Manuscript version
DOI original paper: 10.1109/ITSC48978.2021.9564919
©2021 IEEE

Lane changing behavior recognition based on Artificial Neural Network-based State Machine approach

Ruth David, Sandra Rothe, and Dirk Söffker[1]

Abstract— **Developing driving behavior prediction and recognition models play a crucial role in Advanced Driving Assistance Systems (ADAS). Developing these models generally requires the use of Machine Learning approaches. Often, Machine Learning approaches are difficult to interpret. In this contribution, an Artificial Neural Network (ANN)-based state machine driving behavior recognition model is developed to estimate three lane changing behaviors. A state machine topology defining the relationship between the states (driving behaviors) is developed. The state transitions to another state or remains in the same state based on specific conditions defined by estimations of ANN. Two options are developed: using one common ANN or using three ANN (for three states). Design parameters (weights and biases) defined using optimization describe the ANN estimations when trained. Based on this, lane changing behaviors for the models are estimated. Data from three participants were collected. The results show that this approach performs better than the conventional ANN in terms of DR and FAR with improvements up to 46 % for DR and 34 % for FAR. Based on the results, it can be concluded that the approach introduced realizes high accuracy (ACC), high detection rates (DR), and low false alarm rates (FAR).**

I. INTRODUCTION

Road safety when driving is one of the most important issue faced by drivers. According to the Department of Statistics in Germany, 74.4 % of driving accidents are caused by human driving behaviors making it the main cause of road accidents in 2019 [1]. In recent years, the use of Advanced Driving Assistance Systems (ADAS) have been developed to tackle this issue by helping the driver to maneuver in different situations to avoid errors. Driving behavior prediction and recognition models play an important role this development. As the accidents are mainly caused by driving behaviors, prediction and recognition models should be individualized. Thus, incorporating the knowledge of individual driving behaviors into ADAS is able to provide drivers with information on how to maneuver based on the driver's individual driving habits. Recent researches developed driving behavior prediction and recognition models based on Machine Learning algorithms. Some of the Machine Learning algorithms used are Artificial Neural Networks (ANNs) [2], Support Vector Machines (SVM) [3], or Fuzzy Logic (FL) [4]. Two concepts are used to develop these models. The first concept combines two or more Machine Learning algorithms to develop the model. In [5], density-based clustering, SVM, and long short-term memory model (LSTM) are combined to predict lane changing maneuvers. The density-based clustering identifies driving intentions and the SVM uses this results to label the new raw data. The LSTM predicts lane changing maneuvers using the labeled data. The second concept considers the selection of input features, as the features affect the performance of the model. Thus, determining suitable features is important to improve the performance. Common input features are environmental, driver's operational, physiological, and eye-tracker variables. Feature selection methods such as filter and wrapper methods are used in [6] to select suitable features. In [7], textural expressions and feature points are considered to estimate driver's drowsiness. A state machine-based driving recognition model is introduced in [8].

One of the challenges is finding optimal parameters of the recognition and prediction model to improve the performance of the model. In this contribution, a trainable and interpretable driving recognition model is developed by combining ANN with a state machine-based approach [8] to estimate different lane changing behaviors. Here, the state machine's topology is more interpretable than Hidden Markov Model (HMM), while ANN is used as it requires low statistical training. The objective is to develop a model that generates optimal estimations with regards to high accuracy (ACC), high detection rates (DR), and low false alarm rates (FAR). The state machine determines the final estimation based on the output of the ANN in the model. Two approaches are used to develop the model. For training, Non dominated Sorting Genetic Algorithm II (NSGA-II) is used to generate optimal parameters. The aim of this contribution is to prove that this approach performs better than conventional ANN. In comparison to [8], whereby a state machine approach considering thresholds of input variables as parameters is used, in this contribution the state machine is combined with ANN, with weights and biases of the ANN as parameters.

This contribution is organized as follows: in Section II, a methodology about the state machine-based approach and ANN is presented. In Section III, the development of the recognition model is described. The application is presented in Section IV, whereby the experimental design and the method execution are described. The results in terms of ACC, DR, and FAR including comparisons between the proposed method and a conventional ANN are presented and discussed in Section V. Finally, in Section VI, a conclusion is given.

