Segmentation of Bus Driving Data: A Clustering-based Approach to Identify Similar Driving Sections

Anna Schniertshauer[1*], Sven Angerer[2], Andreas Grabow[1], Michael Schlick[1]

[1]Institute for Automotive Systems Engineering, Technische Hochschule Ulm, Prittwitzstraße 10, 89075, Ulm; [*]*anna.schniertshauer@thu.de*

[2] Ulm University, Helmholtzstraße 16, 89081 Ulm, Germany

SNE 35(4), 2025, 203-210, DOI: 10.11128/sne.35.tn.10756
Selected ASIM WS2025 Postconf. Publication: 2025-09-10
Rec. Revised: 2025-11-09; Accepted: 2025-11-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Driving cycles are required for a variety of applications including longitudinal dynamics simulations. For the generation of representative driving cycles, a driving data analysis is indispensable. This paper proposes a method to efficiently segmenting data and subsequently identifying typical trip sections. A first cluster analysis is performed on individual data points using the kmeans++ algorithm. Based on the results, the consecutive data points are segmented into microsegments. Subsequently, these microsegments are being clustered in a second cluster analysis. The results obtained reveal patterns of cluster formations that are similar to those observed in the cluster analysis of individual data points.

Another segmentation, based on the minimum duration of standstill times between two driving sections, enables the identification of typical trips of longer durations. This is achieved by taking the proportions of the microsegments assigned to the same cluster as input variables for the third cluster analysis. Thereby, groups of similar trips can be identified with the typical distribution of microsegment proportions.

Thus, the developed method yields representative trip sections for a driving dataset and thereby forms a basis for generating representative driving cycles both in the research and in the development of simulation-based technologies.

Introduction

To reduce greenhouse gas emissions related to public transit vehicles, such as buses, the transition from conventional internal combustion drive trains to alternative drive trains is necessary. When developing new drive trains, longitudinal dynamics simulations play an important role. In order to fully exploit the capabilities of these simulations, it is essential to use representative driving cycles for different application scenarios as stimuli. This facilitates the evaluation of the vehicle's ability to meet not only the requirements of standardized driving cycles for buses, such as the Standardized On-Road Tests (SORT), but also the requirements of real customer use. To obtain these representative driving cycles, it is necessary to identify typical operating trips by analyzing driving data from buses already operating in the application area. To this end, this paper proposes an approach to effectively segmenting the dataset and finding groups of similar trips using a clustering algorithm. This approach provides a basis for generating representative driving cycles, which can subsequently be used in longitudinal dynamics simulations and support the development of new drive trains.

There are several approaches for analyzing driving data, whereby the dataset is segmented into brief driving sections, hereby referred to as microsegments. One of these approaches involves the establishment of a fixed duration for the segments. Montazeri-Gh et al. use a duration of 150 s [1] whereas Brady and O'Mahony use a duration of 30 s [2]. The disadvantage of a fixed duration is that a single microsegment contains several different driving scenarios. Therefore, in this work, the driving data is segmented into microsegments of variable length which start or end when the driving conditions change.

A similar approach is employed by Langner et al. to extract representative driving scenarios from real-world driving data [3]. The segmentation is based on significant changes in features such as speed limits, street types and curviness [3]. However, there is no detailed description of how these significant changes in the features are identified.

The approach in this paper is to realize the segmentation by clustering individual data points and creating microsegments from consecutive data points that are in the same cluster. Chetouane et al. also perform a cluster analysis of individual data points to identify similar driving episodes from an autonomous driving dataset [4]. However, the post-processing differs from the one chosen in this paper and the identified driving episodes are not used for further characterization of longer driving segments [4].

The methodology employed for the segmentation of driving data and the identification of groups of similar trips is examined first. Then a description of the dataset utilized to develop the method follows. The implementation of the approach is divided into two steps. Initially, data points are segmented into microsegments, which are then clustered. Subsequently, trips are segmented based on a minimum standstill time, followed by a cluster analysis. Finally, a discussion of the results precedes the conclusion of this work.

