Development of reliable machine learning approaches to systems with structural change like human intention behavior and machine diagnosis

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Maschinenbau und Verfahrenstechnik
der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades
einer
Doktorin der Ingenieurwissenschaften
Dr.-Ing.
genehmigte Dissertation

von

Ruth R K David
aus
Kuantan, Malaysia

Gutachter: Univ.-Prof. Dr.-Ing. Dirk Söffker
Prof. Dr. Giuseppe D'Aniello

Tag der mündlichen Prüfung: 28. Mai 2024

DOI: 10.17185/duepublico/82057
URN: urn:nbn:de:hbz:465-20240618-151440-9

For I can do everything through Christ, who gives me strength - Philippians 4:13

Acknowledgements

This thesis is the result of the research work I have carried out at the Chair of Dynamics and Control (SRS) at the University of Duisburg-Essen during 2019 to 2024. I would like to take this opportunity to thank all those who have guided, helped, and motivated me throughout my doctoral journey.

First and foremost, I would like to thank Lord Jesus Christ for His blessings and grace that has led me in this process.

Next, I would like to convey my thanks to my supervisor, Univ.-Prof. Dr.-Ing. Dirk Söffker, for granting me the opportunity to pursue my Ph.D. research at the SRS chair, as well as for his guidance, valuable insights, and advices throughout this process.

I would also like to thank my second supervisor, Prof. Dr. Giuseppe D'Aniello for his constructive feedback and probing questions, which enhanced the quality of this thesis.

My gratitude also extends to the German Academic Exchange Service (DAAD) which has funded my Ph.D research. Also, a thank you to the relevant colleagues of SRS.

I would also like to take this opportunity to express my gratitude and thanks to my parents (my father and late mother) for their sacrifices, support, and love throughout my life, which have resulted in this achievement. A special gratitude to my sister and late brother as well.

Duisburg, May 2024 Ruth R K David

Kurzfassung

Industrie 4.0 konzentriert sich stark auf die Verbindung zwischen Automatisierung, maschinellem Lernen (ML), und künstlicher Intelligenz (KI). Der Einsatz von Automatisierung, ML und KI sind vielseitige interdisziplinäre Bereiche, die in vielen Systemen mit strukturellen Veränderungen und dynamischem Verhalten angewendet werden, beispielsweise in Transport-, Telekommunikations-, Luft- und Raumfahrt- und Militärsystemen. Systeme mit strukturellen Veränderungen zeigen im Laufe der Zeit je nach Betriebsumgebung Veränderungen in ihrem Verhalten, die den Zustand des Systems oder des Betreibers verändern können. Im Transportwesen beispielsweise haben fahrbedingte Unfälle im Straßenverkehr im Laufe der Jahre aufgrund des Verhaltens der Fahrer erheblich zugenommen. Um dieses Problem anzugehen, hat die Weiterentwicklung der Automatisierungstechnologie durch die kontinuierliche Entwicklung fortschrittlicher Fahrassistenzsysteme (ADASs) spielt eine Schlüsselrolle, um Fahrern bei der Durchführung sicherer Manöver zu helfen. Diese Assistenzsysteme unterstützen den Fahrer, indem sie Verhaltensweisen erkennen und abschätzen. Der Einsatz ML-basierter Modelle zur Generierung von Schätzungen und Erkennungen des Fahrverhaltens hat in den letzten Jahren stark zugenommen. Ein weiteres Beispiel für die Anwendung von ML auf andere Systeme mit strukturellen Änderungen ist die Schätzung der Verschlechterung von Lithium-Ionen-Batterien (LIBs). Prognosen und Gesundheitsmanagement (PHM) der Batterien sind daher wichtig für die Überwachung ihres Zustands, da Batterien mit der Zeit schwächer werden. Ähnlich wie bei der Fahrverhaltensanwendung hat die Entwicklung von Schätzmodellen mithilfe ML-basierter Ansätze an Popularität gewonnen. Gängige ML-Ansätze zur Entwicklung der Schätzmodelle sind Artificial Neural Network (ANN), Support Vector Machine (SVM) und Convolution Neural Network (CNN). Obwohl vielversprechende Ergebnisse erzielt wurden, bestehen gewisse Herausforderungen, wie z. B. ungeeignete Parameterwerte, das Extrahieren relevanter Informationen, Modelle haben individuelle Aufgaben und verschiedene Modelle weisen je nach Situation eine unterschiedliche Leistung auf, was zu suboptimalen Ergebnissen führt. In den letzten Jahren wurde der State-Machine-Ansatz aufgrund seiner Fähigkeit, Verhalten mithilfe diskreter Zustände zu modellieren, und seiner Flexibilität in verschiedenen Forschungsbereichen zur Abschätzung des Verhaltens eines Systems eingesetzt. Das Ziel dieser Arbeit besteht daher darin, ein auf Zustandsmaschinen basierendes Modell als neues interpretierbares ML-basiertes Modell für die Schätzung von Verhaltensweisen (als Zustände) in zwei Bereichen zu entwickeln: Schätzungen des Fahrverhaltens und Schätzungen des Verschlechterungsverhaltens von LIBs. Zustandsmaschinen sind im Gegensatz zu einigen ML-Ansätzen wie SVM, bei denen es schwierig ist, eine geeignete Kernelfunktion auszuwählen, aufgrund der einfachen Zustandserreichbarkeit des Modells ohne komplexe Berechnungen interpretierbar. Im ersten Teil der Arbeit wird die Einschätzung des Fahrverhaltens betrachtet. Das Modell wird aus verschiedenen Blickwinkeln betrachtet. Um die Verallgemeinerung des vorgeschlagenen Modells in anderen

Anwendungen mit Strukturvariablensystemen zu beweisen, wird das Modell im zweiten Teil auch für die Schätzung des LIB-Degradationsverhaltens (im Sinne der Schätzung des Kapazitätsschwunds) angewendet.

Für die Fahrverhaltensschätzungen modelliert die Zustandsmaschine drei Spurwechselverhaltensweisen unter Verwendung von Zuständen, die zwischen einander wechseln, um die Schätzungen zu entwickeln. Die berücksichtigten Spurwechselverhaltensweisen sind: Spurwechsel nach rechts (LCR), Spurhalten (LK) und Spurwechsel nach links (LCL). Um das Modell weiter zu erweitern, werden ein ANN und ein Hidden-Markov-Modell (HMM) einzeln mit dem Zustandsmaschinenmodell kombiniert, um das Spurwechselverhalten abzuschätzen. Das KNN ist für seine Robustheit und Lernfähigkeit bekannt, während das HMM für seine stochastischen Eigenschaften bekannt ist. Für den KNN-basierten Zustandsmaschinenansatz werden zwei verschiedene Modelle entworfen. Das erste Modell verwendet nur ein ANN, während das andere Modell drei verschiedene ANNs integriert. Für das HMM-basierte Modell wird ein verbessertes HMM mit einer Vorfilteranwendung in Betracht gezogen. Darüber hinaus wird auch die Implementierung verschiedener Sub-HMMs für unterschiedliche Eingabevariablen (Merkmale im Zusammenhang mit dem Fahrzeug und der Umgebung) durchgeführt. Auch Fahrmerkmale wirken sich auf die Erkennungsleistung aus. Daher werden Bewertungen verschiedener Merkmalstypen (als Modelleingaben) durchgeführt, um deren Wirksamkeit auf die Schätzleistung des Zustandsmaschinenmodells zu bewerten. Für die Bewertung werden Merkmale wie Umgebungs- (ENV) und Eye-Tracking- (ET) Merkmale berücksichtigt. Darüber hinaus ist bekannt, dass die Optimierung von Hyperparametern und Modellparametern die Leistung des Modells verbessert. Die Auswahl der geeigneten Optimierungstechnik und Auswahlbereiche auf der Grundlage verschiedener ML-Modelldesigns ist jedoch eine Herausforderung. Für die Schätzung des Fahrverhaltens werden sowohl die Bayes'sche Optimierung (BO) als auch der genetische Algorithmus (GA) angewendet, um die Hyperparameter zu optimieren. Darüber hinaus werden Modellparameter während des Trainings mithilfe des Non-Dominated Sorting Genetic Algorithm (NSGA-II) optimiert, da diese Parameter direkten Einfluss auf die Leistung des Modells haben.

Für die LIBs werden im Modell unterschiedliche Abbauzustände definiert, sodass die Zustände von einem zum anderen übergehen, um das Abbauverhalten und den Kapazitätsschwund abzuschätzen. Die mit jedem Zustand verbundene Kapazität wird mithilfe eines NARX-Modells (Nonlinear Auto Regressive Neural Network with Exogenous Input) geschätzt. Die Modellparameter werden ebenfalls mit NSGA-II optimiert. Zur Auswertung der Spurwechselverhaltensschätzungen werden Fahrdaten von Teilnehmern verwendet, die auf zwei verschiedenen Experimenten basieren und mit einem Fahrsimulator gesammelt wurden. Ziel ist die Entwicklung eines Modells mit optimaler Genauigkeit (ACC), Erkennungsrate (DR) und Fehlalarmrate (FAR), um die Wirksamkeit der verschiedenen Modelle zu validieren. Für die Schätzung des Kapazitätsschwunds von Batterien werden Daten aus vier Experimenten für die Anwendung des Modells genutzt. Für Leistungsbewertungen werden der mittlere quadratische Fehler (MSE) und der relative mittlere quadratische Fehler (RMSE) verwendet. Die Ergebnisse der Anwendung des Modells auf den Fahrverhaltensbereich zeigen, dass für die Kapazitätsschätzung von LIBs ein hoher ACC, DR und ein niedriger FAR erreicht werden, während im Allgemeinen niedrige RMSE und MSE erreicht werden. In bestimmten Fällen übertreffen die vorgeschlagenen Modelle die herkömmlichen Modelle in beiden Anwendungen. Darüber hinaus liegen die Zustandsschätzungen für beide Bereiche nahe an den tatsächlichen Zuständen. Basierend auf den Ergebnissen zeigt der

State-Machine-Ansatz als neuer ML-basierter Ansatz günstige Ergebnisse.

Abstract

Industry 4.0 profoundly focuses on the link between automation, machine learning (ML), and artificial intelligence (AI). The use of the automation, ML, and AI are versatile interdisciplinary fields applied in many systems with structural changes and dynamical behaviors, such as transportation, telecommunications, aerospace, and military systems. Systems with structural changes display changes in their behavior over a period of time based on the operating environment, which can change the system's or operator's state. For an example, in transportation, driving-related accidents on the road have increased significantly over the years due to drivers' behavior. To tackle this issue, the advancement of automation technology has played a key role in the continuous development of advanced driving assistance systems (ADASs) to assist drivers in performing safe maneuvers. These assistance systems assist drivers by detecting and estimating behaviors. Employing ML-based models to generate driving behavior estimations and detections has surged in recent years. Another example of applying ML to other systems with structural changes is the degradation of lithium-ion batteries (LIBs) estimations. Prognostics and health management (PHM) of the batteries are therefore important for monitoring their health (machine diagnosis), as batteries degrade over time. Similar to the driving behavior application, the development of estimation models using ML-based approaches has gained popularity. Common ML approaches used for developing the estimation models are Artificial Neural Network (ANN), Support Vector Machine (SVM), and Convolution Neural Network (CNN). While promising results have been achieved, certain challenges exist, such as unsuitable parameter values, extracting relevant information, models have individual tasks, and different models have varying performance depending on the situation leading to sub-optimal results. In recent years, the state machine approach has been used in various research areas for the estimation of a system's behavior due to its ability to model behaviors using discrete states and its flexibility. Hence, the objective of this thesis is to develop a state machine-based model as a new interpretable ML-based model for the estimation of behaviors (as states) in two domains: driving behavior estimations and degradation behavior of LIBs estimations. State machines are interpretable due to the model's easy state reachability without complex computations, unlike some ML approaches such as SVM, whereby it is difficult to choose an appropriate kernel function. The estimation of driving behaviors is considered in the first part of the thesis. The model is tackled from different angles. In addition, to prove the generalization of the proposed model in other applications with structural variable systems, the model is also applied for the estimation of LIBs degradation behavior (in terms of capacity fade estimation) in the second part.

For the driving behavior estimations, the state-machine models three lane changing behaviors using states that transition between each other to develop the estimations. The lane changing behaviors considered are: lane change to the right (LCR), lane keeping (LK), and lane change to the left (LCL). To further extend the model, an ANN and a

Hidden Markov Model (HMM) are combined with the state machine model individually to estimate the lane changing behaviors. The ANN is known for its robustness and learning capabilities, while the HMM is known for its stochastic properties For the ANN-based state machine approach, two different models are designed. The first model only uses one ANN, while the other model integrates three different ANNs. For the HMM-based model, an improved HMM with a prefilter application is considered. In addition, implementing different sub-HMMs for different input variables (features related to the vehicle and environment) are also performed. Driving features also affect the recognition performance. Hence, evaluations of different feature types (as model inputs) are performed to assess their effectiveness on the estimation performance of the state machine model. Features such as environmental (ENV) and eye-tracking (ET) features are considered for the evaluation. In addition, optimization of hyperparameters and model parameters is known to improve the performance of the model. However, selecting the appropriate optimization technique and ranges of selection based on different ML-model designs is challenging. For the driving behavior estimation, both Bayesian optimization (BO) and Genetic algorithm (GA) are applied to optimize the hyperparameters. In addition, model parameters are optimized using the Non-dominated Sorting Genetic Algorithm (NSGA-II) during training, as these parameters directly influence the performance of the model.

For the LIBs, different degradation states are defined in the model, such that the states transition from one to another to estimate the degradation behavior and capacity fade. Capacity associate with each state is estimated using a Nonlinear Auto Regressive Neural Network with Exogenous Input (NARX) model. The model parameters are optimized using NSGA-II as well.

For evaluating the lane changing behavior estimations, driving data from participants based on two different experiments collected using a driving simulator are utilized. The aim is to develop a model with optimal accuracy (ACC), detection rate (DR), and false alarm rate (FAR) to validate the effectiveness of the different models. As for the capacity fade estimation of batteries, data from four experiments are utilized for the application of the model. The mean square error (MSE) and relative mean square error (RMSE) are used for performance evaluations. The results from the application of the model to the driving behavior domain shows that high ACC, DR, and low FAR are achieved, while low RMSE and MSE are generally achieved for the capacity estimation of LIBs. In certain instances, the proposed models outperform the conventional-based models in both applications. In addition, the state estimations are close to the actual states for both domains. Based on the results, the state machine approach shows favorable results as a new ML-based approach.

Contents

List of Figures

List of Tables

List of abbreviations

AI	Artificial Intelligence
ACC	Accuracy
ADAS	Advanced driving assistance system
ANN	Artificial Neural Network
ANN-SM	Artificial Neural Network-based state machine
BO	Bayesian optimization
CNN	Convolution Neural Network
DR	Detection rate
ENV	Environmental variables
EoL	End-of-lifetime
EoD	End-of-discharge time
ET	Eye-tracking variables
FAR	False alram rate
FN	False negative
FP	False positive
GA	Genetic algorithm
HMM	Hidden Markov Model
HMM-SM	Hidden Markov Model-based state machine
LCL	Lane change to the right
LCR	Lane change to the left
LIB	Lithium-ion batteries

LK	Lane keeping
ML	Machine learning
NSGA-II	Non-dominated sorting genetic algorithm
NARX	Nonlinear Autoregressive Network with Exogenous Inputs
RuL	Remaining useful lifetime
SM	State machine
SoH	State-of-Health
SVM	Support Vector Machine
TN	True negative
TP	True positive

List of mathematical motations

Symbols

A	State transition probability matrix
B	Observation probability matrix
Bl	Blink
C_x	Screen coordinate (x-axis)
C_y	Screen coordinate (y-axis)
d_f	Distance to vehicle in front
d_{fl}	Distance to vehicle in left-front
d_{fr}	Distance to vehicle in right-front
d_{bl}	Distance to vehicle left-behind
d_{br}	Distance to vehicle right-behind
d_b	Distance to vehicle behind
F_{Blink}	Blink frequency
G	Gearbox
I	Indicator
l	Current lane number
N_{Screen}	Screen number
O	Observation sequence
P	Final probability of different combinations of sub-HMMs
a_{acc}	Accelerator pedal position
a_{brake}	Brake pedal position
a_{st}	Angle of steering wheel

$Saccade$	Saccade
S	Hidden state sequence
TTC_f	Time to Collision to vehicle in front: $d_f/(v_{ego} - v_f)$
TTC_{fl}	Time to Collision to vehicle in left-front: $d_{fl}/(v_{ego} - v_f l)$
TTC_{fr}	Time to Collision to vehicle in right-front: $d_{fr}/(v_{ego} - v_f r)$
TTC_{bl}	Time to Collision to vehicle left-behind: $d_{bl}/(v_{ego} - v_b l)$
TTC_{br}	Time to Collision to vehicle right-behind: $d_{br}/(v_{ego} - v_b r)$
TTC_b	Time to Collision to vehicle behind: $d_b/(v_{ego} - v_b)$
V	Observation sequence
v_{ego}	Velocity of ego-vehicle
v_f	Velocity of vehicle in front
v_{fl}	Velocity of vehicle in left-front
v_{fr}	Velocity of vehicle in right-front
v_{bl}	Velocity of vehicle left-behind
v_{br}	Velocity of vehicle right-behind
v_b	Velocity of vehicle behind
α	Heading angle of ego-vehicle
π	Initial probability distribution
λ	HMM parameter

1 Introduction

Systems with structural change show changes in the behaviors or states over a period of time. For example, in a technical system, this can be interpreted as being at different states of health or different conditions, as in [SR17] which monitors the surface conditions of a hydraulic system. In this thesis, the structurally variable systems are human driving behaviors and degradation behaviors of lithium-ion batteries (LIBs). Driving behaviors change over a period of time (like lane changing behaviors) and are affected by the conditions in which they are executed, the situation, the characteristics of the driver, and the intended goals of the driver. Similarly, the degradation of batteries over a period of time can be interpreted using different health states to show the changing behaviors of the battery's health (machine diagnosis). For both applications, machine learning (ML) plays an important role in the estimation of changing behaviors. For driving behaviors, these estimations are necessary for the development of driving assistance systems, while for the estimations of degradation behavior of LIBs are important for the health management of battery-operated systems. The objective of this thesis is to apply a newly developed state machine model (as an ML model) for estimations in both domains.

1.1 Motivation and problem statement

The first part of thesis focuses on the estimations of human driving behaviors. Fatalities due to road accidents have increased in recent times. According to the World Health organization (WHO), approximately 1.3 million people have faced fatal traffic accidents each year [Wor18]. Another report by the European commission showed that around 2600 people were killed in road accidents in 2022 within the EU, which is a 3 % increase from 2021 [Eur23]. While the National Highway Traffic Safety Administration has released a report showing a small decrease of 0.3 % in road accident fatalities in 2022 compared to 2021 within the US [Nat23], these reports highlights the pressing road safety problems. Upon further analysis, majority of these accidents are due to human driving behaviors. A report by the Department of Statistics in Germany show that one of main causes of road accidents in 2022 are related to entering or leaving road from premises and turning mistakes, which amounts to 15.4 % of accidents [Sta22]. Another major contributor is the failure to yield the right way, which amounts to 13.4 % of the accidents [Sta22]. To tackle this problem, driving assistance systems to assist drivers on the road have been implemented. Thus, the research based on driving behaviors, intentions, and traffic safety is of growing importance for the continuous improvement of these systems. A major part of this thesis focuses on the recognition of driving behaviors/ intentions, particularly lane changing behaviors/ intentions. As driving behaviors are individual, establishing prediction and recognition models into the assistance systems based on the individual behaviors is the focus for the ongoing development of individualized driving assistance systems.

The driving behavior/intention prediction and recognition models are mainly developed using ML approaches, such as Artificial Neural Network (ANN) [LW17], Support Vector Machines (SVM) [KPLL13], Hidden Markov Models (HMM) [BED08], and Convolution Neural Network (CNN) [LMH17]. These ML approaches are used because they are able to learn and recognize patterns from current situation to generate an output/estimations for future instances. However, the ML-based models do not always result in optimal predictions and recognition due to the various factors:

- Lack insight of relevant information
- Development of optimal parameters and hyperparameters
- No free lunch theorem, as one method cannot be definitely stated to be better than others
- Each model has individualized tasks, as certain ML-models work better in limited complexity (such as less number of classes for classification).

