EPiC Series in Computing

Volume 102, 2024, Pages 20–32

Proceedings of Conference on Cognitive and Computational Aspects of Situation Management 2023

A modified Hidden Markov Model (HMM)-based state machine model for driving behavior recognition: Effectiveness of features using different sub-HMMs

Ruth David[1] and Dirk Söffker[2]

[1] University of Duisburg-Essen, Duisburg, Germany
ruth.david@uni-due.de

[2] University of Duisburg-Essen, Duisburg, Germany
soeffker@uni-due.de

Abstract

Driving behavior estimations play a significant role in the development of Advanced Driving Assistance Systems (ADASs). The estimations are often developed using machine learning-based approaches, which are influenced by different factors, such as input variables and design of methods. However, developing a suitable configuration can be complicated. In this contribution, an improved Hidden Markov Model (HMM)-based state machine model is introduced for the recognition of lane changing behaviors. Adapting a previously developed HMM model, the model consists of different sub-HMMs which are fused to develop the HMM estimations. A prefilter is introduced in the HMM to quantize the input variables into segments of observed sequences that distinguish different driving situations. Hence, optimization of the prefilter is performed. Different from the previous work, a state machine model is incorporated to develop the final behavior estimation using the estimations of the HMM model. To evaluate the estimation effectiveness, different driving features (inputs) are evaluated by using different combinations of sub-HMMs. Experimental driving data based on six drivers used for the application of the method show that the approach generates adequate accuracy (ACC), detection rates (DR), and false alarm rates (FAR).

1 Introduction

In recent years, the advancement of driving assistance systems have grown significantly to improve the driving quality. These driving assistant systems are not only able to estimate driving behaviors or trigger warnings to maneuver safely, but can autonomously control the car. An example is the Tesla Autopilot that includes automatic lane change, lane centering, and adaptive cruise control. Machine Learning (ML) methods are particularly useful to develop behavior estimation models for these systems, as traditional methods are inadequate due to the complexity of the data. In addition, ML algorithms are trained based on historical data/experience to generate estimations. Several factors that affect the ability of the ML models to develop accurate estimations are the design of the models, selection of the driving features as inputs, model parameters, and hyperparameters. To improve the performance of the model in terms of design, combining two or more ML approaches is commonly done when a single approach has limitations. This enables each approach to perform specific tasks. For an example, Long short-term memory network (LSTM) and Convolutional neural network (CNN) are combined in [12] to identify abnormal driving behaviors. Driving data are segmented based on human experience and abnormal acceleration points to develop different training samples. The LSTM is used to extract features from the training samples, while the CNN is used to extract features again based on the output of LSTM. In addition, modifying design of conventional methods can improve the performance, such as HMM-derived models [9]. In [9], a driving estimation model with four different Hidden Markov Models (HMMs) representing four different scenarios are trained: driving straight, turning left, right, and intersection stop. Driving behaviors are influenced by driving features, such as environmental variables [7]. Environmental variables describe the relationship between the environmental conditions and ego vehicle [7]. Thus, selecting appropriate features as model inputs is important for estimations and can be challenging. Filters or Wrappers methods are widely used to select the relevant input features [14], [18]. Unlike Filters and Wrappers method, deep learning methods are able to automatically select the appropriate features during the learning process [10]. In [10], spatial and temporal features are automatically selected without manual extraction. Prefilters have been applied to features in recent years as part of the HMM [6], [8] to quantize input variables to observation sequence with specific information/ features. As driving variables changes with time, the prefilter is applied to simplify model with accurate features. The feature vector developed using the prefilter can be used to distinguish various driving situations. As model parameters and hyperparameters also affect the estimation performance, optimization of these values is important. Model parameters affect the estimation model directly and are usually optimized during the training process, while hyperparameters control the training process and are not part of the model. Nevertheless, hyperparameters can influence the selection of model parameters, which affect the performance.