II. METHODOLOGY

A. State machine-based approach

The state machine models the behavior with discrete variable states. The system dynamics in a state machine are

[1] The authors are with the Chair of Dynamics and Control, University of Duisburg-Essen, Duisburg, Germany ruth.david@uni-due.de, sandra.rothe@uni-due.de, and soeffker@uni-due.de

characterized by a sequence of transitions based on a set of inputs in which the system can remain in the same state or switch to another state [9]. A transition from one state to another is based on specific conditions. Changes in the states result from the output of a particular system. State machines are well-known approaches for modeling behaviors. Here, designers define the parameters and variables required for modeling [8]. A state machine model has a fixed number of states. Some of the advantages of the state machine approach include, easy designing process, quick implementation and execution processes. It is also easy to track the event that causes a state change [10].

In this contribution, the state machine approach in [8], is applied. The lane changing behaviors estimated are lane change to right (LCR, state 1), lane keeping (LK, state 2), and lane change to left (LCL, state 3) defined as states. If the driver is in LK (state 2), the possible next options are further LK (remaining in state 2), LCR (transition to state 1), or LCL (transition to state 3). During a lane change maneuver (state 1 or state 3), possible options are further lane change in same direction (remaining in state 1 or 3) or the lane change maneuver is over and the driver is LK (transition to state 2). Here, the conditions for transitioning or remaining are based on environmental and operational variables (inputs) and parameters. Optimal parameters are generated through the training process, thus producing an effective estimation of the driving behaviors.

B. Artificial Neural Network

An Artificial Neural Network (ANN) is a Machine Learning algorithm that acquires, learns, and stores knowledge received from the input to produce an output [11]. As mentioned, Artificial Neural Networks are well-known approaches for classification and prediction as they can be trained quickly and are highly flexible [12]. In this contribution, only one hidden layer of ten neurons is considered. The output layer consists of three neurons each representing the three driving behaviors (also known as classes), LCR, LK, and LCL. The LCR output is also denoted as 1, LK is denoted as 2, and LCL is denoted as 3. The input layer receives data, which is passed to the hidden layer for processing and sent to the output layer to generate an output. Based on the weights, activation functions, and bias values, predicted probabilities are associated with each neurons with exception to the neurons in the input layer. Predicted probabilities of three neurons (classes) in the output layer are generated, the final estimation of the network is based on the output neuron that has the highest predicted probability. The predicted probability is given by

$$Y = f(\sum_{i=1}^{N} w_i x_i + b), \tag{1}$$

with w_i as the weight, x_i as the input at a particular layer, b as the bias value, f as the activation function, N as the number of inputs, and Y as the output. As activation functions, Sigmoid and Softmax activation functions are used.

III. STATE MACHINE-BASED APPROACH COMBINED WITH ARTIFICIAL NEURAL NETWORK DRIVING RECOGNITION MODEL

The aim of this contribution is to establish a model that combines two trainable systems (ANN and state machine-based approach) to estimate lane changing behavior. The idea is to apply the approach to a system with different states. Therefore, the topology of the state machine-based approach is used to model states and transitions and the ANN to model the transition conditions. The model is realized using two concepts with appropriate inputs and design parameters denoted as approach I and approach II. Approach I is based on one common ANN combined with the state machine approach. Approach II is based on three ANN (representing the three driving behaviors) combined with the state machine approach.

A. Integration of ANN in the state machine model

The state transitions are based on certain conditions modeled by ANN estimations. The input data consist of driver's operation and environmental variables as these variables play a major role in driver's behaviors. The operation variables are angle of steering wheel (a_{st}), accelerator pedal position (a_{acc}), brake pedal position (a_{brake}), indicator (i), and current lane (l). The environmental variables are time to collision (TTC) to the vehicle in the front (TTC_f), to the vehicle in the back (TTC_b), to the vehicle in the front left (TTC_{fl}), to the vehicle in the front right (TTC_{fr}), to the vehicle in the back left (TTC_{bl}), and to the vehicle in the back right (TTC_{br}) [8], [13]. The input variables denote the relationship between the ego vehicle and surrounding vehicles. The output of the proposed model is defined by the three driving states, LCR, LK, and LCL. First, the input variables along with the model parameters are used to estimate the driving behaviors using ANN, which are also LCR, LK, and LCL. These estimations are integrated into the state machine topology as conditions for a transition or to remain.

