Clustering Traffic Scenarios Using Mental Models as Little as Possible

Florian Hauer, Ilias Gerostathopoulos, Tabea Schmidt, Alexander Pretschner

Abstract— **Test scenario generation for testing automated and autonomous driving systems requires knowledge about the recurring traffic cases, known as scenario types. The most common approach in industry is to have experts create lists of scenario types. This poses the risk both that certain types are overlooked; and that the mental model that underlies the manual process is inadequate. We propose to extract scenario types from real driving data by clustering recorded scenario instances, which are composed of timeseries. Existing works in the domain of traffic data either cannot cope with multivariate timeseries; are limited to one or two vehicles per scenario instance; or they use handcrafted features that are based on the mental model of the data scientist. The latter suffers from similar shortcomings as manual scenario type derivation. Our approach clusters scenario instances relying as little as possible on a mental model. As such, we consider the approach an important complement to manual scenario type derivation. It may yield scenario types overlooked by the experts, and it may provide a different segmentation of a whole set of scenarios instances into scenario types, thus overall increasing confidence in the handcrafted scenario types. We present the application of the approach to a real driving dataset.**

I. Introduction

Automated and autonomous driving systems (ADAS) are commonly tested in simulation using *scenario-based testing*: testing such driving systems in challenging traffic scenarios. These are (extreme) instances of so-called scenario types. Scenario types, or functional scenarios [17], capture recurring traffic situations. One example is a vehicle following another on the right lane of a two-lane highway when both vehicles are overtaken by a third vehicle. During testing, scenario types are used to generate *scenario instances* which vary in different aspects [3], [4], [9]. In the example, different instances may consider different driving speeds or distances between the cars. The goal of scenario-based testing is to identify instances that stress the autonomous driving behavior (e.g., near crashes, abrupt acceleration or deceleration). Proving that the system works as expected in the challenging instances increases confidence in the system. This requires that the list of known scenario types for test case generation is "complete," e.g. as discussed in [10].

The derivation of a "complete" list of scenario types is challenging. A common approach in industry is to have experts manually create such lists of scenario types. However, no matter how comprehensive such lists may be, the manual creation process poses multiple risks: (i) certain scenario types are overlooked, (ii) the mental model, according to which the derivation is done, is inadequate. Using an expert's mental model for scenario type derivation influences the way that scenario types are structured and at which level of granularity they are located. For instance, an expert could derive scenario types according to the existence of maneuvers like braking or lane changing. Similarly, the derivation could be stopped at the granularity level of *contains a lane change* or *contains a lane change to the right in front of another vehicle on the target lane*. Because this manual derivation process necessarily introduces bias, there is an obvious need to validate the results.

An alternative approach is to derive scenario types from real driving data such as [11], [12], [16]. Such recordings of real driving contain a high number of scenario instances, from which scenario types can be derived in an automated way (but which risk missing relevant scenario types [10]). In this work, we present an approach that derives scenario types in an automated way without relying on handcrafted features and that is nearly independent of the mental model of an expert. Note that because a human has to select features and distance measures, it is impossible to *completely* remove any kind of mental model—but we aim at minimizing the introduced bias. Our approach yields scenario types of various levels of granularity, structured independently of an elaborate mental model. Thus, it can be used to evaluate manually derived scenario types w.r.t. completeness and adequacy. Note that a hand-written set of rules, e.g. to detect a lane change in the data, depends on the mental model of the person(s) providing the rules, similar to manual derivation. We hence believe that both manual and automated derivation of scenario types should be executed redundantly.

Existing works have suggested clustering techniques to group recorded driving data according to specific features extracted from each recorded drive. The technical solutions presented in these existing works come with at least one of the following technical limitations (see §V): (i) they are restricted to scenario instances with only one or two vehicles; (ii) they are not capable of handling multivariate timeseries of variable length; or (iii) are restricted to scenario instances of two seconds duration. Thus, such approaches cannot be applied in general to arbitrary scenario instances. Moreover, some approaches make use of handcrafted features that are based on a mental model. For instance, [13] uses one feature which explicitly encodes whether or not a scenario instance contains a braking maneuver. We propose a solution that overcomes such technical limitations of existing works.