As the HMM and state machine models perform well for the estimation of lane changing behaviors shown in [6], [8], and [1], this study combines both approaches to estimate lane changing behaviors (lane change to the the right (LCR), lane keeping (LK), and lane change to the left (LCL)). The state machine consists of three discrete states (each representing a lane changing behavior) that switches between each other or remain in the same state to estimate the driving behaviors [1]. The transitions are based on conditions defined by the estimation of an improved HMM [6]. Similar to a previously developed HMM-based state machine model in [3], the improved HMM model employed here considers the prefilter application on the input variables, but differs from [3] as the HMM consists of four sub-HMMs with different inputs (based on [6]). The estimation of the HMM model is developed based on the fused probabilities of the sub-HMMs. Optimization of prefilters and hyperparameters are also performed as part of the training process. Different from [6], various combinations of sub-HMMs are fused to determine the suitable input combinations (features).

The remainder of this paper is structured as follows: The methodology of the state machine approach and the improved HMM is given in Section 2. The proposed approach is introduced in Section 3. Furthermore, the model parameter and hyperparameter optimization are detailed in this section as well. Feature selection based on different combinations of sub-HMMs is presented in Section 3. In Section 4, the application of the method is given, while the evaluations of the results are discussed in Section 5. Finally, a conclusion is given in Section 6.

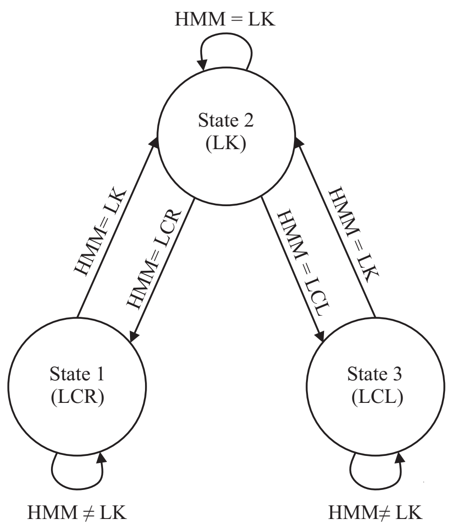

Figure 1: HMM-based state machine model [3]

2 Methodology

In this section, the state machine model and the improved HMM are presented in detail. The prefilter application and the development of the different sub-HMMs are explained as well.

2.1 State Machine Approach

The state machine approach is utilized to describe the multi-switching behavior of a system/application using discrete states [11]. Transition between states or remaining in the same state relies on the conditions attached with the transitions and on the model's input variables. A previously developed state machine model is employed here [1], [3]. Three states are defined in this model, each representing an estimated lane changing behavior: LCR (State 1), LK (State 2), and LCL (State 3) (Fig. 1). Assuming the current estimated state is LK, the model can either switch to LCR or LCL for the next estimation if the transition conditions are met. If the current estimated is LCR or LCL, the only possible transition is to state LK (Fig. 1) [3]. The transition conditions are defined by the estimations of the improved HMM model. Advantages of this method include its non-complex formulations and easy state reachability [13].

2.2 Improved Hidden Markov Model

A standard HMM model (Fig. 2) is described using the probabilistic relationship between the observation sequence (related to the inputs, $V = \{V_1, V_2, ..., V_M\}$) and hidden states (outputs, $S = \{S_1, S_2, ..., S_N\}$), such that M and N are the number of observation variables and hidden states, respectively. In this model, the hidden states are the lane changing maneuvers ($N = 3$). The hidden states are realized using the observation sequence based on the expectation maximization (EM) and maximum likelihood estimation (MLE), which are used to develop the HMM parameter. The HMM parameter consist of different probabilities defined by $\lambda = (A, B, \pi)$. The transition probability ($A = a_{ij}, i, j \in [1, N]$) is the probability of switching from one hidden state to another, observation likelihood ($B = b_{li}, l \in [1, M]$) is the probability an observation is generated from a specific hidden state, while the initial probability π is the probability of starting at a specific hidden state.

To train the HMM, the Baum-Welch algorithm [16] is used to estimate λ that best fits a given observation sequence and the corresponding hidden state sequence. The Viterbi algorithm [16] then utilizes the λ to select the most possible hidden state sequence (lane changing behaviors estimations), based on the hidden state with the highest probability.

Often, standard HMM may not be able to interpret the data well resulting in poor estimation performance, particularly when data are not precise or highly dynamic. Dynamic data that changes with time, changes the observation variables. Hence, a prefilter with thresholds is applied to quantize the data variables (inputs) to develop feature vectors for a better interpretability, as in [6] and [8]. The prefilter divides the input variables into several segments, such that each segment represents an observation with specific information related to the driving data. Prefilter thresholds (ranges of the segments) are defined to develop the observation sequence. In this contribution, a prefilter with five thresholds is applied to each input variable, dividing the variable in to six segments.