Different methods are used to address these issues:

- Combining multiple ML approaches
- Applying a new approach that was not previously used in a particular field to develop an estimation model
- Feature extraction: while certain ML algorithms incorporate feature extraction as part of this procedure (such as basic neural networks), often further feature extraction techniques/ feature extraction layer may be required.
- Parameter and hyperparameter optimization of the model (selection of an appropriate algortihm to optimize the parameters of the model (increasing the search space) to fit the obejctive function well

An example of combining approaches is given in [DF16], whereby ANN is combined with SVM to develop a lane changing behavior prediction model. The estimation performance is further compared with estimations based on Bayes classifier and decision tree, which showed that the proposed approach outperforms the other two. This is also observed in [IQP+19], whereby CNN is combined with Long-Short Term Memory (LSTM) for abnormal driving behavior recognition and lane changing intention recognition, respectively. In addition, LSTM in combination with a regression method is modeled to estimate the time-to-lane change in [DFBH17].

A rather new approach known as the state machine approach, which was previously used for the development of estimation models in tribology experiments [BS17] and modeling the degradation of plant [KS20], is applied in this thesis to develop the driving behavior recognition model (situation recognition). As the model is known to model discrete behaviors, it makes sense to apply this model the discrete driving behaviors. Behaviors are modeled using discrete states that transition between each other based on the model's inputs and transition conditions [Gil62] to estimate the next driving behavior. The state machine model is also flexible and easy to define. Unlike certain ML approaches which require complex computations (difficulty seelcting kernels, requires many layers, black

box nature, etc) which leads to higher computational burden, the state machine can be executed quickly and is interpretable. The previous work in [BS17], applies the state machine approach to model the degradation states of a technical system to evaluate the health of the system. In [KS20], the state machine models the different plant behavior as states based on water stress (water availability). The previous contributions showed that the state machine model is able to model structurally variable behaviors (changes in degradation behavior of a system and changes in plant growth), however this has not been applied to other similar dynamical behaviors such as driving behaviors, which exhibit switching behaviors depending on the situation and environment. To address this, a state machine-based model as a new ML model is developed for the recognition of lane changing behaviors in this thesis. As stated in other contributions like [DS22c], combined approaches tend to perform better than a single conventional approach. The similar technique is applied in this thesis, by extending the state machine approach by combining it with other common approach such as ANN and HMM.

Features that are relevant to the driving scene are important for the development of an optimal lane changing behavior recognition model. Some examples of driving-based features are environmental (ENV), eye-tracking (ET), head tracking, and physiological features. Environmental features are features describing the state of the ego vehicle and driver's operation characteristics, such as time to collision (TTC), vehicle speed, and acceleration. These variables describe the relationship between the ego vehicle and surrounding vehicles in the environment. The ENV features tend to influence driver's decisions the most [DHBS20]. The ET features are features related to the eye movements and gaze, while physiological features can be used to determine the mental state of the driver (fatigues, etc). Previous contributions have shown that different features affect the driving behavior/ intention estimation performance differently. In most research contributions such as [TTK08], ENV features such as acceleration, velocity differences, lateral positions, etc. are mainly used to predict a lane change. Head tracking motions are also proven to be more relevant for driving behavior estimations compared to ET in [DT09]. Nevertheless, in [SL02], the authors emphasize on the importance of ET features. Overall, certain models may perform well using certain features, while other models may under perform using these features as shown in [DHBS20]. Therefore, the selection of appropriate features and models is a recurring challenge for the estimation of driving behaviors. This thesis also aims to study the effect of different features (ENV and ET features, in particular) on the newly developed state machine approach for lane changing intention recognition.

Model parameters and hyperparameters of ML models also affect the performance of the model. Model parameters directly affects the output, while hyperparameters indirectly affect the output. Model parameters are fitted by the data during the training process of the ML model. In this thesis, the Non-dominated Sorting Genetic Algorithm (NSGA-II) is considered for model parameter optimization. The hyperparameters control the training process, thus affecting the selection of model parameters. Hence, optimization of these parameters are important. Many research contributions have already tackled this using the conventional ML methods such as [HK21], whereby hyperparameters of a LSTM are optimized using Bayesian optimization (BO) for detection of driving styles. However, there is still a lack of studies that focuses on hyperparameter optimization on mixed or combined ML-based methods. To bridge this gap, this thesis focuses on the hyperparameter and parameter optimization of combined ML-based approaches, by combining the state machine approach and other ML algorithms. The hyperparameter optimization techniques

used are the Bayesian optimization (BO) and Genetic algorithm (GA), as the techniques are automated hyperparameter tuning methods which are proven to be effective global optimization methods [Hol73] [WCZ+19].

The goals of the first part of this thesis are as follows:

- Development of a state machine-based approach for recognition of lane changing behaviors
- Extension of the state machine approach with ANN and HMM
- Evaluation of different features and their importance for the driving intention estimations
- Optimization of parameters and hyperparameters of the developed state machine-based approach

The second part of the thesis focuses on applying the approach for modeling similar switching behaviors within the prognostics health management (PHM) field, specifically modeling the degradation behavior of LIBs. The use of batteries are increasingly used in many applications, such as electrical vehicles (EV), telecommunication equipment, and military equipment due to its light weight properties, long cycle life, high operating voltage, low self discharge, and low maintenance. The use of batteries in electrical systems leads to material aging with time due to the electrochemical reactions during operation. In addition to material aging, the capacity of the batteries as well as the remaining useful lifetime (RuL) also decreases with time, which leads to performance degradation of the battery (possible damage of the battery-operated system). Several stress factors such as battery temperature, ambient temperature, and C-rate in the loading profiles influence the aging process. Various health indicators such as RuL and capacity fade are determined to evaluate the battery's health. The service life can be extended or a system failure can be avoided by maintenance measures precisely matched to the function loss or by changing usage strategies. The State-of-Health (SoH) of the battery can be determined by the application of lifetime models. The health indicators RuL and capacity fade can be estimated by these models based on the stress factors. Therefore, predicting the health of the battery has gained attention in recent years. In order to maintain the functionality and safety, it is helpful to develop an accurate lifetime model to represent the dynamic properties. The lifetime models are usually established using model-based [XC17] or data-driven approaches [ZZW22]. The model-based approaches are often based on physical and chemical properties, while the data-driven approaches are based on historical data of the battery's charging and discharging process. The use of ML-based models have also gained interest for the estimation of capacity fade. One of the main challenges faced in the development of lifetime models is different LIBs have different lifetime expectancies affected by the stress factors [KJ17] [WWK+14]. In addition, the limited aspects of various approaches make it challenging to select the appropriate approach. For example, certain model-based approaches are developed based on specific operating conditions, as shown in [ZZZ+16] and [ACW16]. Also, some model-based approaches such as in [LSB+17] require complicated electrochemical information and experiments. Therefore, the electrochemical knowledge may not be sufficient in certain instances making it difficult for modeling the degradation of complex systems. Both model-based and data-driven approaches tend to have difficulties in modeling complex non-linear degradation behaviors [XC17]. Hence,

there is a lack of suitable lifetime models in current literature for the estimation of the health indicators. A combination of both model-based approach and data-driven approach is tackled in [Koz03].

Only a few approaches in literature have considered the damage states associated with the multi-switching degradation behavior of the battery. Unlike previous approaches, the switching degradation behavior are modeled using states based on the state machine approach in this thesis. As this approach has shown promising results in previous works, applying this approach to similarly structured systems is the goal. A combined model of Nonlinear Auto Regressive Neural Network with Exogenous Input (NARX) and state machine is considered here. The NARX is selected as it is able to handle complex non-linear data well. The model transitions from one degradation state to another based on defined conditions for the estimation of the capacity fade and the degradation progression.

The goals of the second part of this thesis are as follows:

- Development of a NARX-based state machine model for capacity fade estimation
- Parameter optimization for the proposed model
- Modeling the different states of degradation

1.2 Thesis organization

Some parts of this thesis are published in journal papers [DRS20] [DS22b] [DS23c] or in conference proceedings [DRS21] [DS22a] [DS23a] [DS23b] [DS23d]. This thesis is organized as follows:

In chapter 2, the fundamentals and the background of the driving behavior estimations models and LIB degradation estimation models are given. In the driving behavior estimations models, the different driving behaviors and intentions are distinguished. In addition, the features that play an important role in the driving decision making are discussed. Selecting appropriate features (as inputs) for the developed model is important to improve the estimation performance. As previously stated, driving behavior estimation models are mainly based on ML approaches. Accordingly, the different ML approaches commonly used for the development of these models are discussed in this section. The optimization of model parameters and hyperparameters are introduced. Different techniques used in previous research are summarized. To provide some fundamental understanding about the second part of thesis, the background on the LIB degradation estimation models is given. Here, the health and aging of the LIBs are explained in detailed as well as the different stress factors that contribute to the aging. Furthermore, a review about the lifetime models used for the aging/degradation estimation of the batteries is summarized.

In chapter 3, the fundamentals of the state machine approach (which is the new proposed approach) is introduced. The different contributions that have developed estimation models based on the state machine approach is discussed.

In chapter 4, the application of the state machine approach for the recognition of lane changing behaviors is introduced by describing the model structure. Following that, the optimization of the model parameters is given. In addition, the extension of the state machine model by combining ANN and HMM is introduced. For each of the extension,

different models are established based on the process and input variables. Different features such as ENV and ET are evaluated in chapter 4 as well to study their influence on the performance of the state machine model. In addition, the optimization of hyperparameters related to newly developed model is explained in this chapter. The BO and GA are applied to study the effects on the performance of the model.

In chapter 5, the application of the state machine model for the degradation behavior estimation (capacity fade) of LIBs is described. A combined model of NARX and state machine is developed to estimate the capacity fade. Different degradation states are defined and estimated here.

Experimental design and data processing are detailed in chapter 6 for the both the driving behavior and LIB areas. The training and test process are also included. In addition, the results based on both areas are presented and analyzed.

Finally, in chapter 7, a summary and conclusion about the developed approach's performances for both the driving behavior and capacity fade estimations are summarized. An outlook and limitations related to this thesis are also detailed in this section.

2 Theoretical background

In this chapter, the background about the prediction and recognition of human driving behaviors are discussed. In section 2.1, an overview of driving behaviors/intentions definition and the tasks are given. In addition, the driving behavior and prediction models are presented. The different types of behaviors, the features affecting these behaviors, and the ML approaches developed in various studies are presented. As the models' performance are also affected by the components of the models, the importance of hyperparameters and model parameters is highlighted in this section. In the section 2.2, the model-based and data-driven (includes ML models) models developed for the estimation degradation of the lithium-ions batteries are presented. The theoretical information about the battery aging, the different factors that affect the aging process, known as stress factors are detailed as well.

Parts of the contents and tables presented in this chapter are modified based on publication [DS23c].

2.1 Prediction and recognition of human driving behaviors

In current literature, various prediction and recognition models have been developed for the estimation of driving behaviors. These studies emphasize that driving behaviors are influenced by many factors such as age, gender, environment, driver's characteristics, etc. Driving behaviors (lane changing behaviors, speed change, braking, accelerating), driving styles (normal, aggressive), drunk, drowsiness, and fatigue are typically estimated in literature using the models. For an example in [VLMT13], acceleration, turning, and braking are detected using features related to driver's driving characteristics, such as speed. In addition, the behavior intent estimation is also researched in certain studies, such as [TSLL15]. In these studies various approaches, mainly using ML-based approaches are considered to develop the estimation models. Hence, in this thesis a summary of the developed models will be detailed as well their comparison in terms of functionality and performance.

In this thesis, the focus is on lane changing behaviors as improper lane changing behaviors is one of the major causes of accidents, [Sta22]. Lane changing behaviors can be distinguished into three different types: merge/non-merge behaviors [SZJZ18] [DF16], making a turn at an intersection [VLMT13], and overtaking a vehicle in front [LW17]. Here, the overtaking a preceding vehicle is taken into account. Currently, there are a limited number of review papers that focuses only on different approaches developed for the prediction and recognition of lane changing behaviors. Previous reviews within this field focused on estimations of various driving aspects such as speed, trajectory, driving styles, and drowsiness using ML-based approaches [LZT+14] [MT16] [Kan13]. In [Kan13], different methods to detect drowsiness and distracted driving are discussed. On the other hand, there

are review contributions that focus only on one specific ML-based approach for estimations of driving behaviors, as in [DS22c]. While many studies have considered developing models for the estimations of lane changing behaviors in highways or urban roads, there is still a lack of summary of different of the models based on their functionalities, advantages, disadvantages, performance, driving features' influence, and model components. This summary can help bridge the gap for future research works when selecting and developing driving behavior estimation models based on different characteristics. Therefore, in this section the aim is to summarize the different driving behavior models, the driving features that affect the driving behaviors and estimation models, parameters, and hyperparameters that influence on the estimation performance.

2.1.1 Human driving behaviors and intentions

In this section, the definitions of behaviors and intentions are distinguished, as these are mainly researched in the transportation area.

Driving behaviors are based on human cognition to control the operation and movement of the vehicle, which are influenced by driver's characteristics, emotions, and environment. The key aspects of driving behaviors that are relevant include interaction with the environment, actions that are generated independently, and actions that executed in targeted manner based on the current state. Hence, driving behaviors do not only constitute of the real-time actions, but also reaction of from the behavior patterns and trajectories [Den20]. Here, lane changing behaviors and the estimation of these behaviors are of interest due the aforementioned reasons.

Intention is defined as the mental process that occurs before executing a specific behavior [Car07]. Driving intention recognition evaluates if a sequence of driving actions has a specific intention [BD11]. This enables early detection of behaviors which can improve the driving safety [HHW20]. The recognition/prediction of driving behaviors/intentions is often based on ML algorithms due to the algorithms' ability to learn from given behaviors to provide estimations when similar situations occur.

Both the prediction and recognition driving behaviors and intentions are examined in this thesis for the advancement of the driving assistance systems. The hierarchical cognitive human behaviors in a traffic environment are described in [Mic85], whereby the aim is to show the relation between human beings and the environment to satisfy mobility needs for various tasks [Mic85]. Table 2.1 shows the various roles of human behaviors in a traffic environment to perform various tasks [Mic85]. Four behavioral levels are distinguished (related to the role of the human): road user, transportation consumer, social agent, and psycho-biological organism.

2.1.2 Prediction and recognition models

Prediction and recognition models are also known as estimation models. These models generally consist of three aspects: inputs, processing, and outputs. For a typically driving behavior model, inputs can include the driving variables such as the ego vehicle and surrounding states, the environmental conditions, and the interaction between the ego vehicle and surrounding vehicles as these variables highly influence the driving maneuvers and decision making on the road. In addition, the driver's individual characteristics, such as eye-movements or physiological traits also influence the drivers behaviors. Driver's

Table 2.1: Hierarchical structure of problem solving tasks in traffic and transportation [Mic85]

	Behavioral level			
	1	2	3	4
Human quality (problem solver)	Road user	Transportation consumer	Social agent	Psycho-biological organism
Problem to be solved	Vehicle control	Trip making	Activity pattern (communication)	Basic needs satisfaction
Task environment	Road	Road network	Socio-economic structure	Environment
Task aids	Vehicle, signs, etc.	Transport mode	Transport system	Culture technology

experience or driving patterns also influence the driver's decision when in familiar and unfamiliar driving situations. The estimation models analyze and correlate these information (processing) in order to generate an output (behavior estimations) [Mic85]. The prediction and recognition of the driving behaviors are based on the current driving behavior of the ego vehicle, its interaction with the other vehicles, and environmental conditions.

One of the first ideas of behavioral modeling is developed by [Mic85], which considers two phases of models described in Table 2.2. In the first phase, driving behaviors can be distinguished into: behavioral (input-output) and physiological (internal state). For the second phase, the behaviors can be distinguished into: taxonomic and functional models, representing dynamic and non-dynamic interaction of components. The input-output model represents observable behavior and mathematical function. The physiological model represents the mental process of the driver.

Taxonomic models constitute of a collection of facts, that are grouped in different sets according their relevance. The components of the model do not interact dynamically. Typical taxonomic models for driving behaviors are trait models and task analyses. The trait models focuses on the relationship between the driver's characteristic and occurrence of accidents. These models focus on the influence of external characteristics (age, gender, etc) and internal characteristics (attention, emotions, etc) on accidents [Mic85]. Task analyses tend to focus on the driving tasks, observations, ability requirements, and behavioral requirement for performing a tasks [Mic85], which is typical for current modeling of driving behaviors. Different sets of driving tasks are used to describe a collection of facts as part of the task analyses. Functional models on the other hand have components that interact dynamically with one another.

The Rasmussen model is perhaps the most well-known model for modeling different behaviors [Ras83]. The Rasmussen model consists of three levels of behavioral skills to distinguish between experienced or novice operators (here: drivers). The three levels are skill-based, rule-based, and knowledge-based behaviors. The skill-based behavior is a routine-based behavior that requires very low attention and memory as well as unconscious reasoning. The process of carrying out the behavior is described as automatic, whereby a person reacts instantaneously to a very familiar situation [Ras83]. The behavior has a low error rate, high repetition accuracy, and is learned through sensory. The rule-based behaviors are performed by following specific rules. Based on a recognized/ familiar

Table 2.2: Driver behavior models [Mic85]

Phases	Taxonomic	Functional
Input-output	Task analyses	Mechanistic models Adaptive control models: - Servo-control -Information flow control
Physiological (internal state)	Trait models	Motivational models Cognitive (process) models

situation, the appropriate rules are applied to carry out the relevant tasks [Ras83]. The attention and working memory here are higher and has a certain reasoning. The behavior is also less automated compared to the skill-based. In general, the behavior is learned through practice, which are easily repeated with moderate error rate. The knowledge-based is the behavior when faced with a new situation [Ras83]. The behaviors are hardly repeatable with a high error rate. Carrying out a task requires high attention and consciousness. Hence, the process is not an automated process. Behaviors at knowledge-based is mainly required for problem solving tasks.

2.1.3 Features influencing driving behaviors

Driving behaviors are the actions carried out while driving, affected by features related to environmental conditions and driver's characteristics. To analyze the behaviors, different variables (features) such as environmental, eye-tracking, head-tracking, and physiological variables are considered. The variables provide information about the driver's, ego vehicle's, and surrounding vehicle's state. In research contributions, these variables are used as inputs for prediction and recognition models. For lane changing prediction and recognition, environmental variables are mainly used as the environmental conditions generally influences the driver's decision making process. However, the eye/head-tracking and physiological variables also play a role here. In general, there are two methods of collecting the features, either using sensors installed to the car or using sensors on the side of the road [VLMT13]. For the sensors installed to the car, these are either sensors embedded to the vehicle, such as Controller Area Network (CAN) bus or additional sensors attached to the vehicle like smartphones [LC12]. Feature selection techniques are often used in some contributions to select the most suitable variables, while others use automatic feature selection process within the ML approach/deep learning methods. Hence, here the different input variables used for the development of lane changing prediction and recognition models are summarized.

Environmental

Environmental (ENV) variables provide information that affect the driver's decision based on the relationship between the ego vehicle and surrounding vehicle. The environmental variables are distinguished into two categories: i) variables related to the ego vehicle's and surrounding vehicle's state and ii) variables related to the driver's operational information. The state of vehicle variables provide information about the relationship between the different vehicles as well as the driving environment. Typical vehicle state variables considered in literature are acceleration, vehicle speed, and time to collision (TTC). When

the driver realizes an unexpected maneuver, driver's operational variables are needed to provide vehicle control information for a better understanding of the maneuver. This includes steering wheel angle, position of acceleration/brake pedal, and engine speed. The variables are collected using different sensors such as CAN bus, mobile phones (including GPS), accelerometer or cameras/ LIDARS. As mentioned, most literature research uses environmental variables for the prediction and recognition of lane changing behaviors. In [DSLL15], a HMM-based approach is proposed for predicting lane changing intentions using the driver's operation variables (steering wheel, gas, and brake pedal positions) and the vehicle state variables (acceleration, yaw rate, and velocity). The maneuvers considered include left/right lane change, turn right/left, and stop /non-stop on highway and urban environments using a driving simulator. In [HJ9], a framework is developed for the prediction and recognition of lane changing behaviors. Here, the driver's characteristics are first estimated using an optimization-based technique. Then, based on the estimated characteristics, lane changing behaviors are predicted using a neural network-based approach. Here, environmental variables such as longitudinal position, velocity, and lane-number are considered for the estimation of the driver's characteristics and lane changing behaviors. A lane change intention system is proposed in [MTWR05] using the Bayesian Learning method by considering environmental variables based on vehicle and lane position data. These ENV variables are also considered for the detection of drunk and fatigue driving in [LWWX19] using HMM, which generated an accuracy (ACC) of 88.77 %. A summary of the environmental variables as well as the literature examples that uses these features are given in Table 2.3.

