

Received 31 October 2022, accepted 16 November 2022, date of publication 21 November 2022, date of current version 29 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3224012

RESEARCH ARTICLE

A Study on a HMM-Based State Machine Approach for Lane Changing Behavior Recognition

RUTH DAVID¹ AND DIRK SÖFFKER¹, (Member, IEEE)

Chair of Dynamics and Control, University of Duisburg-Essen, 47057 Duisburg, Germany

Corresponding author: Ruth David (ruth.david@uni-due.de)

The work of Ruth David was supported by the Deutscher Akademischer Austauschdienst (DAAD) (German Academic Exchange Service) through the Chair of Dynamics and Control, University of Duisburg-Essen, Germany.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Chair of Dynamics and Control, University of Duisburg-Essen.

ABSTRACT In recent years, the development of advanced driving assistance systems (ADAS) has grown significantly within the transportation industry to assist drivers for making safe maneuvers. A major component in developing these assistance systems are driving behavior prediction and recognition models. These models aim to infer driving behaviors based on different sources and parameters using complex mathematical models. Machine learning algorithms are being used increasingly to develop these models. In this contribution, two formerly developed trainable models, which are an improved Hidden Markov Model (HMM) and a state machine model, are combined for the recognition of three lane changing behaviors (lane change to the right (LCR), lane keeping (LK), and lane change to the left (LCL)). In the improved HMM, a prefilter is implemented on two sets of observation variables (input variables of HMM): one consisting of distances and velocity deviation, while the other consists of time to collision (TTC) variables. To develop an optimal model, thresholds of the prefilter are optimized using a Non-Dominated Sorting Genetic-Algorithm-II. The aim is to investigate if the proposed model is able to produce estimations with high accuracy (ACC), detection rates (DR), and low false alarm rates (FAR). In addition, the performance based on applying the prefilter on the two sets of variables are compared. Comparisons to an individual improved HMM and an ANN-based state machine approach are also addressed. The obtained results show that the application of prefilter on the TTC variables improves the estimation performance. Furthermore, the proposed approach outperforms other approaches.

INDEX TERMS Advanced driving assistance systems, hidden Markov models, state machine model, prefilter, lane changing behaviors recognition.

I. INTRODUCTION

Based on a recent preliminary report by the European Commission related to road accidents in 2021, an estimate of 19800 lives were lost due to traffic accidents in the EU [1]. Focusing on Germany, road accidents have cost around 2562 lives in 2021, according to the German Federal

Statistical Office (Destatis) [2]. While, this is a decrease of 5.8 % from the previous year, it is still a critical problem faced. Further research shows that majority of the deaths in country roads and motorways involve the passenger vehicles, while in built up areas, most victims are pedestrians [1]. A recent report by Destatis also shows that majority of these accidents are caused by human driving behaviors accounting for 71.8 % of the accidents in Germany in 2020 [2]. This is due to the driver's inability to predict the right action when

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano¹.

driving in a complex or different environment [2]. The development of Advanced Driving Assistance Systems (ADAS) plays an important role to tackle this issue for improvements to road and vehicle safety. The ADAS provide support and assistance for drivers to maneuver safely in different environments. Some of the well-known ADAS used are the adaptive cruise control (controls speed of the vehicle while maintaining safety distances) and the collision avoidance systems (warn and alert drivers to avoid collision). Driving behavior prediction and recognition models are important elements of ADAS. The incorporation of driving behaviors with ADAS allows early prediction of driving behaviors and dangerous situations. Driving behaviors are considered as individualized, thus classifying and predicting behaviors by an assistance system enables the related drivers to get better supervision while driving [3]. A system that is able to adapt to individual behaviors of different drivers is considered to be ideal not only for increasing safety, but for the improvement of functionality (development of smart vehicles) as well [3]. Machine Learning (ML)-based approaches are increasingly used for the development of prediction and recognition models as these approaches are able to learn from given behaviors to predict similar behaviors or the driver's intentions. Some of the well-known ML approaches used in research contributions are Artificial Neural Network (ANN) [4], Support Vector Machine (SVM) [5], and Hidden Markov Model (HMM) [6]. In [4], an ANN-based model is used to predict lane changing maneuvers based on different situations, while in [5], a SVM model is used to detect lane changing intentions. A lane changing prediction model is developed in [6] based on an improved HMM, which includes the use of a prefilter. Most of these contributions analyze lane changing behaviors because a significant proportion of accidents occur during a lane change [2].

However, several challenges exist related to the development of such models, like defining optimal parameters for optimal estimations. A reason for this is the lack of understanding on selecting relevant input variables and parameters to be optimized due to the black box nature of the ML-based approaches. Two methods are usually applied to solve this problem. The first method combines two or more ML-based approaches to develop the model as in [7], whereby ANN is combined with SVM to predict lane changing behaviors on a highway. The results based on this work show that the combined model produced better estimation results than the individual models. A combination of ANN and HMM to develop a prediction model is realized in [8], whereby the HMM is used to predict different driving behaviors such as emergency steering, normal cornering (ability to handle bends), and straight line driving. Based on the predictions, an ANN model is used to obtain specific steering wheel angles for the different behaviors. The second method is based on feature selection techniques. Features used to describe a particular driving situation play an important role in driving behavior predictions. Thus, methods such as filter and wrapper methods are used in [9] to select the most appropriate features as

input variables in the prediction and recognition models. The wrapper method considered in [9] employs the combination of SVM with a recursive feature elimination method to extract features [10], [11]. Different vehicle variables are combined to develop features using a prefilter in [6], as part of HMM for lane changing predictions. On the other hand, deep learning methods have been applied in recent years as in [12], for automatic extractions of temporal and spatial features, eliminating the need for manual extraction.

In this contribution, a trainable model is proposed by combining a state machine-based approach [13] and an improved HMM [6] for the recognition of lane changing behaviors. The lane changing behaviors considered are lane change to the right (LCR), lane keeping (LK), and lane change to the left (LCL). The state machine models the different lane changing behaviors as states, such that it describes the transition from one state to another based on specific conditions. These conditions are defined by the estimations of HMM. A state machine model as a new ML-based model is considered as it is more interpretable than a HMM model, while the HMM is known for its stochastic properties. To improve the performance of HMM, previous works [6] and [14] considered a prefilter application. The prefilter process variables to generate input features. Thus, in this work a prefilter is implemented to two sets of variables as part of the HMM. One set comprises of distance and velocity deviation variables, while the other set comprises of time to collision (TTC) variables. Similar to [6], this work also considers the optimization of the prefilter parameters using Non-Dominated Sorting Genetic-Algorithm-II (NSGA-II). Different from previously mentioned literature, a state machine-based model [13] is combined with HMM to formulate a new ML model for lane changing behavior recognition. The objective is to develop an effective model with improved performance in accuracy (ACC), detection rates (DR), and false alarm rates (FAR). Performance comparisons between the application of prefilter to the two sets of variables are realized as well for evaluations. The aim is also to test the generability of the model such that same parameter values can also be used to obtain optimal estimations for different drivers. In addition, comparisons between the proposed model, an individual improved HMM model [6], and a previously developed ANN-based state machine model [15] are performed to evaluate the effectiveness of the new model.