Our *contributions* are two-fold. From a *technical* perspective, the presented approach generalizes existing clustering approaches to scenario instances that are composed of any

F. Hauer, T. Schmidt, and A. Pretschner are with the Department of Informatics at the Technical University of Munich, Germany. (e-mail: {florian.hauer, tabea.schmidt, alexander.pretschner}@tum.de).

I. Gerostathopoulos is with the Faculty of Science at the Vrije University in Amsterdam, Netherlands. (e-mail: i.g.gerostathopoulos@vu.nl).

number and any kind of timeseries, containing any number of vehicles, and are of any duration. Thus, technical limitations of existing approaches are overcome. From a *methodological* perspective, the presented approach reduces the dependency on mental models for automated scenario type derivation. This way, it can potentially identify scenario types missed during manual derivation improving completeness, and can increase the confidence in the manual derived scenario types, thus validating the mental model of the experts.

In §II, we introduce scenario-based testing. §III explains the technical details of automated scenario clustering. Experiments and insights are discussed in §IV, followed by a presentation of related work in §V. We conclude in §VI.

II. Overview of Scenario-Based Testing

The goal of scenario-based testing of ADAS is to subject the driving system to a variety of traffic scenario types. For each type, "good" test cases are generated, which are test cases that can reveal *potential* faulty behavior [9], [21]. Intuitively, the more complete a set of scenario types is, the more convincingly testing can ensure correct system behavior. Currently, experts derive scenario types manually according to their experience in form of a mental model. This process comes with the described shortcomings. Our work aims at the automated derivation of scenario types as depicted in Fig. 1. The goal is to complement manual derivation by potentially identifying scenario types that were overlooked and by increasing the confidence that the manually derived list of scenario types is complete.

An increasing amount of (publicly available) real driving data (1) serves as foundation. The data sets were designed for different purposes and collected in various ways and locations (see [26] for a survey). While some data sets, e.g. [16], are recorded from a single ego-vehicle's perspective, others are created from bird's eye perspective, e.g. [11].

We focus on the automated clustering approach (2), which aggregates real driving data into scenario types. The desired result (3) is a set of clusters, each representing a scenario type. Scenario instances that have the same structure are hence grouped into the same cluster, e.g. several instances where the ego-vehicle performs a lance change to the left behind a decelerating other vehicle. Ideally, there is not more than one cluster representing the same scenario type. Conversely, clusters that contain scenario instances of two distinct scenario types are not desired either. Determining purity and minimality obviously requires careful inspection.

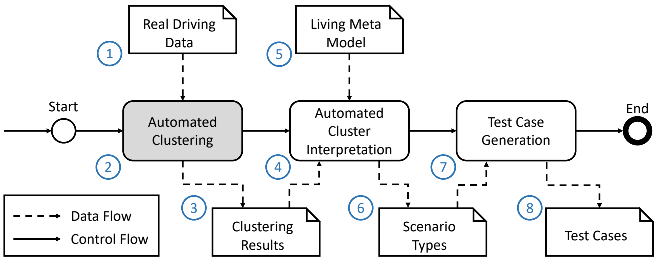


Fig. 1. Automated scenario type derivation for scenario-based testing.

Moreover, automated clustering approaches cluster solely based on *syntactic* features and do not interpret the scenario instances in a *semantic* way, as a human expert would do. The resulting clusters may be perfect in terms of the clustering quality on a *syntactic* level, e.g. measured by silhouette scores; and yet they may not represent the scenario types that a human expert would expect.

As a second step, automated cluster interpretation (4) is applied. It uses a living meta-model (5) for scenarios to interpret clustering results and yield the desired scenario types (6). A starting point for such a living meta model are existing scenario meta models [1], [7], [23]. The idea of a living meta-model for scenarios is to create a meta-model and improve it over time, such that it increasingly resembles the scenario types of real traffic. The meta-model assigns *semantic* meaning to the cluster contents that are merely more than timeseries. For instance, a "lane change" is detected whenever the lateral position of a vehicle exceeds a certain threshold. The cluster may be interpreted as the scenario type shared by most of the scenario instances in the cluster. This process yields a list of scenario types. Ideally, such a list is as complete as possible, in that it contains "all" relevant possible scenario types of real traffic [10].