B. Two concepts for modeling the transition-state relation

For approach I (Fig. 1), transition or remaining conditions are defined by the outputs of one common ANN, which are LCR, LK, and LCL as previously described. For a transition from state 2 (LK) to state 1 (LCR) or state 3 (LCL) in the proposed model, the output of ANN should be LCR (1) or LCL (3) respectively as well, at that time point. Similarly, for a transition from state 1 or state 3 to state 2, the output of ANN is denoted as LK (2). If the conditions are not met, the model remains in the same state.

Following the same integration and transition process as approach I, the conditions in approach II (Fig. 2) are based on the ANN corresponding to the current estimated state, whereby three networks are defined as ANN (right), ANN (keep), and ANN (left) (Fig. 2). The possible outputs for the three networks are listed in Table I. If the estimation of ANN is same as the current state, then the model remains in the same state.

TABLE I: Outputs of the three ANN.

ANN models	Possible outputs (denotations)
ANN(right)	LCR (1), LK (2)
ANN(keep)	LCR (1), LK (2), and LCL (3)
ANN(left)	LK (2), LCL (3)

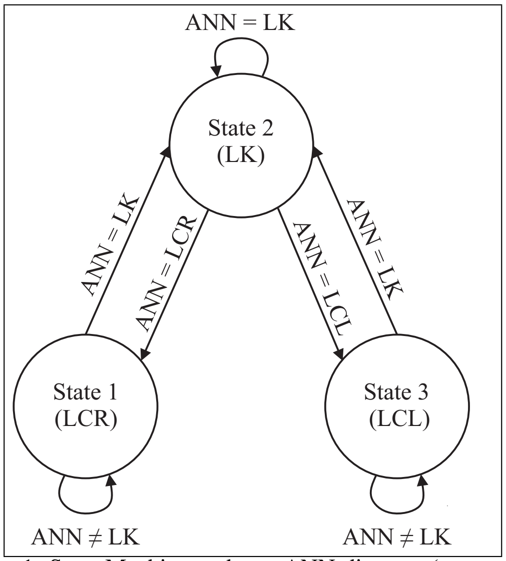

Fig. 1: State Machine and one ANN diagram (approach I).

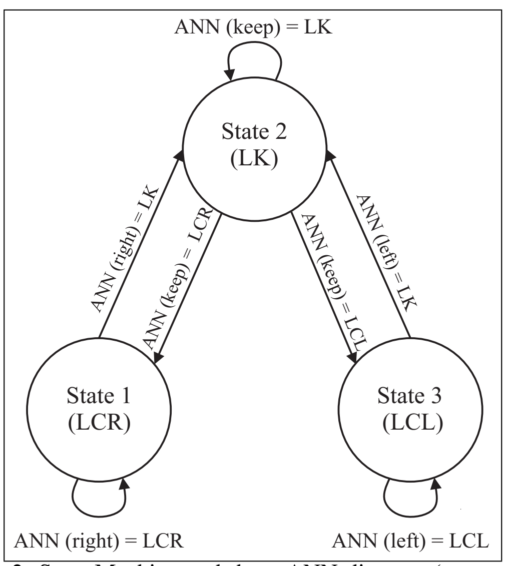

Fig. 2: State Machine and three ANN diagram (approach II).

C. Defining parameters by optimization

The design parameters associated with ANN, which are weights and biases, affect the performance of the overall model to estimate the driving behaviors. Thus, the definition (training) of these parameters by optimization is necessary to develop optimal estimates of the model. In this contribution, NSGA-II[14] is used to define the optimal design parameters in the training process. Some of the important features of NSGA-II include, diversity preserving method (crowding distance), and emphasizing on non dominated solutions [15], [16]. The optimal design parameters are developed with respect to maximal accuracy rates (ACC), detection rates (DR), and minimal false alarm rates (FAR). The equations related to ACC, DR, and FAR are defined as

$$ACC = \frac{TP+TN}{TP+TN+FP+FN}, \tag{2}$$

$$DR = \frac{TP}{TP+FN}, \text{ and} \tag{3}$$

$$FAR = \frac{FP}{TN+FP}. \tag{4}$$

Thus, the performance of the overall model can be evaluated using ACC, DR, and FAR by comparing the estimated and the actual driving states at each time point. Suitable objective functions are chosen to evaluate the optimization process, whereby the actual and estimated driving states are compared to minimize the deviation between them. The objective functions are defined as

$$f_1 = (1 - DR_{right}) + FAR_{right}, \tag{5}$$

$$f_2 = (1 - DR_{keep}) + FAR_{keep}, \text{ and} \tag{6}$$

$$f_3 = (1 - DR_{left}) + FAR_{left}. \tag{7}$$

The objectives are defined for evaluating the three states.