Higher number of input variables increases the number of segments and observation variables, which heightens the complexity of the observation matrix B [6]. Hence, the process is computationally expensive, in terms of training time. To simplify the model, four sub-HMM are developed, such that each sub-HMM is given different inputs: HMM 1 (time to collision (TTC) to vehicles in different directions), HMM 2 (distances), HMM 3 (velocities), and HMM 4 (driving operational) (Table 1) [6]. A point to note is that the prefilter is applied to all the driving variables mentioned, except for the gearbox, current lane, and indicator. These variables are considered on its own as observation variables without adjustments. To obtain the HMM's final estimation, the probabilities of different sub-HMM models are fused using weights to calculate the final probability [6] as

$$P = \sum_{k=1,2,3,\ or\ 4}^{1,2,3,and/or4} w_k \times P_k, \tag{1}$$

whereby, k is the sub-HMM, P is final probability of the HMM, w_k is the weight associated with a specific sub-HMM, and P_k is the probability of a sub-HMM. Here, the hidden state with the highest final probability is selected as the estimated lane changing behavior of the HMM. Different combinations of sub-HMMs are fused to evaluate the effectiveness of features on the performance, which is detailed in the next section.

3 HMM-Based State Machine model

The proposed model which combines the improved HMM and state machine is detailed in this section. In addition, model parameters and hyperparameters optimization are discussed. For evaluation of feature selection, different combination of sub-HMMs are applied.

3.1 Proposed Approach

Based on Fig. 1, a transition from LK to LCR or LCL is realized when the estimation of HMM is LCR or LCL at that time point. Switching from LCR or LCL to LK occurs, if the HMM estimation is LK [3].

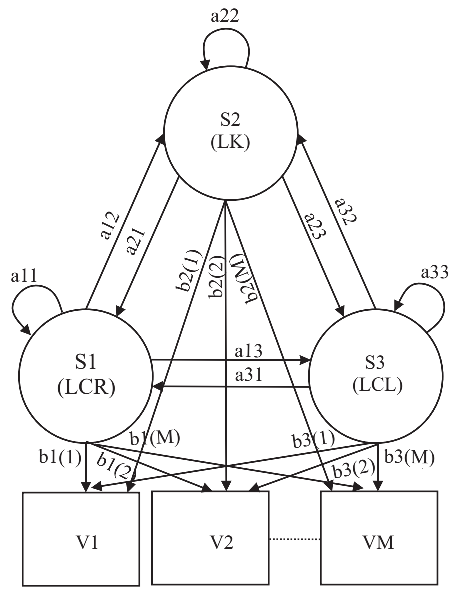

Figure 2: HMM model [3]

3.2 Parameter and Hyperparameter Optimization

Parameters of the model are the prefilter threshold values, while the hyperparameters are the weights associated with sub-HMMs. The weights are considered hyperparameters as the values represent the impact of each sub-HMMs. The threshold values are required to define the observation sequence, ultimately affecting the model's performance. Hence, optimization of these values are necessary. The threshold values and weights are determined automatically during the optimization, however it can be challenging depending on the problem type and the optimization technique used. For an example, Particle swarm optimization (PSO) has a tendency to get a stuck in the local optimum solution making it difficult to find the global optimum solution [17]. In addition, it is not suitable for multi-objective problems, as the one presented here. In contrast, the Non-Dominated Sorting Genetic Algorithm (NSGA-II) can handle multi-objective problems well [4]. The NSGA-II is also well applied due to its ability to find non-dominated solutions (no solution shows performance dominance over the other for all objective functions), elitism (preserving the best solutions from previous iterations), and fast convergence (reducing the number of iterations) [4]. Hence, the NSGA-II is used for the optimization during training.

