Identification of Lane-Change Maneuvers in Real-World Drivings With Hidden Markov Model and Dynamic Time Warping

Lars Klitzke*, Carsten Koch† and Frank Köster*

***Abstract*— For the introduction of new automated driving functions, the systems need to be verified extensively. A scenario-driven approach has become an accepted method for this task. But, to verify the functionality of an automated vehicle in the simulation in a certain scenario such as a lane-change, relevant characteristic of scenarios need to be identified. That, however, requires to extract these scenarios from real-world drivings accurately. For that purpose, this work proposes a novel framework based on a set of unsupervised learning methods to identify lane-changes on motorways. To represent various types of lane-changes, the maneuver is split up into primitive driving actions with a Hidden Markov Model (HMM) and Divisive Hierarchical Clustering (DHC). Based on this, lane-change maneuvers are identified using Dynamic Time Warping (DTW). The presented framework is evaluated with a real-world test drive and compared to other baseline methods. With a F_1 score of 98.01% in lane-change identification, the presented approach shows promising results.**

***Index Terms*— Lane-change Maneuver, Hidden Markov Model, Dynamic Time Warping, Divisive Hierarchical Clustering, Automated Driving**

I. Introduction

In the last decades, the domain of autonomous driving has gained much attention from the research community. The primary focus was on developing and improving automated driving functions, whereas the topic of testing these systems was only a niche. In recent years, however, that changes drastically, and more effort was put into developing new validation methodologies [1]. This was due to vague legal specifications and the enormous overall effort to verify the system's functionality. Indeed, the research focus on testing methods shifted to scenario-based validation approaches.

For a scenario-based validation methodology, however, a set of scenarios is essential as pointed out by Damm *et al.* [2]. Although several studies attempt to identify scenarios [3], [4], [5], [6], [7] and describing them appropriately with formal languages by Bagschik *et al.* [8], this topic is still an open issue and demands further investigation.

A typical scenario or maneuver on motorways are lane-changes. The analysis of lane-changes is broadly discussed in the academic domain for several decades. Vehicles performing lane-changes can affect the overall traffic flow [9] and are one of the main reasons for accidents on highways [10]. Thus, they pose a severe risk to traffic participants.

*Lars Klitzke and Frank Köster are with the German Aerospace Center (DLR), Institute of Transportation Systems, Braunschweig, Germany {lars.klitzke, frank.koester}@dlr.de.

†Carsten Koch is with the Hochschule Emden/Leer, University of Applied Sciences, Emden, Germany carsten.koch@hs-emden-leer.de.

With an increasing level of automation, the Automated Vehicle (AV) has to conduct lane-changes safely—even under uncertainty. Due to this, studies focused on analyzing lane-change maneuvers to derive driving models that can be integrated into the AV for trajectory planning [10] or to assess the AV's performance and safety [11]. For both research fields, however, the identification of lane-changes in large-scale databases from naturalistic field operational tests is a prerequisite.

An established methodology for lane-change detection is to employ supervised learning methods, e.g. Support Vector Machine (SVM) [12], [13], [14] or Artificial Neuronal Network (ANN) [15]. The data needs to be prepared with a sliding window approach, which size is, however, another hyperparameter to tune. Another drawback of using supervised learning methods are the required ground truth information for training. To overcome this, unsupervised learning methods were employed recently. Kruber *et al.* use random forest combined with hierarchical clustering to identify scenarios in simulation data [6]. Probabilistic-based methods such as the HMM were also proposed for, e.g., driving style analysis [16]. Nevertheless, the final cluster interpretation has to either be performed manually by an expert [6] or compared to predefined characteristics [16] since the semantic meaning is lost after clustering.

A. Contribution

This work presents a novel framework for offline identification of lane-change maneuvers in real-world driving data. Instead of using supervised learning methods to extract lane-change maneuvers, in this work, test-drives are clustered with an HMM into more trivial driving actions, the *driving primitives (DP)* and pattern matching is employed for the identification.

Since a significant drawback in clustering with non-deterministic methods such as the HMM is the different labeling, a method is proposed to recover the semantics of DPs. Moreover, to model lane-change maneuvers based on the DPs, this work employs a pattern-recognition based approach. That is, for each maneuver to identify, specific patterns are created and the driving sequence is clustered according to the DPs using Divisive Hierarchical Clustering (DHC). DTW is then employed to find the most likely maneuver for each interval based on the defined patterns.

B. Paper Structure

The remaining paper is structured as follows. In Section II, the proposed framework for lane-change identification based

on DPs is elaborated. At first, a typical lane-change is depicted and a set of DPs is derived. They are the basis for the definition of appropriate features in Section II-A. The features are used to train a HMM for sequence clustering in Section II-B. The semantics of the labels are recovered in Section II-C by employing k-means. The estimated driving primitive (DP) labels are used in Section II-D to partition the driving sequence into intervals with DHC and classify the lane-change with DTW. In Section II-E, an approach is proposed to find the start and end of each lane-change based on the intervals provided by DHC. Afterward, the framework is evaluated with a test drive that includes more than 200 lane changes in Section III. Finally, the work concludes with a summary and an outlook on further works in Section IV.