IV. APPLICATION OF DEVELOPED METHOD

A. Design of the experiment

The driving data sets were obtained using a driving simulator, $SCANEeR^{TM}$ at the Chair of Dynamics and Control (University of Duisburg-Essen) (Fig. 3) to perform driving simulations [8], [13].

Fig. 3: Driving simulator, Chair of Dynamics and Control (SRS).

The scenario in this experiment is based on a four lane highway with two directions. The driving environment is a normal environment. The driver can perform different maneuvers while driving with other surrounding vehicles participating in the driving simulation. For an example, the driver is able to overtake a slow moving vehicle ahead and move back to the initial lane after overtaking. Following the rules in Germany, the driver can only overtake from the left. The data sets were obtained for training and testing from three participants ages between 25 to 30 years, all of which held a valid driving license. The training data set is based on a 40 minute manual drive and the test data set is based on a 10 minute manual drive [8], [13]. Each training and test data set corresponds to one driver. The current lane number of the ego vehicle, l_t is determined through the vehicle's center point. This can be used to determine the actual driving states by comparing the lane numbers at different time points. If the current lane number l_t and the previous lane number l_{t-1} are the same, then the ego vehicle is in the same lane and lane keeping (LK) is defined. If the current lane number is higher than previous lane number, then this suggests a left lane change (LCL), while if the current lane number is lower than the previous, a right lane change (LCR) is defined. Based on Fig. 4, the time at which a lane change occurs is defined by t_{lane} and the time of last significant change in the angle of steering wheel is defined as t_{angle} which is the starting time of a lane changing behavior. Thus, the lane changing behavior is defined by the time interval between t_{angle} and t_{lane} [8], [13].

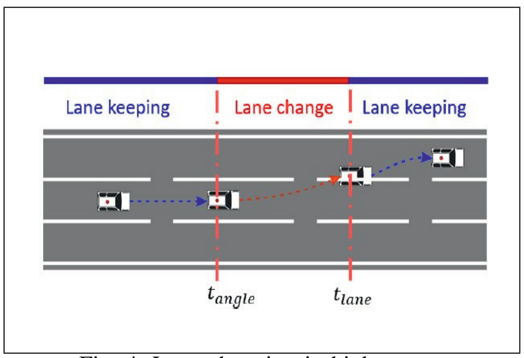

Fig. 4: Lane changing in highway.

B. Training and test process

The training and test processes are based on the input variables and the actual driving states. The data sets were trained in the following manner:

1) Input variables and actual driving states are given into the state machine combined with ANN model.
2) Using NSGA-II, a set of design parameters is generated.
3) Based on the optimal parameters generated, the ANN within the model generates the predicted probabilities for the three different behaviors at different time points. This determines the estimations/outputs of ANN.
4) Based on the estimations of the ANN, the estimations of the proposed model are determined using the state machine topology.
5) For approach II, the NSGA-II determines the design parameters associated with all three neural networks. Using the state machine topology, the transition condition is based on the estimation of the ANN corresponding to the current state to determine the output.
6) The actual states and the estimated states from the proposed model will be compared to derive the ACC, DR, and FAR. The objective functions are evaluated.
7) The process (1) to (6) is repeated until convergence and the optimal model is obtained.

The generation size used is 200 with a population size of 90, which are inputs/options for NSGA-II to describe the number of iterations for training.

Testing is conducted by two steps:

1) The trained model is applied to the test data set for state estimations.
2) The actual driving states (from test data set) and estimated driving states are compared.

V. RESULTS

Comparing the real and estimated lane changing behaviors to evaluate the similarities, enables the evaluation of ACC, DR, and FAR for different test data sets. In Figures 5 to 7, the real states and the estimated states of test data set 1, test data set 2, and test data set 3 corresponding to drivers 1, 2, and 3 are shown based on the trained model, using approach II. Also, training data sets are referred as data sets in the figures and tables.