Table 2.3: Summary of environmental variables

Input variable	Literature
Environmental: operational and vehicle state steering wheel, gas, brake pedal position, lane position , longitudinal position, and velocity	[DSLL15] [HJ9] [MTWR05] [LWWX19]

Eye and head-tracking

Eye and head-tracking variables are also used in studies related to the estimation of lane changing behaviors. Eye-tracking (ET) variables are collected using an eye tracker devices or cameras to provide information on eye movements and gaze in different driving situations. The main ET variables used in driving research are eye fixations and saccades, [VTMB20]. Head-tracking variables are based on the head poses and head movements of the driver which are usually captured using camera sensors and eye-trackers. Head-tracking variables generally consist of head rotation, head angles, and head positions. The aim of these devices is to monitor the behavior of the driver so one can deduce the awareness and intent of the driver. The eye and head-tracking variables provide information on the driver's awareness of the surroundings/situation (such as maneuvering at an intersection) and how the driver intends to proceed in a given situation, [DT09]. Evaluating the reliability of eye gaze and head dynamics to determine the intention of the driver is studied in [DT09]. Based on the results, head dynamics are more useful than eye gaze information for the intention determination.

Head-tracking variables are used in [LW17] to analyze the driving maneuver preparation for a lane change prediction based on ANN. Here, the head's rotation around the vertical axis, lateral movement, and head tracking's confidence value are used as input variables. However, in certain models the inclusion of eye tracking variables increases the estimation performance. For an example in [LRW16], a Bayesian Network-based approach is proposed to predict lane changing intentions. Here, ET and vehicle data are collected to be used as inputs. The performance of the model based on the fusion of both eye tracking and vehicle related variables and individual variable sets are evaluated. From the results, the Bayesian model using the fused set performs better than when using the individual set. The usefulness of eye tracking information for designing a HUD-based warning indicator is studied in [YLS+17]. Eye fixation areas and fixation moving paths frequency are analyzed in the lane changing situations. In [DHBS20], the role of ET information for the prediction of lane changing behaviors based on CNN, HMM, Random Forrest (RF), and SVM are evaluated to develop an assistance system. From the results, it can be concluded that performance of integrating the ET data depends on the prediction ability and choice of the machine learning algorithm. In addition, driver's decisions are mainly based on environmental factors. Using only ET variables with the mentioned approaches produced poor ACC values in [DHBS20]. On the other hand, CNN, HMM, and RF-based models produced slightly better results when using a fusion of ET and ENV variables. However, the SVM-based model performs better when using only environmental variables [DHBS20]. The eye and head-tracking variables such as eye movements, blinking frequency, and head movements are also used for the detection of fatigue and drunk driving in [WY10] [JXP22] [QLH12]. The time series method is used for the detection of fatigue driving in [JXP22], while DBN is used for both fatigue and drunk driving detection in [WY10] which achieved high detection rates. For [QLH12], an embedded HMM is used for the detection of fatigue driving, which generated an ACC of 91.6 %. A summary of the eye and head-tracking variables is given in Table 2.4.

Table 2.4: Summary of eye and head tracking variables

Input variables	Literature
Eye-tracking: eye fixations and saccades	[VTMB20] [DT09] [LRW16] [YLS+17] [DHBS20] [WY10] [JXP22] [QLH12]
Head-tracking: head's rotation, lateral movement head tracking's confidence value	[DT09] [LW17] [WY10] [JXP22]

Physiological

Physiological variables provide information about the driver's characteristics, patterns, and mental state. Common physiological variables in driving research include electrocardiogram (ECG), electroencephalogram (EEG), pulse rate, blood alcohol concentration (BAC), etc. collected using bio- sensors. Physiological information are mainly used as inputs for the detection of fatigue driving and drunk driving. In [SLZ+11] and [PLKR11], physiological variables such as EEG, ECG, heart rate, and BAC are used to estimate fatigue behavior. In [SLZ+11], an SVM-based model is developed which generated an ACC of 87.5 % for the fatigue detection, while in [PLKR11], an ANN-based model is developed which generated an ACC of 90 %. However, in certain cases it can be used for predicting lane changing behaviors as shown in [MKW+15]. The authors state that in previous studies only environmental variables have been considered, thus the aim is to use physiological variables to study and evaluate the prediction performance. In this novel approach, a Granger causality test is used as a feature selection technique and neural network is used for the classification of the lane change. Here, only physiological variables such as ECG, galvanic skin response (GSR), and respiration rate (RR) are considered as inputs. Based on the results, a true positive rate (TPR) of 70 % and a false positive rate (FPR) 10 % are achieved for 30 lane changing and 60 non-lane changing events. In [LWWX19], physiological variables are used also within the context of lane changing behaviors. While this work does not predict lane changes, it considers predicting the risk during a lane change using a HMM-based approach. The authors consider ENV, ET, and physiological variables. The influence of eye movement, heart rate variability, and vehicle dynamic variables on the driving risk are evaluated using a two factor indicator analysis technique. The HMM results showed an ACC value of 90.67 % between the predicted risk and perception of the drivers. The summarized explanation of the variables is given in Table 2.5.

Table 2.5: Summary of physiological variables

Input variables	Literature
Physiological: ECG, EEG, BAC, GSR, RR, pulse rate, and heart rate variability	[SLZ+11] [PLKR11] [MKW+15] [LWWX19]

The review of different features show that ENV, ET, and occasionally physiological variables are generally required for lane changing and speed changing estimations. Nevertheless, most literature consider the use of ENV variables as these variables affect the decisions the most as well as they show the relationship between the vehicles and surroundings. The physiological variables are mainly used for detecting drunk and fatigue behaviors, which are collected using bio-sensors. The ET is relevant for the detection of these behaviors using blink frequency collected by eye-trackers or cameras [JXP22]. The ENV variables related to the lateral and longitudinal movements can be used for the detection of drunk and fatigue driving as well [CRKK15] [DTB+10]. In Table 2.6, the different types of features used for the estimation/detection of different driving behaviors and driving styles are given.

Table 2.6: Summary of different features for estimation or detection of different driving behaviors

Driving behaviors	Feature type (input)	Features	Approach	Performance	Literature
Lane changing behaviors	ENV, head-tracking	Velocity, yaw rate, brake pressure, gas pedal	ANN	AUC (0.972) FAR (0.018)	[LW17]
Lane changing behaviors	ENV	Lane position, traffic volume	Markov chains	Difference (0 %-8.1 %)	[MD20]
Risk of Lane changing behaviors	ENV, ET, physiological	Various	HMM	ACC (90.67%)	[LWWX19]
Aggressiveness	ENV	Lateral and longitudinal acceleration, yaw rate distance between ego and front vehicle, speed between ego and front vehicle	Ensemble learning	ACC (90.50 %) $F_1 Score$(90.42%)	[WWH+22]
Speed	ENV	Vehicle mass, distances, inertia, wheel stiffness	Kalman Filter	ABS (0.126 m/s -0.46 m/s)	[ZXX21]
Drunk and Fatigue	ENV	Distances, lane position, acceleration, velocities	HMM	ACC (88.77 %)	[CRKK15]
Fatigue	ET	Eye closure time, blinking frequency, eye closure rate, mouth opening frequency, head misalignment time	Time series	ACC (97.8 %)	[JXP22]
Fatigue	Physiological	EEG and ECG	SVM	ACC (87.5 %)	[SLZ+11]

2.1.4 Machine learning approaches for prediction and recognition

Artificial Neural Network

Artificial Neural Network (ANN) is a well known ML approach developed for the classification and pattern recognition tasks in recent years. The ANN model is designed based on the functionality of biological neural structure. The network consists of interconnected layers of neurons that processes information from large data sets to produce outputs to classify the data. A basic ANN consist of an input layer, hidden layers, and an output layer. The data are given to the network in the input layer. For the development of the models, each variable is represented as an input neuron. This information is passed to the neurons in the hidden layers for processing to produce outputs. In a driving behavior estimation model, each neuron in the output layer represents a specific class of driving behaviors such as the different lane changing behaviors. The connection between the neurons in different layers have specific weights associated with it, which are adjusted to minimize error margins. All neurons have a bias value with the exception of the input layer neurons. In the output layer, the predicted probability of each class (each neuron) is produced. The final result is determined based on the class with the highest predicted probability. This predicted probability is determined based on the input variables, biases, and weight parameters, given by

$$Y = f(\sum_{i=1}^{N} w_i x_i + b), \tag{2.1}$$

here, Y denotes the predicted probability, w_i the weight, b the bias value, x_i the input value, and f the activation function.

An example of an ANN-based approach for developing the lane changing prediction model is developed in [LW17]. The prediction is based on three consecutive phases: intention to perform a lane change (describing the driving situation), preparation of a lane change (describing the behavior), and lane change maneuver itself (describing the vehicle maneuver). Three sets of variables, each corresponding to the three phases are defined. Environmental and head tracking variables are used in this model as inputs. The environmental variables considered are velocity, yaw rate, gas pedal activation, and break pressure of the ego vehicle. For the head tracking variables, head rotation around the vertical axis, lateral movement, and the confidence value of head tracking are chosen as relevant variables. Two individual neural networks are developed, each to predict the LCL and LCR respectively. Here, different time spans before a lane change (suggesting the latest time for a prediction before a lane change) and different configurations are analyzed to evaluate the effect on the model's performance. Hence, time spans of 2, 4, and 8 s before a lane change with different configurations are selected for predictions. The model is evaluated using the individual and fusion of the variable sets as inputs. The results show that the model performed better with the fused set in all the time spans. Based on the results, a time span of 2 s produced the highest prediction performance with an AUC value of 0.972 and a FAR of 0.018, followed by 4 s and 8 s.

Developing a model using only one ANN model may not be sufficient to handle and interpret certain information to produce optimal estimations. Hence, a method to solve this problem is to combine ANN with another ML algorithm or to combine different types of ANN. In [GBH21], a Recurrent Neural Network (RNN) in combination with

Long Short-Term Memory (LSTM) is used for lane changing predictions. The RNN is implemented with LSTM cells, based on the work of [SMD+18]. Different combination of environmental input variables are tested to be chosen as the input variables, however the combination of steering wheel angle and indicator produced the highest evaluation performance (area under curve (AUC) of 0.93). The outputs are LCR and LCL behaviors. Driving data from 57 drivers who drove a 40 km route in the urban area of Chemnitz, Germany are collected. The results based on this model is compared with the results of an Echo State Network (ESN) model to evaluate the performance. Based on the results, the RNN-based model performs better in predicting LCR with a perfect ACC of 100 %, while the ACC of the ESN model is 92 %. However, the ESN model produces higher ACC for predicting LCL. In [DF16], SVM and ANN are combined to develop a lane changing prediction model. Here, the model predicts the feasibility and suitability of a lane change based on environmental variables such as speed difference, vehicle gap, and positions. By comparing the proposed methods with the Bayes classifier and a decision tree, the proposed method produced the best performance with an ACC of 94 % for non-merge and 78 % for merge events. Integrating the developed models mentioned in future developments of ADAS or autonomous vehicles for behavior estimation can provide new knowledge for better driving assistance or if intervention is needed. This reduces mishaps between drivers as well as between driver and the intelligent vehicle to maintain road safety. The summary of the presented contributions, which includes model types, literature, and performances is given in Table 2.7.

Table 2.7: Summary of ANN-based models' performance for lane changing estimations

Models	Literatures	Performances
Individual ANN	[LW17]	AUC: 0.972, FAR: 0.018
Combined ANN: LSTM-RNN SVM-ANN	 [GBH21] [DF16]	 ACC (LCR): 100 % ACC (merge): 94 % ACC (non-merge): 78 %

Hidden Markov Model

Hidden Markov Model (HMM) was initially used for speech recognition and biological sequences analysis. A HMM defines the stochastic process between a set of unobserved states (hidden) and a set of observed states [Rab89]. In a HMM, the state at time t depends on the state at time $t-1$. The hidden state and observation state sequences are labeled as $Q = \{Q_1, Q_2, ..., Q_L\}$ and $O = \{O_1, O_2, ..., O_L\}$ respectively, whereby L is the length of the sequence. The hidden state sequence can be determined through the observation sequence using a HMM parameter. The Baum-Welch and Viterbi algorithm are used to apply the HMM. The Baum-Welch algorithm is used to estimate the HMM parameter when the HMM is trained. Thus, the HMM parameter is evaluated to best fit a given observation sequence, O and the corresponding hidden state sequence, Q. The most probable hidden state sequence can be determined using Viterbi algorithm based on the estimated HMM parameter. When applying HMM for lane changing behavior prediction as in [DWS18], the observation variables are the input variables, while the hidden states are the different lane changing behaviors. Hence, the most probable driving behavior sequence

is estimated with environmental variables like distances and velocities used as observation variables [DWS18]. A prefilter is introduced to the HMM model and optimized in this work. The aim of the prefilter is to quantize the variables into the segments, which are used to develop oberservation sequence. Comparisons of the prediction performance are done using a general and the optimal prefilter. The ACC and detection rate (DR) values increased when the optimal prefilter is used, generally higher than 80 % with a few exceptions.

In [BED08], HMM is used for continuous recognition of lane changing maneuvers. A total of 100 lane changes are performed in the training data set, which are made up of 50 LCL and 50 LCR . Each maneuver is tested with a few combinations of input variables, HMM grammars, and sub model configurations. Suitable model grammars (number of states in left-right Markov chain) and sub model dimensions are developed using the steering wheel angle. Hence, model grammars with different number of states are tested. Each model grammar is also tested with sub models of different number of states. Here, a sub-model of three states with a grammar model of nine states is chosen as the best model. In addition, the authors examine if additional input variables in combination with the steering wheel angle increases the recognition performance. The highest training performance is achieved with the addition of steering wheel velocity, used as input variables (observation variables). The performance results showed a sensitivity of 71 % and 74 % for the LCL and LCR respectively. As these models are widely applied in ADAS to ensure driving safety is adhered, a recent work that focuses on this application is [YLW18]. The prediction model is developed to optimize the adaptive cruise system that evaluates the vehicle ahead. Here, a HMM-based model is developed to estimate lane changing behaviors of the vehicle in front of the ego vehicle. Input variables used are the distance between the ego vehicle and the front vehicle, the front vehicle's longitudinal velocity, and its lateral velocity. The model is able to predict with a maximum ACC of 97 % for straight roads and 96 % for curved roads within a time window length of 4.5 s and 3.5 s respectively. An intention prediction method based on HMM is developed for autonomous vehicles to predict lane changing intentions of a human driver in [LZZF20]. The aim is to predict the intention of the targeted vehicle based on environmental input variables such as vehicle velocities, accelerations as well as offsets between the lanes and vehicles. Two approaches are introduced, whereby the HMM is trained with discrete and continuous variables (either from targeted vehicle or both targeted and surrounding vehicles). The results show that the prediction of the model trained with continuous variables produces higher ACC than using discrete variables. Furthermore, the ACC is slightly higher when the model is trained using variables related to both the targeted vehicle and surrounding vehicles with some exceptions, such as when the prediction time is closer to the time of the actual intention. For an example, the ACC is less than 78 % when only the targeted vehicle-based variables are involved, while it is 80 % when both variables are considered.

Typically HMM-model within this areas are distinguished into two types. HMM-combined and HMM-derived. Often, research contributions focus on combing HMM with other ML approaches/other mathematical methods to improve the prediction and recognition performance of a model. Besides, if it is unable to model certain information on its own, the HMM requires the use of another algorithm. The HMM and the other ML algorithm have different roles in the model. A Gaussian mixture Hidden Markov model is developed in [JLL20] to characterize lane changing behaviors for autonomous vehicles. The Gaussian mixture model aims to extract variable values to characterize lane changing behaviors, whereby a probability density function is used to describe each variable value.

Then, HMM is used to develop a relationship between the hidden states (lane changing behaviors) and observed states (input variables). A total of nine environmental input variables are considered, consisting of speed and distances to surrounding vehicles. The results showed that the highest ACC achieved is 95.4 %. In [DS18], HMM is combined with Fuzzy Logic (FL) to develop a lane changing prediction model based on the safety level. In this work, the safety level of a scene for a lane change is defined using TTC and distances. Using these variables as input, the FL is used to classify the safety level of a scene into very safe, safe, and dangerous. The HMM is used to estimate a lane change based on the safety of the scene using environmental variables. The obtained ACC and DR are higher than 80 % for the different lane changing behaviors. In [KKK20], a novel approach based on HMM, Divisive Hierarchical Clustering (DHC), and Dynamic Time Wrapping (DTW) is developed to identify lane changes on a highway. Driving maneuvers are clustered using HMM and DHC into primitive driving actions. Based on primitive actions and defined patterns, DTW is employed to identify a lane change. A F_1-score of 0.9801 for lane change identifications is observed. In [LWXW16], a lane changing intention recognition approach is developed based on HMM and Bayesian Filtering (BF) techniques. The HMM produces preliminary behavior classifications, which is then used by the Bayesian Filtering part to develop final lane changing classifications. The input of the model consist of steering angle, lateral acceleration, and yaw rate obtained from the CAN bus. The model successfully recognizes LCR and LCL behaviors with an ACC of 93.5 % and 90.3 %, which are better than using only HMM. Based on the contributions analyzed, the results of other ML approaches/HMM can be used as input or additional information for HMM/other approaches to be trained and estimate lane changing behaviors [JLL20] [KKK20] [LWXW16]. The other approaches can also be used distinguish different driving scenarios/styles, which are then used to estimated lane changing behaviors using HMM [DS18]. As previously stated, the incorporation of these developed models into currently available ADAS (some models have already been integrated) has the potential to improve the traffic safety.

A HMM-derived model consist of multiple layers, such that each layer is used to recognize or predict different driving task, behaviors, variables etc. The results from the each task are combined to determine the final estimated behaviors. An example of the derived method is presented in [DS19b] (Multi-layer HMM), whereby only one specific input can be considered to train a single model in the first layer. The output from the first layer is combined with the models in the second and third layer to develop the final lane changing behavior estimation. The model achieves a high overall accuracy of around 90 %. In [HZW12], a two layer multi-HMM is developed. The first layer consist of three HMM models to recognize braking/acceleration, steering, and speed grading driving behaviors. The output of the first layer (based on the three behaviors) are then passed on to the second layer to estimate the current driving behavior. Different layers are not used in [DS19a], instead different sub-HMMs are considered, such that each sub-HMM are given a specific set of ENV input variables. In addition, the model includes the application of prefilter on the input variables (observations), similar to [DWS18]. The proposed approach generates high ACC, DR, and low FAR. The model achieves an overall ACC of 80.5 % which is an improvement from the conventional HMM's performance of 67.91 %. A summary of the HMM-based models are presented in Table 2.8.

Table 2.8: Summary of HMM-based models' performance for lane changing estimations

Models	Literatures	Performances
Individual HMM	[DWS18]	ACC: higher than 80 %
	[BED08]	Sensitivity (LCL and LCR): 71 % and 74 %
	[YLW18]	ACC (straight roads and curved roads): 97 % and 96 %
	[LZZF20]	ACC (targeted vehicle): 78 %
		ACC (targeted and surrounding vehicles): 80 %
Combined HMM:		
Gaussian-HMM	[JLL20]	ACC: 95.4 %
Fl-HMM	[DS18]	ACC and DR: higher than 80 %
DHC-DTW-HMM	[KKK20]	F_1-score: 0.9801
BF-HMM	[LWXW16]	ACC(LCR and LCL): 93.5 % and 90.3 %
Derived HMM:		
Multi-Layer HMM	[DS19b]	ACC: around 90 %
Multi-Layer HMM	[HZW12]	around 90 %
Improved-HMM	[DS19a]	ACC: 80.5 %

Support Vector Machine

Support Vector Machines (SVM) is a supervised ML algorithm initially developed for the classification of two classes by finding an optimal hyper plane that separates the data points from both classes, [CV04]. The SVM supports both linear and non-linear separable data. The aim of the SVM in a linear separable data set is to find the best hyper plane position such that the margin of separation between both classes is maximized. When data cannot be separated linearly, a non-linear SVM is used instead through the application of a kernel function, which is the case in most real world applications. A lane changing behavior prediction model is based on non-linear data as it has more than two classes. Here, the inputs are transformed into a higher dimension input vector using kernel function. This multi-class classifications are usually realized using one-against-all and one-against-one approach.