The model is evaluated using accuracy (ACC), detection rate (DR), and false alarm rate (FAR) [15], which is common in driving behavior prediction and recognition. Hence, the prefilter thresholds and weights are selected by NSGA-II, such that the model generates high ACC, DR, and low FAR. Suitable objective functions are minimized during the optimization process [1], given by

$$f_1 = (1 - DR_{right}) + FAR_{right}, \tag{2}$$
$$f_2 = (1 - DR_{keep}) + FAR_{keep}, \text{ and} \tag{3}$$
$$f_3 = (1 - DR_{left}) + FAR_{left}, \tag{4}$$

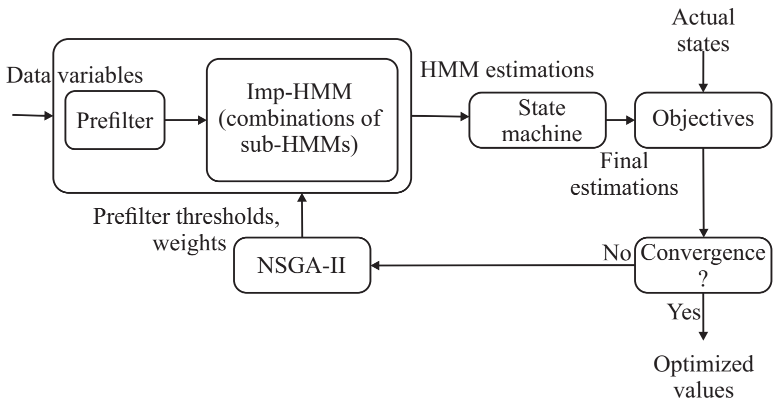

Figure 3: Optimization procedure [6]

whereby each objective function represents a specific driving maneuver. The common objective function used in literature is a loss function (weighted comparisons between actual and estimated behaviors). However, the usual objective function does not consider true positive, true negative, false positive and false negative of the different maneuvers, which are used to define the ACC, DR, and FAR. As the optimal parameters will be selected on the basis of ACC, DR, and FAR during training, the aforementioned objective functions are used. The optimized training procedure is illustrated in Fig. 3.

3.3 Feature Selection

As previously stated, four sub-HMMs are defined with different inputs (Table 1) [6]. The HMM model estimations are calculated using the fused probability of the different sub-HMMs. A total of eleven combinations of sub-HMMs are evaluated to examine the effects of the different features, given in Table 2.

Sub-HMM models	Input variables
HMM 1	TTC to vehicle in the front (f), back(b), front left (fl), front right(fr), back right(br), back left (bl)
HMM 2	Distances to the vehicle in f , b, fl, fr, br, bl
HMM 3	Velocities of the ego vehicle, vehicle in f , b, fl, fr, br, bl
HMM 4	Driving operational variables: Ego vehicle's steering wheel angle accelerator pedal position, brake pedal position, heading angle, gearbox, indicator, current lane

Table 1: Input variables for the four sub-HMMs

HMM models	Combination of sub-HMM models
HMM I	1, 2, 3, 4
HMM II	1, 2, 3
HMM III	1, 2, 4
HMM IV	1, 3, 4
HMM V	2, 3, 4
HMM VI	1, 2
HMM VII	1, 3
HMM VIII	1, 4
HMM IX	2, 3
HMM X	2, 4
HMM XI	3, 4

Table 2: Combination of different sub-HMM models

Figure 4: Driving simulator at Chair of Dynamics and Control, Uni-DuE

4 Application of the New Method

In this section, the experimental design and data processing are explained. A description on the definition of the lane change is illustrated as part of the data processing. In addition, the training and test procedures are described.

4.1 Experimental Design and Data Processing

Experimental data are collected based on an experiment conducted using a driving simulator ($SCANeR^{TM}$) in the Chair of Dynamics and Control (University Duisburg-Essen, Germany) which mimics a driving environment (Fig. 4). The driving environment consist of a four lane highway in two directions. A total of six drivers' data are used, whereby each driver performed a 40-minute drive to collect the training data and another 10 minute drive for the test data [5]. The driver can perform various maneuvers such as driving straight and overtaking vehicles ahead. Overtaking is performed from the left based on the German traffic rules. The lane change duration is the time interval between the t_{angle} (time point of last significant change in steering wheel angle) and t_{lane} (time point of lane change) as illustrated in Fig. 5 [5]. All drivers were in a sober state (not drunk, not delusional, etc) and the simulation has other vehicles to mimic an actual driving environment (not congested).