Finally, for each scenario type in the list, a variety of different approaches for test case generation (7) can be used to generate "good" test cases (8) [8].

III. Automated Clustering Approach

An overview of the approach is provided in Fig. 2. The first step is data preparation (1). Real driving datasets usually consist of n data segments or scenario instances $d_i, i \in [1, n]$. We assume this segments to be given; a real drive recording of many kilometer length has to be segmented first. Two such scenario instances are shown in Fig. 2, "car following" and "cut-in." A scenario instance d_i consists of a list of m timeseries $ts_{j,d_i}, j \in [1, m]$ that describe the evolution of m object attributes related to an ego-vehicle over time. For instance, a timeseries can be the evolution of the distance $s_{c_1} - s_e$, where s_e is the position of the ego-vehicle and s_{c_1} the position of a preceding vehicle (Fig. 2, top). Since our technique can cope with any number of timeseries m, it overcomes the technical limitations of existing works that only allow small or specific numbers. We will assume that each scenario instance consists of the same list of timeseries. The time interval of the m timeseries of a single scenario instance is equally long, but usually differs among scenario instances. This way, we overcome the technical limitations of very short or fixed lengths scenario instances. Depending on the dataset, the number of timeseries (attributes) m may range from a dozen to even a hundred, capturing e.g. the difference in longitudinal and lateral positions, velocities, or accelerations between an ego-vehicle and vehicles on its left lane, its right lane, its own lane, etc.

Once the data is prepared, normalization is applied. We chose z-normalization (2) as suggested in [6], since it emphasizes the structure of the timeseries and neglects the absolute

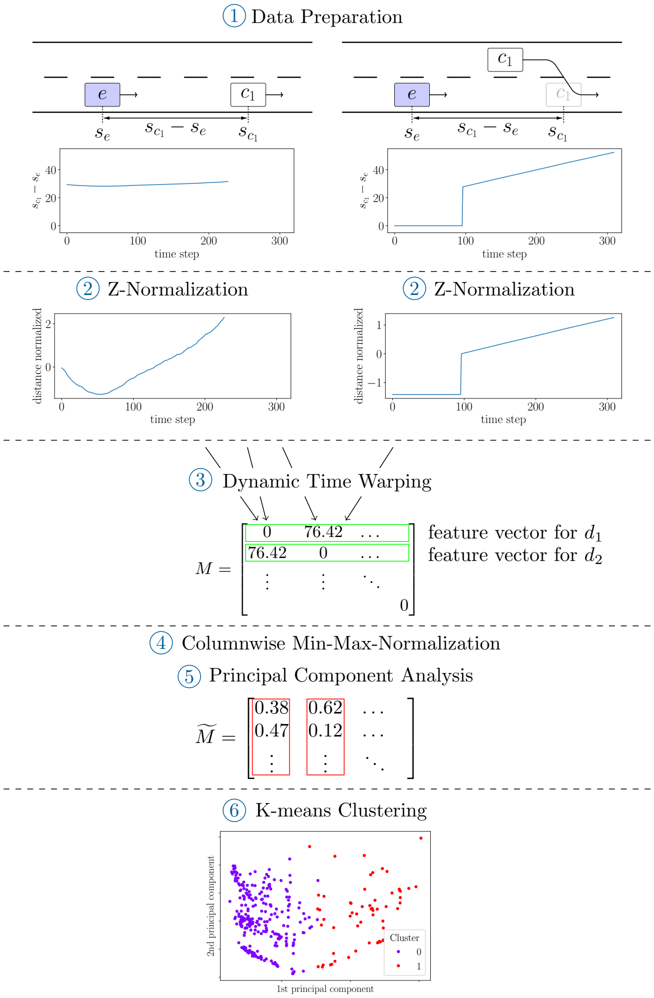


Fig. 2. Overview of the proposed approach.

values, desired as described above. For each timeseries ts_{j,d_i}, z-normalization is applied individually.