II. Lane-change identification Framework

The proposed approach for lane-change identification is a multi-level framework that consists of a set of unsupervised learning methods for clustering (HMM, DHC and k-means) into driving primitives (DP) and DTW for the final identification (see Fig. 2).

At first, a typical lane-change maneuver is decomposed into unambiguous stages (see Fig. 1) to identify the DPs that are used for clustering. In the first state *Idle*, the vehicle drives in the center of the lane and transitions into the state *Approach* if it approaches either the left or right lane marking. If any vehicle side crosses the street marking, the vehicle is in the state *Cross* until the vehicle's center crosses that marking and changes to the *Change* state. Hence, the vehicle is changing the lane if the majority of the vehicle is on the new lane. The maneuver finishes if the second side of the vehicle crosses that marking, which is indicated by the *Depart* state, followed by the *Settle* state denoting the vehicle driving in the middle of the new lane.

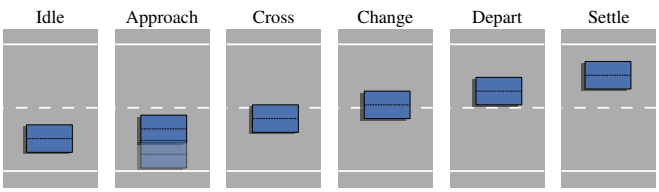

Fig. 1: A simple lane-change maneuver consists of six stages. They are used for maneuver sequence partitioning and classification.

From a lane-oriented point of view, a typical lane-change maneuver does not consist of six but four states. The states *Idle* and *Settle*, as well as *Approach* and *Depart*, are more or less the same since the absolute relative position of the vehicle in the lane equal. Hence, the set S of DPs used in this work only consists of four states.

$$S = \{Idle, Approach, Cross, Change\} \tag{1}$$

Based on this concept of partitioning a lane-change maneuver into DPs, a multi-level framework is proposed to perform maneuver classification (see Fig. 2). In the following, each level is described briefly.

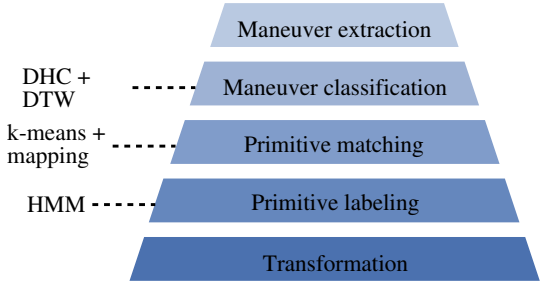

Fig. 2: The proposed framework for lane-change identification based on driving primitives.

A. Transformation

To divide the lane-change maneuver into the set of driving primitives and associate each sample in the sequence with a driving primitive, features need to be selected that appropriately represent the vehicle's state. In this work, the distances from the vehicle's center to the left d_{cl} and right d_{cr} lane marking are employed that are provided by a camera-based ADAS. However, instead of using these distances, features are derived that are independent of the vehicle and lane width. That allows the presented approach to function on roads with arbitrary lane widths and for different vehicles.

The first feature is the normalized distance from the vehicle center to the lane center d_c (see Fig. 3). This feature represents the lateral displacement of the vehicle in the lane. Based on the distance between the vehicle's center and the left lane marking, the normalized lane-center distance d_c is defined as

$$d_c = \frac{\frac{w_l}{2} - d_{cl}}{\frac{w_l}{2}} = 1 - \frac{2d_{cl}}{w_l} \tag{2}$$

with d_{cl} as the distance to the left marking and w_l the lane width. The latter is estimated using the distances provided by the on-board system (d_{cl} and d_{cr}). The lane width $w_l = |d_{cr} - d_{cl}|$ is the absolute sum of the distances $d_{cl} \in \mathbb{R}$ and $d_{cr} \in \mathbb{R}$ to the left and right lane markings since $d_{cr} < 0$ and $d_{cl} > 0$.

Although the normalized distance d_c may be used for partitioning only, we propose to use another feature $crossing$ to divide the four states further onto two clusters. Due to this, the HMM will more likely be able to represent the driving primitives. The first cluster represents the vehicle moving within a lane (states *Idle* and *Approach*), whereas in the second one, the vehicle moves between two lanes (*Cross* and *Change*). For that purpose, the normalized distances d_l and d_r (see Fig. 3) are used defined as $d_l = \frac{1}{w_l}\left(d_{cl} - \frac{w_v}{2}\right)$ and $d_r = \frac{1}{w_l}\left(d_{cr} + \frac{w_v}{2}\right)$ since the maximum value of d_l and d_r is $w_l - w_v/2$. That is because if the center of the vehicle crosses a lane marking, the on-board system changes the left and right lane markings to represent the new lane boundaries. In Fig. 1, for instance, the left marking is the dashed one and the right marking the solid line until the vehicle transitions in the *Change* state. Afterward, the left lane is the top solid line and the right lane marking the dashed one.