1 Methodology

With the aim of identifying similar trips from a driving dataset, the method shown in Figure 1 is developed. The raw dataset undergoes a series of preprocessing steps to obtain the desired data quality prior to segmentation. Two segmentation levels are employed for this purpose. On the first level, the data is segmented into microsegments, which are characterized by similar driving conditions. A new segment is created when the considered variables change significantly. To achieve this segmentation, all data points are categorized with a first cluster analysis. For all cluster analyses in this paper the kmeans++ algorithm by Arthur and Vassilvitskii is used [5]. It is an augmentation of the well-known cluster algorithm kmeans, which was developed by Lloyd [6].

Kmeans is a partitioning cluster algorithm and is commonly used to cluster driving data [1, 2, 4, 7]. The algorithm takes the number of clusters k as an input and chooses k cluster centers randomly from the dataset. In the next step the Euclidean distances from each point to the cluster centers are calculated and the data points are assigned to the cluster center with the minimal distance to the point. New cluster centers are calculated as the average of all assigned points and the procedure is repeated. Kmeans++ chooses the initial cluster centers based on the distribution of the input data, which leads to better results and a faster convergence [5]. To choose the number of clusters k, three metrics are considered. The Silhouette score compares the difference of the average distances between a given point and the points within the same cluster and the average distance between that point and the points in the nearest cluster with the maximum of these two values [8]. The values are in a range of -1 and 1, where a higher value indicates a better cluster result [8]. The Davies-Bouldin index compares the size of the clusters with the distance between them. Better partition and separation are achieved with a lower Davies-Bouldin index [9]. The third index used is the Calinski-Harabasz index, which validates the cluster validity by calculating the proportion obtained by dividing the total dispersion between clusters by the total dispersion within clusters across all clusters [10]. A higher index indicates better results [10]. All three metrics are also used by Chetouane and Wotawa to evaluate the results of clustering driving data [11].

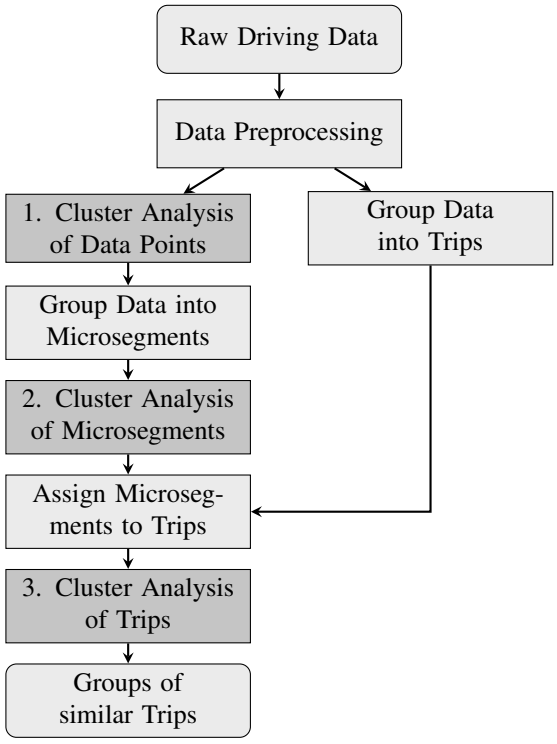


Figure 1: Segmentation and clustering process.

The first cluster analysis assigns to each data point a cluster index, ci_p. It is imperative to ensure that the occurrence of single data points, possessing a different cluster index compared to their surrounding data points that share the same cluster index, does not result in the initiation of a new microsegment. To this end, a smoothing method is developed. Furthermore, driving sections are identified that consist of data points with highly divergent cluster indices. They are grouped as self-contained microsegments. The values of the individual data points are aggregated so that each microsegment has one value for each cluster variable. The start and end times of each microsegment are taken from the first and the last data point of this microsegment. Subsequently, a second cluster analysis is performed on the aggregated microsegments, which assigns to each microsegment a cluster index ci_m.