For the computation of the feature vectors used for clustering, we use (3) Dynamic Time Warping (DTW) [19]. DTW is one of several methods to measure the difference between two timeseries. Those timeseries may be of different lengths, and the key structural characteristics may be shifted and stretched over time without affecting the final score of DTW (contrary to e.g. Euclidean distance). We use DTW based on the L1-norm; see e.g. [2]. In our example of Fig. 2, the DTW distance between the timeseries of the two scenario instances is 76.42, while the DTW distance between two identical timeseries is 0. Based on this, a distance vector q between two scenario instances d_i and d_j can be defined as $q_k = DTW(ts_{k,d_i}, ts_{k,d_j})$ with $k \in [1, m]$. This means q contains the pairwise DTW distances of the timeseries of the two scenario instances. In our example, since we used $m = 1$ timeseries of each scenario instance, q has length 1, in particular $q = [76.42]$. The final feature vector of d_i is created by concatenating all distance vectors between $d_j, j \in [1, n]$ and d_i. Such a feature vector of a scenario instance can be understood as the difference to all other scenario instances based on the individual timeseries. In total, each feature vector has a length of $n * m$. This results in a feature matrix M of dimension $n \times n * m$.

Using such a similarity measure instead of handcrafting features, our approach intuitively generalizes well. This concept of feature computation—and with that the clustering—does not depend on the mental model of an expert. Moreover, by comparing timeseries in a direct way using DTW, instead of comparing their summary statistics such as moving averages or min-max values, we preserve important patterns that may be lost in aggregation.

The next step, columnwise min-max-normalization (4), ensures that all features are scaled to $[0..1]$. Remember that the feature vectors' length is linear in the number of scenario instances to be clustered. To reduce the dimensionality of the feature vectors and facilitate clustering, we use Principal Component Analysis (PCA) (5), a well-known statistical procedure for dimensionality reduction. For our experiments, the PCA was parameterized to keep 95% of the variance in the features, which resulted in 15 dimensions (§IV).

For the clustering itself, we chose classic k-means (6), since it allows for an easier interpretation of the clustering results compared to other techniques. We also experimented with hierarchical clustering, which yielded very similar results; and density-based clustering, which resulted in clusters of undesired structure and quality and were difficult to interpret. When using k-means, we do not prescribe the number k of clusters. Instead, we run k-means for every k from 2 to the number of scenario instances n and let a state of the art knee/elbow detector [22] choose the best k based on the inertia of the clustering results.

IV. Experiments

A. The highD Dataset

We applied our approach to the highD dataset [11], which is just one dataset containing well-structured highway traffic data. The data was recorded from a bird's-eye perspective with the help of a drone-mounted camera. Fig. 3 shows an exemplary picture. Each vehicle's trip from end to end of the field of view of the recording camera corresponds to one scenario instance with this vehicle as ego-vehicle.

1) Data Preparation: Surrounding vehicles may be very far away from the ego-vehicle due to the recording in bird's-eye perspective. Therefore, we apply pre-processing to the data in form of a *range of interest*. Other vehicles outside of this range are not considered to be neighbors of the ego-vehicle. This region of interest should be the maximum range at which other cars still influence the scenario type.

We chose 60m with the intuition that the scenario type from the perspective of the ego-vehicle mainly depends on the next and not so much on the second next car ahead or

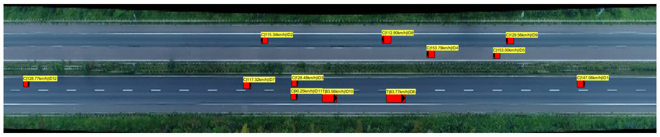


Fig. 3. Exemplary image of the highD dataset

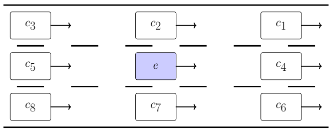


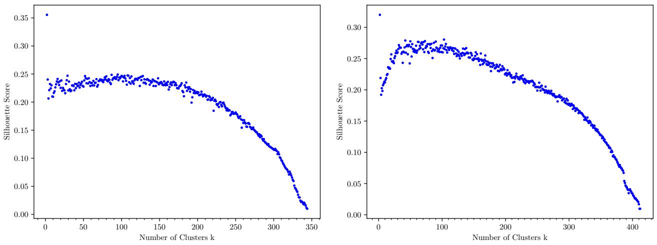
Fig. 4. Eight car model for environment modelling of the ego-vehicle

behind. Another pre-processing step is to filter the scenario instance for those where the ego-vehicle is of type "car," since we are interested in scenario types to test automated and autonomous driving systems for passenger cars. Trucks and other vehicle types may still be part of the environment.