The distances from the vehicle sides to the lane markings

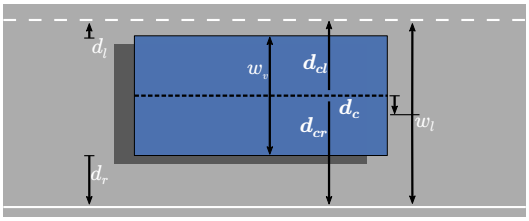

Fig. 3: The features used for the identification of lane-change maneuvers are derived from the distances between the vehicle center and the left d_{cl} and right d_{cr} lane markings.

(d_l and d_r) are employed to mark the vehicle's current sub-state with the feature $crossing \in \{-1, 0, 1\}$. That feature denotes if the vehicle crosses the left -1, the right 1 or no marking 0.

B. Primitive labeling

To partition the lane-change maneuver into sequences, each of which represent one of the four states of the vehicle defined previously and to associate each sample in the driving sequence with a DP, this work employs an HMM.

A HMM is typically used if the system's internal state is not accessible, but the system emits information. This information can be used to derive the system's internal state under the condition that the number of states is known. In this work, the system is the vehicle, the internal states are the driving primitives and the emitted information are the distance to the next lane markings (see (2)) and the $crossing$ flag. Hence, the aim is to derive the HMM's model parameters based on the driving sequence. For that purpose, the Baum-Welch algorithm can, according to Bilmes [17], be employed. Furthermore, since the emitted information is not discrete, the emissions are modeled with Gaussians [17]. With the derived parameter set and the given sequence, the aim is to find the optimal matching sequence of driving primitives. For that purpose, the Viterbi algorithm can be employed. [17] The result of an exemplary drive with a duration of fifteen minutes on a motorway that contains 39 lane-changes is shown in Fig. 4. The figure illustrates the vehicle's lateral displacement d_c distribution from (2) for each DP with each DP representing a hidden state of the HMM associated with a label $l \in \{0, 1, 2, 3\}$.

C. Primitive matching

The estimated model parameters of the HMM can be different for several runs. This is due to the inherent clustering nature of the used approach. Hence, the DP *Idle* may not always be labelled with 0 as in Fig. 4. That is, however, required in the follow-up steps to model lane-changes based on DPs. Hence, the aim of the primitive matching step is to recover the semantic of the labels so that, regardless of the used driving sequence, the labels $\{0, 1, 2, 3\}$ always represent the primitives S as defined in (1).

Let $P = \{p_1, p_2, p_3, p_4\}$ represent the DP clusters with $p_i = \left\{d_c^1, d_c^2, \ldots, d_c^n\right\}$ as the normalized distances defined in (2) for each DP. Let $L = \{0, 1, 2, 3\}$ denote the labels for

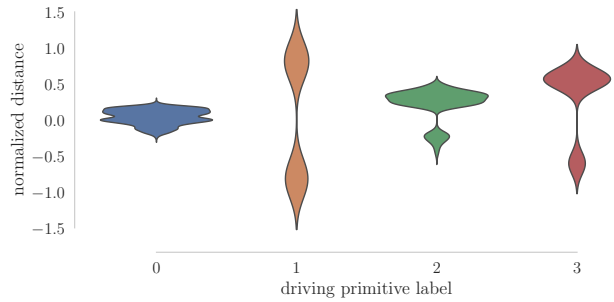

Fig. 4: The result of the Viterbi algorithm applied to a sequence using HMM model parameters estimated with the Baum-Welch algorithm assuming four hidden states. Each hidden state represents a DP. For each DP, the distribution of lateral displacement d_c is shown.

the DP and $M_L = \{(l_i, p_i) | \, l \in L, p \in P, 1 \leq i \leq |P|\}$ a set of tuples and each tuple consists of a label l and a DP p. Furthermore, let $M_L(p) \rightarrow l$ be the label $l \in L$ associated with the DP p. The aim is to determine the correct set of labels $\bar{L}$. For that purpose, the mean absolute normalized distances $\widetilde{d}_c = \left\{\widetilde{d}_{c,1}, \ldots, \widetilde{d}_{c,4}\right\}$ with $\widetilde{d}_{c,i} = \frac{1}{|p_i|} \sum_{j=1}^{|p_i|} |p_i^j|$ are estimated. The correct DP labels $\bar{L} = \left\{\bar{l}_1, \ldots, \bar{l}_4\right\}$ are the indices of the sorted set of means $\bar{L} = \arg\operatorname{sort} \widetilde{d}_c$ with $\arg\operatorname{sort}$ giving the index of the centers in $\widetilde{d}_c$ to sort the set in ascending order. The set of tuples $M_{\bar{L}}$ now associates each DP label correctly to its cluster. For the example in Fig. 4, the correct labels are $\bar{L} = \{0, 2, 3, 1\}$.