A SVM-based model is developed for the detection of lane changing intentions in [MS05]. Different combinations of window sizes, overlapping vs non overlapping, and input variable sets are evaluated to analyze which combination resulted in the best classification performance. Environmental variables are extracted from the driving data for detecting the intentions. Five sets of variables are selected to validate the set that produced the best classification performance. In set 1, acceleration, lane positions, and heading are included. In set 2, the importance of the lead car distance is considered, while set 3 considers the effect of longitudinal and latitudinal variables. Set 4 examines the influence of the steering wheel angle. Finally, only lane positions are included in set 5 [MS05]. Based on the performance, set 5 produced the best classification results in all time windows in the over-lapping representation. The authors specify that the model has a good performance, whereby the highest ACC obtained is 97.9 % in a 1.2 s time window.

Similar to previously introduced ML-based models, combining SVM with another ML algorithm is a common approach to obtain extra information and to produce a better performance. In [IPMB$^+$17], ANN and SVM are combined to develop a lane changing prediction model. Two types of ANN (Nonlinear Auto Regressive Neural Network (NARNN) and Feed-Forward Neural Network (FFNN)) are used to predict the trajectory of the vehicle

(lateral position) a few seconds ahead of the ego vehicle. The NARNN is used for predicting time series data, while FFNN (the simplest form of ANN) is used for the mapping of input and output. A SVM is then used to predict if a lane change will occur. The environmental feature variables extracted for the trajectory and lane changing predictions are lateral position, lateral speed, and heading error. The NARNN uses the same variable it predicts as an input (here the lateral position), while the FFNN uses any variable as input to predict an output that could be same or different to the input. For SVM, the estimations of ANN is used as an extra variable for better classification of lane changes. From the prediction of the ANNs, the NARNN does not show an improved prediction when compared with a baseline method, however, the FFNN is able to improve this by reducing the error values. Based on the SVM classification, it is able to predict a lane change 3 seconds prior to a lane change, however no evaluation metric scores are provided here. Probabilities of a situation-based and a SVM-based movement approach are combined to predict LCR, LK, and LCL in [WTG+17]. The situation-based probability is based on current traffic situation to predict a possible lane change, while the movement-based probability is defined using SVM based on vehicle movements in the lane. Both probabilities are fused to estimate an upcoming lane change. The situation-based approach considers environmental variables that describe the inter vehicle relation to analyze the current traffic situation, while the movement approach considers environmental variables related to the vehicle movements in lane such as distances and velocities. The presented approach is able to predict LCR with a recall value of 0.93, while LCL with a recall value of 0.72. The authors in [KPLL13] combine SVM and Bayesian Filter (BF) to develop a lane changing intention prediction model. A multiclass SVM is used for the classification of the trajectories that belong to three different classes such as LCR, LK, and LCL. The SVM outputs are then used an inputs for the Bayesian filter for developing an improved prediction results. Bayesian filter aims to provide a smoothing effect to reduce FAR and missed detections in contrast to the SVM-based results. Hence, a comparison of the prediction performance between the combined SVM and BF model and only SVM is presented. Environmental variables such as lane information, speed, and steering wheel angle are considered as inputs for the SVM. The results show the average precision increases from 0.2857 to 0.7154 when using the combined model instead of the individual SVM model. Similar precision values are obtained when the model is trained and tested with different combination of drivers' data, with the highest precision at 0.8235, proving its robustness. In Table 2.9, the summary of the SVM-based models is given.

Table 2.9: Summary of SVM-based models' performance for lane changing estimations

Models	Literature	Performances
Individual SVM	[MS05]	ACC: 97.9 %
Combined SVM:		
ANN-SVM	[IPMB+17]	Able to predict 3 prior to lane change
Situation based-SVM	[WTG+17]	Recall (LCR and LCL): 0.93 and 0.72
BF-SVM	[KPLL13]	Precission: 0.7154

Advantages and Disadvantages

As mentioned, the advantages and disadvantages of the approaches affect the prediction/recognition performances. One of the advantages of ANN is its ability to handle multi-class problems, in contrast to SVM, whereby the performance decreases with increasing number of classes [LRL07]. The contribution [LW17] shows this as well as its ability to combine different information. Here, the different driving features describing the driving phases of a lane change are combined for estimations. Another advantage, particularly with deep neural network is its ability to extract features automatically. However, a major disadvantage of ANN is its tendency to overfit. Increasing the number of neurons in the hidden layer shows a low performance due to over fitting, hence the authors in [LW17] limited the hidden layers neurons to three. Contrarily, SVM posses certain advantages over ANN. For example, SVM does not have the tendency to over fit, by choosing a hyperplane which minimizes the distance between the hyperplane and the nearest training sample point [LRL07]. To avoid over fitting, the authors in [DF16] also combine ANN with the SVM. In the reviewed paper [KPLL13], this advantage takes effect as it is able to accurately classify trajectory belonging to different classes. Another observed advantage, is its ability to transform a low dimension nonlinear classification problem into a high dimension linear classification problem by mapping input data from a low dimensional space to a lane changing behavior in a high dimensional space. Moreover, training of SVM models only require few samples in high dimensional spaces compared to training ANN [LRL07]. In [MS05], choosing an optimal data representation of the input features is challenging in SVM. Rather than using the original values, variance of features over others are used to solve this, reducing the input size whilst extracting the change in features. Nevertheless, SVM posses certain drawbacks. As stated in [KPLL13], an extra BF is needed to improve the performance of SVM. While the authors state that the false detection rates are mainly caused by lane tracker inaccuracies, there could be reasons related to the SVM model and its parameters. A possible reason is the development of optimal weights and kernel function is challenging when applying SVM individually. Advantages of HMM are its stochastic properties, ability to mange time series data, and ability to deal with temporal pattern recognition,[LZT$^+$14] [JF15] [MDPB09]. These benefits make it feasible for the prediction of driving behaviors, since future driving behaviors are described as stochastic and dependent on the current behavior [JF15]. However, HMM has several limitations as in [DS18] whereby, models with different parameterizations are required for different drivers to generate accurate estimations as driving behaviors are individual. Thus, the HMM model has poor generalization capabilities. The HMM approach is also not suitable for long term prediction as the number of hidden states must be specified prior to training as observed in [DS22c]. In [LZZF20], the limitations of using discrete characterization of features in comparison to continuous characterization with HMM are observed, as the performance is reduced due to information loss. A major disadvantage is individual ML methods are often combined with another ML method or other methods to yield better performances which only increases the complexity, as using only a single method don't always generate optimal results.

In Table 2.10, a comparison of machine learning-based approaches for the estimation of lane changing behaviors in terms variables, traffic environment, and contributions are presented. The approaches compared are ANN, CNN, HMM, SVM, DBN (Dynamic Bayesian Network), and RF-based approaches. Based on the comparisons of different

models discussed in this work and illustrated in Table 2.10, the following conclusions are summarized.

- Environmental variables have the highest influence in developing machine learning-based models for the estimations of lane changing behaviors (all the contributions listed in Table 1), followed by head/eye-tracking variables ([LW17] [DHBS20]), and physiological variables ([MKW+15] [WMK16]). The eye-tracking and physiological variables do not significantly contribute to the prediction and recognition of lane changing behaviors as stated in the work of [DT09]. In [DHBS20], the estimation models using only eye-tracking variables produced poor ACC values. As mentioned, a reason for this is because decision making is mainly based on environmental information, which describe the relationship between ego vehicle and surrounding vehicles. Only a small number of contributions consider the role of physiological variables [SLZ+11] [PLKR11] [LWWX19] .

- Combined ML approaches often have better performance than using an approach by itself. The purpose of using more than one algorithm is to employ the advantages from different approaches as well as to achieve different tasks which cannot done by using a single approach. In lane changing estimation models, the outputs of one approach can be used as an input of another as an extra information for predicting the behaviors. Also, one approach can be used to recognize driving patterns, styles, and scenarios which is then used with another approach to estimate the lane changing behaviors.

- To evaluate the performance of lane changing behavior models, common metrics used in contributions are ACC, DR, precision, FAR, AUC, and F1-score. These metrics produce the most accurate evaluation between the estimated and actual behaviors. However, different metrics places different importance on the different classes depending on the classification problem. The metric best focusing on the importance of each class should be selected to evaluate the lane changing behavior classifications. For example in an imbalanced data set, the metric accuracy tends to attach more significance to the majority class than the minority class, making it difficult to show good performance on minority class. This is known and avoided by more suitable metrics like F1-score.

- From the results of different contributions, the performance values of different approaches do not differ much. For example, most models tend to have good ACC, DR values which are higher than 80 % with low FAR values. High AUC and F1-scores are also achieved ranging from 0.8 to 0.98.

2.1.5 Model parameter optimization

Model parameters of a ML model are internal to the model, that are part of the training process. These parameters are learned during the training process/optimization process, which are used to develop the appropriate estimations based on the inputs. The optimizer select the appropriate model parameters such as, weights, bias values, clusters, etc. Common optimization methods used to select the optimal model parameters, are Gradient descent, Stochastic Gradient Descent (SGD), Mini Batch Stochastic Gradient Descent (MB-SGD),

Table 2.10: Comparisons between different models for lane changing behavior prediction and recognition

Algorithms	ANN/combined	HMM/combined	SVM/combined	CNN/combined	DBN/combined	RF/combined
References examples	ANN: [LW17] [LW18] Combined: [GBH21] [SMD+18] [DF16]	HMM: [DWS18] [BED08] [YLW18] [DHBS20] [LZZF20] Combined: [JLL20] [DS18] [KKK20] [LWXW16]	SVM: [MS05] [DHBS20] [TVT11] Combined: [IPMB+17] [WTG+17] [KPLL13]	CNN: [DJ+18] [WMK16] [IQP+19] [DHBS20]	DBN: [UM15] [KWB+11] [KRH17] [WTZK16]	RF: [SWWB15] [DHBS20] [GHHW19]
Input variables						
Environmental-state of vehicle (acceleration, vehicle speed, trajectory, yaw rate, TTC)	yes	yes	yes	yes	yes	yes
Environmental- operational information (angle of steering wheel, brake pressure, gearbox, indicator)	yes	yes	yes	yes	yes	yes
Eye/Head tracking (eye gaze, blink frequency, PERCLOS, head movements- yaw, pitch, roll)	yes	yes	yes	yes	-	yes
Physiological (ECG, GSR, RR)	yes	-	-	yes	-	-
Traffic environment						
Highway(traffic)	yes	yes	yes	yes	yes	yes
Urban/city(traffic)	yes	-	-	yes	-	-
Results	AUC: 0.972, FAR: 0.018 - 0.095 [LW17]	$F_1 Score = 0.9801$[KKK20]	Precision: 0.55 -0.8235 [KPLL13]	Recall: 0.67 [IQP+19]	AUC: 0.986, Recall: 0.927 [KRH17]	ACC: 0.9772 [GHHW19]

Adaptive Gradient (AdaGrad), and Adam. These are usually built-in within a specific programming tool/software, like Matlab. The NSGA-II is well-known for solving multi-objective problems as it has an advantage of emphasizing on non-dominated solutions, it's an elitist approach, and diversity preserving method (crowding distance) [DPAM02] [Deb01]. The non-dominated solutions are solutions that generate a compromise for the different objective functions, such that none of the objective functions can be improved without depraving the others [DPAM02]. Here, one solution does not dominate the other. Previous work in the driving behavior research that have considered the optimization of model parameters using NSGA-II include [DS19a] [DS19b] [DS18]. In [DS19a], model parameters of ANN, SVM, HMM, and Random Forrest (RF) are optimized, while in the [DS19b] parameters of a multi-layer HMM are optimized. In [DS18], model parameters of a combined approach of Fuzzy Logic and Hidden Markov Model (FL-HMM) are optimized. The parameters optimized are weights, threshold values of a prefilter, etc.

2.1.6 Hyperparameter optimization

Hyperparameters are parameters that control the training/learning process of a ML algorithm, thus affecting the performance of the model [YS20]. While hyperparameters are not part of the model and training process, these parameters do affect the choice of the model parameters, ultimately influencing the performance [YS20]. Hyperparameters are set prior to the training process of ML model, such that the set of hyperparameters can be used to fit the input to the output well, through selection of appropriate model parameters [YS20]. This guarantees the model's effectiveness remains at a high standard. Examples of hyperparameters are, the number of neurons in an ANN, activation function type, number of iterations, pooling size, batch size, number of hidden layers, and learning rates.

In general, two techniques are employed to select the appropriate hyperparameters: manual search and automated search. Manual search include manually selecting values based on one's experience and expertise to select the values that improve estimations by determining if a specific value develops a good fit between the input data and target values. In order to select optimal values, the search method requires professional experience and background knowledge of the problem presented (similar problem encountered in the past). For an example, researchers may start with 0.1 as the learning rate for training an ANN-based model when estimating driving behaviors, as this rate this is known as the optimal rate based on previous research contributions. From the generated performance of the model, researchers will increase or decrease this value accordingly to achieve optimal estimations. However, this technique will not work well for users with less experience leading to a poor model performance. In addition, the process of manual search can be time consuming and expensive.

To tackle this, another type of technique known as grid search is used. Here, every possible set of hyperparameter combinations are trained to select the combination that yields the best results. This method only works well if the the possible combinations are small and the number of hyperparameters are less [BB12]. For more complex models with higher number of hyperparameters, the process is computationally expensive as it requires separate models for each combination [BB12]. Despite the mentioned disadvantages, this method is still widely used for less complex models.

Random search is another manual search method. Here, random sets of hyperparameter combinations are selected for training the ML model from a predefined set of hyperparameter

values. The user predefines a set of possible values for the different hyperparameters [BB12]. The algorithm then selects a set of combination at random [BB12]. The combination that generates the best performance values is chosen as the optimal set. However, like the grid search, it can be computationally expensive for complex models [BB12].

Manual search methods can be tedious, time consuming, as well as it does not always generate optimal values. There is also a lack of logical reasoning for the choice of values. Rather than using expert knowledge, automated hyperparameter optimization is commonly performed for better results. For improved results and a trade off to manual search, automated hyperparameter optimization often require large amounts of data to select an optimal search space. The automated hyperparameter optimization serves the following purposes:

- Reduce the costly expert of manual influence.
- Improve the efficiency of the model training.

Common automated hyperparameter optimization techniques are Bayesian optimization (BO) and Genetic algorithm (GA). The BO is an automated method that develops a surrogate model (typically using Gaussian process) to represent the objective function and develops an acquisition function from the surrogate to select optimal point [MTZ78]. Unlike the random search and grid search, it uses previous values to select the next point. As for the GA, the method is an automated method that considers a set of possible solutions and repeatedly modifies a population of solutions to develop a better solution [Hol73]. It is also a type of Evolutionary algortihm.This method makes good use of parallelism.

These well-known hyperparameter optimization/selection methods (both manual and automated) are summarized:

- Manual: manual search by designers.
- Grid search: evaluates all possible hyperparameter combination in a specified grid (Cartesian product). Suitable for a small number of hyperparameters.
- Random search: randomly select points from a defined search space.
- Bayesian optimization (BO): An automated method that develops a probabilistic model (typically using Gaussian process) of the objective function using current hyperparameter values. The model is used to predict the next set of hyperparameters based on the expected improvement of the model [MTZ78].
- Evolutionary: an automated method (as well as a meta heuristic method) that repeatedly modifies solution to develop a better solution that fit well with the model [Hol73], such as GA.

As mentioned in the previous chapter, the use of hyperparameter optimization has already been tackled in certain contributions, such [HK21]. In [HK21], the hyperparameters of LSTM are optimized using BO for the detection of aggressive behaviors. Other research contributions that have considered the optimization of the hyperparameters within the driving behavior area is [DS19a], whereby hyperparameters of ANN, SVM, HMM, and RF are optimized to improve the prediction performance of lane changing behaviors

using NSGA-II. In [DS19b], the same optimization technique is used for the optimization of a Multi-layer HMM's hyperparameters to estimate lane changing behaviors, showing improved performance. Nevertheless, there is still a lack of contributions that focuses on hyperparameter optimization within the driving behavior estimation area, especially for models developed based on combined approaches.

2.2 Estimation of degradation behavior in lithium-ion batteries (LIBs)

In recent years, the use of lithium-ion batteries (LIBs) in various systems have increased, such as in hybrid electrical vehicles (HEVs), unmanned aerial vehicles, and telecommunication systems. This is due to its light weight, long cycle life, and low charge rate loss [HZZL17] [Xia15]. These batteries provide energy through electrochemical processes during charging and discharging cycles. However, increasing the number of cycles causes aging as well as stability deterioration of the batteries from side reactions and electrochemical processes. The degradation of batteries leads to efficiency reduction of battery-powered systems as well as catastrophic events. Stress factors such as battery temperature, ambient temperature, and c-rate in the loading profiles influence the degradation. Some of the common aging reactions include solid-electrolyte interface (SEI) layer growth, corrosion of lithium, and lithium platting [DK13]. The degradation of the batteries are prone to catastrophic events such as the breakdown of a battery operated system and thermal runway. Therefore, monitoring the battery's health is important for maintaining safety, the system's performance, and avoiding unexpected maintenance. Several health indicators such as remaining useful lifetime (RuL), capacity fade, end-of-discharge time (EoD), and end-of-lifetime (EoL) are used for monitoring the health.

As part of the battery's prognosis and health management, many approaches have been proposed to estimate the health indicators using lifetime models. These approaches are typically distinguished in two types: model-based [CN03] [DLH$^+$19] and data-driven approaches [ZWWC21] [ZZW22]. The main challenge in the development of lifetime models is LIBs have different lifetime expectancy affected by stress factors. Another challenge is model-based approaches tend to only be designed for specific conditions making it difficult for generalization purposes [ZZZ$^+$16] [ACW16]. While certain models tend to yield optimal estimations, these modelsl require the complicated experimental design as well electrochemical information [LSB$^+$17]. Overall, there is still a lack of optimal lifetime models developed for the degradation estimation of LIBs. Considering the continuous and increasing use of these batteries in everyday life as well as many industries, developing an optimal model is a vital aspect of this field. In this thesis, the aim is to develop a model for the estimation of capacity fade.

2.2.1 Aging of LIB

A cell consist of a positive and negative electrode, such that lithium-ions can move from one electrode to another through the electrolyte diffusing through the separator. In the positive electrode, the active material present is known as lithium cobalt oxide (Li_xCoO_2), while the active material present in the negative electrode is lithiated carbon (Li_xC). Based on the movement of lithium-ions from one electrode to another, these ions insert (intercalation) or leave (deintercalation) the active material depending if the cell is charging or discharging [DK13]. A fully charged battery has lithium-ions in the negative electrode.

When the battery is discharging, the electrons and ions are liberated from the negative electrode and move to the positive electrode, due to the oxidation (loss of electrons) in the negative electrode

$$Li_xC \rightarrow C + xLi^+ + xe^-. \tag{2.2}$$

This results in the reduction gain (gain of electrons) in the positive electrode, given by

$$Li_{1x}CoO_2 + xLi^+ + xe^- \rightarrow LiCoO_2. \tag{2.3}$$

As for the charging process, the lithium-ions resides at the positive electrode initially. As the battery charges, the oxidation process takes place in the positive electrode resulting in the liberation of electron and ions. The ions move from the positive electrode to the negative electrode (deintercalation),

$$LiCoO_2 \rightarrow Li_{1x}CoO_2 + xLi^+ + xe^-, \tag{2.4}$$

whereby the reduction reaction occurs in the negative electrode

$$C + xLi^+ + xe^- \rightarrow Li_xC. \tag{2.5}$$

As mentioned, this charging and discharging process causes battery aging due to chemical reactions from different stress factors eventually leading to the end of life of the battery. The different types of aging reactions/mechanism are summarized [DK13]:

- Solid-electrolyte interface (SEI) layer growth: the SEI layer is formed due to cycling and storage at high temperature. This layer forms at the negative electrode causing the electrode to degrade and an increase in impedance [DK13].