This paper is organized as follows: in Section II the methodology of the state machine model, HMM, and improved HMM are described. Thereafter, the HMM-based state machine model is introduced in Section III. The feature variables used and optimization process of design parameters are described in this section as well. In Section IV, the application of the method is described, which includes the experimental setup, data processing as well as training and test process. The experimental results are discussed in section V. Finally, a conclusion is summarized in section VI.

II. METHODOLOGY

A. STATE MACHINE MODEL

State machines are used to model behaviors using a discrete number of states. A typical state machine model can either remain in the same state or describe the transition to another state based on a set of inputs and conditions. In this contribution, the transition conditions and parameters of the model are defined by the designers. One of the benefits of using a state machine model is its easy designing process and flexibility [16]. Another advantage includes its easy state reachability as the states can be defined finitely [17]. Nevertheless, a state machine-based model as a ML model for behavior estimations has not been widely applied in research. By far, only a few research works have applied this model in various areas, such as in tribology experiments to develop a lifetime model based on acoustic emission data [18] and in driving behavior experiments to develop driving behavior recognition models [13], [15]. Typical state machine models are developed in [18] and [13] whereby, the conditions for state transitions are based on threshold limits of certain variables. For an example, in [13] different threshold conditions of driving variables (values of input variables are higher or lower than the threshold values) define the state transitions.

In this contribution, the state machine-based ML model structure developed in [13] for the recognition of lane changing behaviors is adapted. The three lane changing behaviors estimated are represented by three states. In the developed model, the states are the output of the model at a particular time point. Based on Fig. 1, if the current estimated state is LK, the next possible estimations are either switching to states LCR, LCL (estimating the driver performs a lane change maneuver) or remaining in the same state (estimating the driver remains in the same lane) depending on the transition conditions. On the other hand, if the current estimated state is LCR or LCL, the next possible estimations are remaining in the same state (driver performs further lane changes in the respective direction) or to switch to state LK (the lane change is over).

B. HIDDEN MARKOV MODEL

The HMM is a probabilistic graphical model used to represent behavioral changes. As shown in Fig. 2, Hidden Markov Models (HMMs) are sequential models describing the relationship between the observation sequence (inputs, $V = \{V_1, V_2, \dots, V_M\}$) and hidden state sequence (outputs, $S = \{S_1, S_2, \dots, S_N\}$), whereby M and N are the number of observations and hidden states respectively. The hidden states are the lane changing behaviors in the driving behavior recognition model, hence, $N = 3$. The observation sequence is used to realize the hidden state sequence based on expectation maximization (EM) and maximum likelihood estimation (MLE). The HMM model used for the recognition of the lane changing behaviors is based on [6] (Fig. 2). The model consists of interconnected nodes describing the probabilistic relationship between the nodes. Parameters of HMM

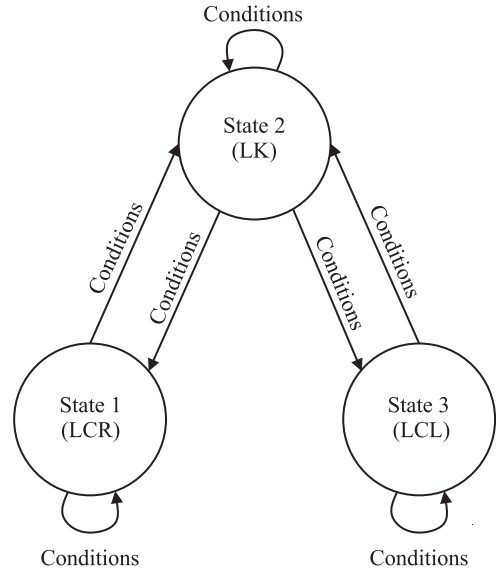


FIGURE 1. State machine model.

include transition probabilities, observation likelihoods, and an initial probability distribution. The transition probability ($A = a_{ij}, i, j \in [1, N]$) is the probability of switching from one hidden state, S_i to another, S_j . The observation likelihood ($B = b_{ki}, k \in [1, M]$) is the probability of an observation, V_k generated from a particular hidden state, S_i . The initial probability distribution, π_i defines the probability of the Markov chain starting in state S_i . Thus, the HMM model can be defined by the maximum likelihood parameter, $\lambda = (A, B, \pi)$.

To apply the HMM-based model for the recognition of the lane changing behaviors, the model is first trained with the Baum-Welch algorithm (a special case of EM) to estimate λ . Given an observation sequence and the possible hidden states sequence, λ is estimated through learning to best fit both sequences. Then, using the Viterbi algorithm the most probable lane changing behaviors sequence is estimated based on the saved parameters.

The transition probability of switching from S_i to S_j (a switch from one behavior to another) can be formulated as

$$a_{ij} = \frac{\text{Expected number of transitions from } S_i \text{ to } S_j}{\text{Expected number of transitions from } S_i} \tag{1}$$

The probability of being at S_i at time t and changing to S_j at time $t + 1$, given the observation sequence is defined as

$$\eta_t(i, j) = P(q_t = i, q_{t+1} = j | V, \lambda), \tag{2}$$

whereby, q_t is the state at time t . Hence, the expected number of transition from S_i to S_j is the the sum of η_t for all time steps, while the expected number of transitions from S_i is the sum of all transitions from S_i . Thus, the a_{ij} is formulated as

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \eta_t(i, j)}{\sum_{t=1}^{T-1} \sum_{h=1}^N \eta_t(i, h) \eta_t}, \tag{3}$$

whereby, T is the time length of the full drive.

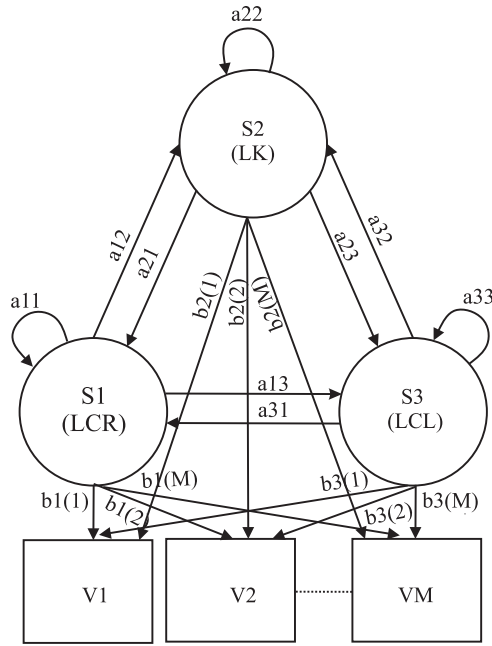


FIGURE 2. Hidden markov model for driving behavior estimations.