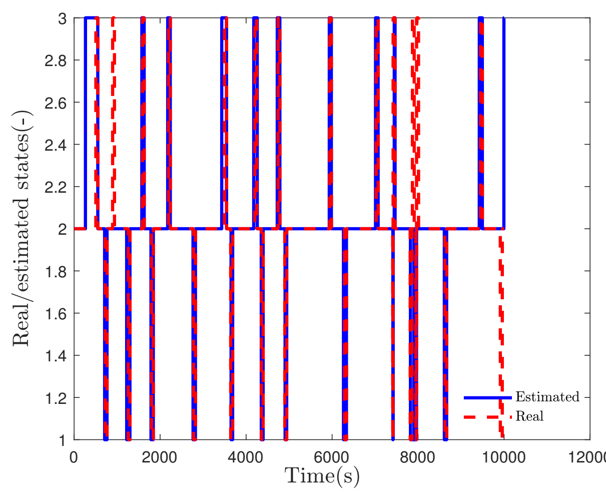

Fig. 5: Real and estimated driving states for test data set 1 (training based on data set 1).

The drivers make the choice by assessing the traffic conditions and of the drivers' free will. Based on the results presented, a close fit between the estimated and real states for all three data sets can be observed with some inconsistencies. A close fit between the estimated and real states are also achieved when approach I is applied. The training and test process were repeated several times.

Tables II to IV, show the ACC, DR, and FAR results of different test data sets based on a model using approach II. When a model is trained using a data set from a driver, the corresponding test data set and whole data sets (combined training and test data sets of a driver) from other drivers are used for test. This is done to prove the generalibility of the proposed approaches. For reference, training, test, and whole data sets 1, 2, and 3 correspond to drivers 1, 2, and 3 respectively.

Table II shows ACC, DR, and FAR for test data set 1, whole data set 2 (combined training and test data set 2), and

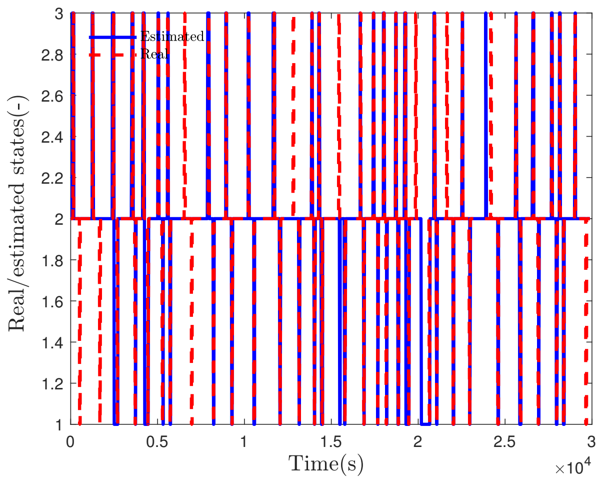

Fig. 6: Real and estimated driving states for test data set 2 (training based on data set 2).

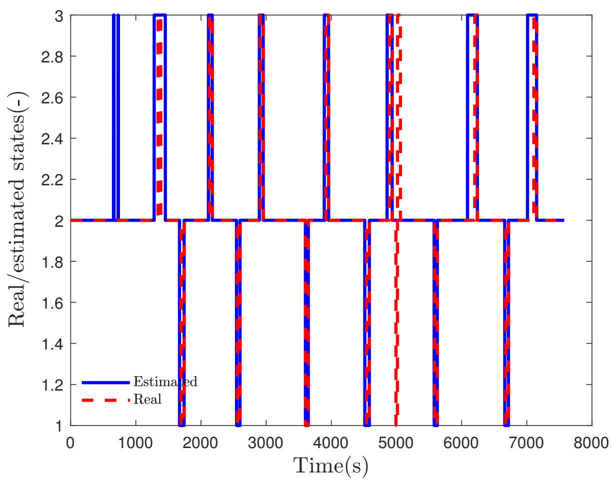

Fig. 7: Real and estimated driving states for test data set 3 (training based on data set 3).

whole data set 3 (combined training and test data set 3) when the model is trained using data set 1.

TABLE II: Evaluation of metrics of different test data sets (data set 1).