On the second level of segmentation, the preprocessed dataset is segmented into trips based on a defined threshold for standstill time. If the vehicle stands still for a duration exceeding the threshold, the current trip ends and a new trip starts as soon as the vehicle moves off again. This results in larger segments than those extracted from the individual data point clustering. Therefore, each trip consists of several microsegments. The start and end time of each trip and microsegment facilitates the temporal assignment of the microsegments to the trips. For each trip, the proportions of time spent by the vehicle in a microsegment of the clusters ci_m are calculated and taken as new variables for the third cluster analysis at the trip level.

The quantity of cluster variables at this level is defined by the cluster number that is derived from the second cluster analysis conducted at the microsegment level. The analysis yields groups of trips with similar proportions of microsegments assigned to the same cluster index ci_m.

2 Data Source and Preparation

The data source used for this paper is the Zurich Transit Bus dataset [12]. It contains driving data from two electric city buses over a period of 3.5 years. To prepare the data for the cluster analysis and to ensure sufficient data quality, a variety of filtering and data preprocessing steps are carried out preceding the clustering. The signals used from the dataset are the time of recording, the latitude and longitude of the vehicle and the velocity of the vehicle.

To filter out incomplete data, the dataset is split into individual days. A day is only included in the analysis if the positional information *or* the velocity signal is not missing for more than 60 s. If the positional information *and* the velocity signal is missing, the day is included since this corresponds to a break. This results in a dataset of 7.5 million data points with a sampling rate of less than 10 s for 99.2% of the data points and a sampling rate of exactly one second for 93.1% of the data points. If the positional information is missing, it is interpolated linearly.

As described by Widmer et al., the dataset contains the raw measurements of the sensors [12]. To counteract on inaccuracies of the latitude and longitude signal introduced by the GNSS sensor, the Valhalla Map Matching API [13] is used to map the coordinates onto the road.

The altitude information for the matched coordinates is retrieved from NASA's publicly available SRTM data [14]. To smooth the altitude signal, the average altitude of the 15 seconds preceding and succeeding each data point is calculated and used for further processing. The velocity of the vehicle is calculated by Widmer et al. based on the signal of rotational speed sensors mounted on the motor shafts of the vehicle and then multiplied with a transmission ratio γ [12]. To counteract slight deviations introduced through this estimation, all values with negative velocity values are set to zero.

To calculate the slope, the difference in altitude is divided by the distance traveled between two data points. For the slope of one data point, the average slope of the preceding and succeeding data point is used. To avoid dividing by a small value of the traveled distance, the slope is set to zero if the velocity of a data point or the preceding or succeeding data point is less than 0.2 $\frac{m}{s}$.

3 Segmentation and Clustering of Microsegments

In order to identify similar microsegments, individual data points are clustered. For this, the slope and velocity of the vehicle are used, as these are typical input variables for longitudinal dynamics simulations. To filter out any outliers, the interpercentile range (ipr) between the 5 and 95 percentile is calculated. Values smaller than $p_5 - 1.5 \cdot ipr$ or larger than $p_{95} + 1.5 \cdot ipr$ are neglected. This filters out 0.2% of the values corresponding to 14 385 data points.

The data is then scaled using a Standard Scaler to remove the mean and scaling it to unit variance [15]. Each data point is clustered using the kmeans implementation by Scikit learn [15] with the kmeans++ algorithm for choosing the initial clusters [5]. This results in each data point being assigned to one of k clusters.

To determine the number of clusters, the scores in Table 1 are considered. As the Silhouette score and Calinski-Harabasz index indicate, the best results are achieved with $k = 4$. The cluster index a data point is assigned to is called ci_p. As can be seen in Figure 2, the clusters with $ci_p = 2$ and $ci_p = 4$ contain data points with moderate slope but differ in velocity. The clusters with $ci_p = 1$ and $ci_p = 3$ contain data points with moderate to high absolute slope values regardless of the velocity.

k	Silhouette	Davies–Bouldin	Calinski–Harabasz
2	0.370	1.099	398668
3	0.374	0.929	435960
4	0.405	0.876	470390
5	0.356	0.885	459099
6	0.360	0.866	447477

Table 1: Scores to determine the number of clusters.