The observation likelihood of observation V_k generated by S_i is given by

$$b_{ki} = \frac{\text{Expected number of times in } S_i \text{ and observation } V_k}{\text{Expected number of times in } S_i} \quad (4)$$

Here, $\chi_t(i)$ is the probability of S_i at time t formulated as

$$\chi_t(i) = P(q_t = i | V, \lambda). \quad (5)$$

Thus, b_{ki} can be finally defined as

$$b_{ki} = \frac{\sum_{t=1}^T \sum_{s.t. O_t = V_k} \chi_t(i)}{\sum_{t=1}^T \sum_{h=1}^N \chi_t(i)}, \quad (6)$$

such that the sum of $\chi_t(i)$ for the entire time length of the drive for a given observation is divided by the sum of $\chi_t(i)$ for all time steps. Here, $\sum_{t=1}^T \sum_{s.t. O_t = V_k}$ is the sum for all t when O_t is V_k .

Based on the defined a and b , the HMM parameter is generated using the Baum-Welch algorithm. The next driving state is determined based on the saved HMM parameter and the given observation sequence by employing the Viterbi algorithm. Here, the maximum probability of all previous state sequences leading to the next state ($q_t(j)$) is considered to denote the most probable path at time t . So, $q_t(j)$ is formulated as

$$q_t(j) = \max_{i=1}^N q_{t-1}(i) a_{ij} b_j(O_t). \quad (7)$$

The HMM possesses certain benefits such as its ability to analyze time series data and its stochastic characteristics [19], [20]. Since upcoming behaviors are stochastic and only depend on the present state, the HMM is a suitable choice

for driving behavior estimation. The HMM can also handle temporal pattern recognition [20].

C. IMPROVED HMM WITH PREFILTER

Using a conventional HMM may have an impoverished performance if the features used are not accurate enough. Therefore for performance improvement, various approaches have been established such as a combination of HMM with other methods and HMM-derived methods [21]. The combination of HMM with other methods includes ANN-HMM [22], Fuzzy Logic (FL)-HMM [22], and Gaussian Mixture Model (GMM)-HMM [23]. These methods use results from one method as input to the other to determine the final behavior estimation. The other method is also used for determining the parameters and classifying different driving styles, behaviors, or situations. Thus, the combined methods consider the advantages from both the HMM and other methods to determine the final outcome. On the other hand, the HMM-derived methods such as Hierarchical HMM [24] and Bayesian Nonparametric HMM consider the time series property of HMM [25]. The general idea of HMM-derived methods include partitioning behaviors into several task layers. In these methods, the initial layer is used for determining different driving variables like acceleration, while the higher layer uses the results from the initial layer to estimate the corresponding driving behavior. A new and improved HMM-derived method developed in [6], which includes the application of a prefilter is utilized in this work. The prefilter is applied to the observation variables of HMM.

In the HMM model, the observations variables are dynamic which changes with time. Changes in the observation parameters changes the observation vector. To simplify the model, a prefilter is applied on the data of the observation variables to quantize the variables with a feature vector. The feature vector is used to determine different driving situations [6]. The prefilter divides the driving variables into segments with thresholds, such that each segment represents an observation. Thresholds are defined using optimization to develop the observations and ultimately the observation sequence.

III. HMM-BASED STATE MACHINE MODEL

A new HMM-based state machine model is introduced in this section. Here, two trainable systems, a state machine and an improved HMM are combined to develop a model that recognizes lane changing behaviors. The state machine model describes the transition between the states, while the estimations of an improved HMM define the transition conditions. The transition conditions differ from [13] which uses threshold conditions instead. The structure of this model is similar to the ANN-based state machine model [15], which uses the ANN estimations instead as the transition or remaining conditions. Driving decisions depend on environmental variables as well as individual driving behaviors. Hence, environmental variables describing the relationship between the ego vehicle and surrounding vehicles are selected as inputs.

TABLE 1. Transition conditions.

Transitions	Estimations of HMM
LK to LCR/LCL	LCR or LCL
LCR/LCL to LK	LK

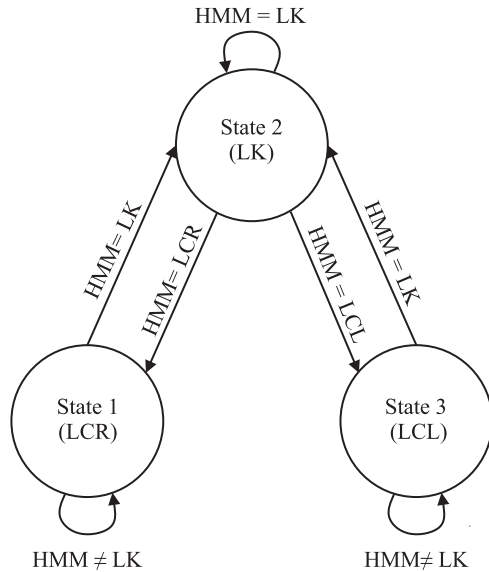


FIGURE 3. HMM-based state machine model.

A. HMM-BASED STATE MACHINE APPROACH

Based on Fig. 3, for a transition from LCR or LCL to LK to occur, the estimation of HMM should also be LK. On the other hand, for a transition from LK to LCR or LCL, the HMM estimation should be LCR or LCL respectively. If the HMM estimation is same as the current state or the conditions are not met, the model remains in the same state. The conditions for state changes are based on the aforementioned HMM mathematical process. The transition conditions are summarized in Table 1.

1) DATA SELECTION

As mentioned previously, only environmental variables are used as inputs for the model as these variables provide information that mainly affect the human driving decisions. Environmental variables are distinguished into two types: state of ego/surrounding vehicles and driver’s operational information. Accordingly, two models are developed using two sets of input variables. Model I uses distances, velocity deviation, and current lane as inputs, while model II uses TTC and current lane as inputs. The selected variables based on both models best describe the current driving situation, thus affecting the driving behaviors (given in Table 2 and 3). The variables are selected as they have a higher influence on the driver’s decision to make a lane change as well as the ability to describe the relationship between the ego vehicle and surrounding vehicles [3], [6].

A prefilter using two thresholds is applied on the distance and velocity variables for model I and on the TTC variables

TABLE 2. Environmental variables of model I.

Symbol	Definitions
l	Current lane number of ego vehicle
$v_{deviation}$	velocity deviation between the ego vehicle and the vehicle in the front
d_f	distance to the vehicle in the front
d_{fl}	distance to the vehicle in the front left
d_{fr}	distance to the vehicle in the front right
d_{bl}	distance to the vehicle in the back left
d_{br}	distance to the vehicle in the back right

TABLE 3. Environmental variables of model II.