State	Metrics	Test data set 1 [%]	Whole data set 2 [%]	Whole data set 3 [%]
Overall	*ACC*	92.71	93.69	82.40
Right	*ACC*	97.41	96.80	87.04
	DR	89.92	79.24	92.88
	FAR	1.64	2.15	13.24
Keep	*ACC*	92.71	93.69	82.40
	DR	93.42	94.90	81.14
	FAR	13.30	17.05	5.85
Left	*ACC*	94.77	96.79	95.35
	DR	83.87	86.71	95.35
	FAR	4.58	2.67	4.65

Based on Table II, the ACC and DR of right, keep, and left maneuvers are generally higher than 80 %, sometimes higher than 90 %. The highest accuracy is ACC_{right} at 97.41 %. Low FAR are generally achieved for the test data sets. Thus, when model is trained using data set 1 and tested using the test data sets, high ACC, DR, and low FAR are achieved for the different states.

Table III shows ACC, DR, and FAR for test data set 2, whole data set 1, and whole data set 3 when training using data set 2.

TABLE III: Evaluation of metrics of different test data sets (data set 2).

State	Metrics	Whole data set 1 [%]	Test data set 2 [%]	Whole data set 3 [%]
Overall	*ACC*	89.16	92.43	74.89
Right	*ACC*	96.67	95.41	85.06
	DR	84.86	72.86	87.88
	FAR	2.75	3.54	15.08
Keep	*ACC*	89.27	92.43	74.90
	DR	89.32	94.30	73.21
	FAR	11.24	26.64	9.31
Left	*ACC*	92.38	97.02	89.83
	DR	90.26	73.86	93. 24
	FAR	7.51	1.90	10.35

Here, the overall accuracy for test data set 2 is 92.43 %. The ACC and DR for the right, keep, and left maneuvers are also higher than 80 % (some are higher than 90 %) with a few exceptions which have values higher than 70 %. The highest ACC is ACC_{left} with an accuracy of 97.02 % in test data set 2. The FAR are also low in most test data sets, with a few exceptions like FAR_{keep} in test data set 2, at 26.64 % and FAR_{right} in whole data set 3, at 15.08 %.

Next, Table IV shows ACC, DR, and FAR for test data set 3, whole data set 1, and whole data set 2 when the model is trained using data set 3.

TABLE IV: Evaluation of metrics of different test data sets (data set 3).

States	Metrics	Whole data set 1 [%]	Whole data set 2 [%]	Test data set 3 [%]
Overall	*ACC*	94.19	92.06	93.32
Right	*ACC*	97.99	96.49	98.59
	DR	87.39	73.07	91.55
	FAR	1.48	2.25	1.14
Keep	*ACC*	94.20	92.06	93.32
	DR	94.80	93.58	93.71
	FAR	11.39	21.39	11.04
Left	*ACC*	96.18	95.57	94.73
	DR	89.47	84.23	86.80
	FAR	3.47	3.83	4.89

The highest ACC in this table is ACC_{right}, at 98.59 %. Similar to others, the DR values are also higher than 80 % (higher than 90 % for DR_{keep}) with some exceptions. The FAR values are low for for FAR_{right} and FAR_{left}, however it tends to be higher for FAR_{keep}. From the analysis of all results, high ACC, DR, and low FAR are generally achieved when the model is tested using different data sets resulting in an optimal model. This proves the generability of this method. Approach I also produces ACC and DR values higher than 80 % and low FAR values when the model is trained and tested.

Comparisons between the mean performances of approach I, approach II, and a conventional ANN (from Matlab toolbox) are presented using the same three test data sets in Table V. For an example, the overall accuracy here, is the mean overall accuracy obtained from the test data sets 1, 2, and 3 based on different approaches when the training data sets 1, 2, and 3 are used for training. The rest of the metric means are also evaluated the same way.

TABLE V: Mean comparison of ANN, ANN-based state machine approaches I and II.

States	Metrics	Conventional ANN [%]	Approach I [%]	Approach II [%]
Overall	*ACC*	81.95	84.09	92.82
Right	*ACC*	93.72	91.58	97.31
	DR	46.79	77.97	84.78
	FAR	4.17	7.03	2.11
Keep	*ACC*	82.37	84.67	92.82
	DR	85.77	86.88	93.81
	FAR	51.39	23.93	16.99
Left	*ACC*	87.81	91.97	95.51
	DR	40.98	69.29	81.55
	FAR	9.90	5.07	3.79

Based on mean performances shown in Table V, approach II has better performances than the conventional ANN and approach I for all metrics. Approach I also performs better than the conventional ANN for most of the metrics, with the exception of ACC_{right} and FAR_{right}. Approaches I and II perform significantly better particularly in DR_{right}, FAR_{keep}, and DR_{left}. Thus, this shows that the proposed approaches perform better than the conventional ANN. In addition, the mean elapsed times of each approach are also evaluated based on training process. The conventional ANN is the fastest one (16 seconds) and the developed approaches take longer time (approach I: 641 seconds, approach II: 2413 seconds).

