a comparative understanding of algorithm effectiveness in modeling driver behavior patterns. The distance function was adapted for categorical variables and modified to consider legality weightings. Redundant features and rarely occurring activity patterns were excluded to optimize clustering relevance. The analysis strategy followed these key steps:

- Activities encoded as ”words” and grouped by day
- Clustering applied using unsupervised methods
- Distinct attention to non-deterministic and deterministic breaks

IV. Implementation

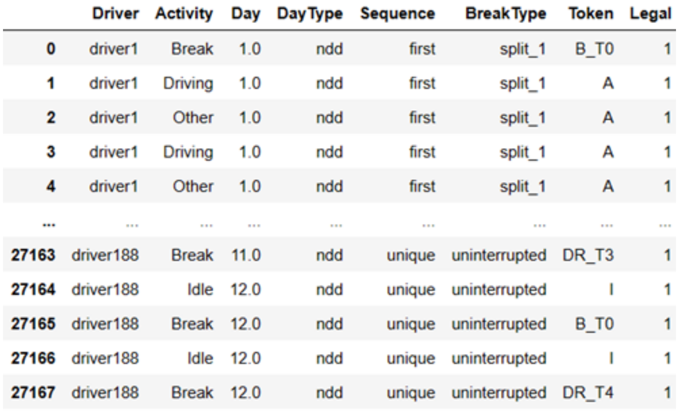
During the pre-processing phase, it is advisable to remove commonly occurring words before transforming the text into either a Bag-of-Words (BoW) or a Document-to-Vector (Doc2Vec) representation. The inclusion of activities that are highly probable and commonly seen across multiple clusters does not provide valuable information in the context of clustering analysis. There should be a distinction between non-deterministic deadlines (NDD) and exact deterministic deadlines (EDD). It is likely that more favourable outcomes can be achieved by directing our attention towards breaks and periods of rest.

Additionally, it is advisable to exclude words that are rarely used, as they are likely referring to suggestions.

- Load and preprocess the CSV log file, converting timestamps and cleaning columns.
- Encode categorical columns (activity, sequence, break type, token, legality) into ordinal values.
- Apply lambda functions to group and format sequences as strings.
- Aggregate sequences by driver and day, forming a list of daily encoded activities.
- Compute unique daily activity sequences (152 found).
- Visualize high-dimensional embeddings using t-SNE, PCA, and 3D PCA

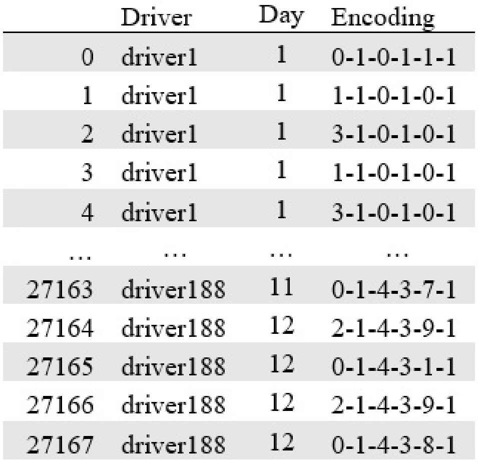
The Terminology used are word and dictionary. Word represents the encoded driving activity. Document represents the full sequence of encoded activities for one day.

Figures 3 and 4 demonstrate the encoded dataset and its graphical representation, illustrating the workflow from raw sequence data to clustered driving patterns.



	Driver	Activity	Day	DayType	Sequence	BreakType	Token	Legal
0	driver1	Break	1.0	ndd	first	split_1	B_T0	1
1	driver1	Driving	1.0	ndd	first	split_1	A	1
2	driver1	Other	1.0	ndd	first	split_1	A	1
3	driver1	Driving	1.0	ndd	first	split_1	A	1
4	driver1	Other	1.0	ndd	first	split_1	A	1
...	...	...	...	...	...	...	...	...
27163	driver188	Break	11.0	ndd	unique	uninterrupted	DR_T3	1
27164	driver188	Idle	12.0	ndd	unique	uninterrupted	I	1
27165	driver188	Break	12.0	ndd	unique	uninterrupted	B_T0	1
27166	driver188	Idle	12.0	ndd	unique	uninterrupted	I	1
27167	driver188	Break	12.0	ndd	unique	uninterrupted	DR_T4	1

Fig. 3. Dataframe of log file



	Driver	Day	Encoding
0	driver1	1	0-1-0-1-1-1
1	driver1	1	1-1-0-1-0-1
2	driver1	1	3-1-0-1-0-1
3	driver1	1	1-1-0-1-0-1
4	driver1	1	3-1-0-1-0-1
…	…	…	…
27163	driver188	11	0-1-4-3-7-1
27164	driver188	12	2-1-4-3-9-1
27165	driver188	12	0-1-4-3-1-1
27166	driver188	12	2-1-4-3-9-1
27167	driver188	12	0-1-4-3-8-1

Fig. 4. Encoding

This enhanced methodology provides a robust, data-driven foundation for analyzing and predicting driver compliance with HOS regulations, improving both road safety and regulatory adherence.