2) Choosing Relevant Timeseries: There are several kinds of timeseries in the data, including longitudinal and lateral positions as well as velocities, and meta data, e.g. vehicle type. Following the intuition of the *range of interest*, we consider the eight cars around the ego vehicle at every time step (Fig. 4) similar to [11]. Based on the available data, we computed the longitudinal and lateral distances from the ego-vehicle to those eight other vehicles. As long as there is no such other vehicle at one of those eight positions, the respective timeseries is set to 0. This results in $m = 2 * 8$ timeseries per scenario instance. Note that the eight car model and the choice of time series can be understood as a mental model. We argue, however, that this choice constitutes a minimalistic model. In principle, the presented technique can be applied to more cars and more timeseries to further reduce the influence of the mental model. Scenario instances are of different lengths, since vehicles pass the field of view with different velocities. The number of time steps varies between 200 for fast vehicles and 300 for slow ones.

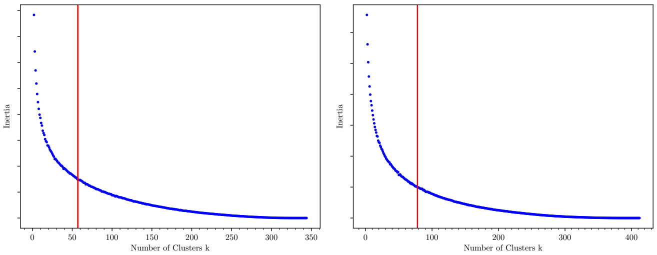
B. Experiment Results

We cluster the data for a two-lane and a three-lane recording, monitored over stretches of 420m, containing 346 and 414 scenario instances respectively. Clustering is done for every number of clusters k from two to the number of scenario instances n. A common way to identify the best k is the one with the highest so-called silhouette score [24]. Fig. 5 shows the silhouette scores for both datasets and all k. The maximum silhouette scores for both datasets is at $k = 2$, which is not surprising: In the two-lane dataset, all scenario instances have either other vehicles on the left or on the right. Clearly, the best clustering is to divide the scenario instances into two clusters, one for driving on each lane. Potential lane changes are assigned to the cluster of the two lanes on which the ego-vehicle drives for a longer duration. For most of the scenario instances of the three-lane data, this explanation still holds. Even though the best k equals 2, this is not a helpful clustering result; we therefore seek the next best k. However, for both datasets, a wide range of k provide similar silhouette scores, making a clear decision impossible. Therefore, we rely on the elbow (or *knee*) method [18] to identify a good k. We apply this method to the inertias of the clustering results (Fig. 6) and let the state-of-the-art elbow detector Kneedle [22] identify the elbow, providing us with $k = 57$ for two-lane data and $k = 78$ for three-lane data.



(a) Two lane data (b) Three lane data

Fig. 5. Silhouette score for each number of clusters in $[2..n]$



(a) Two lane data (b) Three lane data

Fig. 6. Inertias for each number of clusters in $[2..n]$; red line is the chosen number of clusters by the kneedle algorithm [22]

To understand the experiment results, we manually inspected the 57 and 78 clusters and interpreted the clusters as the scenario types that are shared by most of the instances contained. Intuitively, one would watch the video recordings to manually interpret the cluster representatives or run an automated interpretation. We cannot present videos in this paper nor can we show all the scenario type descriptions of all clusters. Instead, we present the 16 timeseries of one exemplary cluster of the two-lane data in Fig. 7. Additionally, we provide the clustering results for the presented experiments online.[1] For the two-lane case, the 57 clusters represent, upon manual inspection, 38 different scenario types. $38 < n = 57$ is a result of the nature of traffic data: Some scenario types are very rare compared to others, which leads to more than one cluster for a single common scenario type. For the three-lane data, we manually identified 67 distinct scenario types among the 78 clusters.