The next step is to determine the direction of the DP since we currently only know the type. Despite the *Idle* DP, all other follow a bimodal distribution with each cluster representing the direction of that DP (see Fig. 4). Since the clusters do not overlap, k-means is employed to cluster each DP $p_i \in P, i > 0$ despite the first one, into two clusters $c_i = \left\{c_i^1, c_i^2\right\}$ with $c_i^j, \; j \in \{1, 2\}$ denoting the cluster mean and $c_i^1 > c_i^2$. The DP direction of a sample x is encoded in the sign of the label l of the sample's primitive p_x with

$$l(x, p_x) = \begin{cases} M_{\bar{L}}(p_x) & \text{if} \quad |x - c_x^1| < |x - c_x^2| \\ -M_{\bar{L}}(p_x) & \text{else} \end{cases} \tag{3}$$

according to the distance of the sample to the primitive's cluster means $\left(c_x^1, c_x^2\right)$ so that $l \in \{-3, -2 \ldots, 2, 3\}$. The result of the label matching and primitive direction estimation is depicted in Fig. 5 with the primitives sorted by their mean distance and the cluster colors denoting the direction.

D. Maneuver classification

The next step in the framework is to find lane-changes based on the DP and to classify them as left or right lane-changes. Remark that each driving primitive is represented with a label $l \in \{-3, -2, \ldots, 2, 3\}$ and the sign of the label denotes the direction according to (3). Hence, the time-series of labels is divided into lane-change intervals and each interval is classified as left or right lane-change. That is,

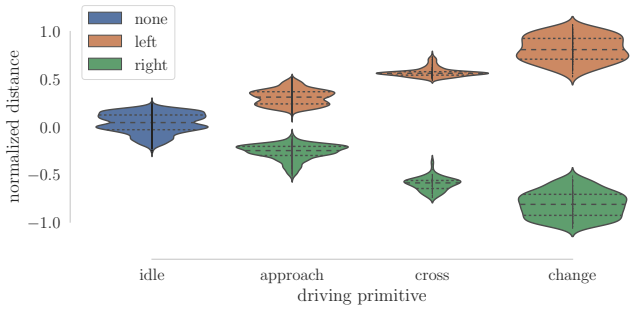

Fig. 5: The partitioned driving sequence with the correctly matched primitives, indicated with the names on the ordinate, and direction information encoded in the cluster color.

we want to find the pattern of a specific lane-change in the sequence.

For the first step, Divisive Hierarchical Clustering (DHC) is employed to partition the sequence into intervals w.r.t the absolute primitive label. Based on a starting label l_{min}, the sequence is recursively splitted until a maximum label l_{max} is reached. Only the intervals that contain the maximum label are selected for further processing. An example lane-change interval is depicted in Fig. 6 with the original sequence (top) and the sequence after partitioning with $l_{min} = 2$ and $l_{max} = 3$ (bottom). The next step is to find lane-change pattern in the extracted intervals. For pattern matching in time-series, Dynamic Time Warping (DTW) and more specifically, the nearest neighbor algorithm with DTW, is, although already proposed in the 70s [18], still the benchmarking algorithms for time series classification. It allows to compare signals with different lengths and provides a distance metric for comparison. That metric allows to select the most likely pattern that matches the interval. In this work, the *fastDTW* implementation [19] for DTW is used, which speeds up the process.

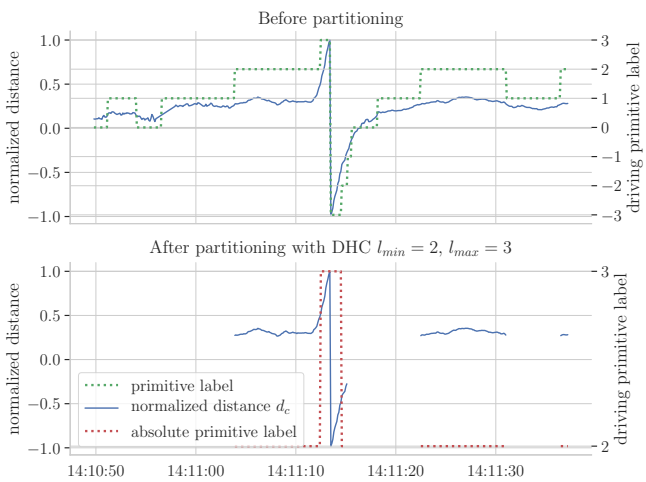

Fig. 6: Divisive Hierarchical Clustering is employed to split up the sequence into intervals based on the absolute driving primitive label. *Top:* The normalized distance d_c and label l before partitioning. *Bottom:* The same interval after partitioning.