Symbol	Definitions
l	Current lane number of ego vehicle
TTC_f	Time to collision to the vehicle in the front
TTC_{fl}	Time to collision to the vehicle in the front left
TTC_{fr}	Time to collision to the vehicle in the front right
TTC_b	Time to collision to the vehicle in the back
TTC_{bl}	Time to collision to the vehicle in the back left
TTC_{br}	Time to collision to the vehicle in the back right

for model II. Each variable is divided by the prefilter into three segments [6]. As for the current lane number, the values are fixed indicating the specific lane of the ego vehicle.

B. OPTIMIZATION

The prefilter thresholds are design parameters of the model that needs to be optimized to generate an optimal $\lambda = (A, B, \pi)$ for performance improvements. Here, the Non-Dominated Sorting Genetic-Algorithm-II (NSGA-II) is chosen to optimize these parameters. This technique is used due to its ability to handle Multi-objective optimization problems (MOPs) [26] and its fast convergence [26], [27]. The parameters are defined using this approach, such that the objective functions are minimized.

To evaluate the model’s performance, ACC, DR, and FAR [28], [29] metrics are used by comparing the actual and estimated behaviors. The metric values are defined based on the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values. Using LCR as an example to illustrate these values, TP is the number of events when the actual and estimated maneuvers are positive (LCR), while FP is the number of events when the estimated maneuver is positive, but the actual maneuver is not. Similar concept applies to TN and FN. The formulations for the metrics are given as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{8}$$

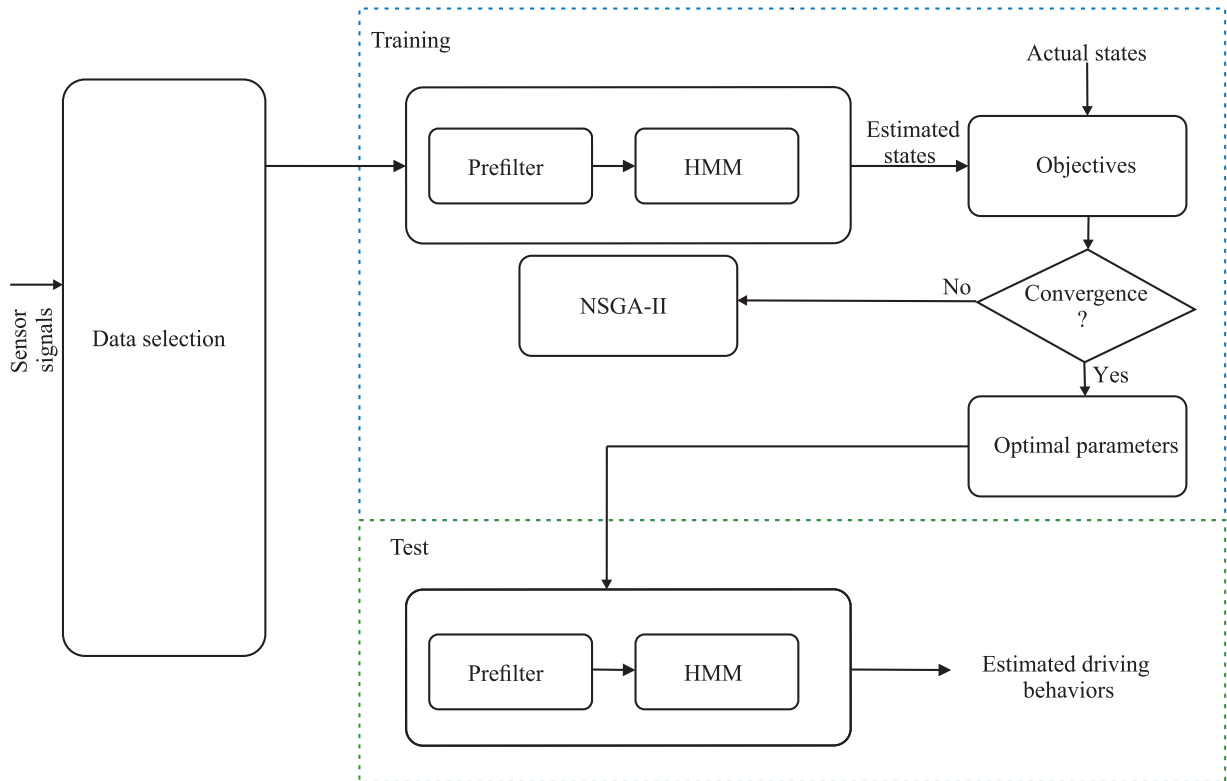


FIGURE 4. Optimal prefilter development.

$$DR = \frac{TP}{TP + FN}, \quad \text{and} \quad (9)$$

$$FAR = \frac{FP}{TN + FP}. \quad (10)$$

Therefore, the parameters are determined such that the model achieves in parallel high ACC, DR, and low FAR. Similar objective functions as in [13] and [15] are chosen for the NSGA-II. As objective functions

$$f_1 = (1 - DR_{right}) + FAR_{right}, \quad (11)$$

$$f_2 = (1 - DR_{keep}) + FAR_{keep}, \quad \text{and} \quad (12)$$

$$f_3 = (1 - DR_{left}) + FAR_{left} \quad (13)$$

are used, whereby each function corresponds to a specific lane changing behavior. In Fig. 4, the optimization process of the prefilter thresholds is shown.

IV. APPLICATION OF THE METHOD

The application of the HMM-based state machine approach is realized in this section. The experimental setup for data collection is first described followed by the process of labeling lane changing behaviors. Next, the training and test processes are explained. The training is done using the driving data from each individual driver for defining the optimal design parameter values. Then, the test is done based on the trained model using different data by the same driver and other drivers.

A. DESIGN OF THE EXPERIMENT

The driving experiments are conducted using a driving simulator, *SCANeRTM* in a laboratory environment (Fig. 5). The simulator has a base-fixed seat equipped with a steering wheel, pedals, and a gear. Five screens are used to simulate a real driving experience and environment. The left, right, and rear view mirrors are placed in the corresponding positions of the screens, which are essential when performing a lane change maneuver. The scenarios in the experiment are based on a four lane highway in two directions. A traffic environment with other vehicles to simulate an actual driving environment is utilized. Different maneuvers such as overtaking the vehicle ahead are performed by the drivers. Based on the rules in Germany, overtaking can only take place from the left lane. In the experiment, 9 drivers are considered such that each driver performed a 25 minutes drive to generate data. The split ratio for training and test data of each driver is 70:30. Hence, 70 % of the data is used for training, while the remaining 30 % is used for test. All participants provided their consent to take part in the experiment.

B. DATA PROCESSING

To determine the current lane l of the ego vehicle, the vehicle's centre point position is taken into account [6]. By comparing the values of l at different time points, the driving states are determined. A change in the value of l indicates that a lane change has occurred at a time point denoted by



FIGURE 5. Driving simulator, chair of dynamics and control (SRS, Uni-DuE).

TABLE 4. Average performance based on different lane changing duration.