C. Threats to Validity

All experiments face threats to validity. We applied the presented approach to the highD dataset, which is limited in terms of data diversity, since the data is recorded on a straight highway section of 420m length without ramps. To transform the data from bird's-eye perspective to ego-vehicle perspective, we applied the discussed *range of interest*. By inspection of the data, we chose a suitable value, which might not be perfect to identify scenario types. For the clustering, longitudinal and lateral distances are experimentally selected as information source. It might be that there exists a better set of timeseries. Similarly, there might be more suitable clustering techniques than k-means, even though we experimentally

[1] `https://drive.google.com/open?id=1JApX49mbT-zULq3uFmiRRm4ja5SWFG23`

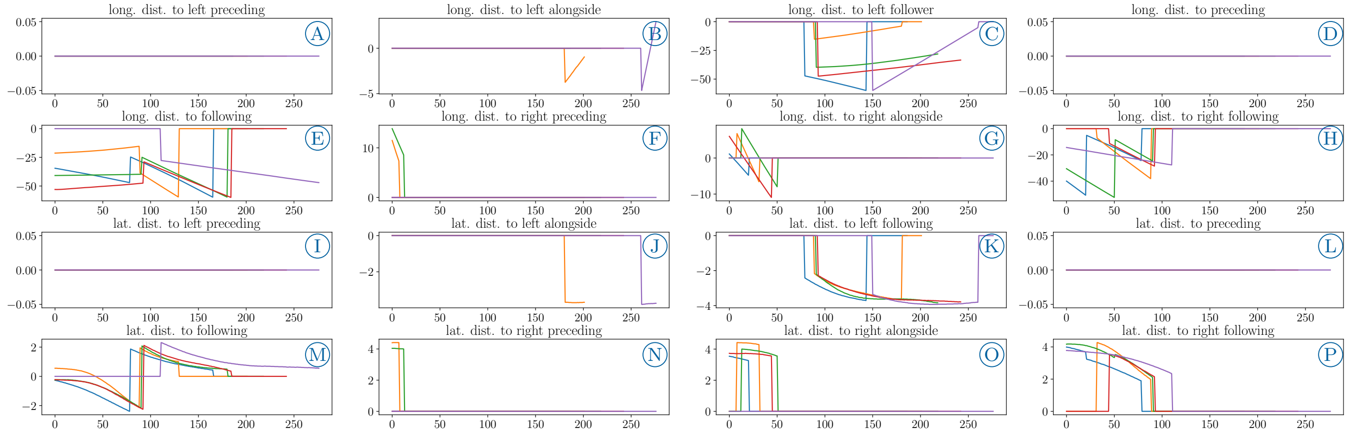


Fig. 7. Shown are the 16 timeseries of five scenario instances that are contained in a cluster of the two-lane dataset. The five scenario instances are plotted in blue, orange, red, green, and purple. The scenario instance plotted in blue is a good representative for this cluster. In this scenario instance, the ego-vehicle drives on the left lane with a vehicle behind it, as seen in plot (E). Then, the ego-vehicle drives by another vehicle on the right lane, which can be seen in the plots (G) and (H). Finally, the ego-vehicle performs a lane change to the right in front of the overtaken vehicle, which is indicated by the jump from -2 to 2 in plot (M). This behavior defines the scenario type for this cluster.

found k-means to perform better than others. The quality of clustering results strongly depends on the correct number of clusters. We used the elbow method, which leaves room for interpretation. We tried to mitigate eye-balling by using the Kneedle [22] algorithm. However, there might still be better numbers of clusters. The final experiment results have been analyzed and interpreted manually.

D. Discussion

Our feature vectors (§III) are defined solely by the distance between one scenario instance and all other scenario instances, based on the individual timeseries. These features hence do not encode any further “semantics.” The clustering depends on the timeseries data only. Information that is not contained in the timeseries data cannot impact clustering: scenario instances cannot be grouped according to this missing information, and missing scenario types cannot be found. For instance, if weather conditions should intuitively impact the resulting clusters but weather information is not input to the clustering algorithm, then the resulting scenario types cannot distinguish different weather conditions, unless of course this weather information correlates with other information provided as input (in this case, however, “weather” cannot be identified as a relevant feature). The choice of data in time series hence constitutes a mental model. Similarly, we are aware that the eight-car-model constitutes a mental model, but argue that it encodes a minimum amount of information needed for clustering.

Selecting the number of clusters k can be understood as the selection of a distance threshold up to which scenario instances are put into the same cluster. Choosing more or fewer clusters allows the adjustment of the granularity of the resulting scenario types. We let the Kneedle algorithm [22] automatically perform this choice to avoid further bias.