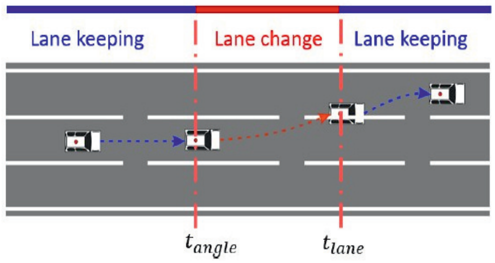

Figure 5: A lane change illustration [2]

4.2 Training and Test

Optimal prefilter thresholds and weights are developed during the training. Using the optimal values, the model is tested.

4.2.1 Training

The steps for training are given as follows:

1. Input variables based on the training data and actual lane changing behaviors are given to the model.

2. The prefilter thresholds of the input variables for the sub-HMMs are optimized using NSGA-II to develop the observation sequences. Using the TTC features as an example, five threshold values are generated automatically by NSGA-II for each feature. Depending on the current TTC value and threshold values, it is assigned to one of the six segments (as the threshold values divide the feature into six segments). Based on the selected segment of the different TTC variables, the observation sequences are calculated (as input sequences) for the TTC sub-HMM.

3. Using the observation sequences, the sub-HMMs are trained to develop the optimal HMM parameter λ, for each sub-HMM.

4. Using the HMM parameters, the probability for each of the sub-HMMs are calculated. In addition, the weights are optimized as well. The probabilities are fused using the optimal weights to develop the final probability for each state.

5. The hidden state with the highest probability is selected as the final HMM estimation.

6. Depending on the current estimated state, the state machine generates the final lane changing estimation of the model based on the HMM estimation. The ACC, DR, and FAR are evaluated based on the comparisons between the actual and estimated behaviors. These metrics are used to develop the objective functions.

7. Steps (1) to (5) are repeated until convergence of the optimizer and iteration limits are reached (generation size: 100, population size: 20).

4.2.2 Test

1. The trained model with the optimized prefilter thresholds and weights are used for the recognition of lane changing behaviors based on the test data sets .

2. The estimated and actual behaviors are compared using ACC, DR, and FAR to validate the model's performance.

5 Evaluation of Results

In this section, the evaluation of the state machine-based HMM model is presented. To that verify the sub-HMM combinations develop effective performances for the proposed model, the actual and estimated behaviors are compared in terms of ACC, DR, and FAR. The average performance based on six drivers using different sub-HMM combinations (in the proposed approach) is presented in Table 3.

States	**Metrics**	**Sub-HMM combinations (in the proposed approach)**										
		I [%]	**II** [%]	**III** [%]	**IV** [%]	**V** [%]	**VI** [%]	**VII** [%]	**VIII** [%]	**IX** [%]	**X** [%]	**XI** [%]
Overall	ACC	79.14	79.79	71.15	76.36	83.46	55.34	74.68	72.43	73.51	72.03	80.65
Right	ACC	91.35	93.74	87.47	92.23	93.13	54.12	89.22	89.06	90.97	91.10	92.52
	DR	44.48	58.85	72.04	48.61	34.65	49.88	59.77	83.96	41.11	65.20	66.35
	FAR	5.20	4.87	11.39	6.81	3.16	46.06	8.74	10.03	6.34	7.35	5.80
Keep	ACC	80.51	80.76	72.10	77.41	85.81	52.01	76.02	73.51	75.03	72.39	81.63
	DR	83.46	81.55	71.04	75.95	88.01	48.77	77.29	72.09	75.33	72.06	81.68
	FAR	50.10	43.50	26.21	49.15	47.34	44.25	39.65	18.59	42.44	25.02	31.07
Left	ACC	86.42	85.09	82.72	83.07	87.98	49.36	84.12	82.29	81.03	79.67	87.14
	DR	43.30	43.64	63.57	51.43	47.06	35.82	45.83	65.97	55.16	65.99	61.56
	FAR	11.53	13.06	16.56	16.85	10.29	22.41	13.99	17.01	18.11	19.80	12.04

Table 3: Average metric values of different models based on six test data sets

The values in green indicate the best performing values of the metrics for a particular state, while the values in red indicate the worst. The following statements can be made from the evaluations:

1. The model with HMM V's sub-HMM combinations outperforms other models in most metrics (highest number of green values), while the model with HMM VI has the worst performance in most metrics (highest number of red values). Nevertheless, it cannot be concluded that the combinations of HMM V generated the best results. This is because the DR_{right}, FAR_{keep}, and DR_{left} show rather low performances.