States	Metrics	Model I variables [%]			Model II variables [%]		
		2	2.5	3	2	2.5	3
Overall	ACC	71.27	83.74	73.72	82.82	84.14	82.88
Right	ACC	87.05	91.67	87.28	90.42	91.24	90.58
	DR	78.86	84.97	68.86	81.89	90.76	72.40
	FAR	12.42	7.95	11.10	9.07	8.70	7.82
Keep	ACC	71.43	83.74	74.31	82.91	84.14	83.26
	DR	71.16	83.57	75.20	84.02	84.89	86.22
	FAR	27.85	15.57	31.24	25.82	21.22	32.66
Left	ACC	84.06	92.08	85.84	92.32	92.90	91.92
	DR	62.93	83.83	61.72	64.84	66.16	58.07
	FAR	14.66	7.37	12.02	5.91	5.35	5.03

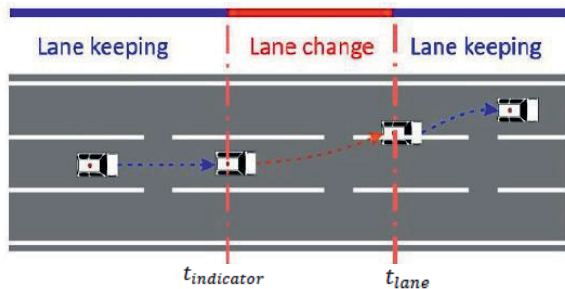


FIGURE 6. Lane changing description.

t_{lane} (Fig. 6). A LCL is defined by an increase in l , while a LCR is defined by a decrease. When l is the same as the previous time point, a LK is denoted.

The beginning of a lane change is defined by the time of the indicator activation, $t_{indicator}$. Here, the interval between $t_{indicator}$ and t_{lane} is defined as the lane change duration denoted by t_{change} [3], [6]. From the experiments, this duration is between 2 to 3 seconds, whereby the driver activates the indicator 2 to 3 seconds before performing a lane change. Different preset t_{change} values of 2 s, 2.5 s, and 3 s are therefore tested for labeling behaviors to evaluate the impact of t_{change} on the lane change recognition abilities. To do so, the HMM developed in [6] is used to estimate the lane changing behaviors based on the variables of models I and II. Previously mentioned metrics are used to evaluate the estimations. The training and test split ratio is also 70:30. In Table 4, the average performance values based on all drivers using the different t_{change} values are given. Using 2.5 s

to label the behaviors, generates results closest to the actual behavior for both models as most metrics have the highest values. Hence, t_{change} of 2.5 s is used to define a lane changing behavior in this research. Inaccurate data are removed as part of the labeling processing. For an example, when the driver does not intend to change lanes, but drives over the white lines or slightly overlaps the lines to the next lane due to driving errors. A lane change is detected consequently, when it does not reflect the actual driver's behavior. Hence, these inaccuracies are removed [6].

C. TRAINING AND TEST

The training and test process are explained here.

Training phase: The purpose of the training is to develop a model with optimal parameters. The training process is described in the following manner,

- 1) Data (input variables) and the actual lane changing behaviors are loaded into the model.
- 2) The design parameters (prefilter thresholds) are developed using NSGA-II, which are then used to define the observation sequence in the HMM. Based on the defined observation sequence and actual driving behaviors, the HMM model is trained by optimization to define the optimal HMM parameters.
- 3) Next, the hidden states are estimated using the HMM parameters.
- 4) The state machine defines the final estimations using the estimations of HMM from the previous step.
- 5) The actual and estimated behaviors are compared to evaluate the ACC, DR, and FAR values. Using these values, the objective functions are calculated.
- 6) Steps (1) to (5) are repeated until convergence, such that optimal parameters are defined minimizing the objective functions. For the number of iterations, a generation size of 200 and population size of 90 is used in NSGA-II.

Test phase: The test is performed using data not used in training as mentioned previously. Thus, based on the trained models for each driver, the lane changing behaviors are estimated using test data of the corresponding driver. Estimation performances are then evaluated using the mentioned metrics. To analyze the generability of the model, the trained parameters based on a specific driver are not only tested using the corresponding driver's test data, but also using test data of other drivers.

V. RESULTS

The performance based on the proposed method is presented in this section. Here, the performance based on models I and II are given and compared to analyze the effects of the prefilter on the different variable sets. The results presented are based on using 2.5 s for the lane changing duration, as it generated the best performance values when tested with the proposed approach for both models. This shows that the estimations are closest to the actual behaviors. Performance comparisons between the proposed approach and other

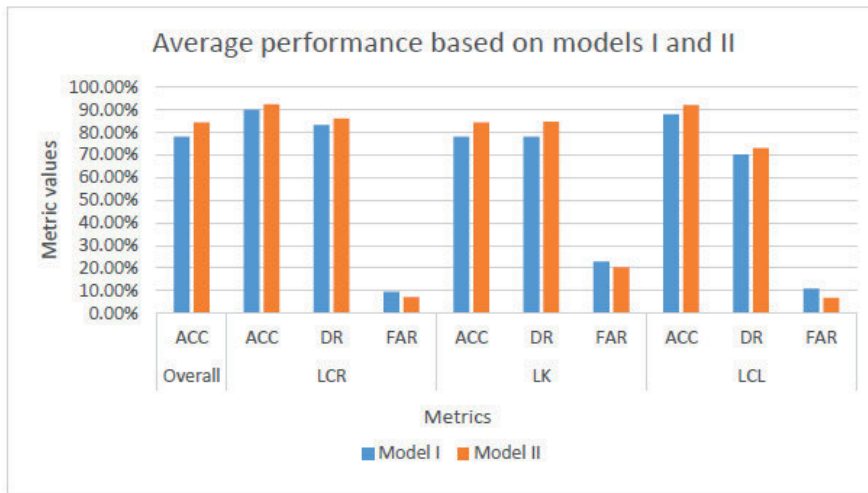


FIGURE 7. Test results of model I and II.

TABLE 5. Average performance values based on test data.

States	Metrics	Model I: Average values (test data) [%]	Model II: Average values (test data) [%]
Overall	ACC	78.13	84.48
Right	ACC	90.08	92.44
	DR	83.27	86.09
	FAR	9.48	7.13
Keep	ACC	78.13	84.48
	DR	78.08	84.87
	FAR	22.97	20.27
Left	ACC	88.04	92.04
	DR	70.03	73.03
	FAR	10.91	6.89

approaches (HMM and ANN-based state machine model) are presented as well to assess the capabilities of the approach.

A. EVALUATION OF RESULTS

The average values of ACC, DR, and FAR based on the test data for both models are given in Table 5 and Fig. 7. In the table, the average values (test data) are the average metric values when each trained model is tested with the corresponding driver’s test data.

As an example, the actual and estimated lane changing states corresponding to test data set of driver 2 (based on trained model of driver 2) from both models are plotted in Fig. 8 and Fig. 9. The red dotted lines represent the actual driving states, while the blue lines are the estimated states. The different driving states are represented in the vertical axis, whereby 1 is LCR, 2 is LK, and 3 is LCL, while the horizontal axis represents the time length of the drive in seconds, s. The data are recorded every 0.05 s. The figures show the proximity between the actual and estimated states.