At first, let $H = \{h_1, \ldots, h_k\}$ be a set of patterns to match with the previously extracted intervals $X = x_1, x_2, \ldots, x_n$ with each $x \in X$ representing only the driving primitive labels of that time-series. The pattern $h_l = \{1, 2, 3, -3, -2, -1\}$ represent a lane-change to the left and $h_r = \{-1, -2, -3, 3, 2, 1\}$ to the right. Furthermore, two patterns are defined to represent aborted lane-changes to the left and right. These are required to prevent misclassification of lane-changes attempts. DTW is now employed to map the patterns on each extracted time-series $x_i \in X$. Let $\mathrm{dtw}(h, x_i)$ return the distance mapping the pattern h on the time-series x_i. The overall aim is to find the pattern with the smallest distance

$$\arg\min_{h \in H} \mathrm{dtw}(h, x_i) \tag{4}$$

for all intervals. An example is depicted in Fig. 7 showing a lane-change to the left with the normalized distance to the lane center (in blue) and the driving primitive (green).

The matrix with the accumulated cost to map the value h_i of the pattern h at index i to the value x_j at the j index in the time-series is shown on the right with x as the time-series of driving primitive labels. The top image represents the matrix for the left lane-change pattern h_l and the bottom for the right h_r with the path of lowest cost represented in red. It is evident that the path for the left lane-change pattern has a smaller cost than the one for the right pattern (2.0 vs 85.0), and thus, the time-series is classified as left lane-change. The proposed approach for maneuver classification has the benefit that maneuvers of interest can be identified by merely adding further appropriate patterns to the list H.

E. Maneuver extraction

The intervals generated by the previous steps may not represent the real lane-change accurately. A typical case is if the vehicle is in the state *Approach* or *Cross* for a lengthy period before the actual lane-change (as in the example

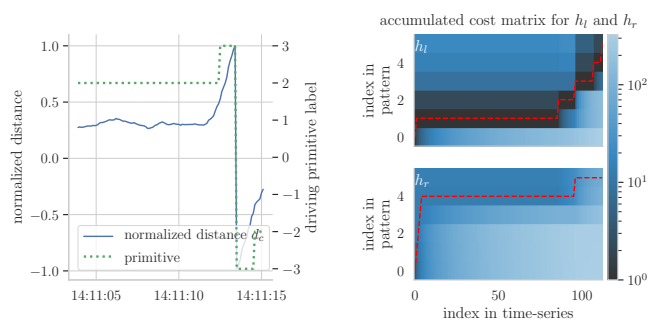

Fig. 7: Lane-change classification using Dynamic Time Warping. *Left:* The enlarged time series of the lane-change and the driving primitives. *Right:* The accumulated cost matrix for mapping the left (top) and the right lane-change (bottom) with the chosen path (in red).

depicted in Fig. 7). Thus, a post-processing step is required for maneuver interval extraction.

The proposed maneuver extraction approach is elaborated in the following for the lane-change maneuver depicted in Fig. 8 with the blue line as the normalized distance from the vehicle to the lane center. The approach aims at extracting the interval (t_s, t_e) represented with the two vertical lines in Fig. 8. Let $d_c(t)$ be the normalized distance from the vehicle center to the lane center as defined in (2) at time t in a time window $w = \{t - \Delta t, \ldots, t + \Delta t\}$ with a duration of $2\Delta t$ so that $t \in w$. Since the aim is to fine-tune the point in time of the maneuver start and end, the window is equally and generously enlarged in both directions by $\widetilde{\Delta t} = 8$ seconds or until the vehicle crosses a marking. The latter allows to divide two consecutive lane-changes properly. Then, let $|d_c(t)|$ represent the absolute normalized distance denoted as the dotted green line in Fig. 8. The first step is to split up the signal into two parts representing the time before and after the lane crossing.

For that purpose, the time and value t_{max}, d_{max} of the signal peak is estimated in the window denoting the point in time of the lane crossing. To prevent the selection of a maximum value that does belong to another lane-change in the enlarged time-series, the search area is restricted to the interval before enlargement.

Afterwards, the signal is split up into the left $w_l = \{t_1, \ldots, t_l\}$ and right $w_r = \{t_r, \ldots, t_{|w|}\}$ sub-window. Since the vehicle may drive for a longer period between the two lanes, the start of the windows (t_l, t_r) is determined by

$$t_l = \max\left(\{t | t \in w, t < t_{max} \wedge |d_c(t)| < \zeta\}\right) \tag{5}$$

and

$$t_r = \min\left(\{t | t \in w, t > t_{max} \wedge |d_c(t)| < \zeta\}\right) \tag{6}$$

that is the earliest and latest time where the signal is smaller than the threshold $\zeta = 0.9$ (see the upper horizontal line in Fig. 8). Note that choosing ζ is not critical as long as the threshold is lower than the maximum distance and greater than ϑ.