V. Enhanced Clustering Methodology and Evaluation

A. Algorithm for Clustering Prerequisite

The clustering pipeline begins with text preprocessing and vector representation for unsupervised learning. The complete algorithm is as follows:

1) Extract a dictionary from the corpus.
2) Remove words that appear in fewer than 20
3) Convert the filtered corpus into a bag-of-words representation.
4) Apply term frequency-inverse document frequency conversion.
5) Convert the TF-IDF into a dense matrix using corpus2dense function.
6) Tag documents for downstream training.
7) Initialize the Doc2Vec model with a vector size of 300, epoch count of 70, and length equal to the corpus.
8) Train the model, compute similarity scores, and evaluate ranks.
9) Generate and normalize document embeddings using cosine distance.

The document plotting is shown in Fig. 5 for BoW and Doc2Vec models.

B. K-Means Clustering

K-Means assumes convex, equally sized clusters [28]. Initial centroids are randomly selected and iteratively refined using mean distance. Figure 6 demonstrates clusters derived from K-Means. The steps involved in this algorithm are:

- Define a function to tune k.
- Apply clustering.
- Store model and labels.
- Aggregate results for performance scoring.
- Plot silhouette values.
- Construct a 1×2 subplot layout for visualization.
- Store scores in a dataframe indexed by $n - clusters$.

C. Silhouette Coefficient

The Silhouette Coefficient is a metric used to evaluate the quality of clusters in a clustering analysis [27]. It provides a measure of how well-separated the clusters are and helps in determining the optimal number of clusters. The Silhouette Coefficient ranges from -1 to 1, with higher values indicating better-defined and more distinct clusters. The formula $(numberofclusters(n) + 1) \times 10$ is utilized to incorporate empty space among silhouette plots of distinct clusters, hence facilitating obvious demarcation. The silhouette-score provides the mean value across all samples. This provides an understanding of the density and spatial distribution of the clusters that have been generated.

In cases where the ground truth labels are unavailable, the evaluation process necessitates the utilization of the model itself. The Silhouette Coefficient serves as an illustration of such an assessment, wherein a greater Silhouette Coefficient score corresponds to a model that exhibits more distinct clusters. The Silhouette Coefficient(sc) is calculated individually for each sample and consists of two distinct scores (c,n):

$$sc = (n - c)/\max(c, n) \tag{1}$$

Here, the variable c represents the average distance between a given sample and all other points within the same class. The variable ”n” represents the average distance between a given sample and all other points inside the closest neighboring cluster.

1) Silhouette Analysis Algorithm: The Silhouette is calculated and interpreted as follows.

1) Compute the silhouette scores for each sample.
2) The silhouette values for instances assigned to cluster i are aggregated and subsequently sorted.
3) The silhouette plots should be labeled with their respective cluster numbers positioned at the center.
4) Calculate the updated lower bound for the y-axis in the subsequent plot.
5) The vertical line represents the average silhouette score for all the variables.
6) Generate a plot of the silhouette graph for the KMeans algorithm, utilizing the supplied number of clusters (clusters(n)).
7) Construct a subplot with a layout consisting of a single row and 2 columns.
8) The cluster is initialized with a specified value for clusters(n) and a random generator.

The score is constrained within the range of -1 for erroneous clustering and +1 for clustering with high density. Scores in close proximity to zero suggest the presence of clusters that overlap with one another. Higher silhouette scores indicate more distinct, dense clusters. The score exhibits an upward trend in instances when clusters demonstrate high density and clear separation, aligning with the conventional notion of a cluster. Convex clusters tend to have a greater Silhouette Coefficient compared to alternative cluster concepts, such as density-based clusters formed by DBSCAN.

D. Calinski-Harabasz Index

The Calinski-Harabasz Index (CHI), also known as the Variance Ratio Criterion (VRC), is a metric used to evaluate the quality of clusters in a clustering analysis. It measures the ratio of between-cluster variance to within-cluster variance and can help in determining the optimal number of clusters. The higher the Calinski-Harabasz Index, the better the clustering result. A higher Calinski-Harabasz score is indicative of a model that exhibits more distinct and well-defined clusters.

The index can be defined as the ratio between the sum of inter-clusters dispersion and the sum of within-cluster dispersion for all clusters. In this context, dispersion refers to the sum of squared distances. The steps to calculate CHI are as follows:

- Calculate the total variance of the dataset, which is the sum of squared distances between all data points and the dataset’s mean.
- Calculate the between-cluster variance, which is the sum of squared distances between cluster centroids and the dataset’s mean, weighted by the number of data points in each cluster.
- Calculate the within-cluster variance, which is the sum of squared distances between data points and their respective cluster centroids.
- Calculate the Calinski-Harabasz Index (CHI) using the formula:

  $$CHI = [\frac{variationbetweencluster}{variationwithincluster}] \times [\frac{N - K}{K - 1}] \tag{2}$$

  Where, N is the total number of data points and K is number of clusters.