As mentioned, a generability test is performed as well in which the results are presented in Table 6 and Fig. 10. In this test, the trained model of a specific driver is tested with test data of other drivers with the aim to analyze if the performance values are close to the values obtained in Table 5. The average values (other test data) are the average

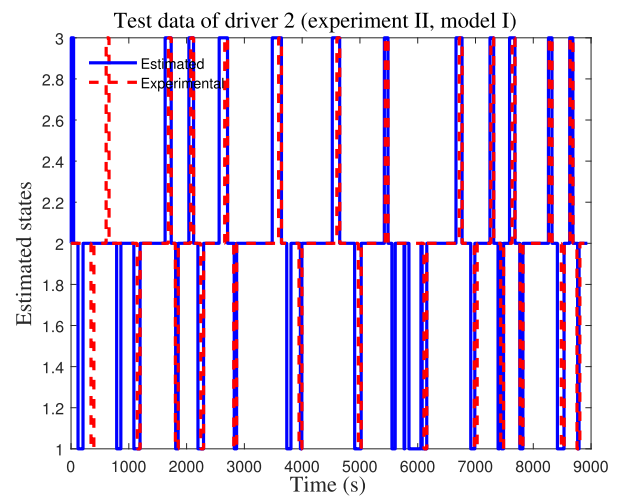


FIGURE 8. Test data of driver 2 (model I).

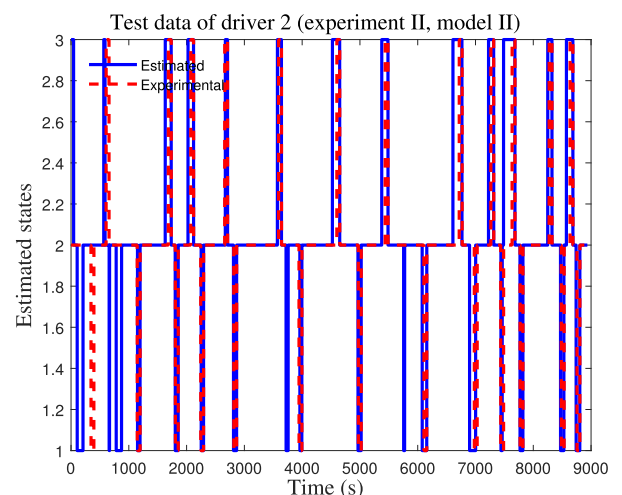


FIGURE 9. Test data of driver 2 (model II).

values when the test data of other drivers are used for testing the trained model of each specific driver.

Using drivers 1 and 2 as examples for the generability test, the estimated and actual lane changing states based on test

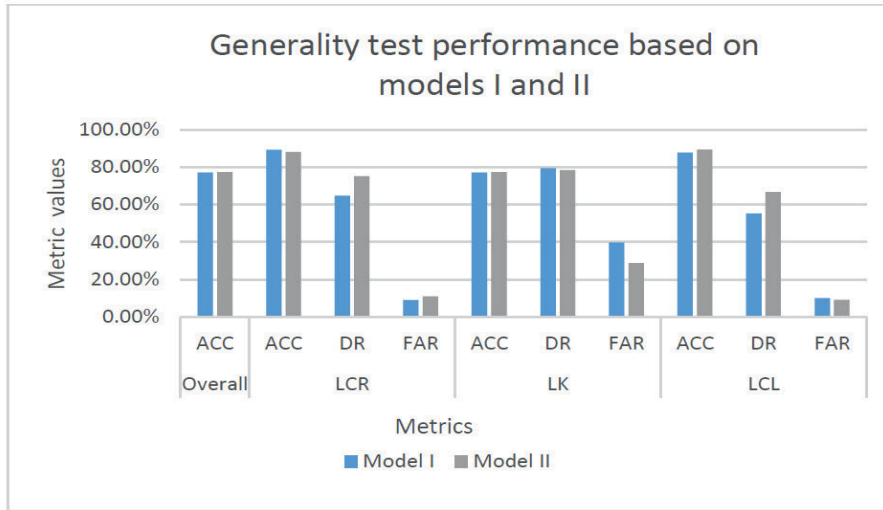


FIGURE 10. Generability test results of models I and II.

TABLE 6. Average performance based on generability test.

States	Metrics	Model I: Average values (other test data) [%]	Model II: Average values (other test data) [%]
Overall	ACC	77.06	77.47
Right	ACC	89.27	88.05
	DR	64.77	75.16
	FAR	9.00	10.99
Keep	ACC	77.06	77.47
	DR	79.50	78.41
	FAR	39.67	28.68
Left	ACC	87.78	89.42
	DR	55.27	66.75
	FAR	10.02	9.09

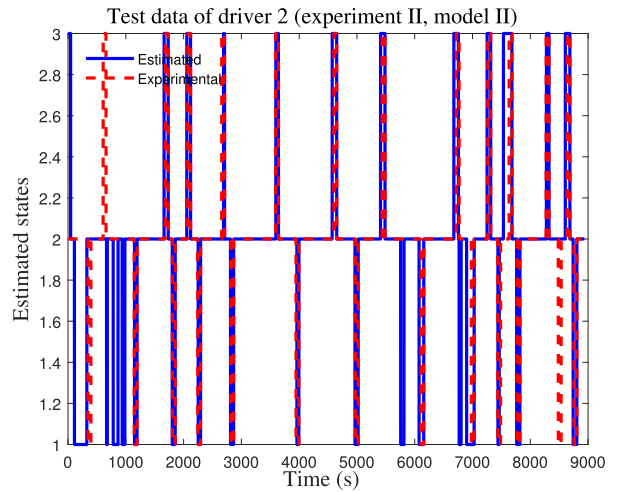


FIGURE 12. Generability test based on driver 2 (model II).

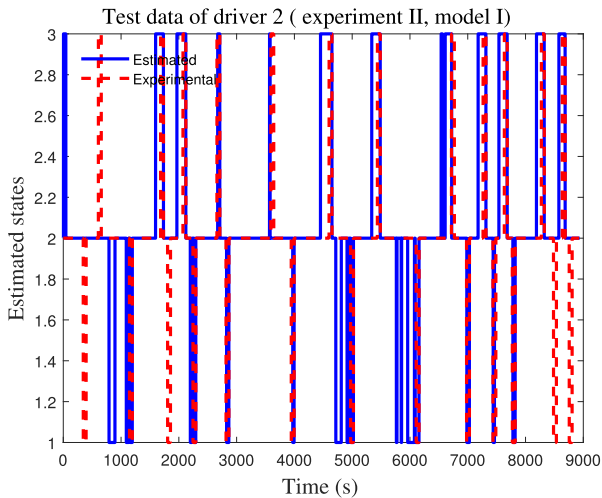


FIGURE 11. Generability test based on driver 2 (model I).

data of driver 2 (trained model of driver 1 tested) are plotted in Fig. 11 and Fig. 12.