The next step is to find the situation where the vehicle's heading recovers to the lane's heading. For that purpose, the windows are stripped down to $w_l = \{t_0, \ldots, t_l - t_{o,l}\}$

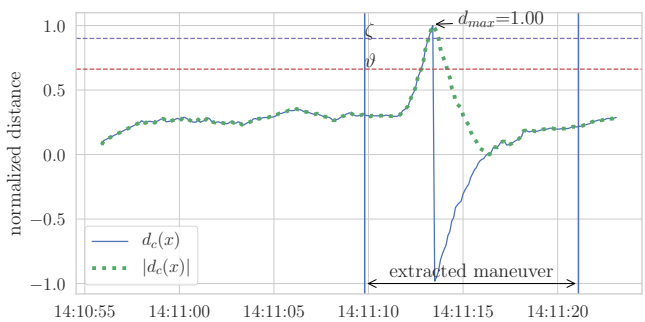

Fig. 8: The absolute normalized distance is used to fine-tune the extracted maneuver interval.

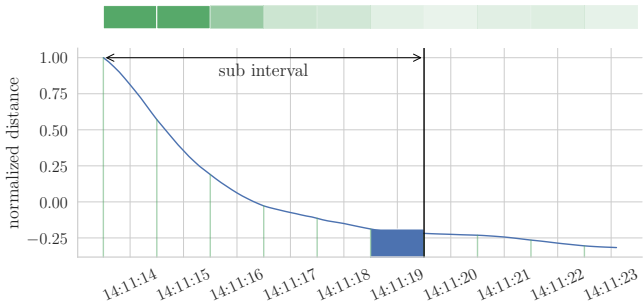

Fig. 9: The maneuver end is represented as the partition with the lowest average signal change (blue box). The mean signal change of each partition is depicted with the green boxes.

and $w_r = \{t_r + t_{o,r}, \ldots, t_{|w|}\}$ with $t_{o,l}, t_{o,r}$ as the offsets in both sub-windows likewise determined by (5) and (6) with the threshold ϑ (see Fig. 8). Since it is assumed that the interval contains a lane-change maneuver and that lane-changes will always end after the vehicle crosses the lane marking, ϑ is the mean of the cluster in the HMM represented by the state *Crossing*.

To find the start and end of the maneuver, each sub-window $w \in \{w_l, w_r\}$ is partitioned into $m_w = \left\lfloor \frac{|w|}{k} \right\rfloor$ bins $B_w = \{b_w^1, b_w^2, \ldots, b_w^m\}$ with $|w|$ as the size of the sub-window w and $k = \max([3, t_o])$ as at least three or the previously determined offset. The latter ensures, that partitions have a valid size for the follow up steps. Also note that before partitioning, a convolution is applied to the signal with a box blur kernel of size 7 for smoothing. Furthermore, the signal is normalized, so that the first value is one. For each bin $b \in B_w$ of the sub-window w the average signal derivation is determined by $\bar{b} = \frac{1}{k}\sum_{j=ik+1}^{(i+1)k} f'(w^j)$ with $f'(t)$ as the derivation of $f(t)$, which is the smoothed normalized distance d_c of the window w and w^j as the jth time in the sub-window w since we are interested in the partition with the lowest average signal change. The process is illustrated exemplarily for the right sub-window w_r in Fig. 9 with the green vertical lines denoting the partition borders. The mean signal change for each partition is illustrated with the color-coded boxes at the top of the figure. The brighter the box, the lower the average signal derivation.

The last step is to find the ith bin with the lowest average signal change in both windows $i_{l,r}$

$$i_l = \underset{j \in \{1,\ldots,k_l\}}{\arg\min}\ \bar{b}_{l,j} \qquad i_r = \underset{j \in \{1,\ldots,k_r\}}{\arg\min}\ \bar{b}_{r,j} \tag{7}$$

to finally estimate the maneuver interval (t_s, t_e) by

$$\begin{aligned} t_s &= t_{max} - t_l - t_{o,l} - (i_l + 1)k_l \\ t_e &= t_{max} + t_r + t_{o,r} + (i_r + 1)k_r \end{aligned} \tag{8}$$

depicted as the horizontal arrow in Fig. 8. The images of the left front-facing camera for the estimated start (left), maximum normalized distance (middle) and estimated end (right) are depicted in Fig. 10.

Fig. 10: Images of the front-facing camera for the start (left), the point in time with the maximum normalized distance (middle) and the end of the extracted lane-change maneuver.

III. Evaluation

For the evaluation of the proposed lane-change identification framework, a test drive on a motorway with a duration of approximately 2.5 hours was chosen. The data was collected as part of the research project FASva [20]. During the drive, 205 lane-changes took place with 101 lane-changes to the left and 104 to the right. The lane-changes were labeled manually by inspecting the images of the vehicle's front camera. The start of a lane-change is defined as that moment where the vehicle's heading is changing towards the target lane or, in case of multiple consecutive lane-changes, the vehicle position is near the lane center. The maneuver finishes if the vehicle is on the target lane and the vehicle's heading is recovered to that lane or, in case of multiple consecutive lane-changes, it reaches the center of the lane.