1) Merits: The score exhibits an increase when clusters demonstrate high density and clear separation, aligning with the conventional notion of a cluster. The computation of the score is rapid.

2) Demerits: Convex clusters tend to yield larger values of the Calinski-Harabasz index compared to alternative cluster concepts, such as density-based clusters exemplified by those generated by the DBSCAN algorithm.

E. Davies-Bouldin Index

In cases where the ground truth (true) labels of the data are unknown, the Davies-Bouldin index can serve as a suitable metric for assessing the performance of a model. A lower value of the Davies-Bouldin index indicates a higher

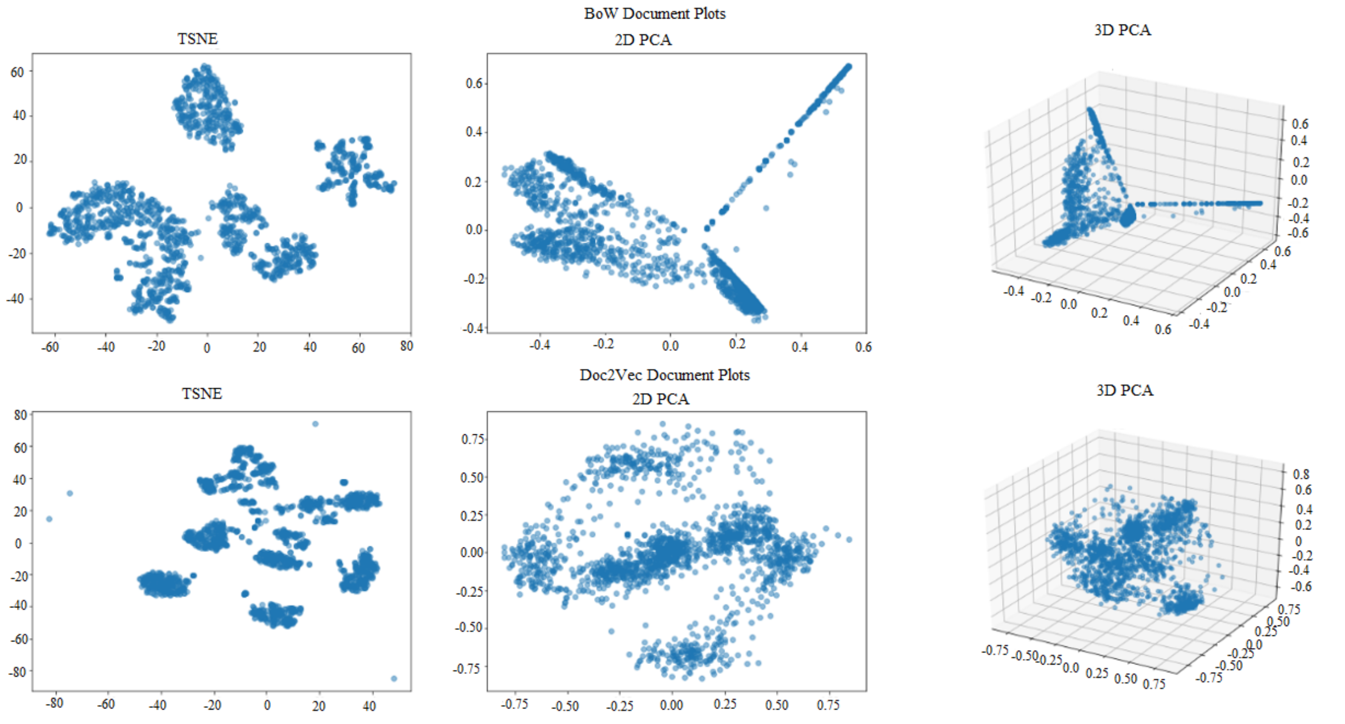


Fig. 5. BoW and Doc2Vec document plot

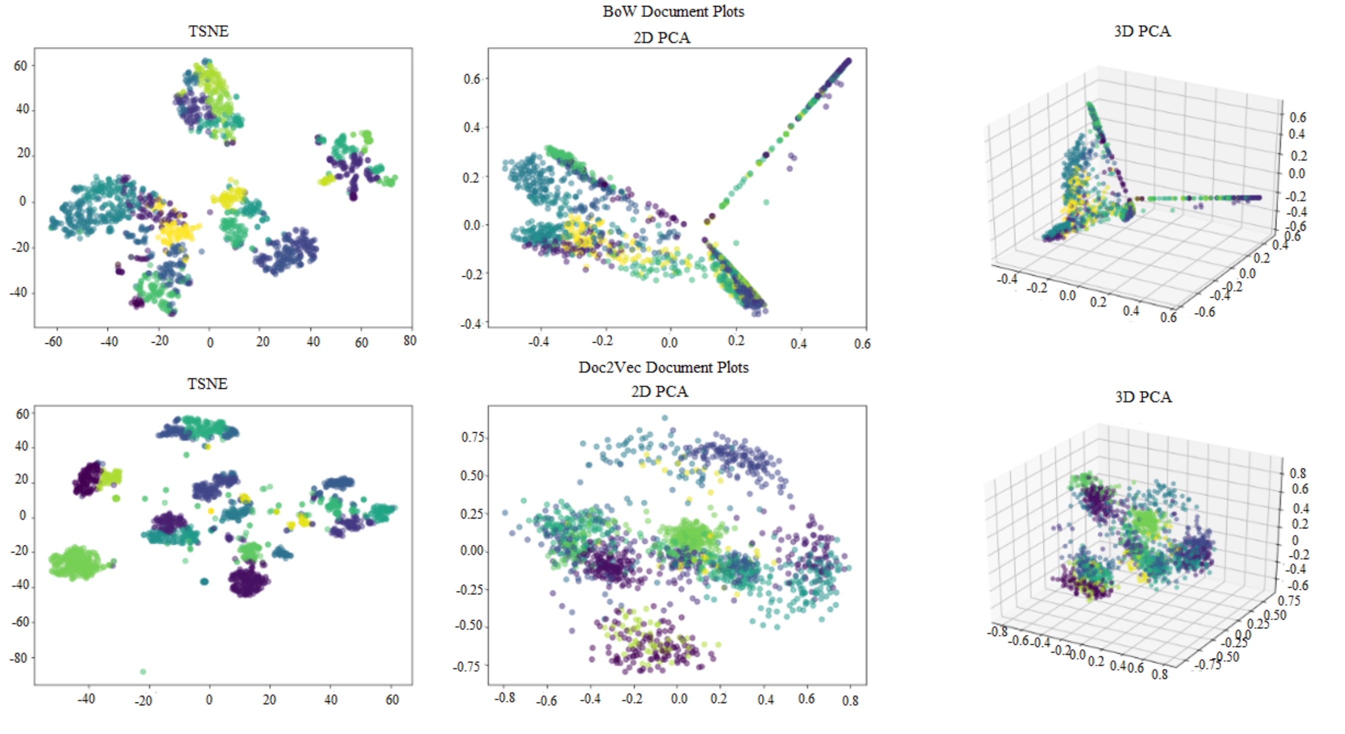


Fig. 6. K-Means Clustering

degree of separation between the clusters, hence indicating a superior model.

The index represents the mean ”similarity” among clusters, where similarity is a metric that evaluates the distance between clusters relative to their respective sizes (Glassen et al.,2018). The numerical value of zero represents the minimum attainable score. Partitions with values closer to 0 are indicative of higher quality.

1) Merits:

- The calculation of the Davies-Bouldin index is less complex in comparison to that of the Silhouette scores.
- The calculation of the index relies exclusively on the numbers and characteristics intrinsic to the dataset, as it is computed solely using point-wise distances.

2) Demerits:

- Convex clusters tend to exhibit greater values on the Davies-Boulding index compared to alternative cluster concepts, such as density-based clusters derived using DBSCAN.
- The utilization of centroid distance restricts the applica-

TABLE II
SILHOUETTE ANALYSIS

No of clusters (n)	Silhouette score	Calinski-Harabasz score	Davies-Bouldin Score
20	0.1419	85.591	2.212
21	0.1458	85.247	2.051
22	0.1448	83.042	2.014
23	0.1384	81.349	2.028
24	0.1422	78.985	1.974
25	0.1520	77.363	1.986
26	0.1478	76.254	1.983
27	0.1539	75.392	1.959
28	0.1481	74.231	1.941
29	0.1479	72.840	1.967

TABLE III
PERFORMANCE OF DBSCAN CLUSTERING

EPS	No of clusters (n)	Silhouette score	Calinski-Harabasz score	Davies-Bouldin Score
0.1	1	0	0	0
0.2	4	-0.1061	21.4831	1.6314
0.3	3	-0.0795	59.9756	1.9035
0.4	5	-0.0413	64.8754	1.9688
0.5	5	0.0201	72.9785	2.6031
0.6	3	0.0910	95.2415	3.2289
0.7	2	0.1571	83.9764	3.6987
0.8	2	0.1678	44.6218	3.2197
0.9	2	0.2269	22.0147	1.7958
1.0	2	0.3124	3.2031	0.5498

TABLE IV
CLUSTERING PERFORMANCE RESULTS

Clustering Method	Performance Metric	Result
K-Means	Silhouette (BoW)	0.152
	Silhouette (Doc2Vec)	0.349
	Adjusted Rand Index	0.4299
	Adjusted Mutual Info	0.6462
DBSCAN	Silhouette (BoW)	0.099
	Cluster Labels	[-1,0...6]
	Silhouette (Doc2Vec)	0.148
	Cluster Labels	[-1,0]
HDBSCAN	Silhouette (Doc2Vec)	-0.036
	Cluster Labels	[0...43]

bility of the distance metric solely to Euclidean space. Table II illustrates the score analysis. Silhouette and Davies scores start to increase at 24 but past 30 could be an excessive quantity. Fig.7 and Fig.8 illustrate the clusters formed using K-Means clustering.

F. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

The DBSCAN is a density-based clustering algorithm used for grouping data points into clusters. A data point is considered as a core point if it has atleast minimum data points (minpts) within a distance of epsilon (EPS) from itself. A data point can be classified as a border point if it satisfies two conditions: first, it must be located within a specified distance, denoted as EPS, from a core point; second, it must not have enough neighboring points to qualify as a core point.

Noise points refer to data points that do not fall within the categories of core points or border points. DBSCAN selects an arbitrary unvisited data point and if it is core point, it forms a new cluster. It then identifies all data points in the EPS neighborhood of the core point and adds them to the cluster. This process is continued recursively. It considers clusters as regions characterized by a significant concentration of entities, delineated by regions with comparatively lower entity density. As a consequence of this broad perspective, the clusters identified by DBSCAN exhibit the potential to adopt many shapes, in contrast to k-means clustering which presupposes that clusters possess a convex shape. The performance analysis is shown in Table III.

G. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

HDBSCAN is an extension of traditional DBSCAN offers advantages in handling varying cluster desnsities and providing a comprehensive hierarchical cluster structure. It groups the data points together based on proximity in feature space , regions of high data point density separated by areas of lower density. It determines the appropriate density threshold based on the data. Here the minimum cluster size chosen is 5. It is often more stable and robust than DBSCAN.

The hierarchical structure of clusters is visualized in Fig. The leaf size is set as 40, minimum sample size is 1 and Euclidean distance metric is chosen to perform HDBSCAN clustering. The hierarchical structure of HDBSCAN is illustrated in Fig.9. Fig.10 illustrates the clusters visualized.

VI. DISCUSSION

The clustering performance results are shown in Table IV. It is plausible for the value of K to exceed 20 while maintaining coherence. It is crucial to acknowledge that while there are only a limited number of non-delivery days (NDD), estimated delivery days (EDD), or no specific delivery days (NONE), there exists a significant distinction in the concluding letters, namely those ending in WR and DR. The findings obtained from the Doc2Vec model indicate that using a vector-size smaller or larger than 200 leads to suboptimal performance. However, after applying Principal Component Analysis (PCA), the data appears to be better separated when using a vector space of size 200. It is worth noting that this improvement is observed only after normalizing the data, but the reason behind this phenomenon is not yet clear. The duration of time from a specific starting point to the age of 70.

Achieving improved outcomes in KMeans clustering can be accomplished by utilizing the cosine distance metric in conjunction with normalization techniques. The Latent Dirichlet Allocation (LDA) algorithm also demonstrates promising outcomes; however, the interpretation of topics is more challenging due to the absence of ordering and the probabilistic nature of the results.

This comprehensive evaluation and visualization demonstrate the efficacy and trade-offs of multiple clustering ap-