- Lithium corrosion: lithium-ions in negative electrode corrodes over time causes the loss of lithium-ions in that electrode. This phenomena leads to degradation of capacity and battery [DK13].

- Lithium plating: a platting layer is formed on the negative electrode due to the t low temperatures, high charge rates and low cell voltages, causing the loss of lithium-ionss [DK13].

- Contact loss: the SEI layer dislodges from the negative electrode causing contact loss and increase in impedance [DK13].

Generally, there are two types of aging: calendar aging and cycling aging. Calendar aging occurs while the battery is at storage and not in use, which means the time period the battery is not going thorugh charging-discharging cycling. Cycling aging occurs when the battery is in use and operating (dependent on the charging and-discharging cycling) [SDBD12]. To measure the health or degradation, health indicators such RuL, SoH, EoD, EoL, and capacity degradation calculations are utilized.

2.2.2 Stress factors

As stated, the degradation of the battery are often the direct impact of the operating and storage conditions (stress factors), such as temperatures (ambient, surface), c-rates, depth of discharge (DoD), state of charge (SoC), etc. These stress factors tend to affect the performance of the battery. The correlation between various stress factors and battery aging is investigated in several research contributions. In [KJ17], the capacity fade dependence on different operating temperatures is studied. For instance, low temperatures generate an accelerated and increased capacity degradation during cycling aging (when load is applied). On the other hand, capacity degradation decreases at low temperatures during calendar aging (when no load is applied). Based on [WLHG+11], high ambient temperatures (above room temperature) or very low temperatures (lower than zero) accelerate the aging process. The correlation between different stress factors and degradation are given in Table 2.11.

2.2.3 Lifetime models: diagnosis and prognosis of LIB

As mentioned, the battery's health can be stipulated by estimating the health indicators using lifetime models, which are developed using either model-based or data-driven approaches. The model-based approaches require information about the physical and chemical properties of the battery, as shown in [DLH+19]. A non-linear least squares method with dynamic bounds is employed for the estimation of RuL in [DLH+19]. These models are developed using mathematical models and parameters. It is also based on observer design and parameter estimations as developed in [Ple04]. As for [Ple04], a Kalman-based filtering process is used to estimate SoC and other model states for a lithium-ion polymer battery (LiPB) pack. In [HWMP11], a particle filter-based model (as model-based approach) is used to predict the RuL. The authors employ a regression model to estimate the aging dynamics and parameters. While in [WYT+16], a state-space model is employed to model the capacity degradation. Thereafter, a Spherical Cubature Particle Filter (SCPF), a combination of particle filter and Spherical Cubature integration-based Kalman Filter (SCKF) is integrated into the state space model to predict the RuL. The model-based methods can be further distinguished into physics of failure (PoF) [LSB+17] and empirical models [SG09]. The PoF model is generated based on material properties, loading conditions, and failure mechanisms [LSB+17]. An example of the PoF model in the battery community is developed in [LSB+17], whereby an improved particle learning is used to predict the RuL. The particle learning re-sample state particles by considering the current measurement information and propagating them avoiding particle degradation. Empirical models on the other hand, do not consider the battery's electrochemistry aspects and tend to be computationally efficient, as in [SG09]. In [SG09], the authors fit functions to experimental data without the use of electrochemical information. A particle filtering framework is utilized for the estimation of the capacity degradation, RuL of discharge cycles, and EoL. Another example of the empirical model is given in [WYZT17], whereby a discharge-rate dependent model is developed. A limitation of model-based approaches is their potential lack of suitability for complex systems, as they may not effectively capture degradation behavior stemming from physical and chemical factors. [WKC+22]. In addition, implementing parameters based on the electrochemical models is time consuming due to need of elaborate experimental setups, as stated in [WWZ19]. The generalization of the models is also limited according to [WKC+22].

Data-driven approaches (such as ML approaches) are often based on historical measured

Table 2.11: Summary of the stress factors' effects

Stress factors	Affects
Temperature (ambient)	High: accelerated aging, capacity degradation Low: accelerated aging, capacity degradation
Temperature (surface)	High: For calendar aging, increase in capacity fade, increase in aging - For cycling aging, increase in capacity fade Low: For calendar aging, there is capacity reduction, however not as much when temperature is high - For cycling aging, increase in capacity fade
C-rates	High c-rates, high temperature: increase in capacity aging High c-rates, low temperature: lithium platting
DoD	Increase in capacity loss with capacity High c-rates, high DoD: more capacity loss Low c-rates, low DoD: more capacity loss
SoC	High temperatures, low SoC:) increased degradation

data. These models have the benefit of being computationally and time efficient, making the models suitable for larger systems [ZZW22]. In addition, these models do not require the use of electrochemical information, which makes these models desirable. The ML-based approaches have gained popularity in recent years. Some of the ML models used are ANN [ZZW22] and SVM [PTH+15]. In [ZZW22], an LSTM is combined with Broad Learning System (BLS) algorithm to estimate the capacity fade and RuL. Here, the BLS is first used to create feature nodes based on the capacity data. These nodes are then used as an input layer to the combined model. A SVM-based model is developed for the RuL prediction using a minimal set of features from the battery cycle data in [PTH+15]. Another example of the data-driven model is presented in [LPZ+13], whereby a Gaussian process regression model is developed to estimate State-of-Health (SoH) with means and variance as the uncertainty of the SoH. In [LZX+20], a LSTM is used as well as a SoH estimator. However, poor quality information collected can affect performance of data-driven models negatively. Hence, appropriate techniques needs to be implemented when collecting data.

Combined/mixed approaches of model-based and data-driven are also developed in certain research work to combine the abilities and tasks of both methods. A quintessential example is [Koz03], which combined three different models: Autoregressive Moving Average (ARMA), neural network, and Fuzzy Logic (FL) to predict the RuL. In [CP12], the authors integrate the model-based and data-driven approaches using a particle filter framework. Another combined example is developed in [CCF+21], such that a Recurrent Neural Network (RNN) is combined with a state space model for the estimations of RuL. The state space model is used to generate the data set for the RNN to be trained.

A summary of different research contributions that have developed both model-based, data-driven, and combined approaches for modeling the degradation of LIBs is presented in Table 2.12. The different model types, approaches, health indicators, and stress factors as inputs are highlighted in the table.

2.3 Open research questions

As an end to the review of the estimation of lane changing behaviors and battery degradation estimation, open research questions are posed in this section to shape the remaining part of the thesis.

The previous sections introduced and defined the prediction and recognition of driving behaviors and intentions. As ML-based approaches are mainly used in current literature to establish models for the prediction and recognition of these behaviors, the common ML-based driving behavior estimation models are summarized. Based on the literature review, different factors affect the performance of these models, such as the type of approach used, features selected as input for ML models, hyperparameters, and parameters. In fact, there is no single approach that performs better than others according to the no free lunch theorem as shown in [WM97]. In addition, certain features (as input variables) tend to improve the performance of model, while others do not always generate satisfactory results as stated in [DT09]. The influence of parameters and hyperparameters is also studied in some contributions [HK21] [DS19a] through the use of optimization methods. However, selecting optimal parameter values is challenging. Therefore, to develop a suitable driving behavior recognition model, several questions are posed based on existing literature:

- Common ML-based approaches have been extensively utilized in literature for the

Table 2.12: Summary of different lifetime models

Model type	Approaches	Estimated health indicator	Reference
Model-based	Non-linear least square	RuL	[DLH+19]
	State-space	Capacity, EoD times, RuL, voltage	[XC17]
	Extended Kalman filter	SoC, power fade, capacity fade	[Ple04]
	Particle filter	RuL	[HWMP11]
	State-space, SCPF	Capacity fade, RuL	[WYT+16]
Model-based (PoF)	Particle filter	RuL	[LSB+17]
Model-based (empirical)	Particle filter	RuL, EoL, EoD	[SG09]
	Discharge-rate dependent	RuL	[WYZT17]
Data-driven	ANN (LSTM and BLS)	Capacity fade, RuL	[ZZW22]
	SVM	RuL	[PTH+15]
	Gaussian regression model	SoH	[LPZ+13]
	LSTM	RuL	[LZX+20]
Combined	ARMA, ANN, FL	RuL	[Koz03]
	Particle filter framework	Capacity fade, RuL	[CP12]
	RNN and state space	RuL	[CCF+21]

development of driving behavior estimation models, however different models tend to generate different performance and there is no single model that outperforms others. Therefore, what other methods can be utilized to develop a driving behavior model? While many contributions have tried to tackle this by combining different approaches as it is well-known that combined approaches tend to perform better, the problem of generating an optimal model still exists. This leaves researchers to continue their venture to build an optimized model. Taking into consideration the limitations of ML approaches, other approaches such as the state machine approach has shown promising results in various domains which exhibit similar characteristics to driving behaviors, particularly for structural health monitoring [BS17] and modeling plant growth behaviors [KS20]. In the state machine model, different states are used to define the behaviors. In this thesis, the aim is to tackle this by developing a state machine-based model for lane changing behavior recognition. The questions that exist are how can the state machine approach be applied to the lane changing behavior estimation model? How are the lane changing behaviors modeled using states? How are the transitions between states defined? Also, how does the model's performance compare to other models?

- As combined models tend to perform better, the aim is to combine the state machine approach with another ML-based approach. Each approach tend to have their advantages, thus combining the methods helps. Due to the performance and capabilities as proven in [DS19a] [LW17] [YLW18], ANN and HMM are selected here. The ANN has been used widely in this area due to its ability to handle multi-class problems, such as different driving behaviors. Also, its ability to extract relevant information and features through the use of the hidden layers (to trigger the relevant node), makes this technique desirable. The HMM is known for its ability to handle time series data and its stochastic features, hence it is applied for the estimation of driving behaviors in research contributions through development of HMM-combined or HMM-derived models. Therefore, what impact does integrating both ANN and HMM into a state machine have on the performance? Can a HMM-derived model be incorporated with a state machine model? Can the combined models generate an improved performance?

- Previous studies have shown that using certain features as input variables such as ENV features improves the performance of the driving behavior models, while ET features do not necessarily improve the results for certain models. In fact, impoverished performances are generated. However, this does not seem to be case for ANN, HMM, and RF as shown in [DHBS20], whereby the inclusion of ET features slightly improves the prediction performance of the model. Nevertheless, the use of ET features in a SVM model does not improve the performance. Here, the aim is to study the effect of the ENV and ET features on the lane changing recognition performance of the state machine-based model.

- As proven by other approaches and contributions, model parameters play an important role in ML-based models. Therefore, the optimization of these parameters are important. This leads to a question of what model parameters of the proposed model needs to be optimized? How does the parameters influence the a newly developed state machine model? As driving behaviors are unique and individual, typically a single model is trained for each driver resulting in different parameter values for

different driver data and models. Is it possible to use the same parameter values for different drivers to generalize the approach? To develop an effective driving behavior estimation model, generalization is important. So, the aim is to develop a model that can generate optimal estimation performance using the same parameter values for different drivers. Also, setting the parameter values manually to develop an optimized model can be time consuming and can increase the inaccuracies. Therefore, parameters needs to be developed automatically by an optimization method based on the different data sets. Here, the NSGA-II is employed to tackle this.

- Similar to existing contributions, how does the hyperparameters of the new model influence the performance? What would be the adequate method to optimize the specific hyperparameters in the new model? Here, the BO and GA are selected due to the methods' benefits (explained in previous sections) and their ability to develop optimal values in research contributions [HK21]. In addition, as there is a lack of studies that focuses on the hyperparameter optimization of model developed based on combined approaches, this thesis aims to bridge that gap.

Applying the newly proposed state machine model to other systems that exhibit dynamic behaviors, such as estimation models for the degradation of the lithium-ion batteries is important to prove the versatility of the model. Majority of the degradation estimation models are based on model-based or data-driven approaches. While some models are difficult to develop due to need of complex experimental setups and electrochemistry information, other models are easier to develop as these models do not require physical and chemistry information of the battery. Hence, it is challenging to select an appropriate model for the estimation of capacity fade, RuL, etc. In addition to the questions related to the state machine approach (such as parameter optimization, types of models to be selected, etc), there are several other questions which are specifically related to the degradation estimation of LIBs domain:

- As capacity fade is estimated as the degradation behavior, how can the degradation behavior be modeled using the state machine model? Current research do not consider different degradation states for modeling the degradation behavior. Can degradation states be defined using the state machine model? In addition, how are the degradation states defined? How are the transition between states defined?

- Besides the degradation states, the estimation of the capacity is also needed. Many literature tend use an Arrhenius equation to calculate the capacity. However, can another method be used instead without the need of a mathematical equation?

- Does the current model require the use of electrochemical information or only historical data of experimental data (from stress factors), like capacity, temperature, etc.

- How does the proposed model differs from other well-known models in terms of estimation performance?

3 State machine-based approach

In this chapter, a review of the state machine approach in various applications are discussed.

3.1 Review of state machine-based approach

In this contribution, a trainable and interpretable state machine-based approach is introduced for the first time as a machine learning algorithm applied to situation recognition. A state machine (in case of deterministic modeling) models behaviors using a finite number of states. The system dynamics are characterized by a sequence of transitions, whereby the system can either remain in the current state or shift to another state based on a set of transition conditions and inputs. In the classical approach, parameters and variables used for modeling are defined by designers or are related to processes to be modeled. State machines are well-known approaches used for modeling, analysis, and control. They were initially introduced for describing computing systems [Gil62], however the use of these have extended to various applications, such as robotics and telecommunication fields. Typical operations of a state machine include scheduling system's task sequence, switching between different modes of a system, and fault detection [Bö10].

The state machine can be defined by (S, I, X, O, t) whereby,

- S: set of states, $\{s_1, s_2, ...s_n, \}$
- I: initial state, s_0
- X: input of system, $X : \{x_1, x_2, ...x_n, \}$
- O: output of system, $O : \{o_1, o_2, ...o_n, \}$
- t: next state transition, $t(x, s)$, which state the machine should switch to if current state is s and input is x

Generally, a state machine follows these three conditions [Bö10]:

- Has an initial state (s_0)
- State transitions from one state to another $< s_i, s_{i=1} >$. The transitions are part of the subset of transitions or next state relations.
- An end state exist if there are no further states, showing that the behavior is finite.

State machines are deterministic if the transition is a function $t(x, s)$ (it can transition to at most one other state for each states). Non-deterministic can occur when there is an end state (no transition possible), as behavior is finite [Bö10].

Generally, they are two types of state machines:

- Mealy: Output of the state machine is dependent on the states and input [Mea55]
- Moore: Output of the state machine is dependent on the states [Moo56]

One of the advantages of this approach is its easy tracking abilities (tracking which event/data/condition is causing a change) [Gol93]. The state machine systems are easy to design, hence the implementation and execution are quick [Gol93]. This is in contrast with other approaches (like SVM), whereby the final model and weights may be hard to interpret. The state machine is also more interpretable than HMM. For example, the HMM compute output based on the porbabilitic relationship between the observable states and hidden states. The state machine does not consider such complexity, which simplifies the model. Another advantage is its flexibility, enabling a finite state-based system using a topology [Gol93]. The state machine structure is able to determine the next possible state easily based on a set of conditions and inputs. This differs from some traditional ML-based models that require higher complexity and computational burdens due to different layers in the model or black box nature of the model such as ANN. On the other hand, the disadvantage of the state machine is that the approach may not be suitable for all types of dynamical systems; it can only be used when a system has defined and crisp conditions for transitions. For the development of prediction/estimation models, the application of state machines has already been done in different research areas such as tribological experiments [BS17], to describe a lifetime model expressing the relationship between wear degradation and RuL based on Acoustic Emission (AE) data for state selection [BS17]. The idea first published in [BS17] uses the state machine approach with parameters defining state and transitions (the topology is given by designers) as part of an optimization loop to develop models describing wear degradation behaviors. Other contributions include [KS20], whereby the state machine model is used for modeling plant growth and the prediction of leaf elongation [KS20]. Thus, the aim is to use this previously introduced approach as a interpretable new approach within the driving behavior/ transportation field.

3.1.1 States

The states are used to define/represent discrete situation or behaviors. The authors in [RBEL91] define states as sets of values that are grouped together based on their properties affecting the behavior of an object (which can be a system, human beings, etc.) [RBEL91]. The focus in this thesis is on Finite State Machines (FSM), thus the model has finite number of states. The FSM is more practical than infinite states when modeling driving behaviors or damage states. Common characteristics of a state are the model is in a specific state for a certain time period, the state changes from one state to another, and the state is a consequence due to a satisfied condition in an object/system (where certain action or event occurs) based on the current inputs or state [Gil62] [RBEL91]. A state machine model consists of an initial state, as the starting point and progresses through a series of states (events/situations/behaviors). An open loop state machine model consists of a final state, which means the system terminates at the end state (before the end of the system). In contrast, the closed loop state machine does not consist of a final state, whereby the system terminates at the end of the system (entire system terminates) [Gil62].

3.1.2 Transition

The transition is described as a switch from one state to another. The switch is usually associated with certain condition known as transition conditions. For example, a state changes from one state to another if a certain action/event occurs. In Figure 3.1, a simple representation of elevator operation using the state machine is given. Two states, ground and first are defined in this example indicating the different levels. The conditions specified in the transitions indicate the input values (requirements) for a switch from one state to another. If the current state is ground and the input is up (up button in the lift is activated), the next state is first. If the input is down, then it remains in the same state. Similarly, if the current state is first and the input is down, the next state is first.

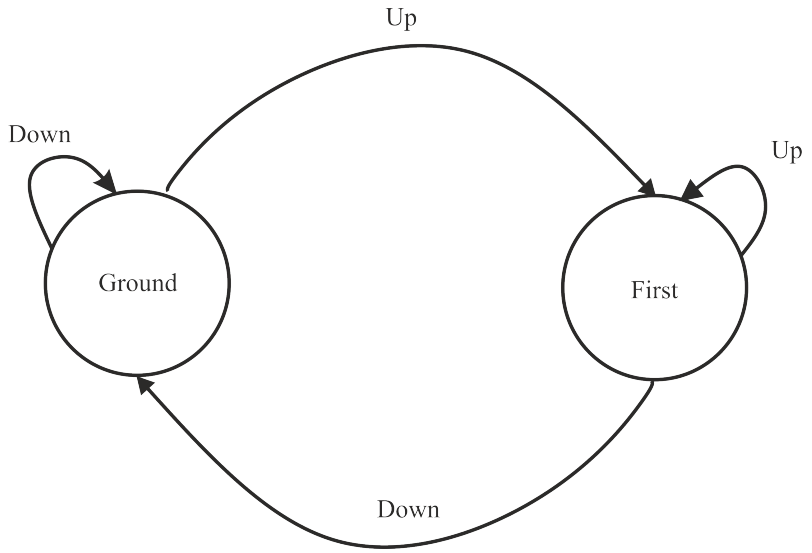

Figure 3.1: Elevator example 1

A more precise example of the elevator operation related to the door control is given in [ACH+09]. Here, the states are the status of the of the elevator door (opening, closing, and waiting). The states transition are dependent on time, elevator door status, and control buttons [ACH+09].

3.1.3 State machine estimation model examples

As mentioned, the state machine has been previously used to develop estimation models in [BS17] [SR17] for the estimation of the tribological systems' health (in terms of RuL), plant growth in [KS20]. As technical systems undergo degradation with use, the system's reliability and functionality deteriorates as well. The state of health (SoH) and health prognosis needs to be observed so that the appropriate maintenance can be performed in a timely manner to avoid break down of a system. In addition, it is also crucial for extending the lifetime of systems. To do this, health estimation models can be develop using lifetime models for RuL and SoH estimations. In [BS17], a state machine-based approach is used to develop the lifetime models which shows relationship between the damage increments and consumed lifetime of a triboligical system. Different states are defined, representing the damage phase, and the transition conditions between states are based on the threshold

exceedance related to AE measurements. The AE measurements are used as input variables to the model. Using the AE data, the RuL can be estimated. Each state is modeled using a lifetime model (a mathematical equation/parametric model) calculating the consumed lifetime associated with a specific state. The model's output is dependent on the current SoH. Two models are introduced in [BS17]:

- Approach I: Parameters are defined by optimization, while thresholds for transitions are fixed and predefined.
- Approach II: Parameters and thresholds associated with the transitions are defined by optimization.