Since the SVM and ANN are widely used in the literature for lane-change identification, they are selected for comparison. To also respect the temporal dependencies in the classification, the driving sequence X with the same features as used in Section II-C is partitioned into n windows $w_0, \ldots, w_n$ according to the window size Δ_w so that $n = |X| - \Delta_w + 1$ with $|X|$ as the number of measurements in the sequence. That is, a window of size Δ_w is shifted over the signal (with a shift length of one and without padding) and each window represents the feature vector of the window's center with the features d_c and $crossing$. The predicted label $l_i \in \{-1, 0, 1\}$ is either $l_i = 1$ for a left, $l_i = -1$ for a right or $l_i = 0$ for no lane-change. Since the performance of both approaches is related to the proper selection of parameters, grid-search is employed to find the best parameter combination (see Table I). The maneuver intervals are extracted based on series of equal labels with the first label denoting the start and last one the end time.

TABLE I: The model parameters found using grid-search for the SVM and ANN with a subset of the evaluation dataset.

SVM		ANN	
Window size	39	Window size	35
Kernel	rbf	Hidden Layers	3
Gamma	0.001	Neurons per Layer	10
C	100	Activation function	tanh
-	-	Alpha	1.0

A. Metrics

To qualitatively evaluate the performance of the presented approach, lane-change classification is treated as a binary classification problem. For that purpose, the approach used in a previous work [21] is adopted to state whether a lane-change was correctly classified or not. Let $\mathbf{t}_{c,i} = [t_c^s, t_c^e]$ denote the ith interval represented as start and end time of the lane-change identified by a classifier c with $c \in \{\texttt{ANN}, \texttt{SVM}, \texttt{HMM+DTW}, \texttt{HMM+DTW-Ex}\}$ and $\mathbf{t}_g = [t_g^s, t_g^e]$ the manually identified start and end times. HMM+DTW represent the intervals using only the primitive labels and HMM+DTW−Ex the extended interval extraction method from Section II-E.

A maneuver $\mathbf{t}_{c,i}$ is correctly identified, a *true positive* (tp), if both interval overlaps and the difference between the ground truth and extracted interval is smaller than a threshold Δt. If the time difference is, however, greater than Δt or do not overlap, the maneuver is a *false positive* (fp). Furthermore, all maneuvers that were not found by a classifier are *false negatives* (fn). Based on these metrics, the *precision* and *recall* are estimated

$$\textit{precision} = \frac{\texttt{tp}}{\texttt{tp} + \texttt{fp}} \qquad \textit{recall} = \frac{\texttt{tp}}{\texttt{tp} + \texttt{fn}} \tag{9}$$

for each classifier and combined using the harmonic mean (F_1 score) given with

$$F_1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \tag{10}$$

for the final performance evaluation.

B. Results

The classification performance is evaluated w.r.t the deviation of the extracted interval from the ground-truth interval since it affects the classification precision as described in the previous section. In Fig. 11, the F_1 score for each classifier and different deviations Δt is depicted.

Contrary to our expectations, the figure shows that the MLP and SVM lack precision compared to the proposed HMM-based approaches for all Δt with a maximum F_1 score of 38.66% and 37.35%, respectively. Possible reasons for this discrepancy might be that maneuvers are extracted based on series with equal labels. Hence, the interval is created incorrectly if there is only one misclassification in the series. Additionally, the features used by the SVM and ANN are the same as by the HMM. Although the feature set seems to be well-designed for the proposed approach, this may not be the case for the SVM and ANN.

The proposed approach with the interval extraction method HMM+DTW−Ex as described in Section II-E is superior to the HMM+DTW method, although both approaches tend to the same maximum F_1 score of 98.01% with an increasing Δt. This is consistent with our expectation since the HMM+DTW uses the results after clustering with DHC as shown in Fig. 6 and tend to shorten the interval compared to HMM+DTW−Ex. This is the reason for the shift between the results in Fig. 11 and that both approaches reaches the same maximum F_1 score with an increasing Δt.

Concluding, the results show that the proposed framework is able to identify lane-change with high accuracy. Our dataset was, however, limited to find lane-changes of the ego vehicle since only the signals of the next left and right

marking relative to the ego-vehicle are available. Hence, future studies will have to explore if the framework can be employed for other datasets with multiple vehicles as well. Additionally, a more detailed analysis of the results is part of future work.

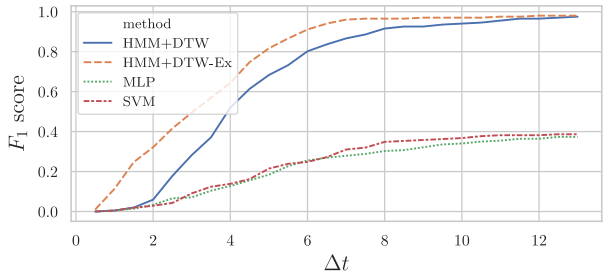

Fig. 11: The classification performance of the proposed approach compared with two state-of-the-art approaches for different maximum maneuver interval deviations Δt.