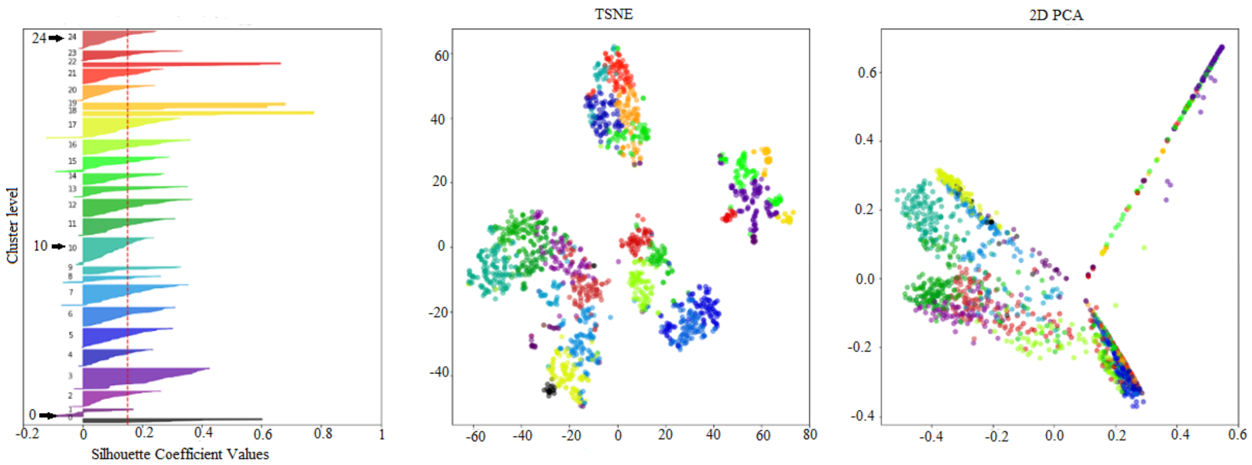


Fig. 7. Silhouette analysis for K Means clustering with 25 clusters

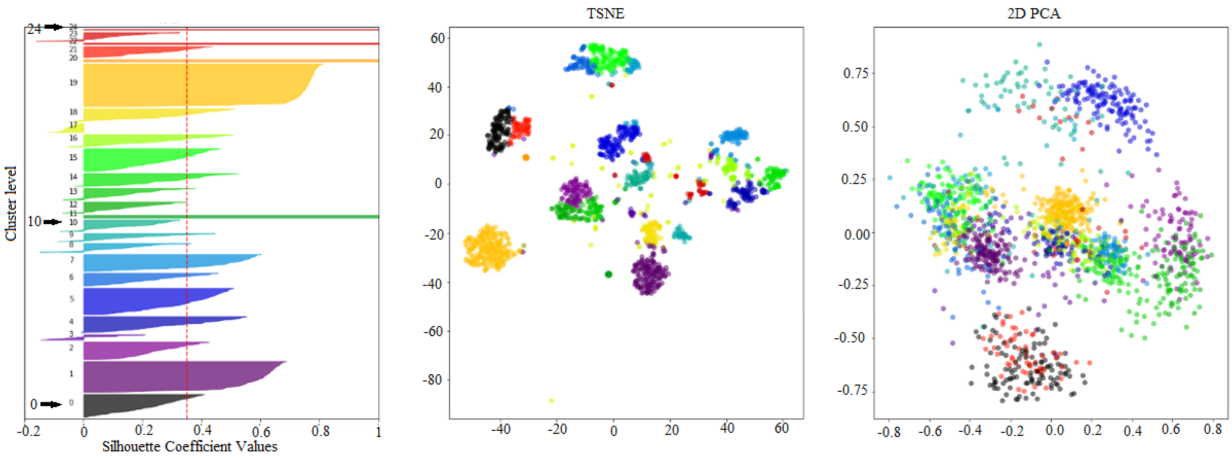


Fig. 8. After tuning KMeans clustering

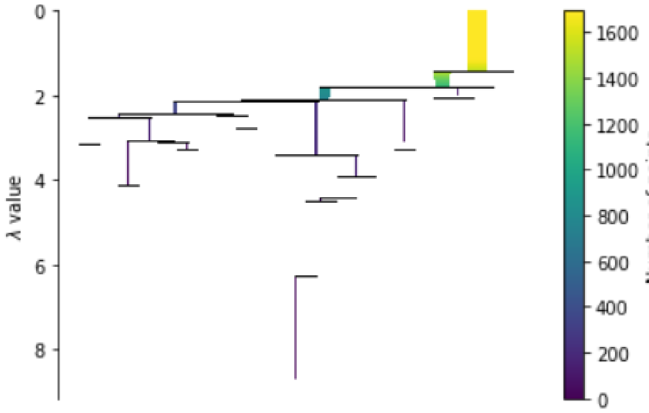


Fig. 9. Hierarchical structure of HDBSCAN Clustering

proaches, aiding in the selection of suitable models for text-based activity pattern recognition.

VII. Conclusion

To minimize the development effort of advanced driver assistance systems, it is necessary to have a comprehensive set of test scenarios that effectively cover a wide range of scenarios in the database. This paper presents a comprehensive discussion of multiple clustering algorithms. DBSACAN and HDBSCAN can be powerful tools for identifying complex driving behavior patterns on considering varying densities of sequences. It provides a more adaptive and detailed clustering structure which can be advantageous for understanding and responding to different driving scenarios. The average silhouette width was employed as a metric for cluster validation. The average silhouette width, at which the ideal number of clusters was computed, does not exhibit exceptional values when compared to alternative cluster sizes. Hence, the study focused on examining the various cluster sizes. It has been observed that each data point belonging to a cluster without any additional data points has a negative impact on the average silhouette width. When considering scenario clustering, it is recommended to revise the calculation technique for the average silhouette width. Currently, this value is determined by taking the mean of all silhouette values in the dataset. However, it is important to exclude outliers, specifically rare scenarios, from being merged with the nearby cluster.

References

[1] Aghabozorgi, S., Shirkhorshidi, A. & Wah, T., ”Time-series clustering–a decade review,” *Information Systems*, vol. 53, no. 1, pp. 16-38. 2015.