In approach I, the same equation (as lifetime models) with different parameters for four damage states are defined. However, in approach II, different equations (as lifetime models) for three damage states are defined. The optimization method used to define the parameters and thresholds is the Non-dominated Sorting Genetic Algorithm-II (NSGA-II).

As for [SR17], the sliding wear of a hydraulically driven machine is monitored to distinguished abnormal and normal operation and to determine the RuL. In this hydraulic machine, sliding wear is described to consists of parts that slide over each other, leading to degradation of the system (due to stress from friction). The aim is to evaluate the wear of the system by monitoring the surface conditions. As one of the challenges faced for monitoring conditions based on classification and filtering approaches is extracting features without prior knowledge of the system, the authors aim to develop for optimal state classification using the measurements based on realistic operations by adjusting the mechanical and thermal conditions. The AE measurements and hydraulic pressure are used for the application of the model. The state machine model is employed to develop four states, defining different conditions of the surface. Classifying the different states (switch between states) are based on threshold values related to the measurement data.

The state machine developed in [KS20] models plant growth based on water stress. The goal of this work is to control water consumption and plant growth using different water stress sequences levels. Thus, the state machine models the plant growth subject to different irrigation treatments. The different states in this model show the plant growth state in terms of water stress (water availability/demand), memory level (function of water stress), damage level (function of time and stress), time between successive states, and duration in a specific state. A total of seven states are defined. The transition between states are also based on threshold conditions related to the water stress level defined by NGSA-II [KS20].

In [RL23], a FSM is used to develop a control system for the autonomous regeneration machine to perform a full tree planting cycle autonomously. Each state represents a composite task performed by the machine, such as crane movements, seedling transfer, etc. The aim of the machine is to facilitate the communication between the different subsystems (hardware and software) [RL23].

3.1.4 Difference between previous models and proposed model

Driving behaviors and LIBs' degradation exhibit similar structurally variable behavior to the technical system based on tribological experiments [BS17], sliding wear of hydraulic system [SR17], and plant behavior [KS20]. Hence, the state machine is used to model

the lane changing behaviors and degradation of LIBs. The main difference between the previous contributions and this thesis is

- The previously introduced state machine models do not incorporate other approaches to the model. On the other hand, in this thesis, an ANN and HMM are utilized for the estimation of driving behaviors. As for the batteries, the NARX is incorporated into the state machine to estimate the capacity and degradation states.
- The transition conditions also differ in certain models. The previous works only consider transition conditions related to the variables that describe the operating conditions of the system. On the other hand, here, different models are used to define the transition conditions in the combined state machine models.

4 Proposed state machine approach for driving behavior estimations

In this chapter, the state machine model developed for driving behavior estimations is introduced. The chapter is distinguished by the different state machine-based models, the evaluation of features on the proposed state machine model, and hyperparameter optimization of the model.

Parts of the contents, figures, and tables presented in this chapter are modified based on previous publications [DRS20] [DRS21] [DS22b] [DS23b] [DS22a] [DS23a].

4.1 State machine-based approach for driving behavior recognition

As the state machine has shown promising results in other areas of research, the aim is to apply this approach for the recognition of driving behaviors. To establish a driving behavior model, first inputs and outputs of the considered system are necessary to be defined. Three different driving maneuvers such as lane keeping (LK), lane change to the left (LCL), and lane change to the right (LCR) are modeled as the states, which are also the model's output. The variables affecting driver's decisions are used as inputs, which are measurable. In general, different states of the ego vehicle (position, speed, acceleration, steering wheel angle, etc.) and information about surrounding vehicles are used as inputs. The driver's decision is often based on the ego vehicle's relationship with surrounding vehicles, current environmental conditions, and individual driving styles. Decisions rely on individual's perception of environmental variables and their combination, like velocity of ego vehicle and actual angle of steering wheel.

The state machine approach is assumed as given in Figure 4.1. The topology shown consists of the three states transitioning from one state to another based on specific parameters/thresholds conditions. The threshold conditions are related to the input variables, which are the different environmental variables [DRS20]. The variables used are lane number (l), angle of steering wheel (a_{st}), accelerator pedal position (a_{acc}), brake pressure (a_{br}), indicator (i), time to collision (TTC) to the vehicle in front (TTC_f), TTC to the vehicle in the back (TTC_b), TTC to the vehicle in the front left (TTC_{fl}), TTC to the vehicle in front right (TTC_{fr}), TTC to the vehicle in the back left (TTC_{bl}), and TTC to the vehicle in the back right (TTC_{br}), given in Table 4.1.

In this case, when the current estimated state of the vehicle is LK (denoted as state 2), the model can switch to LCR (denoted state 1) or to LCL (denoted as state 3) if the model satisfies a set of threshold conditions. State LCR and LCL can only switch to LK if the current lane is not the same as the previous time point. In Table 4.1, the first set of thresholds for each variables are used to define a transition from LK to LCR, while the second set of thresholds is used to define a transition from LK to LCL. These threshold

values are generated automatically through the Non-dominated sorting genetic algorithm II (NSGA-II) optimization [DPAM02]. For a transition from LK to LCR at a specific time point, values of either one of the variables at that time point have to be within the first set of thresholds. For a transition from LK to LCL, the values of either one of variables generated should be within the second threshold set. If these threshold conditions are not met, the state machine remains in the same state. The T_n denotes threshold values and the maximum number of thresholds/parameter is 40, hence n = 40.

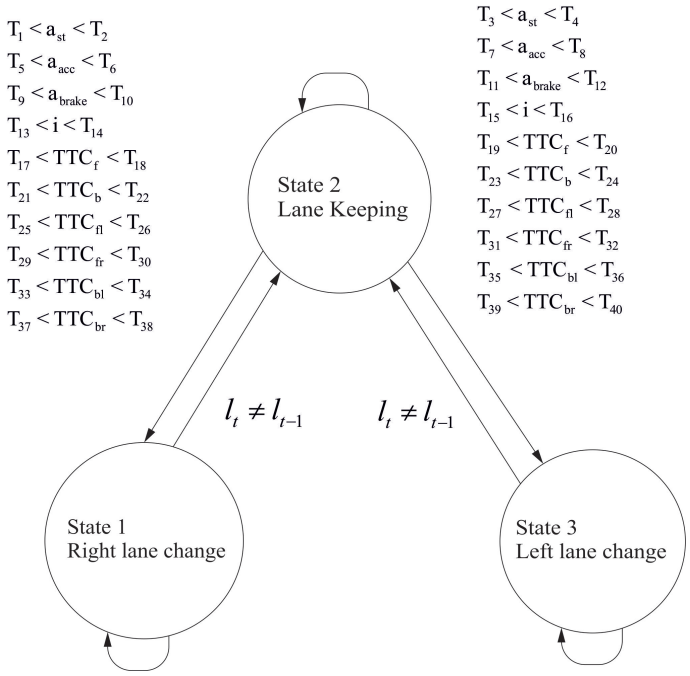

Figure 4.1: State machine topology for the driving behavior prediction [DRS20]

Table 4.1: Description of driving variables and related optimization thresholds [DRS20]

Input	Variables	Design parameters
a_{st}	Angle of steering wheel	$[T_1..T_2]$ $[T_3..T_4]$
a_{acc}	Accelerator pedal position	$[T_5..T_6]$ $[T_7..T_8]$
a_{brake}	Brake pedal pressure	$[T_9..T_{10}]$ $[T_{11}..T_{12}]$
i	Indicator	$[T_{13}..T_{14}]$ $[T_{15}..T_{16}]$
TTC_f	Time to collision (TCC) to the vehicle in the front	$[T_{17}..T_{18}]$ $[T_{19}..T_{20}]$
TTC_b	TTC to the vehicle in the back	$[T_{21}..T_{22}]$ $[T_{23}..T_{24}]$
TTC_{fl}	TTC to the vehicle in the front left	$[T_{25}..T_{26}]$ $[T_{27}..T_{28}]$
TTC_{fr}	TTC to the vehicle in the front right	$[T_{29}..T_{30}]$ $[T_{31}..T_{32}]$
TTC_{bl}	TTC to the vehicle in the back left	$[T_{33}..T_{34}]$ $[T_{35}..T_{36}]$
TTC_{br}	TTC to the vehicle in the back right	$[T_{37}..T_{38}]$ $[T_{39}..T_{40}]$

4.1.1 Optimization

The state machine model introduced here has two major components. The first part consists of determining the thresholds (design parameters) through optimization. As mentioned, a

total of 40 parameters are optimized. The second part focuses on determining/recognizing the driving states based on the optimal thresholds. The thresholds related to the input variables determine whether a state transition can occur. The driving behaviors determined using this model and the actual driving behavior (by a human driver) will be compared to evaluate the accuracy and the reliability of this model. The main aim is to establish a suitable recognition performance with respect to the well-known metrics accuracy (ACC), detection rate (DR), and false alarm rate (FAR) [Pow11].

As transitions from one state to another are dependent on thresholds values, these values are defined by NSGA-II [DPAM02] subjected to maximal DR, maximal ACC, and minimal FAR (or maximal 1-FAR). The ACC, DR, and FAR are determined based on true positive (TP), false positives (FP), true negative (TN), as well as false negative (FN) values. True positive (TP) is calculated based on the number of events when an estimated maneuver is positive (right lane change) and the actual maneuver is positive as well. False positive (FP) is based on the number of events when an estimated maneuver is positive but the actual driving behavior is not [DWS18]. This concept is applied to the true negative (TN) and false negative (FN) as well. Thus, this enables the evaluation of the metrics given by

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.1}$$

$$DR = \frac{TP}{TP + FN}\ , \text{and} \tag{4.2}$$

$$FAR = \frac{FP}{TN + FP}. \tag{4.3}$$

Suitable objective functions are selected to evaluate the optimization process by comparing the actual states (actual driving behaviors) and the calculated states at each moment. In the state machine model introduced in this paper, the metrics ACC, DR, and FAR are used to describe the objective functions with respect to minimizing the deviation between actual and estimated driving behaviors. The termination criteria is based on the maximum generation of the NSGA-II. The optimal values of parameters are generated when the conditions are fulfilled, which will then be used to calculate the driving behaviors. The objective functions are defined as

$$f_1 = (1 - DR_{right}) + FAR_{right}, \tag{4.4}$$

$$f_2 = (1 - DR_{keep}) + FAR_{keep}, \text{and} \tag{4.5}$$

$$f_3 = (1 - DR_{left}) + FAR_{left} \tag{4.6}$$

such that, each equation represents each state.

Defining the optimal threshold can be challenging depending on the problem type and the optimization technique used. For 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 [SLY+17]. In addition, it is not suitable for multi-objective problems, as the one presented here. In contrast, the NSGA-II is used for multi objective optimization as it consist of three main features: uses an elitist preserving method, diversity preserving which involves crowding distance and highlights the non-dominated

results [DPAM02] [Deb01]. Due to the conflicting objectives, NSGA-II is used to handle the multi-objective problem presented. As previously mentioned, the non-dominated solutions developed show a set of solutions that are not dominated by other solutions, hence these solutions do not improve the objective functions without degrading any of the functions. The design parameters generated from NSGA-II are used to minimize the objective function. In Table 4.2, the list of arguments/inputs required for NSGA-II are shown. These configurations were selected as the model developed the most effective estimation performance using these values. Nevertheless, certain parameters have default values such as the mutation fraction, as it based on the number of design parameters.

Table 4.2: Description of NSGA-II options [DRS20]

Parameter	**Value**
Maximum Population	20
Maximum Generation	50
Cross over fraction	10
Mutation fraction	1/number of design parameters=1/40
Cross over variable	Intermediate 1.2
Mutation variable	Gaussian, 0.1, 0.05

The optimal threshold values selected by the optimizer presented in Figure 4.2 are based on one driver's data set.

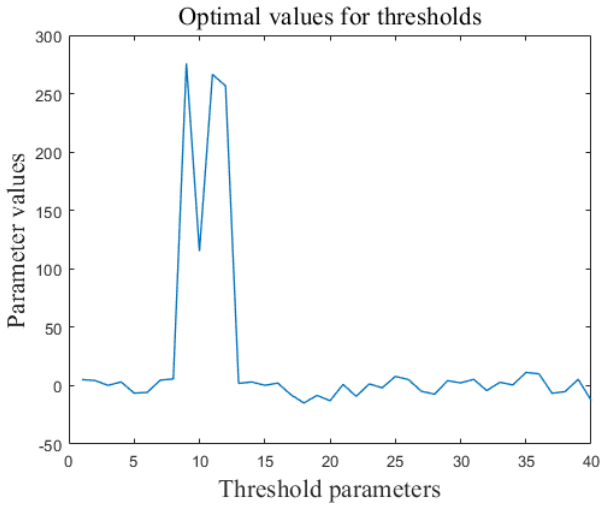

Figure 4.2: Optimal threshold values

4.1.2 Summary

The proposed state machine approach is developed for the recognition of the lane changing behaviors. Three different states are defined, LCR, LK, and LCL to represent the lane changing behaviors. Transitions between different states are established to determine the behavior estimation. The transition from one state to another is defined by threshold conditions related to environmental variables/driving features (input variables). If a

threshold condition is fulfilled, such that the value of a specific input variable is within the threshold values at a specific time point, a transition to another state can occur. If the conditions are not fulfilled, the model remains in the same state. The threshold values are model parameters defined by the NSGA-II optimization technique to develop an optimal model.

4.2 Extension of the state machine model

The approach described previously is only based on the state machine approach and its threshold conditions. A limitation that exists is that certain complex driving patterns performed by the driver may not be captured/represented by these conditions, hence affecting the estimation performance of the model.. As known from [JF17] and [ZWL15], combined approaches tend to have an extra advantage for better performance as characteristics from multiple methods are considered. In addition, each method can be used to perform different tasks in the model. In this thesis, the goal is to combine the newly developed state machine ML approach with other well-known approaches. Here, two separate models are developed, whereby one combines the state machine with ANN, while the other with HMM. The ANN is chosen due to its ability to handle complex non-linear data, parallel processing, and extract relevant features automatically. The model learns from given behavioral patterns and estimates future driving behaviors for similar situations. On the other hand, the HMM is chosen due to its ability to handle time series data and its stochastic properties.

4.2.1 ANN-based state machine model

The aim is to establish a model that combines two trainable systems (ANN and state machine-based approach) for the recognition lane changing behavior [DRS21]. 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, while the ANN's estimations model the transition conditions. The transition conditions differ from the state machine model in section 4.1 which uses threshold conditions instead. 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.

As input variables, the similar environmental features used for the state machine model is selected as these features play a major role in the driver's behaviors, given in Table 4.1. The output is similar to the state machine model: LCR , LK, and LCL. First, the input variables along with the model parameters are used to estimate the lane changing behaviors using ANN. These estimations are integrated into the state machine model as conditions for a state transition or to remain in the same state.

For approach I (Figure 4.3), transition or remaining conditions are defined by the outputs of one common ANN. For a transition from LK to LCR or LCL in the proposed model, the output of ANN should be LCR or LCL respectively as well, at that time point. Similarly, for a transition from LCL or LCR to LK , the output of ANN should be LK. If the conditions are not met, the model remains in the same state.

Following the same integration and transition process as approach I, the transition conditions in approach II (Figure 4.4) are based on the ANN's estimation corresponding

to the current estimated state, whereby three networks are defined as ANN (right), ANN (keep), and ANN (left) (Figure 4.4). The possible outputs for the three networks are listed in Table 4.3. If the estimation of a specific ANN is same as the current state, then the model remains in the same state.

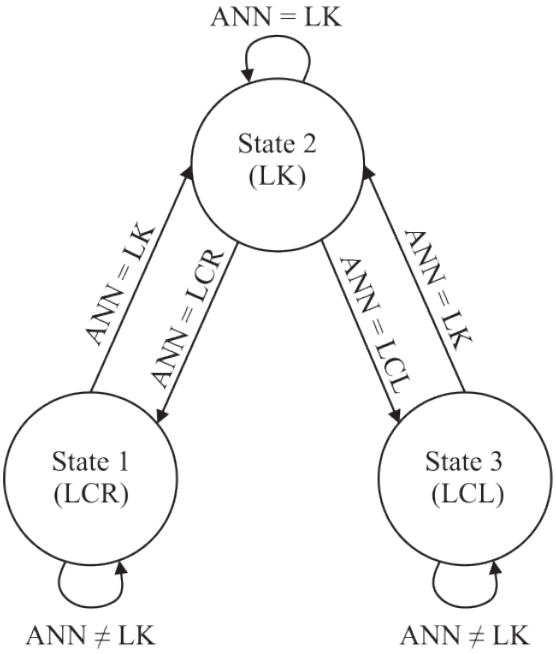

Figure 4.3: State machine and one ANN diagram (approach I) [DRS21]

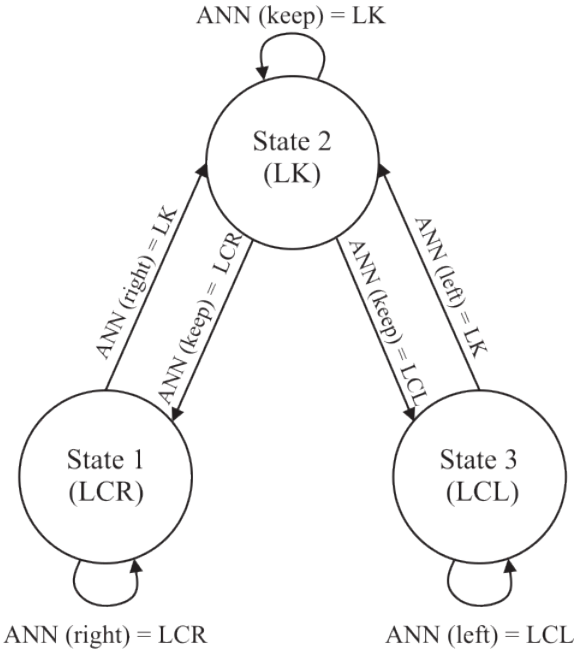

Figure 4.4: State machine and three ANN diagram (approach II) [DRS21]

Table 4.3: Outputs of the three ANN [DRS21]

ANN models	Possible outputs (states)
ANN(right)	LCR , LK
ANN(keep)	LCR, LK, and LCL
ANN(left)	LK , LCL

Optimization of parameters

The model parameters are the weights and biases associated with the ANN that affect the estimation performance of the overall models. A total of 153 parameters are optimized for the model in approach I, while a total of 459 parameters are optimized for the model in approach II. Thus, defining these parameters by optimization is necessary to develop optimal lane changing estimations. As the previous model, the NSGA-II [DPAM02] is used to define the optimal parameters in the training process. The NSGA-II uses a generation size of 200 and a population size of 90. The optimal parameters are developed with respect to maximal ACC, DR, and minimal FAR. 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 similar objective functions used for the optimization of the state machine model, defined in section 4.1.1 are used. The performance of the overall model can be evaluated using ACC, DR, and FAR .

The optimal weights, and bias values for approach I are given in Figures 4.5-4.6 based on one data set. Two weight and bias sets for the different layers are shown, whereby the weight 1 shows the weight connections between input layer and hidden layer. On the other hand, weight 2 shows the weights between the hidden layer and output layer. The bias values only corresponds to the neurons in the hidden layer and output layer.