2. On the other hand, HMM III, VIII, X, and XI generated balanced performances throughout the metrics in contrast to the rest. For an example, the DR_{right} and DR_{left} are higher than 60 %, while the FAR_{keep} are lower than 35 %. Overall, poor performance values are not observed based on these sub-HMMs.

3. Sub-combinations HMM I, II, IV,VII, and IX do not generate a balanced performances throughout the metrics. Certain metrics tend to underperform, such as FAR_{keep} in HMM I.

4. All five HMMs (HMM III, V, VIII, X, and XI) consist of driving operational variables. Thus, this shows the usefulness of the driving operational on the performance.

To further verify the sub-HMM combinations in the HMM-based state machine model which are effective for the recognition of lane changing behaviors, comparisons between a conventional HMM (based on [6]) and the proposed approach are performed using different sub-HMM combinations. The comparisons are based on HMM III, V, VIII, X, and XI only, as these sub-HMM combinations in the models developed estimations closest to the actual behavior. The conventional HMM uses default weights and prefilter threshold values, instead of optimized values. The average performance based on six drivers are evaluated.

States	Metrics	Models	
		Conventional HMM [%]	Proposed approach [%]
Overall	ACC	79.23	71.15
Right	ACC	90.82	87.47
	DR	80.37	72.04
	FAR	8.61	11.39
Keep	ACC	79.40	72.10
	DR	80.91	71.04
	FAR	32.50	26.21
Left	ACC	88.24	82.72
	DR	51.58	63.57
	FAR	9.54	16.56

Table 4: Comparisons based on HMM III

States	Metrics	Models	
		Conventional HMM [%]	Proposed approach [%]
Overall	ACC	90.88	83.46
Right	ACC	96.08	93.13
	DR	60.53	34.65
	FAR	2.72	3.16
Keep	ACC	90.89	85.81
	DR	93.95	88.01
	FAR	46.22	47.34
Left	ACC	94.79	87.98
	DR	46.98	47.06
	FAR	3.03	10.29

Table 5: Comparisons based on HMM V

The conventional HMM outperforms the HMM-based state machine approach when HMM III and V combinations are used. On the other hand, the results based on HMM VIII, X, and XI show that the proposed approach outperforms the conventional HMM in most metrics. Thus, using the HMM VIII, X, and XI combinations show the effectiveness of the proposed approach as well as the relevance of the specific input features.

6 Conclusion

In this work, an improved HMM-based state machine model is developed for the recognition of lane changing behaviors. Three lane changing behaviors are considered, hence the state machine consists of three states. For the HMM model, a prefilter with five thresholds is applied to data variables to quantize each variable into segments. Each segment is a corresponding observation, thus the prefilter thresholds define observation sequence. A modified HMM with four sub-HMMs

States	Metrics	Models	
		Conventional HMM [%]	Proposed approach [%]
Overall	ACC	68.34	72.43
Right	ACC	86.10	89.06
	DR	95.65	83.96
	FAR	14.46	10.03
Keep	ACC	68.58	73.51
	DR	66.40	72.09
	FAR	16.12	18.59
Left	ACC	81.99	82.29
	DR	68.58	65.97
	FAR	17.34	17.01

Table 6: Comparisons based on HMM VIII

States	Metrics	Models	
		Conventional HMM [%]	Proposed approach [%]
Overall	ACC	70.85	72.03
Right	ACC	85.70	91.10
	DR	76.23	65.20
	FAR	13.85	7.35
Keep	ACC	70.88	73.29
	DR	71.07	72.06
	FAR	30.57	25.02
Left	ACC	85.13	79.67
	DR	60.80	65.99
	FAR	13.44	19.80

Table 7: Comparisons based on HMM X

is introduced, such that each sub-HMM is given different input variables (TTC, distances, velocities, and driving operational variables). The probabilities developed by different sub-HMMs are fused to develop the final probability. The hidden state with the highest probability is selected as the estimation of HMM. The state machine transitions between the states to estimate the final lane changing behaviors of the model using the HMM estimations as transition conditions. Different sub-HMM combinations are fused to determine feature combinations that are effective for the recognition. Based on the results, the HMM V combination outperforms other models in most metrics, however performs poorly in DR_{right}, FAR_{keep}, and DR_{left}. Further verification show that feature combinations of HMM VIII, X, and XI generated balanced performances and outperform the conventional HMM in most metrics. These models have driving operational variables in common, indicating the importance of these variables for the recognition. The results show the potential of this new method for estimation problems with specific input features. In future, using other environmental features and increasing the prefilter thresholds (instead of five) can be considered to improve the specifications of features/variables.