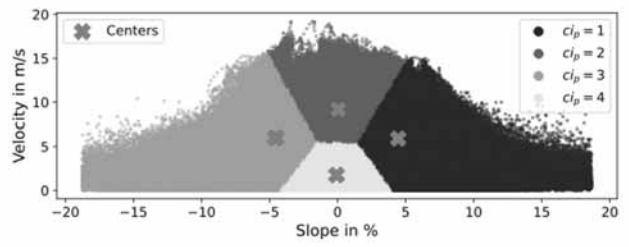


Figure 2: Clustering result of individual data points

In the next step, consecutive data points with the same cluster indices ci_p are grouped together into microsegments.

As can be seen in Figure 3(a), there are data points with a different cluster index than the cluster index that is dominating in this driving section. To address this issue a smoothing process is introduced.

For this, each data point is first transformed into a binary vector representation v, where only the value at the index of the cluster (ci_p) is set to one.

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_k \end{bmatrix} \qquad v_i = \begin{cases} 1 & \text{if } i = ci_p, \\ 0 & \text{otherwise.} \end{cases}$$

Since each data point can only be assigned to one cluster, each vector always only has one entry not equal to zero, which means that all cluster indices have the same Euclidean distance to each other and the smoothing can be performed. The average is calculated rowwise in a rolling window. The row with the highest average is then set to one and all other values are set to zero. In a last step the vector can be transformed back into the cluster assignment. An example of this process is demonstrated in Table 2 with a fixed window size of 3 data points. The actual implementation of the algorithm uses a window size of 9 s centered around the current data point. Since the majority of data points have a sampling rate of 1 s, this corresponds to a window size of 9 data points. As shown in Figure 3(b), this method successfully assigns the dominating cluster index to individual points that differ from the dominating cluster index. Depending on the size of the rolling window, even multiple deviating points can be adjusted this way. An important aspect which this method also fulfills is that an already clear transition between microsegments remains intact.

2	2	1	1	3	1	1

↓ Transformation in $\{0,1\}^k$ ↓

$$\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}$$

↓ Rolling Average ↓

$$\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}\frac{1}{3}\\\frac{2}{3}\\0\end{bmatrix}\begin{bmatrix}\frac{2}{3}\\\frac{1}{3}\\0\end{bmatrix}\begin{bmatrix}\frac{2}{3}\\0\\\frac{1}{3}\end{bmatrix}\begin{bmatrix}\frac{2}{3}\\0\\\frac{1}{3}\end{bmatrix}\begin{bmatrix}\frac{2}{3}\\0\\\frac{1}{3}\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}$$

↓ Assignment↓

$$\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}$$

↓ Transforming to Cluster↓

2	2	1	1	1	1	1

Table 2: Example on how to smooth the cluster result with a rolling window of length 3.

(a) Before Smoothing (b) After Smoothing

Figure 3: Correcting missmatched points after clustering.

In a next step the driving sections are handeled in which a dominant cluster is not apparent. For this, the variance of the smooth clusters is calculated.

If in a window of 9 s the cluster index changes more than four times or the variance is smaller than 0.1 and the cluster index changes more than two times, the data point is labeled as diverse and assigned to an additional cluster index value. After the introduction of this additional cluster index value, the smoothing is performed again. If a microsegment has a duration of less than 5 s, it will be assigned to the microsegment closest to the start or end time of the current microsegment. If the time difference to the previous and next microsegment does not differ, the microsegment will be assigned to which ever has the shorter duration. The entire dataset now consists of 325 311 microsegments with a duration of at least 5 s. To uniquely identify each microsegment, a microsegment index is assigned to each microsegment. This index is employed to aggregate the data and calculate the metrics later used for clustering. The aggregated table also contains the start and end time of each microsegment, which is later used to match the microsegments to the corresponding trips.

To not only capture the average or median slope of a microsegment, the cumulative sum of altitude gain and altitude loss are calculated while aggregating. These variables are now used to calculate the altitude gain per distance and altitude loss per distance. Together with the median velocity of a microsegment these two variables are used to cluster the microsegments.