VI. CONCLUSIONS

In this contribution, an ANN-based state machine driving behavior recognition model is developed to estimate lane changing behaviors based on the optimal parameters. Here, the estimations from ANN serves as a condition if a transition between states can occur in the state machine, producing the estimate of the proposed model. The model is trained using NSGA-II to generate optimal parameters. From the results, high ACC, high DR, and low FAR are achieved. The ACC and DR achieved for most of the test data sets are better than 80 % up to 98 %. While low FAR for different states are generally achieved, FAR_{keep} tends to be higher. Two approaches of the model are developed, whereby one model uses one ANN with the state machine based approach, while the other model uses three ANN. Comparisons between the conventional ANN and the proposed approaches are done. The proposed models perform significantly better than the conventional ANN especially in DR_{right}, FAR_{keep}, and DR_{left}. For future works, incorporating eye tracking variables like gaze quality to evaluate the model's performance will be considered. Another aspect to consider is applying the model for classifying driving styles during lane changes, such as aggressive, etc.

ACKNOWLEDGMENT

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

REFERENCES

[1] Driver-related causes of accidents involving personal injury in road traffic. Accessed: 2021-03-04. [Online]. Available: https://www.destatis.de/EN/Themes/Society-Environment/Traffic-Accidents/_Graphic/_Interactive/traffic-accidents-driver-related-causes.html

[2] V. Leonhardt and G. Wanielik, "Neural network for lane change prediction assessing driving situation, driver behavior and vehicle movement," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

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

[4] D. S. Hurwitz, H. Wang, M. A. Knodler, D. Ni, and D. Moore, "Fuzzy sets to describe driver behavior in the dilemma zone of high-speed signalized intersections," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 15, no. 2, pp. 132–143, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847811001021

[5] V. Mahajan, C. Katrakazas, and C. Antoniou, "Prediction of lane-changing maneuvers with automatic labeling and deep learning," *Transportation Research Record*, vol. 2674, no. 7, pp. 336–347, 2020. [Online]. Available: https://doi.org/10.1177/0361198120922210

[6] W. Mostert, K. Malan, and A. Engelbrecht, "Filter versus wrapper feature selection based on problem landscape features," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, p. 1489–1496.

[7] T. Nakamura, A. Maejima, and S. Morishima, "Driver drowsiness estimation from facial expression features computer vision feature investigation using a cg model," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2, 2014, pp. 207–214.

[8] R. David, S. Rothe, and D. Söffker, "State machine approach for lane changing driving behavior recognition," *Automation*, vol. 1, no. 1, pp. 68–79, 2020. [Online]. Available: https://www.mdpi.com/2673-4052/1/1/6

[9] A. Gill, *Introduction to the theory of finite-state machines*. McGraw-Hill, 1962.

[10] A. Fernández-Isabel, P. Peixoto, I. M. de Diego, C. Conde, and E. Cabello, "Combining dynamic finite state machines and text-based similarities to represent human behavior," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 504–516, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197619301708

[11] S. Haykin, *Neural networks a comprehensive introduction*. Prentice Hall, New Jersey, 1999.

[12] K. Förderer, M. Ahrens, K. Bao, I. Mauser, and H. Schmeck, "Towards the modeling of flexibility using artificial neural networks in energy management and smart grids: note," in *Proceedings of the Ninth International Conference on Future Energy Systems*, 2018, pp. 85–90.

[13] Q. Deng and D. Söffker, "Improved driving behaviors prediction based on fuzzy logic-hidden markov model (fl-hmm)," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 2003–2008.

[14] S. Lin. NGPM – A NSGA-II Program in Matlab v1.4. Accessed: 2021-02-11. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4

[15] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. USA: John Wiley Sons, Inc., 2001.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.