[2] Ruiz, A., Flynn, M., Large, J., Middlehurst, M. & Bagnall, A., ”The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances,” *Data Mining And Knowledge Discovery*, vol. 35, no. 2, pp. 401-449, 2021.

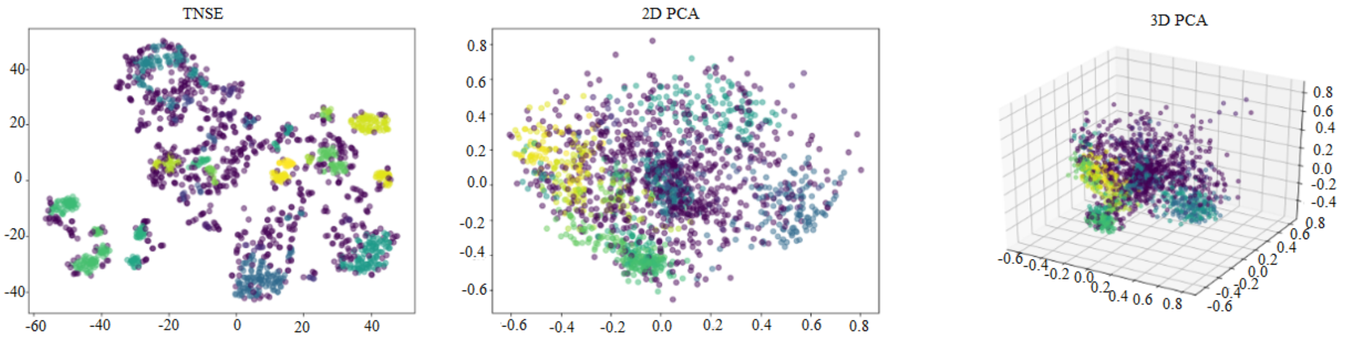


Fig. 10. Comparative HDBSCAN Clustering

[3] Bender, A., Agamennoni, G., Ward, J., Worrall, S. & Nebot, E., An unsupervised approach for inferring driver behavior from naturalistic driving data, *IEEE Transactions On Intelligent Transportation Systems*, vol. 16, no.6, pp. 3325-3336, 2015.

[4] Besse, P., Guillouet, B., Loubes, J. & Royer, F., "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Transactions On Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3306-3317, 2016.

[5] Eftekhari, H. & Ghatee, M., "Hybrid of discrete wavelet transform and adaptive neuro fuzzy inference system for overall driving behavior recognition," *Transportation Research Part F: Traffic Psychology And Behaviour*, vol. 58, no. 1, pp. 782-796, 2018.

[6] Fugiglando, U., Massaro, E., Santi, P., Milardo, S., Abida, K., Stahlmann, R., Netter, F. & Ratti, C., "Driving behavior analysis through CAN bus data in an uncontrolled environment," *IEEE Transactions On Intelligent Transportation Systems*, vol. 20, no. 2, pp. 737-748, 2018.

[7] Funke, J., Brown, M., Erlien, S. & Gerdes, J., "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Transactions On Control Systems Technology*, vol. 25, no. 4, pp. 1204-1216, 2016.

[8] Glassen, T., Oertzen, T. & Konovalov, D., "Finding the mean in a partition distribution," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1-10, 2018.

[9] He, Y., Ciuffo, B., Zhou, Q., Makridis, M., Mattas, K., Li, J., Li, Z., Yan, F. & Xu, H., "Adaptive cruise control strategies implemented on experimental vehicles: A review," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 21-27, 2019.

[10] Mantouka, E., Barmpounakis, E. & Vlahogianni, E., "Identifying driving safety profiles from smartphone data using unsupervised learning," *Safety Science*, vol. 119, no. 1, pp. 84-90, 2019.

[11] Martinussen, L., Møller, M., Prato, C. & Haustein, S., "How indicative is a self-reported driving behaviour profile of police registered traffic law offences?," *Accident Analysis & Prevention*, vol. 99, no. 1, pp. 1-5, 2017.

[12] Ryder, B., Gahr, B., Egolf, P., Dahlinger, A. & Wortmann, F., "Preventing traffic accidents with in-vehicle decision support systems-The impact of accident hotspot warnings on driver behaviour," *Decision Support Systems*, vol. 99 no. 1, pp. 64-74, 2017.

[13] Sattar, S., Li, S. & Chapman, M., "Road surface monitoring using smartphone sensors: A review," *Sensors*, vol. 18, no. 11, pp. 3845-3855, 2018.

[14] Tselentis, D., Vlahogianni, E. & Yannis, G., "Temporal analysis of driving efficiency using smartphone data," *Accident Analysis & Prevention*, vol. 154, no. 1, pp. 1-12, 2021.

[15] Vaiana, R., Iuele, T., Astarita, V., Caruso, M., Tassitani, A., Zaffino, C. & Giofrè, V., "Driving behavior and traffic safety: an acceleration-based safety evaluation procedure for smartphones, *Modern Applied Science*, vol. 8, no. 1, pp. 88-98, 2014.