Based on the results in Table 5, both models generate high ACC, DR, and low FAR, with the exception of FAR_{keep} . Based on the obtained results, it can be stated that model II has a higher performance than model I in most metrics. From this observation, it can be concluded that the prefilter application

on TTC variables tends to have a positive effect on the performance. The generability test results (Table 6) show that the model II also outperforms model I here except for ACC_{right} , FAR_{right} , and FAR_{keep} . However, the metric values based on Table 5 are higher.

B. COMPARISONS WITH OTHER APPROACHES

Comparisons between the proposed approach with an improved HMM approach and an ANN-based state machine approach are also part of the evaluation process. The ANN-based state machine approach is developed in [15], while the HMM is based on [6]. The prefilter thresholds are optimized as well in the improved HMM, while the ANN-based state machine use biases and weights (as ANN parameters) defined by optimization to develop estimations. Model II is used for comparisons as it has a better performance than model I. All other approaches also uses the same input variables as model II.

TABLE 7. Comparisons between different approaches.

States	Metrics	HMM-based state machine [%]	HMM [%]	ANN-based state machine [%]
Overall	ACC	84.48	84.14	83.27
Right	ACC	92.44	91.24	93.57
	DR	86.09	90.76	61.57
	FAR	7.13	8.70	3.95
Keep	ACC	84.48	84.14	84.27
	DR	84.87	84.89	88.08
	FAR	20.27	21.22	38.54
Left	ACC	92.04	92.90	88.70
	DR	73.03	66.16	45.85
	FAR	6.89	5.35	8.23

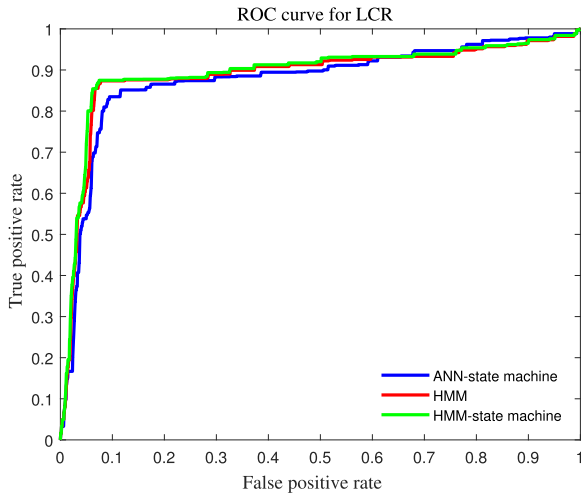


FIGURE 13. LCR ROC curve.

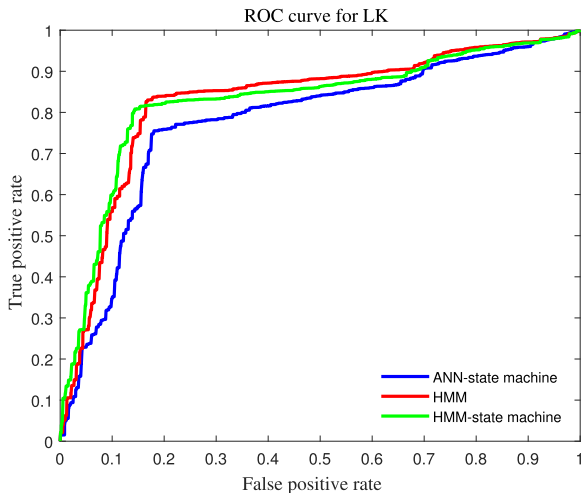


FIGURE 14. LK ROC curve.

In Table 7, the average metric values based on each driver’s test data are shown. In addition, the receiver operating characteristic (ROC) curves for the three approaches based on the different lane changing behaviors are given in Fig. 13 to Fig. 15. The area under curve (AUC) values of each method based on the different behaviors are also presented in Table 8.

From the results, the HMM-based state machine approach has a better performance than the other two approaches in most metrics. On the other hand, the individual HMM approach outperforms the ANN-based approach, except for

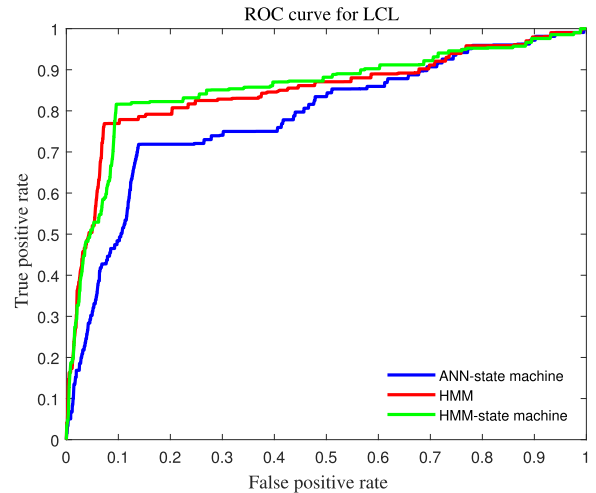


FIGURE 15. LCL ROC curve.

TABLE 8. AUC values of different approaches.

States	Approaches	AUC
LCR	HMM-based state machine	0.8939
	HMM	0.8885
	ANN-based state machine	0.8768
LK	HMM-based state machine	0.8275
	HMM	0.8281
	ANN-based state machine	0.7775
LCL	HMM-based state machine	0.8557
	HMM	0.8482
	ANN-based state machine	0.7860

ACC_{right} , FAR_{right} , ACC_{keep} , and DR_{keep} . An observation based on the results are the FAR_{keep} values tend to be high in all approaches. The ANN-based state machine approach also produces a lower DR_{left} value compared to other HMM-based approaches. Based on the ROC curves, the HMM-based state machine model has the best performance in LCR and LCL, as the model generates the highest true positive rate corresponding to a related low false alarm rate. The AUC values of the the proposed approach are also the highest in LCR and LCL (Table 8), while the AUC of the individual improved HMM approach is the highest fin LK. In general, it can be concluded that the proposed approach improves the performance of the individual improved HMM approach and the ANN-based state machine approach showing its effectiveness.

VI. SUMMARY AND CONCLUSION

Driving behavior prediction accuracy often rely on the structure of ML approaches or the input features used. In this contribution, a new approach combining a previously developed state machine model and an improved HMM for the recognition of lane changing behaviors is introduced. The state machine models the lane changing behaviors using three states, such that the model can express the transition between states or remain in the same state for estimations of the final

lane changing behaviors. The modeling of the transition is realized based on specific conditions defined by the HMM estimations. Here, the HMM estimates lane changing behaviors as well. The HMM differs from a conventional HMM as a prefilter with unknown thresholds values is applied to the features used with it. Two feature sets consisting of distance as well as velocity deviation variables (model I) and TTC variables (model II) are considered for the application of this method.