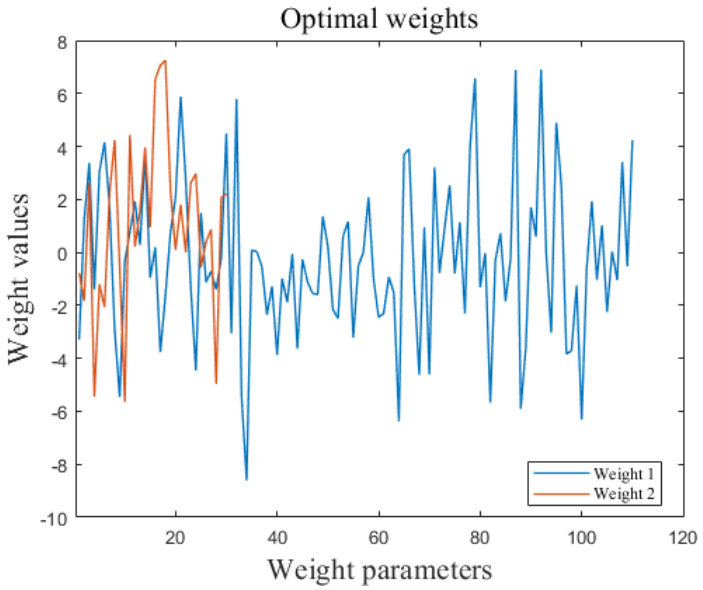

Figure 4.5: Optimal weights

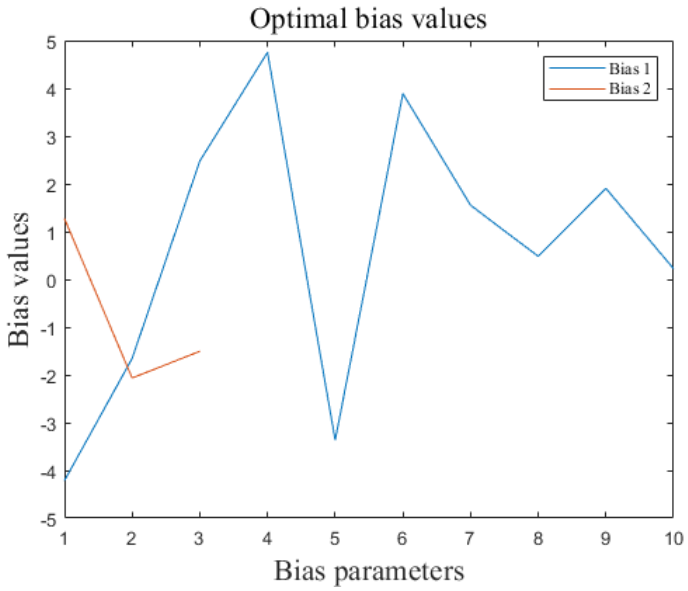

Figure 4.6: Optimal bias values

4.2.2 HMM-based state machine model

A new HMM-based state machine model is introduced in this section [DS22b] [DS23b]. Here, 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 structure of this model is similar to the ANN-based state machine model [DRS21], which uses the ANN estimations instead as the transition or remaining conditions. Two improved HMM are utilized in this work.

Review of improved HMM

The HMM model is given in Figure 4.7. The observation sequence (inputs, $V = \{V_1, V_2, ..., V_M\}$) and hidden state sequence (outputs, $S = \{S_1, S_2, ..., S_N\}$), whereby M and N are the number of observations and hidden states respectively. The hidden states are the lane changing behaviors, hence, N=3. 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, $\lambda = (A, B, \pi)$. The HMM model is trained using the Baum-Welch algorithm to develop the λ that fit a given observation and the corresponding hidden state sequence. Using λ, the hidden state sequence with the highest probability is determined using the Viterbi algorithm to generate the estimated behaviors.

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. For performance improvements, various approaches have been established such as a combination of HMM with other methods and HMM-derived methods [DS22c]. The combination of

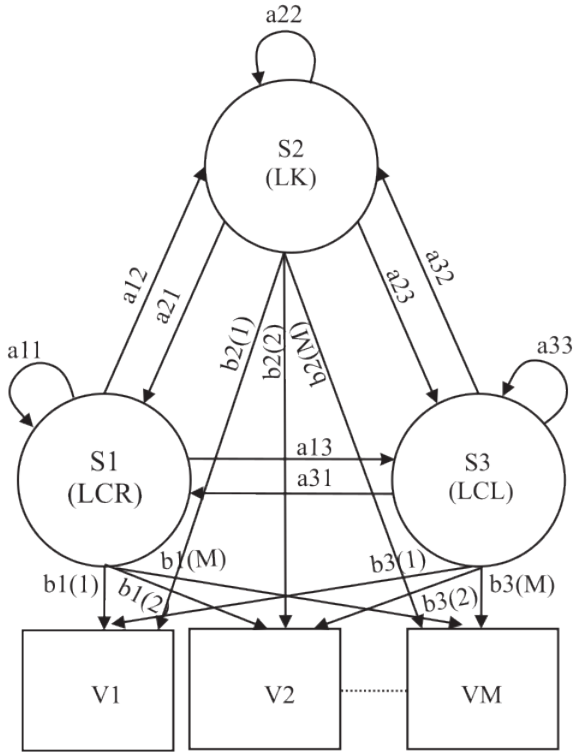

Figure 4.7: Hidden Markov Model for driving behavior estimations [DS22b]

HMM with other methods includes ANN-HMM [JF17], Fuzzy Logic (FL)-HMM [JF17], and Gaussian Mixture Model (GMM)-HMM [WZHX18]. These methods use results from one method as input to the other to determine the final behavior estimation. In the combined HMM, one of method can also be used to determine the parameters, while the other to classify different driving styles, behaviors, or situations. On the other hand, the HMM-derived methods such as Hierarchical HMM [FS98] and Bayesian non-parametric HMM consider the time series property of HMM [NTTB14]. 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 features 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 [DWS18], includes the application of a prefilter for performance efficiency. The prefilter aims to quantize the driving features (measured input variables) into observation sequences. 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. For example, an input variable (distance to the vehicle in front feature) is divided by the prefilter into three segments with two thresholds [DWS18], given in Figure 4.8 and Table 4.4. In this thesis, the same prefilter application is considered in the HMM model.

Table 4.4: Prefilter segments

Segments	Thresholds (m)
S1	$d_f < a_{df}$
S2	$a_{df} \geq d_f \leq b_{df}$
S3	$d_f > b_{df}$

Improved HMM I

Dynamic driving data (driving features) that changes with time, changes the observation variables in the HMM model. Hence, a prefilter with thresholds is applied to quantize the data variables (inputs) to develop feature vectors for a better interpretability, as in [DS19a] and [DWS18]. The feature vector is used to determine different driving situations [DWS18]. In this model, the prefilter divides each feature into three segments with two thresholds [DWS18]. Each segment is then used as an observation for the HMM.

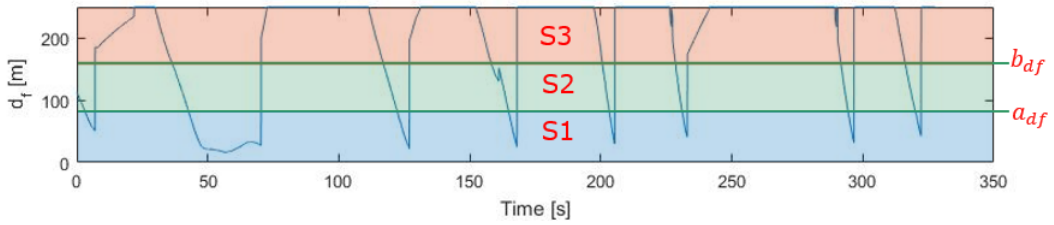

Figure 4.8: Prefilter application to the distance top the vehicle in front feature

Improved HMM II: Sub-HMM

With the prefilter application, higher number of input variables (driving features) increases the number of segments and observation variables, which heightens the complexity of the observation matrix B [DS19a]. Hence, the process is computationally expensive, in terms of training time. To simplify the model, four sub-HMMs are developed the improved HMM II, such that each sub-HMM is given different inputs: HMM 1 (TTC to vehicles in different directions), HMM 2 (distances), HMM 3 (velocities), and HMM 4 (driving operational) [DS19a]. A prefilter is applied to the driving features here as well. The prefilter considered in this model divides each feature into six segments with five thresholds. To obtain the HMM's final estimation, the probabilities of different sub-HMM models are fused using weights to calculate the final probability given as

$$P = \sum_{k=1,2,3,\,or\,4}^{1,2,3,and/or4} w_k \times P_k[\text{DS19a}], \tag{4.7}$$

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 section 4.2.4.

4.2.3 Development of HMM-based state machine model I

The improved HMM I is combined with the state machine approach here [DS22b]. The estimations of improved HMM I are considered as the transition conditions. Based on Figure 4.9, LCR or LCL transitions to LK when the estimation of improved HMM I is LK. On the other hand, LK switches to LCR or LCL when the HMM estimation is 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 transition conditions are summarized in Table 4.5.

Table 4.5: Transition conditions [DS22b]

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

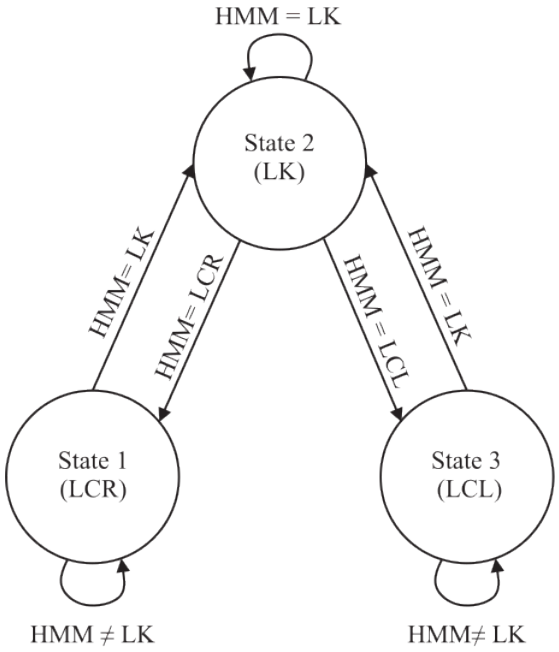

Figure 4.9: HMM-based state machine model [DS22b]

Feature selection: HMM-based state machine I

As mentioned previously, only environmental variables are used in the development of the models. Accordingly, two models (models I and II) are developed using two sets of input variables based on the proposed model. 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 4.6 and Table 4.7). The prefilter is applied on the distance and velocity variables for model I and on the TTC variables for model II (dividing

each feature into three segments with two thresholds). As for the current lane number, the values are fixed indicating the specific lane of the ego vehicle.

Table 4.6: Environmental variables of model I [DS22b]

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 4.7: Environmental variables of model II [DS22b]

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

4.2.4 Development of HMM-based state machine model II

For the HMM-based state machine model II [DS23b], the improved HMM model II will be integrated instead. The similar topology and transition process as HMM-based state machine I is considered .

Feature selection: HMM-based state machine II

As previously stated, four sub-HMMs are defined with different environmental features as inputs. The different features for the different sub-HMM are described in Table 4.8. A total of eleven combinations of sub-HMMs (given in Table 4.9) are evaluated to examine the effectiveness of the different features as well as the relevance of the features for the estimations.

4.2.5 Optimization of parameters

The prefilter thresholds (HMM-based state machine models I and II) and weights (HMM-based state machine models II) are optimized using NSGA-II to generate an optimal HMM parameter $\lambda = (A, B, \pi)$ for performance improvements. For HMM-based state machine I, a total for 12 parameters are optimized for both models I (distances, velocity) and II (TTCs). The optimal threshold values for models I and II are given in Figures 4.10- 4.11 based on the training of one data set. As for the HMM-based state machine

Table 4.8: Input variables for the four sub-HMMs [DS23b]

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 4.9: Combination of different sub-HMM models [DS23b]

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

II, the number of parameters optimized ranges from 62 to 114 depending on the number of sub-HMM combinations. The weights represent the impact of each sub-HMMs. The threshold values are required to define the observation sequence, ultimately affecting the model's performance. Similar objective functions defined in section 4.1.1 are minimized for the NSGA-II. The generation size for the NSGA-II is 200, while the population size is 90. In Figure 4.12 and Figure 4.13, the optimization process for the different models is given.

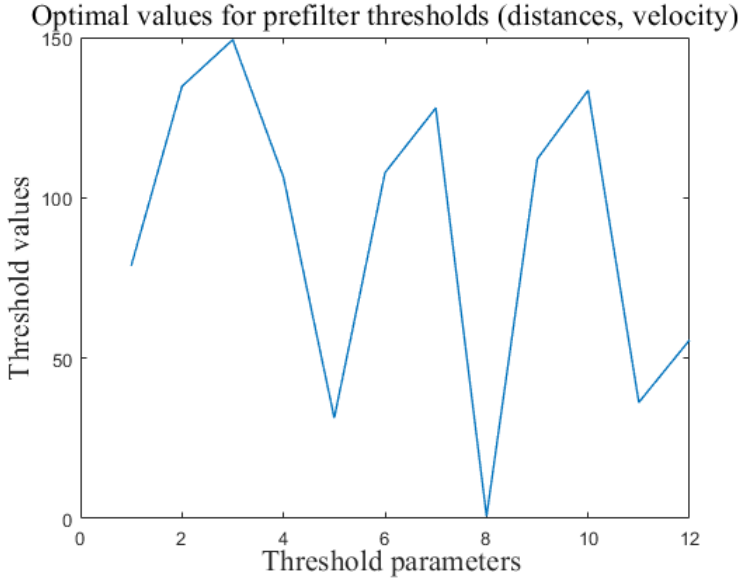

Figure 4.10: Optimal threshold values (model I)

4.2.6 Summary

In this section, the extension of the state machine model is introduced, whereby the ANN and HMM are combined with the state machine model to develop two separate model extensions. The estimations of the ANN and HMM serve as transition conditions.

In the ANN-based state machine model, two models are developed, whereby one has a single ANN model, while the other consists of three ANN models. For the HMM-based state machine model, two models are developed as well. An improved HMM model is used, with the application of a prefilter. The HMM-based state machine model I considers the prefilter application for two driving feature sets (as inputs), distances, and TTCs. As for the HMM-based state machine model II, the prefilter is applied to four feature sets. Four sub-HMMs are developed, such that each sub-HMM is given a specific feature set as inputs. The different sub-HMMs are fused to develop the estimation of the overall HMM model. Therefore, different combination of sub-HMMs are fused to evaluate the most effective feature combination for the lane changing behavior recognition. In Table 4.10, a summary of the models is given.

4.3 Evalution of features

Different driving features as input to the model have different effects on the estimation performance, showing the significance of a particular feature. Extracting the appropriate

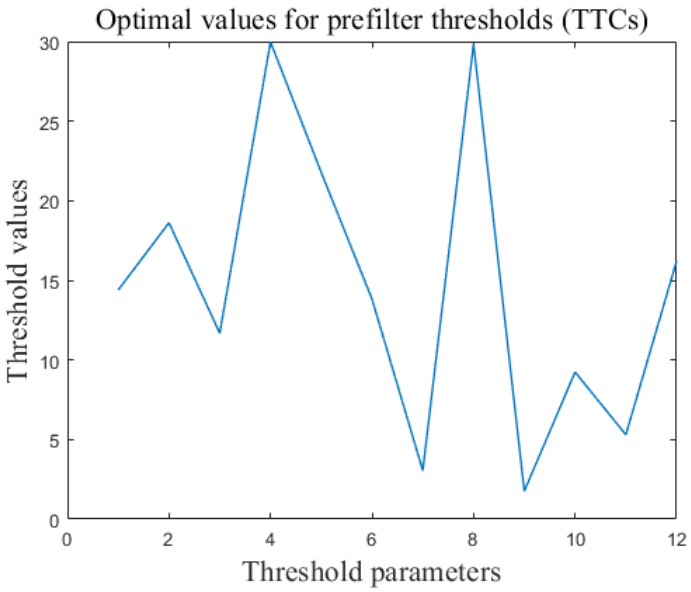

Figure 4.11: Optimal threshold values (model II)

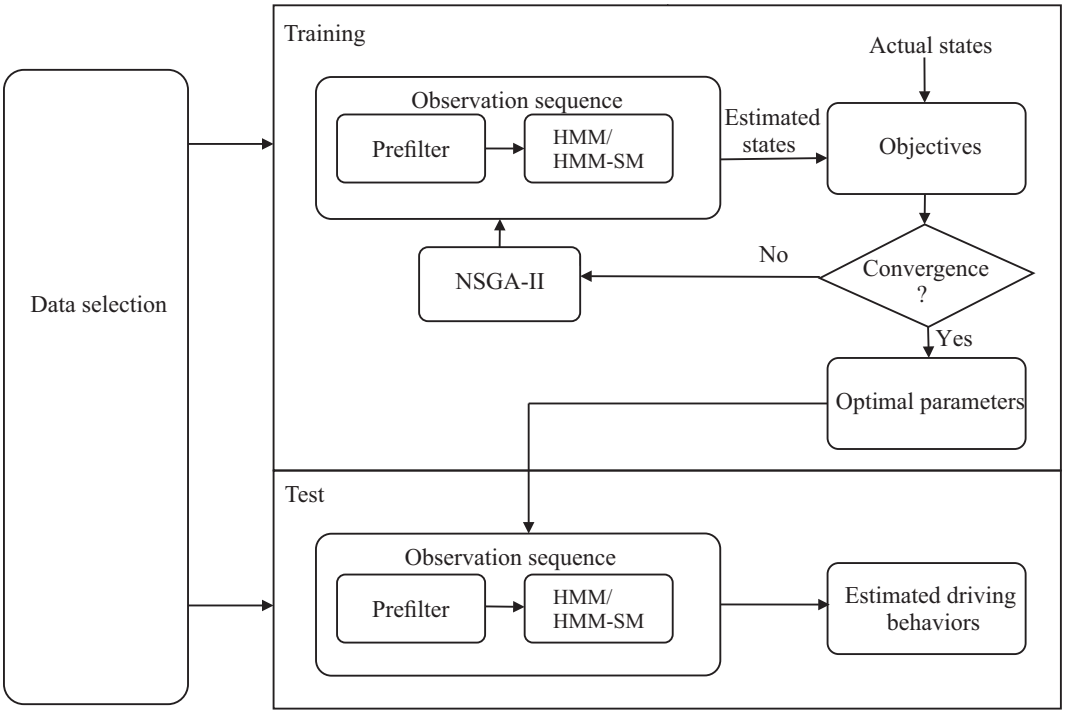

Figure 4.12: Optimization procedure for HMM-based state machine model I [DS22b]

Table 4.10: Summary of the ANN and HMM-based state machine models

Models	Definition	Parameters optimized	Prefilter threshold
ANN-based state machine I	Transition conditions: Estimations of one ANN model	Bias values and weights	-
ANN-based state machine II	Transition conditions: Estimations of three ANN (representing the different behaviors)	Bias values and weights	-
HMM-based state machine I	Prefilter application to distances (model I) and TTCs (model II)	Prefilter thresholds	Three segments with two thresholds
HMM-based state machine II	Sub-HMMs developed for TTCs, distances, velocities, and operational	Prefilter thresholds and weights	Six segments with five thresholds

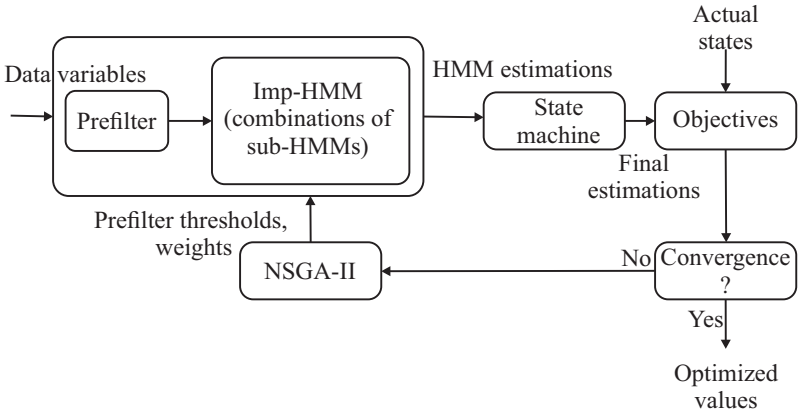

Figure 4.13: Optimization procedure for HMM-based state machine model II [DS19a], [DS23b]

features as inputs is a continuous and important part of research in developing driving behavior models. Feature extraction can be performed manually or it can part of ML-based algorithm, such as deep neural networks. Accordingly, the aim of this contribution is to study effects of common feature types on the newly developed state machine approach. The influence of environmental (ENV), eye-tracking (ET), and physiological features play a significant role in driving behavior, thus evaluating the influence of these variables on the the proposed approach is of interest. The impact of these feature on the driving behavior predictions and recognition have been previously studied in other research contributions such as [SLZ+11] [DT09]. The limitations of these studies are the influence of the features is only evaluated using single conventional ML-based models. The goal here is to study the effect of these features on a combined ML-based model, more specifically the newly developed state machine-based model.