The resulting scenario types together with the provided level of granularity are meant to provide redundancy w.r.t. the scenario types (& granularity) of the experts’ manual derivation. This raises the question of what the “correct” level of granularity is for testing automated and autonomous driving systems, since it is crucial for the safety argumentation. Manual derivation of scenario types relies on the correctness of the mental model of the expert performing this derivation, which motivates the need for redundancy. Our work provides a perspective on scenario types that is barely influenced by mental models and can be used to identify further scenario types and to validate the scenario types yielded by manual derivation, both in terms of correct granularity and completeness.

V. Related Work

A multitude of existing works are concerned with clustering timeseries; see [15] for an overview.

In the domain of traffic engineering, the goal is to understand the usage and demand of the road network [2] or of single road sections [5]. Both works cluster two-dimensional GPS position timeseries of individual vehicle trips from start to destination. Since the position timeseries of a single vehicle does not contain information about surrounding vehicles and since such a trip is composed of a multitude of scenario instances, their approach is not suitable to cluster single scenario instances with traffic interactions to scenario types. From a technical perspective, their approaches are limited to the two-dimensional GPS position timeseries.

In [14] and [25], “driving encounters” between two vehicles are clustered based on vehicle position trajectories. The approach yields four [14] and ten [25] clusters where one cluster contains driving encounters at crossings, another clusters contains driving encounters where vehicles approach each other on opposing lanes, and so on. Arguably, this level of granularity does not provide a sufficient level of detail to yield fine-grained scenario types. For instance, it ignores the various different ways how two or more vehicles may interact at a crossing. From a technical perspective, the approach is limited to two-vehicle interactions while in reality scenario instances take place with more than two vehicles.

In [20], an approach is presented that clusters collision data to identify different types of collisions. Abstract, categorical information is used as input for the clustering, e.g. the gender of the driver or three categories of injuries. Intuitively, this approach is limited to the specific use case of collision data composed of categorical data. It is not applicable to (non-collision) driving data presented as timeseries.

The goal of [13] is to extract traffic scenario types from simulated driving data. The approach is limited to scenario instances of two seconds' length and interactions between two vehicles. The clustering is based on handcrafted features, such as aggregations and characteristic points within the timeseries, e.g. whether or not a braking maneuver took place as well as the velocity of both vehicles at the start and at the end of the two second time span. In reality there are traffic scenarios with (i) more than two interacting vehicles and (ii) usually such scenarios are longer than two seconds. Further, handcrafted features come with the discussed shortcomings.

In sum, this paper closes the following gap: It overcomes technological limitations, i.e. restrictions in the number of vehicles, the total duration of the scenario, and the type, number, and length of timeseries. Moreover, our approach clusters without handcrafted features and, thus, arguably relies on a minimum mental model of an expert.

VI. Conclusions

We motivated the need for scenario types by test scenario generation, highlighting that the current manual derivation by industrial experts poses a risk of incompleteness and inadequacy. We proposed an automated clustering approach to extract scenario types from real driving data. It solely relies on the difference between the timeseries of recorded scenario instances. We applied the presented approach to the highD dataset [11]. The presented experiment results show the application of the clustering to both a two-lane and a three-lane highway recording. We discussed how the presented feature creation allows the clustering to yield different levels of granularity, and how the clusters are influenced by the choice of data, and distance measures according to the eight-car-model, used for clustering. We have argued that the partitioning of the scenario instances into types is barely influenced by a mental model and the level of granularity is not pre-set by an expert. As the clustering results can therefore be used to evaluate handcrafted scenario types, this makes the presented approach valuable from a methodological perspective, in addition to overcoming the technical shortcomings of existing works. However, further research is necessary to understand what an adequate level of granularity is for scenario types. It heavily influences the test case generation and, therefore, also the overall assessment of the driving system. We believe that the presented approach is a first step in this direction.

VII. Acknowledgements

This work was supported by the Intel Collaborative Research Institute "Safe Automated Vehicles."

References

[1] J. Bach, S. Otten, and E. Sax. Model based scenario specification for development and test of automated driving functions. In *IEEE Intelligent Vehicles Symposium*, pages 1149–1155, 2016.