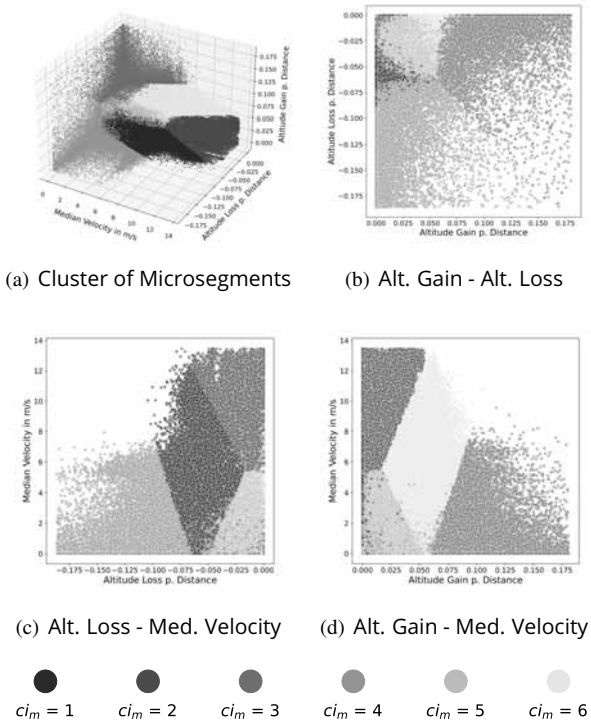
To filter out any outliers, microsegments in which the velocity falls within the bottom or top 5% of values or the altitude loss per distance or altitude gain per distance falls within the top or bottom 3% of values, are not considered. To determine the number of clusters, the metrics in Table 3 are considered. Based on that, a cluster number of $k = 6$ is chosen.

k	Silhouette	Davies–Bouldin	Calinski–Harabasz
2	0.324	1.292	139185
3	0.378	1.008	179274
4	0.385	0.909	207663
5	0.379	0.942	198766
6	0.404	0.876	213039
7	0.391	0.908	202768
8	0.390	0.922	200846

Table 3: Scores to determine the number of clusters.

As can be seen in Figure 4, the clusters are separated by the median velocity, as well as the altitude gain and altitude loss within a microsegment.

The cluster with $ci_m = 2$ contains microsegments with the highest velocities but only little to moderate changes in altitude gain and loss. The clusters with $ci_m = 1$ and $ci_m = 6$ contain microsegments with moderate velocities and are separated by altitude gain and altitude loss. While the cluster with $ci_m = 1$ contains microsegments with moderate altitude loss and almost no altitude gain, the cluster with $ci_m = 6$ contains microsegments with moderate altitude gain and only very little altitude loss. The cluster with $ci_m = 5$ contains the microsegments with low velocity and almost no altitude gain or loss. In contrast to that, the clusters with $ci_m = 3$ and $ci_m = 4$ contain microsegments with velocities up to 6.5 $\frac{m}{s}$. While the cluster with $ci_m = 3$ contains segments with moderate to high altitude gain and moderate altitude loss, the cluster with $ci_m = 4$ contains segments with moderate altitude gain and moderate to high altitude loss.



(a) Cluster of Microsegments (b) Alt. Gain - Alt. Loss

(c) Alt. Loss - Med. Velocity (d) Alt. Gain - Med. Velocity

Figure 4: Cluster Result Microsegments.

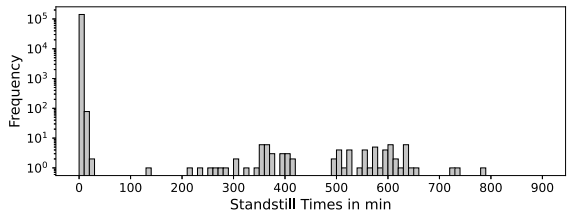
4 Segmentation and Clustering of Trips

In addition to segmenting the dataset based on changes of the velocity or the slope, segmentation is performed at a higher level based on a minimum time of standstill.