7 Acknowledgments

This research is partly supported by the German Academic Exchange Service (DAAD) scholarship received by first author for her Ph.D. study at the Chair of Dynamics and Control, University of Duisburg-Essen, Germany.

States	Metrics	Models	
		Conventional HMM [%]	Proposed approach [%]
Overall	ACC	74.79	80.65
Right	ACC	88.56	92.52
	DR	82.91	66.35
	FAR	11.46	5.80
Keep	ACC	74.92	81.63
	DR	73.86	81.68
	FAR	24.85	31.07
Left	ACC	86.10	87.14
	DR	64.98	61.56
	FAR	13.09	12.04

Table 8: Comparisons based on HMM XI

References

[1] Ruth David, Sandra Rothe, and Dirk Söffker. State machine approach for lane changing driving behavior recognition. *Automation*, 1(1):68–79, 2020.

[2] Ruth David, Sandra Rothe, and Dirk Söffker. Lane changing behavior recognition based on artificial neural network-based state machine approach. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3444–3449, 2021.

[3] Ruth David and Dirk Söffker. A study on a hmm-based state machine approach for lane changing behavior recognition. *IEEE Access*, 10:122954–122964, 2022.

[4] Kalyanmoy. Deb, Amrit. Pratap, Sameer. Agarwal, and Thirunavukarasu . Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[5] Qi Deng and Dirk Söffker. Improved driving behaviors prediction based on fuzzy logic-hidden markov model (fl-hmm). In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 2003–2008, 2018.

[6] Qi Deng and Dirk Söffker. Classifying human behaviors: Improving training of conventional algorithms. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1060–1065, 2019.

[7] Qi Deng, Jiao Wang, Kevin Hillebrand, Christoper Ragenold Benjamin, and Dirk Söffker. Prediction performance of lane changing behaviors: A study of combining environmental and eye-tracking data in a driving simulator. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3561–3570, 2020.

[8] Qi Deng, Jiao Wang, and Dirk Soffker. Prediction of human driver behaviors based on an improved hmm approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 2066–2071, 2018.

[9] Vijay Gadepally, Arda Kurt, Ashok Krishnamurthy, and Ümit Özgüner. Driver/vehicle state estimation and detection. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 582–587, 2011.

[10] Jun Gao, Yi Lu Murphey, and Honghui Zhu. Multivariate time series prediction of lane changing behavior using deep neural network. *Applied Intelligence*, 48:3523 – 3537, 2018.

[11] Arthur Gill. *Introduction to the Theory of Finite-state Machines*. Electronic science series. McGraw-Hill, 1962.

[12] Shuo Jia, Fei Hui, Shining Li, Xiangmo Zhao, and Asad J. Khattak. Long short-term memory and convolutional neural network for abnormal driving behaviour recognition. *IET Intelligent Transport Systems*, 14(5):306–312.

[13] Giovanni De Micheli, Robert K. Brayton, and Alberto L. Sangiovanni-Vincentelli. Optimal state

assignment for finite state machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 4:269–285, 1985.

[14] Werner Mostert, Katherine Mary Malan, and Andries Petrus Engelbrecht. Filter versus wrapper feature selection based on problem landscape features. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018.

[15] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2:37–63, 2011.

[16] Lawrence Rabiner and Biing Hwang Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[17] Wei Sun, Anping Lin, Hongshan Yu, Qiaokang Liang, and Guohua Wu. All-dimension neighborhood based particle swarm optimization with randomly selected neighbors. *Information Sciences*, 405:141–156, 2017.

[18] Yu Zhang, Wangkai Jin, Zeyu Xiong, Zhihao Li, Yuyang Liu, and Xiangjun Peng. Demystifying interactions between driving behaviors and styles through self-clustering algorithms. In *HCI in Mobility, Transport, and Automotive Systems*, pages 335–350. Springer International Publishing, 2021.