[2] P. Besse, B. Guillouet, J.-M. Loubes, and F. Royer. Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Trans. on Intelligent Transportation Systems*, 17(11):3306–3317, 2016.

[3] A. Calo, P. Arcaini, S. Ali, F. Hauer, and I. Fuyuki. Generating avoidable collision scenarios for testing autonomous driving systems. In *IEEE Intl. Conf. on SW Testing, Verification and Validation*, 2020.

[4] A. Calo, P. Arcaini, S. Ali, F. Hauer, and I. Fuyuki. Simultaneously searching and solving multiple avoidable collisions for testing autonomous driving systems. In *Genetic and Evolutionary Computation Conference*, 2020. to appear.

[5] M. Y. Choong et al. Modeling of vehicle trajectory clustering based on lcss for traffic pattern extraction. In *IEEE International Conference on Automatic Control and Intelligent Systems*, pages 74–79, 2017.

[6] D. Goldin and P. Kanellakis. On similarity queries for time-series data: constraint specification and implementation. In *Intl. Conf. on Principles and Practice of Constraint Prog.*, pages 137–153, 1995.

[7] L. Hartjen, F. Schuldt, and B. Friedrich. Semantic classification of pedestrian traffic scenarios for the validation of automated driving. In *IEEE Intelligent Transp. Systems Conf.*, pages 3696–3701, 2019.

[8] F. Hauer, B. Holzmüller, and A. Pretschner. Re-using concrete test scenarios generally is a bad idea. In *IEEE Intelligent Vehicles Symposium (IV)*, page to appear, 2020.

[9] F. Hauer, A. Pretschner, and B. Holzmüller. Fitness functions for testing automated and autonomous driving systems. In *Intl. Conf. on Computer Safety, Reliability, and Security*, pages 69–84, 2019.

[10] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner. Did we test all scenarios for automated and autonomous driving systems? In *IEEE Intelligent Transportation Systems Conf.*, pages 2950–2955, 2019.

[11] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2118–2125, 2018.

[12] A. Krämmer, C. Schöller, D. Gulati, and A. Knoll. Providentia - a large scale sensing system for the assistance of autonomous vehicles. *Robotics Science and Systems Workshops (RSS Workshops)*, 2019.

[13] F. Kruber, J. Wurst, and M. Botsch. An unsupervised random forest clustering technique for automatic traffic scenario categorization. In *IEEE Intelligent Transp. Systems Conf.*, pages 2811–2818, 2018.

[14] S. Li, W. Wang, Z. Mo, and D. Zhao. Cluster naturalistic driving encounters using deep unsupervised learning. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1354–1359, 2018.

[15] T. W. Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[16] W. LLC. Waymo open dataset. online at `https://waymo.com/open/`, retrieved 6th December 2019, 2019.

[17] T. Menzel, G. Bagschik, and M. Maurer. Scenarios for development, test and validation of automated vehicles. *arXiv:1801.08598*, 2018.

[18] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, June 1985.

[19] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[20] P. Nitsche, P. Thomas, R. Stuetz, and R. Welsh. Pre-crash scenarios at road junctions: A clustering method for car crash data. *Accident Analysis & Prevention*, 107:137–151, 2017.

[21] A. Pretschner. Defect-based testing. In: Dependable Software Systems Engineering, 2015.

[22] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan. Finding a kneedle in a haystack: Detecting knee points in system behavior. In *IEEE Intl. Conf. on Distr. Comp. Systems Workshops*, pages 166–171, 2011.

[23] J. J. So, I. Park, J. Wee, S. Park, and I. Yun. Generating traffic safety test scenarios for automated vehicles using a big data technique. *KSCE Journal of Civil Engineering*, 23(6):2702–2712, 2019.

[24] A. Starczewski and A. Krzyżak. Performance Evaluation of the Silhouette Index. In *Artificial Intelligence and Soft Computing*, pages 49–58. Springer International Publishing, 2015.

[25] W. Wang, A. Ramesh, and D. Zhao. Clustering of driving scenarios using connected vehicle datasets. *arXiv:1807.08415*, 2018.

[26] H. Yin and C. Berger. When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets. In *IEEE Intelligent Transportation Systems Conf.*, pages 1–8, 2017.