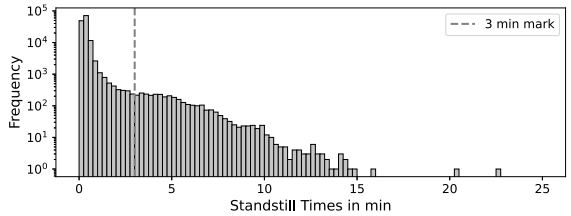
A trip ends when the velocity signal is below 0.5 $\frac{m}{s}$ for a certain amount of time. As soon as the velocity signal exceeds this value again, a new trip begins. Figure 5(a) shows a histogram of the standstill times up to a duration of 900 minutes.

This exposes that most standstill times are below 20 minutes, which correspond to short breaks and traffic flow interruptions during daily operation. The standstill times between 200 and 800 minutes most probably reflect the nighttime breaks. However, choosing a minimum standstill time of 200 minutes would result in a total of only 229 trips because only 0.16 % of standstills have a minimum duration of 200 minutes.

Consequently, the minimum standstill time is set at a lower threshold. As can be seen in Figure 5(b) a high proportion of standstill times are below a duration of three minutes. Standstills of less than three minutes are therefore attributed to normal standstill times during operation, for example at bus stops or traffic lights. Standstills that last longer than three minutes account for 2.44% of all standstills and are defined as a segmentation criterion for the trips. This results in a total number of 3 475 trips, 99.42% of which have a duration of less than 300 minutes.



(a) Histogram of Standstill times until 900 minutes.



(b) Histogram of Standstill times until 25 minutes.

Figure 5: Histograms of Standstill times.

A closer look reveals that 10.16% of the trips are shorter than 30 s and 11.08% are not longer than 5 minutes. However, a trip duration of less than 5 minutes is not reasonable. Therefore, all trips of less than 5 minutes are excluded from further analysis. This leaves 3 090 trips for the cluster analysis.

After segmentation, all 325 311 microsegments are temporally assigned to the trip that includes the microsegment. Since trips with a duration of less than 5 minutes are neglected, 0.97% of microsegments cannot be assigned to trips. For each trip, the temporal proportions of microsegments with cluster index ci_m are calculated. As the best result within the cluster analysis of the microsegments is achieved when $k = 6$ is set, it leads to six new input variables for the cluster analysis of the trips. Outliers are identified and excluded by neglecting data points if the value of any of the used variables fall within the 1% lowest or highest values. From 3 090 trips before filtering there are 2 866 left after filtering for the cluster analysis. To determine the number of clusters k with the best cluster results, the metrics in Table 4 are calculated for different values of k. As the Davies-Bouldin and Calinksi-Harabasz Index indicates best results for $k = 6$, a cluster number of six is chosen. In order to show the results of the cluster

k	Silhouette	Davies–Bouldin	Calinski–Harabasz
2	0.240	1.804	731.8
3	0.175	1.740	631.8
4	0.183	1.625	598.5
5	0.185	1.534	559.7
6	0.189	1.441	540.9
7	0.174	1.450	504.3
8	0.163	1.459	479.6

Table 4: Scores to determine the number of clusters within cluster analysis on trip level.

analysis with six input variables, a radar chart is plotted in Figure 6. The chart visualizes the centers of the six clusters with the cluster indices $ci_t = 1$ to $ci_t = 6$ from the trip-level cluster analysis. Each axis of the radar chart represents one cluster variable. These variables represent the proportions of microsegments that are assigned to the cluster indices $ci_m = 1$ to $ci_m = 6$. The proportions range from 0% in the center of the chart to 50% in the outermost circle. Each cluster on trip-level is represented by one line. As an example, the trip corresponding to the cluster center of $ci_t = 6$ consists of 26% microsegments which are assigned to $ci_m = 6$ and 5% of microsegments which are assigned to $ci_m = 3$. Figure 6 also shows that the proportions for trips represented by $ci_t = 2$ and $ci_t = 4$ overlap significantly. Likewise $ci_t = 1$ and $ci_t = 5$ have similar curves, but differ in the axes $ci_m = 5$ and $ci_m = 6$. The cluster $ci_t = 6$ clearly stands out due to higher values for $ci_m = 6$ and the cluster $ci_t = 3$ differs by an overall flatter distribution. The proportions of microsegments assigned to $ci_m = 2$ are

the highest, ranging between 35% to 45%, whereas the proportions of $ci_m = 3$ are the lowest with proportions under 9%.