[16] Wang, W., Han, W., Na, X., Gong, J. & Xi, J., "A probabilistic approach to measuring driving behavior similarity with driving primitives," *IEEE Transactions On Intelligent Vehicles*, vol. 5, no. 1, pp. 127-138, 2019.

[17] Wang, X. & Xu, X., "Assessing the relationship between self-reported driving behaviors and driver risk using a naturalistic driving study," *Accident Analysis & Prevention*, vol. 128, no. 1, pp. 8-16, 2019.

[18] Wang, W., Ramesh, A., Zhu, J., Li, J. & Zhao, D., "Clustering of driving encounter scenarios using connected vehicle trajectories," *IEEE Transactions On Intelligent Vehicles*, vol. 5, no. 3, pp. 485-496, 2020.

[19] Wangdi, C., Gurung, M., Duba, T., Wilkinson, E., Tun, Z. & Tripathy, J., "Burden, pattern and causes of road traffic accidents in Bhutan, 2013–2014: a police record review," *International Journal Of Injury Control And Safety Promotion*, vol. 25, no. 1, pp. 65-69, 2018.

[20] Warren, J., Lipkowitz, J. & Sokolov, V., "Clusters of driving behavior from observational smartphone data," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 3, pp. 171-180, 2019.

[21] Yao, Y., Zhao, X., Wu, Y., Zhang, Y. & Rong, J., "Clustering driver behavior using dynamic time warping and hidden Markov model," *Journal Of Intelligent Transportation Systems*, vol. 25, no. 3, pp. 249-262, 2021.

[22] Yi, D., Su, J., Liu, C. & Chen, W., "Trajectory clustering aided personalized driver intention prediction for intelligent vehicles," *IEEE Transactions On Industrial Informatics*, vol. 15, no. 6, pp. 3693-3702, 2018.

[23] Zhong, Z., Lee, E., Nejad, M. & Lee, J., "Influence of CAV clustering strategies on mixed traffic flow characteristics: An analysis of vehicle trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 115, no. 1, pp. 1-13, 2020.

[24] Zhou, J., Gao, D. & Zhang, D., "Moving vehicle detection for automatic traffic monitoring," *IEEE Transactions On Vehicular Technology*, vol. 56, no. 1, pp. 51- 59, 2007.

[25] Zhu, L., Yu, F., Wang, Y., Ning, B. & Tang, T., "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions On Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383-398, 2018.

[26] Constantinescu, Z., Marinoiu, C. & Vladoiu, M., "Driving style analysis using data mining techniques," *International Journal Of Computers Communications & Control*, vol. 5, no. 5, pp. 654-663, 2010.

[27] Zepeda, M., Meng, F., Su, J., Zeng, X. & Wang, Q., "Dynamic clustering analysis for driving styles identification," *Engineering Applications Of Artificial Intelligence*, vol. 97, no. 1, pp. 1-13, 2021.

[28] Förster, D., Inderka, R. & Gauterin, F., "Data-driven identification of characteristic real-driving cycles based on k-means clustering and mixed-integer optimization," *IEEE Transactions On Vehicular Technology*, vol. 69, no. 3, pp. 2398-2410,2019.

[29] Yang, N. & Zhao, J., "Dangerous driving behavior recognition based on improved YoloV5 and Openpose," *IAENG International Journal Of Computer Science*, vol. 49, no. 4, pp. 1112-1122, 2022.

[30] Sun, J. & Wang, Z., "Vehicle and pedestrian detection algorithm based on improved YOLOv5," *IAENG International Journal Of Computer Science*, vol. 50, no. 4, pp. 1401-1409, 2023.

[31] Wang, Y., Cheng, X. & Meng, X., "Sentiment analysis with an integrated model of BERT and bi-LSTM based on multi-head attention mechanism," *IAENG International Journal Of Computer Science*, vol. 50, no. 1, pp. 255-262, 2023.

Dr. Gunasekaran Megala (M'2022) received Ph.D. degree in Computer Science and Engineering from VIT University: Vellore Institute of Technology, Vellore. She has more than 20 publications in national and international journals and conferences. She has more than 11 years of experience in teaching and 5 years of experience in research. Her current research includes multimedia security, lightweight cryptography, image and video processing, and deep learning. Currently, she is an Assistant Professor in the School of Computer Science and Engineering, Presidency University, Bangalore.

Dr. Rudhrakoti Venkatesan received Ph.D. degree in Computer Science and Engineering from VIT University : Vellore Institute of Technology, Vellore. He has more than 30 publications in national and international journals and conferences. He has more than 16 years of experience in teaching and 9 years of experience in research. He is a Member of IAENG. His current research includes satellite image processing and neural networks. Currently, he is an Assistant Professor in the School of Computing, SASTRA University, Thanjavur.