Based on the results obtained, model II produces better ACC, DR, and FAR than model I. Nevertheless, both models generates acceptable results. In general, it can be concluded that using TTC variables improves the recognition performance in both experiments. A generality test is performed as part of this work, which shows that model II has a better generalization ability than model I. For further evaluations, the developed approach is compared to an individual HMM approach and an ANN-based state machine approach using model II. The results show that the HMM-based state machine model outperforms the other two methods.

ACKNOWLEDGMENT

The authors acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

REFERENCES

- [1] *Mobility & Transport-Road Safety: Care Database*. Accessed: Sep. 3, 2022. [Online]. Available: https://road-safety.transport.ec.europa.eu/statistics-and-analysis/methodology-and-research/care-database_en
- [2] (2021). *Road Safety Statistics: What is Behind the Figures*. Accessed: Sep. 3, 2022. [Online]. Available: https://www.destatis.de/EN/Themes/Society-Environment/Traffic-Accidents/_Graphic/_Interactive/traffic-accidentsdriver-related-causes.html
- [3] Q. Deng, J. Wang, K. Hillebrand, C. R. Benjamin, and D. Soffker, "Prediction performance of lane changing behaviors: A study of combining environmental and eye-tracking data in a driving simulator," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3561–3570, Aug. 2020.
- [4] V. Leonhardt and G. Wanielik, "Neural network for lane change prediction assessing driving situation, driver behavior and vehicle movement," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
- [5] H. M. Mandalia and M. D. D. Salvucci, "Using support vector machines for lane-change detection," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 49, 2005, pp. 1965–1969, doi: 10.1177/154193120504902217.
- [6] Q. Deng, J. Wang, and D. Soffker, "Prediction of human driver behaviors based on an improved HMM approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 2066–2071.
- [7] Y. Dou, F. Yan, and D. Feng, "Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2016, pp. 901–906.
- [8] C. Zong, C. Wang, D. Yang, and H. Yang, "Driving intention identification and maneuvering behavior prediction of drivers on cornering," in *Proc. Int. Conf. Mechatronics Autom.*, Aug. 2009, pp. 4055–4060.
- [9] S. Wang, X. Zhao, Q. Yu, and T. Yuan, "Identification of driver braking intention based on long short-term memory (LSTM) network," *IEEE Access*, vol. 8, pp. 180422–180432, 2020.
- [10] K. B. Duan, J. C. Rajapakse, H. Wang, and F. Azuaje, "Multiple SVM-RFE for gene selection in cancer classification with expression data," *IEEE Trans. Nanobiosci.*, vol. 4, no. 3, pp. 228–234, Sep. 2005.
- [11] P. A. Mundra and J. C. Rajapakse, "SVM-RFE with MRMR filter for gene selection," *IEEE Trans. Nanobiosci.*, vol. 9, no. 1, pp. 31–37, Mar. 2009.
- [12] J. Gao, Y. L. Murphey, and H. Zhu, "Multivariate time series prediction of lane changing behavior using deep neural network," *Appl. Intell.*, vol. 48, no. 10, pp. 3523–3537, Oct. 2018.
- [13] R. David, S. Rothe, and D. Söffker, "State machine approach for lane changing driving behavior recognition," *Automation*, vol. 1, no. 1, pp. 68–79, Nov. 2020.
- [14] Q. Deng and D. Soffker, "Improved driving behaviors prediction based on fuzzy logic-hidden Markov model (FL-HMM)," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 2003–2008.
- [15] R. David, S. Rothe, and D. Soffker, "Lane changing behavior recognition based on artificial neural network-based state machine approach," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3444–3449.
- [16] S. F. Lopes, S. Silva, and J. L. Monteiro, "An easy-to-use and flexible object-oriented framework for extended finite state machines," in *Proc. IEEE 10th Int. Conf. Ind. Informat.*, Jul. 2012, pp. 35–40.
- [17] G. D. Micheli, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Optimal state assignment for finite state machines," *IEEE Trans. Comput.-Aided Design Integr.*, vol. CADI-4, no. 3, pp. 269–285, Jul. 1985.
- [18] N. Beganovic and D. Söffker, "Remaining lifetime modeling using state-of-health estimation," *Mech. Syst. Signal Process.*, vol. 92, pp. 107–123, Apr. 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327017300390>
- [19] B. Jiang and Y. Fei, "Traffic and vehicle speed prediction with neural network and hidden Markov model in vehicular networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1082–1087.
- [20] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, "An overview on study of identification of driver behavior characteristics for automotive control," *Math. Problems Eng.*, vol. 2014, pp. 1–15, Mar. 2014.
- [21] Q. Deng and D. Soffker, "A review of HMM-based approaches of driving behaviors recognition and prediction," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 1, pp. 21–31, Mar. 2022.
- [22] B. Jiang and Y. Fei, "Vehicle speed prediction by two-level data driven models in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1793–1801, Jul. 2017.
- [23] Y. Zhang, Q. Lin, J. Wang, S. Verwer, and J. M. Dolan, "Lane-change intention estimation for car-following control in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 276–286, Sep. 2018.
- [24] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Mach. Learn.*, vol. 32, pp. 41–62, Nov. 1998.
- [25] S. Nagasaka, T. Taniguchi, K. Hitomi, K. Takenaka, and T. Bando, "Prediction of next contextual changing point of driving behavior using unsupervised Bayesian double articulation analyzer," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 924–931.
- [26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2000.
- [27] K. D. Tran, "Elitist non-dominated sorting GA-II (NSGA-II) as a parameter-less multi-objective genetic algorithm," in *Proc. IEEE South-eastCon*, Apr. 2005, pp. 359–367.
- [28] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv: 2010.16061*.
- [29] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 4–19, Feb. 2014.



RUTH DAVID is currently pursuing the Ph.D. degree with the Chair of Dynamics and Control, University of Duisburg-Essen, Germany. Her current research interests include recognition and prediction of driving behaviors based on applications of machine learning methods. She currently focuses on developing a state machine-based model as a new machine learning approach for the prediction and recognition of behaviors.



DIRK SÖFFKER (Member, IEEE) received the Ph.D. degree (Dr.-Ing.) in mechanical engineering and the Habilitation degree in automatic control/safety engineering from the University of Wuppertal, Germany, in 1995 and 2001, respectively. He has been the Head Professor of the Chair of Dynamics and Control, University of Duisburg-Essen, Germany, since 2001. His current research interests include diagnostics and prognostics, modern methods of control theory, safe human interaction with technical systems, safety and reliability control engineering of technical systems, and cognitive technical systems.

• • •

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub

universitäts
bibliothek

This text is made available via DuEPublico, the institutional repository of the University of Duisburg-Essen. This version may eventually differ from another version distributed by a commercial publisher.

DOI: 10.1109/ACCESS.2022.3224012

URN: urn:nbn:de:hbz:465-20230623-091548-0



This work may be used under a Creative Commons Attribution 4.0 License (CC BY 4.0).