IV. Conclusion

For the validation of automated driving functions, a scenario-driven approach is a widely accepted method. Therefore, solutions are required to find specific scenarios in real-world drivings. Due to the adverse impact of lane changes on the overall traffic, this work proposes a multi-level framework to identify lane changes on motorways providing information for follow-up analysis. Therefore, a drive is partitioned into driving primitives which are the basis for the maneuver identification using a set of unsupervised learning methods and DTW for lane-change classification.

The proposed approach is evaluated using a test drive on a motorway with 204 lane-changes, and the results are compared with two baseline methods showing the efficacy and superiority of the proposed framework with an identification F_1 score of $98.01\,\%$.

In follow-up works, the scalability of the approach and adaptation to other data sources is verified. Therefore, information of the Testfeld Niedersachsen [22] will be employed to find lane changes for multiple and different types of vehicles with the proposed approach and to assess the criticality of the maneuvers by relating it to other vehicles. Furthermore, the extracted maneuver intervals of lane-changes will be analyzed in future studies to derive parameter distributions that can be employed to model lane-changes on motorways for, e.g., simulation-based testing.

References

[1] H. Winner, K. Lemmer, T. Form, and J. Mazzega, *PEGASUS—First Steps for the Safe Introduction of Automated Driving*. Switzerland: Springer International Publishing, 2018, ch. Vehicle Systems and Technologies Development, pp. 185–195.

[2] W. Damm, E. Möhlmann, T. Peikenkamp, and A. Rakow, *A Formal Semantics for Traffic Sequence Charts*. Cham: Springer International Publishing, 2018, pp. 182–205.

[3] C. Sippl, F. Bock, B. Huber, A. Djanatliev, and R. German, "Identifying relevant traffic situations based on human decision making," in *19. Internationales Stuttgarter Symposium*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 1003–1017.

[4] A. Erdogan, B. Ugranli, E. Adali, A. Sentas, E. Mungan, E. Kaplan, and A. Leitner, "Real-World Maneuver Extraction for Autonomous Vehicle Validation: A Comparative Study," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, June 2019, pp. 267–272.

[5] X. Peng, R. Liu, Y. L. Murphey, S. Stent, and Y. Li, "Driving Maneuver Detection via Sequence Learning from Vehicle Signals and Video Images," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug 2018, pp. 1265–1270.

[6] F. Kruber, J. Wurst, and M. Botsch, "An Unsupervised Random Forest Clustering Technique for Automatic Traffic Scenario Categorization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 2811–2818.

[7] C. Roesener, F. Fahrenkrog, A. Uhlig, and L. Eckstein, "A Scenario-Based Assessment Approach for Automated Driving by Using Time Series Classification of Human-Driving Behaviour," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1360–1365.

[8] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based Scene Creation for the Development of Automated Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 1813–1820.

[9] Yiguang Xuan and B. Coifman, "Lane Change Maneuver Detection from Probe Vehicle DGPS Data," in *2006 IEEE Intelligent Transportation Systems Conference*, Sep. 2006, pp. 624–629.

[10] H. Zheng, J. Zhou, Q. Shao, and Y. Wang, "Investigation of a Longitudinal and Lateral Lane-Changing Motion Planning Model for Intelligent Vehicles in Dynamical Driving Environments," *IEEE Access*, vol. 7, pp. 44 783–44 802, 2019.

[11] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, March 2017.

[12] P. Kumar, M. Perrollaz, S. Lefvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 797–802.

[13] R. Izquierdo, I. Parra, J. Muoz-Bulnes, D. Fernndez-Llorca, and M. A. Sotelo, "Vehicle trajectory and lane change prediction using ANN and SVM classifiers," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[14] S. Ramyar, A. Homaifar, A. Karimoddini, and E. Tunstel, "Identification of anomalies in lane change behavior using one-class SVM," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2016, pp. 004 405–004 410.

[15] N. Monot, X. Moreau, A. Benine-Neto, A. Rizzo, and F. Aioun, "Comparison of rule-based and machine learning methods for lane change detection," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 198–203.

[16] W. Wang, J. Xi, and D. Zhao, "Driving Style Analysis Using Primitive Driving Patterns With Bayesian Nonparametric Approaches," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2986–2998, Aug 2019.

[17] J. A. Bilmes *et al.*, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, February 1978.

[19] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[20] L. Klitzke, C. Koch, A. Haja, and F. Köster, "Real-world Test Drive Vehicle Data Management System for Validation of Automated Driving Systems," in *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 2019, pp. 171–180.

[21] L. Klitzke, J. Meyer, T. Leune, C. Koch, and F. Köster, "DAGMaR: A DAG-based Robust Road Membership Estimation Framework for Scenario Mining," in *The Second International Workshop on Intelligent Transportation and Connected Vehicles Technologies (ITCVT 2019)*, Granada, Spain, Oct 2019, pp. 358–365.

[22] F. Köster, J. Mazzega, and S. Knake-Langhorst, "Automatisierte und vernetzte Systeme Effizient erprobt und evaluiert," *ATZextra*, vol. 23, no. 5, pp. 26–29, 2018.