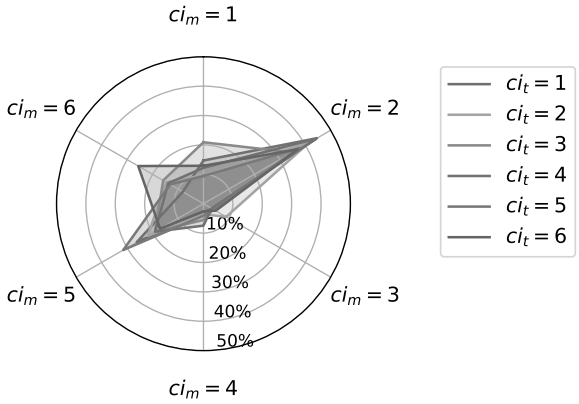


Figure 6: Radar chart of trip-level cluster centers of clusters $ci_t = 1,...,6$ representing temporal proportion of microsegment clusters $ci_m = 1,...,6$.

5 Results and Discussion

The clustering of individual points of driving data from electric city buses leads to the differentiation of four driving scenarios. Cluster $ci_p = 4$ represents scenarios in which the vehicle is not moving or moving at a very slow velocity, for instance at bus stops. By contrast, $ci_p = 2$ indicates normal driving in urban traffic at velocities ranging from 6 to 19 $\frac{m}{s}$. The remaining two clusters contain scenarios in which the bus is either ascending or descending an incline, irrespective of its velocity. When consecutive data points are combined into microsegments based on the preceding cluster results and the microsegments are then clustered again, a similar result is obtained. Again there is a cluster containing microsegments with low median velocity and a cluster containing microsegments with high median velocity. Furthermore, there are two clusters, each predominantly containing microsegments with either altitude gain or loss. However, two additional clusters are identified, representing microsegments with high altitude gain or loss, or both, and low to moderate values for the median velocity. The additional clusters can be attributed to the replacement of slope with the altitude gain and loss and the fact that a driving section is now considered instead of individual points. Therefore sections with altitude gain as well as altitude loss can occur.

Analyzing the results of the clustering concerning trips, it can be seen that the most common microsegments occurring in a trip are microsegments with $ci_m = 2$ and $ci_m = 5$ with over 30% for some trips. Those microsegments correspond to segments with only moderate values for altitude gain and loss. This aligns with the topography of Zurich, as the city center is predominantly flat, while the surrounding areas feature hilly terrain [14]. This also corresponds to the observation that all clusters contain less than 9% of microsegments from $ci_m = 3$ and $ci_m = 4$ which represent microsegments with high altitude difference. The distribution of microsegments across trips is relatively balanced. One reason contributing to this finding is that trips have a duration of at least five minutes and can span over multiple hours and therefore increase the probability of many different microsegments being contained in one trip. Another reason might be that the dataset is limited to city buses within a single city, resulting in routes that are inherently similar. Because of the variables available and to mitigate the complexity of the clustering results only two or three variables, respectively, are considered in the first step. However, these variables are only capable of representing the actual driving sections to a limited extent.

6 Conclusion

The aim of this research is to develop an approach to effectively segmenting driving data and finding groups of similar driving sections using a clustering algorithm. For this purpose a dataset of electric city buses operating in Zurich is used. After data preparation, all data points are being clustered in a first cluster analysis. The categorization of the data points using the cluster results enables a segmentation of the dataset into microsegments. The dataset is aggregated on these microsegments and a second cluster analysis is conducted. The results show, that the characteristic patterns obtained in the first cluster analysis of points can also be seen in the second cluster analysis of microsegments. Additionally, the dataset is segmented into longer driving sections, referred to as trips, based on the duration of standstills between two consecutive driving sections. All microsegments are temporally assigned to the trips and the proportion of microsegments with the same cluster index are computed and used as an input variable for the cluster analysis on trip-level. The results demonstrate minimal variation in the distribution of microsegments across the trips. Considering the setting of Zurich, the segmentation of microsegments and distribution of trips achieves plausible results.

In summary, this approach enables to identify representative driving sections by segmentation of driving data based on cluster analyses. Thereby a basis for generating representative driving cycles is provided, which are essential to fully exploit the capabilities of longitudinal dynamics simulations. In a further study, more variables, for example a congestion index could be taken into account, to potentially represent driving sections more accurately. Another subject of interest could be including driving data from other cities to gain information on the typical trips of city buses in general. Moreover, an investigation on data of intercity buses or coaches could discuss the transferability of the approach described in this paper to data of other bus classes.

Acknowledgement

This research is part of the project "HyCo", which is funded by the Federal Ministry for Digital and Transport under the grant number 03B10305B.

Publication Remark. This contribution is the revised version of the workshop version in ASIM Workshop GMMS/STS 2025 Tagungsband, p. 29-36; ISBN ebook: 978-3-903347-66-3, DOI 10.11128/arep.48

References

[1] Montazeri-Gh M, Fotouhi A, Naderpour A. Driving patterns clustering based on driving features analysis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. 2011;225(6):1301–1317. doi: 10.1177/2041298310392599.

[2] Brady J, O'Mahony M. Development of a driving cycle to evaluate the energy economy of electric vehicles in urban areas. *Applied Energy*. 2016;177:165–178. doi: 10.1016/j.apenergy.2016.05.094.

[3] Langner J, Grolig H, Otten S, Holzäpfel M, Sax E. Logical Scenario Derivation by Clustering Dynamic-Length-Segments Extracted from Real-World-Driving-Data. In: *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*. INSTICC, SciTePress. 2019; pp. 458–467. doi: 10.5220/0007723304580467.

[4] Chetouane N, Klampfl L, Wotawa F. Extracting information from driving data using k-means clustering. In: *Proceedings - SEKE 2021*, Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE. Knowledge Systems Institute Graduate School. 2021; pp. 610–615. doi: 10.18293/SEKE2021-118.

[5] Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding. In: *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. 2007; pp. 1027–1035.

[6] Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*. 1982; 28(2):129–137. doi: 10.1109/TIT.1982.1056489.

[7] Berzi L, Delogu M, Pierini M. Development of driving cycles for electric vehicles in the context of the city of Florence. *Transportation Research Part D-transport and Environment*. 2016;47:299–322. doi: 10.1016/j.trd.2016.05.010.

[8] Rousseeuw P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Comput. Appl. Math. 20, 53-65. *Journal of Computational and Applied Mathematics*. 1987; 20:53–65. doi: 10.1016/0377-0427(87)90125-7.

[9] Davies DL, Bouldin DW. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1979;PAMI-1(2):224–227. doi: 10.1109/TPAMI.1979.4766909.

[10] Caliński T, Harabasz J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*. 1974;3(1):1–27. doi: 10.1080/03610927408827101.

[11] Chetouane N, Wotawa F. On the application of clustering for extracting driving scenarios from vehicle data. *Machine Learning with Applications*. 2022; 9:100377. doi: 10.1016/j.mlwa.2022.100377.

[12] Widmer F, Ritter A, Onder CH. ZTBus: A large dataset of time-resolved city bus driving missions. *Scientific Data*. 2023;10(1). doi: 10.1038/s41597-023-02600-6.

[13] Valhalla contributors. Valhalla: Open Source Routing Engine for OpenStreetMap. `https://github.com/valhalla/valhalla`. Accessed: 2024-11-17.

[14] NASA. Shuttle Radar Topography Mission (SRTM). `https://www.earthdata.nasa.gov/data/instruments/srtm`. 2000. Accessed: 2024-11-22.

[15] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011;12:2825–2830.