The ENV variables can influence driving styles (aggressive/non-aggressive/risky), drunk driving, speed changes, trajectory changes, and lane changing behaviors making it important to incorporate these information as inputs into the estimation models. In general, ET features seem to be more effective for the recognition of drunk and fatigue driving [JXP22]. On the other hand, it is known from other research contributions such as [DHBS20] [DT09], that the incorporation of ET features does not necessarily improve the lane changing estimation performance. In [DHBS20], the combination of ET with ENV features in SVM-based model generates a poor estimation performance when compared to the model using only ENV features. The researchers suggest that combining a feature set with lower accuracy feature set results in impoverished performance compared to using the feature set individually, as proven in the classifier fusion study by [RS16]. Nevertheless, improved performances were observed using the combination of ENV and ET features in [LW17] and in some models developed in [DHBS20]. Hence, the the effectiveness of ENV and ET features are examined in this thesis. While physiological variables are used in [MKW+15], there are still a lot of open questions on the effectiveness of these variables for lane changing behavior estimations.

The ANN-based state machine II is utilized to study the effects of using only ENV and a combination of ENV and ET features on the recognition of lane changing intentions [DS22a]. The ET features provide information about the human driver's eye movements

(saccade).

4.3.1 Features variables: data processing

A total of nine ENV and six ET features are selected for the intention recognition process (Table II). These nine ENV features have been validated to accurately describe the driving environment (ego vehicle state and relationship with other vehicles). On the other hand, the selected ET features are proven to show accurate saccadic eye movement information [DHBS20]. To evaluate the effect of the different features on the model's performance, two models are trained for each participant: model I using only ENV features and model II using both ENV and ET features.

Table 4.11: Description of input variables [DS22a]

Symbol	Descriptions	Feature types
l	Current lane number	Environmental (ENV)
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	
a_h	Heading angle of the ego vehicle	
TTC_f	Time to collision to the vehicle in the front	
v	Velocity of the ego vehicle	
$Saccade$	Saccade	Eye-tracking (ET)
Bl	Blink	
F_{Bl}	Blink Frequency	
N_{screen}	Screen Number	
C_x	Screen coordinate (x-axis)	
C_y	Screen coordinate (y-axis)	

Optimization of parameters

Similar to the previous ANN-based state machine model, the parameters are the weights and biases. Hence, the NSGA-II is used with the same objective functions to develop the optimal parameters. For model I (ENV), a total of 133 parameters are optimized, while for model II (ENV and ET) 193 parameters are optimized. The generation and population size for the optimizer are 150 and 90, respectively.

4.3.2 Summary

In this section, the ANN-based state machine II model is utilized to study the effectiveness of ENV and ET features (as model inputs) on the performance of the model for lane changing intention recognition. As inputs only ENV and a combination of ENV and ET are used separately to determine the feature types that are most effective. In addition, the aim is to study if the ET features improve the recognition abilities of the proposed model.

As known from previous research, ET features don't always improve the performance of a conventional ML-based model [DHBS20].

4.4 Hyperparameter optimization of the state machine approach

As hyperparameters tend to affect the performance of an estimation model, selecting optimal values are important. The optimization of these values can be performed through manual methods or automated methods as discussed in section 2.1.6. The automated methods (BO and GA) are applied to the develop hyperparameter values for the ANN-based state machine II model [DS23a], as both generated better results compared to other methods in research contributions, such as [HK21]. Hyperparameters associated with the ANN model are optimized using the BO and GA first, then the ANN model structure (with optimized hyperparameters) is combined with the state machine to develop the estimations.

4.4.1 Bayesian optimization

Bayesian optimization (BO) is an iterative approach for optimizing parameters of an objective function f_1. The objective function is expensive to compute which can be challenging for optimization. Therefore, BO aims to develop a probabilistic model (a surrogate model which is cheaper to compute) to approximate the objective function. Two key factors in the BO: prior (for developing the surrogate model) and the acquisition function. Bayesian optimization integrates prior belief about the objective function and updates the prior in order to generate better approximations of the objective function. Hence, when building the surrogate model, a Gaussian process is usually used to include prior belief. The acquisition function is part of the surrogate model used to select the next optimal point (where the acquisition function is maximize). Generally, there are two types of acquisition functions: confident-based criteria and improvement-based criteria [JRGV16] [MTZ78]. Here, an improvement-based criteria known as 'expected-improvement-per-second-plus' function is chosen. The acquisition function takes into account exploration (space of high uncertainty) and exploitation (space with high objective value-area of current best hyperparameter values) when searching the search space for the next point [MTZ78]. Past evaluations are considered to select the next set of hyperparameter values defined by the acquisition function [WCZ+19] [MTZ78]. Selecting parameter combinations based on past evaluations, allows the optimization to focus on the space which generates the most accurate validation. The objective function for this optimization technique shows the measure of loss, given as

$$f_1 = \sum \frac{1}{n}(y_i \neq \hat{y}_i), \tag{4.8}$$

whereby n is the total number of observations, y_i is the actual lane changing behavior, and $\hat{y}_i$ is the estimated behavior. The hyperparameters optimized are the number of hidden layer neurons, activation function of the ANN's first layer, learning rate, and number of epochs. These variables are selected as they are known to affect the selection of model parameters [YS20].

4.4.2 Genetic algorithm

The GA is a heuristic optimization method. This technique is a stochastic global search optimization technique that changes the population of individual solutions repeatedly

[Hol73]. The solutions are the hyperparameters to be optimized. A similar objective function (known as fitness function) as used in Bayesian optimization is selected here. The fitness function describes how well a solution fits the optimal solution [Hue97] [CJ15]. Two existing solutions can be combined to develop a new solution (known as crossover) [Hol73]. A new solution can also be obtained by making random changes to an individual existing solution (known as mutation) [Hol73]. On the other hand, an elite solution has the best fitness value in the current iteration, and is automatically passed on to the next iteration. Thus, the GA process can be described as follows [EDSD96]:

1. Initialize the population and generation
2. Calculate the fitness function for each individual solution
3. Select individual solutions based on the fitness function
4. Certain solutions have better (lower) fitness values, known as elites, which are passed to the next iteration
5. Perform the crossover or mutation between the solutions
6. Repeat steps (2) to (4) until convergence (or the iteration limit has been reached)

To generate an optimal ANN model, the same hyperparameters used in the BO are considered.

4.4.3 Advantages and disadvantages

One of the main benefits of BO is less iterations are required to reach the optimal parameter set [MTZ78] [WCZ+19]. This is because it only updates the posterior information, unlike the GA which needs train the same model on various hyperparameters from one iteration to another [MTZ78] [MTM05]. The BO only needs to store prior distribution which does not require a large amount of memory, while for GA storing information about multiple individual solutions when the model has large number of hyperparameters may not be feasible, which can slow down the running time of the algorithm. On the other hand, GA posses parallelism capabilities, whereby multiple samples can be trained in parallel and the best solution is selected based the evaluations [MTM05]. In BO, as current experiments depend on the previous experiments based on the acquisition function and surrogate model, it is unable to utilize parallelism. While both methods have the capability to develop optimal search spaces, both methods may get stuck in the local optima rather than reaching the global optima [MTM05].

4.4.4 Application of hyperparameter optimization methods

In this work, the number of hidden layer neurons, the activation function of the first layer, learning rates, and the number of epochs are the hyperparameters considered for optimization using both methods to develop the ANN model [DS23a]. The selection ranges of the hyperparameter values for the techniques are detailed in Table 4.12. Also, the number of iterations used for the BO is 30 (as it requires fewer number of iterations), while the population and generation sizes for GA are 25 and 100, respectively. Different optimal values are developed for each set. The ANN model structure is based on the

optimal number of hidden neurons and the activation function. In contrast, the learning rate and number of epochs only aids the selection of other hyperparameters (in terms of speed). For each data set, the three ANN models are based on the same optimized hyperparameters. The ANN models with the optimized hyperparameters are combined with the state machine to generate the final estimations. The ANN-based state machine II is employed for the estimation process. The objective function values based on BO is presented in Figure 4.14.

Table 4.12: Hyperparameter optimized using both methods [DS23a]

Selected hyperparameters	Ranges
Hidden layer neurons	1-30
Activation functions	Hyperbolic tangent sigmoid (tansig), Log-sigmoid (logsig)
Learning rates	0.001-1
Epochs	100-10000

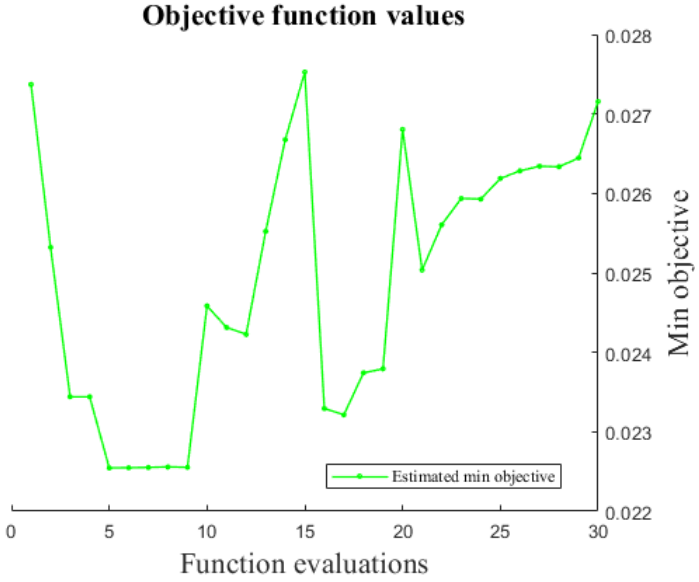

Figure 4.14: Objective function values for BO

Optimization of parameters

For the optimization of the model parameters, the NSGA-II is used here as well to optimize the weights and biases. In addition, the similar objective functions are applied. Here, 459 parameters are optimized, same as the ANN-based state machine II. Generation and population size for the execution of NSGA-II are 200 and 90. The hyperparameter and model parameter optimization scheme is presented in Figure 4.15.

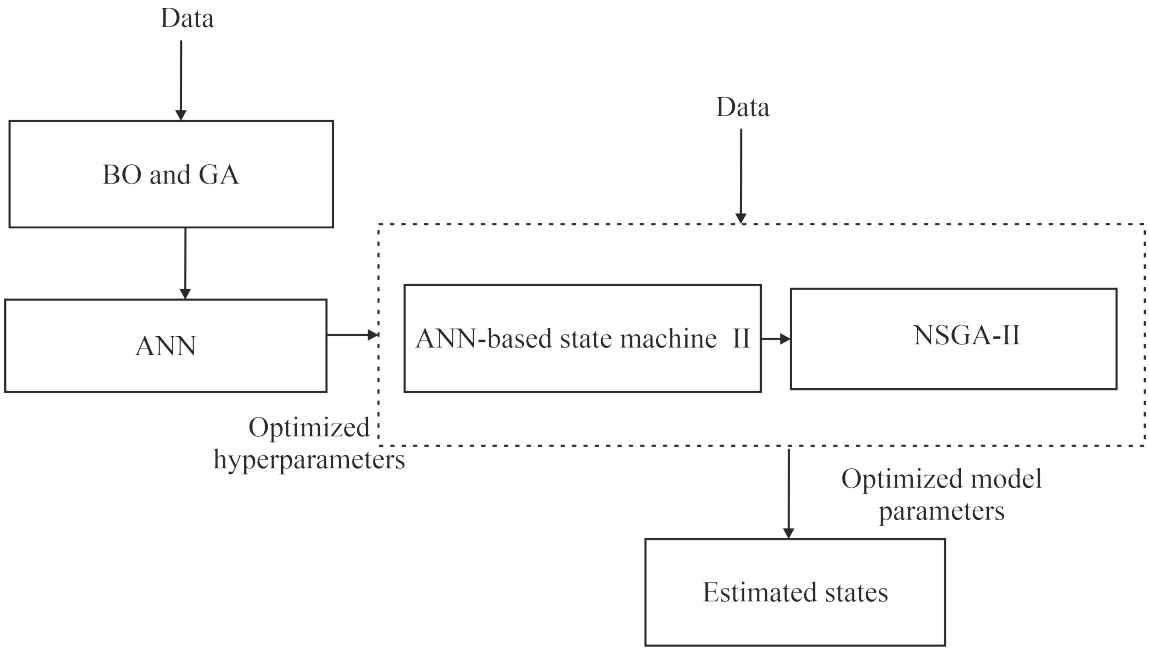

Figure 4.15: Hyperparameter and model parameter optimization

Feature selection

The features addressed in Section 4.1 for the development of the ANN-based state machine are considered for this application.

4.4.5 Summary

The BO and GA as methods for optimizing the hyperparameters of the ANN-based state machine II model are employed here. The hyperparameters optimized are hidden layer neurons, activation function, learning rates, and epochs. The ranges for the optimization are selected based on previous experience and research within the area of optimization. The ANN models with the optimized hyperparameters are combined with the state machine to develop the final lane changing estimation.

5 Proposed state machine approach for lithium-ion batteries (LIBs) degradation estimations

In this chapter, a Nonlinear Auto Regressive Neural Network with Exogenous Input (NARX)-based state machine model is introduced for the capacity fade estimation. Different states of degradation are modeled using discrete states, such that capacity fade associated with each state is estimated using the NARX model. Transition from the first state to the last are defined to show the damage progression. These transitions are based on specific threshold conditions (design parameters) associated with the temperature (input) and actual capacity. As parameters of the model affect the estimation performance, developing optimal parameters are important. The parameters are optimized using the NSGA-II method.

Part of the contents, figures, and tables presented in this chapter are modified based on previous publications [DS23d].

5.1 Nonlinear Auto Regressive Neural Network with Exogenous Input model

The estimated capacity is calculated using different NARX models for each state, presented in Figure 5.1 (NARX 1, NARX 2, and NARX 3). Nevertheless, the structure of the model is the same for all states. Only the input and parameter values differ depending on the state. The NARX model describes the input-output mapping using a multi-layer percepton [CYLA15]. A usual NARX model also incorporates the time delays and feedback (target values or output, depending on the network type) in the input layer. The target values are the actual capacity values. In the model utilized an open loop network is used, which means only the input data variables (normalized battery's temperature) and target values (priori information of capacity) as feedback are used. The output is not fed back to the network (to the input layer), unlike a closed loop. The time delays of the input and target values are two time steps (1:2), which means the estimation starts at the third time step. The NARX network is trained to perform estimations of the capacity fade based on the past battery's temperature and target capacity values. The NARX input-output relationship can be describe as [Ven05]

$$\begin{aligned}\hat{y}(t) = f[i(t), i(t-1), ..., i(t-m_i), y(t-1), ..., \\ y(t-m_k)] + e(t),\end{aligned} \tag{5.1}$$

whereby $\hat{y}(t)$ and $y(t)$ are the estimated and target values, $i(t)$ is the input, m_i and m_k are the time delays of the input and target variables respectively, t is the time step, and $e(t)$ is the error between the estimated and target values.

The network consists of inputs, targets, a hidden layer of ten neurons, and an output layer. In the NARX model, the input layer consists of temperature and target capacity

as input neurons. The activation function considered in this model is the tansig function (Hyperbolic tangent sigmoid transfer function). The value of each hidden layer neuron is calculated based on the inputs, targets, and weights associated between the neurons. In addition, the bias values are also taken into consideration. The capacity estimations are calculated based on the values of hidden layer neurons and weights associated between the hidden layer neurons and output neuron.

5.2 NARX-based state machine model

In this section, two NARX-based state machine models are introduced, which differ in terms of the topology and transition conditions [DS23d].

5.2.1 NARX-based state machine approach I

As mentioned, behaviors of multi-state switching systems can be modeled using state machine models with discrete states. The states transition from one state to another or remain in the same state determined by the model's inputs and transition conditions [Gil62] [WM13]. The transition conditions are defined by designers. The model consists of three states, whereby each state represents different degradation levels/behaviors describing the aging [DS23d]. As input, only the normalized temperature of the battery is taken into account, due to the fact that the temperature change contributes to the capacity loss. The normalization is done with the $z - score$ normalization. The transition conditions are defined using the normalized temperature and the end-of-lifetime (EoL) capacity. The estimated loss of capacity associated with each state is modeled using a neural network model (NARX) [DS23d].

Based on Figure 5.1, when the current estimated state is state 1, it can transition to state 2 (EoL reached) if the transition conditions are met to estimate the next state and capacity fade. Otherwise, the model remains in state 1. Similarly, possible estimations when the current estimated state is state 2 is transitioning to state 3 or remaining in state 2. If state 3 is estimated, the model can only remain in state 3 for the next estimation.

As for the transition conditions, if the normalized temperature is higher than threshold $tr1$ and the capacity is less than or equals to the capacity at EoL (here: reaches 80 % of the nominal capacity), a state transition from state 1 to state 2 occurs (Figure 5.1). This indicates, the battery has reached EoL at state 2. On the other hand, a transition from state 2 to state 3 occurs when the temperature is higher than $tr2$ and the capacity is less than or equals to the capacity measured at the final time point. Once state 3 has been reached, the model can only only remain in the same state, as the final state has been reached. If conditions are not met, the model remains in the same state.

Optimization

The model parameters of the network are the weights and biases related to the neural network as well as the temperature thresholds associated with the transition conditions in the state machine, affecting the performance of the model. Thus, optimization of these parameters are important. The $tr1$, $tr2$, weights and biases (parameters of the model) are selected automatically using NSGA-II. In total, 124 parameters are optimized. These are given as unknown values to the algorithm initially. Based on different temperature ranges

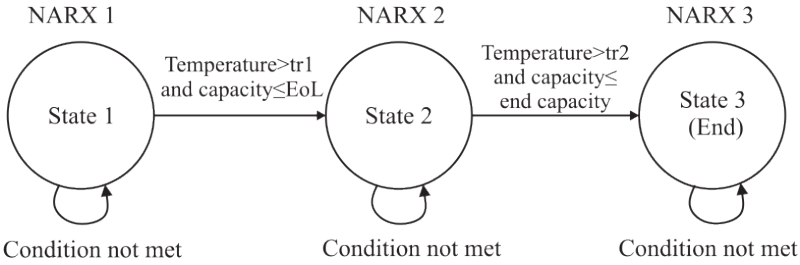

Figure 5.1: NARX-based state machine model I [DS23d]

for the specific states and capacities, the algorithm selects and optimizes the threshold values. The objective function of the optimization is given as

$$obj(t) = (|actual\ capacity(t) - estimated\ capacity(t)|). \tag{5.2}$$

The objective function is chosen with respect to minimizing the deviation between the actual and estimated discharge capacities. The generation and population size of NSGA-II is 200 and 90, respectively. To evaluate the overall performance of the model, the mean square error (MSE) and relative mean square error (RMSE) are often used, as done in [CCF+21].

5.2.2 NARX-based state machine model II

The topology of the NARX-based state machine model II differs from the previous model. In this model, the ambient temperature as well as the surface temperature of the battery are considered as inputs. Four damage states are defined in this model that transitions between each other, as shown in Figure 5.2. Here, state 0 is the initial state, state 1 and state 2 are the intermediate states, and state 3 is the EoL state. Similar to the previously introduced model, each state is modeled by NARX for the estimation of the capacity associated with each state. Four threshold values are defined , such that $tr1$ and $tr2$ are associated with the ambient temperature (T1) and $tr3$ and $tr4$ are associated with the surface temperature (T2). State 0 as a starting point transitions to state 1. Based on the transition conditions associated with both temperatures, state 1 can either transition to state 2, 3 or remain in the same state. The transition from state 1 to 2 occurs if the ambient temperature is less than or equal to $tr1$ and the capacity is greater than EoL capacity at that time point. State 1 transitions to state 3 if the T1 is greater than $tr1$ and the T2 is greater than $tr4$ and the capacity is less than or equal to EoL capacity. The model remains in state 1 if T1 is less than equal $tr1$, T2 is greater than $tr3$, and capacity is greater than EoL capacity. The model also remains in state 1 if T1 is greater than $tr1$, T2 is between $tr3$ and $tr4$, and the capacity is greater than EoL capacity. Once in state 2, it can either transition to state 1, 3 or remain in the same state based on the specified transition conditions. A switch from state 2 to 3 occurs if the EoL capacity is reached (less than or equal to EoL capacity). As state 2 is an intermediate state, it can also switch back to state 1 if T2 is greater than $tr4$ or T1 is greater than $tr4$ and T2 less than $tr4$. The model remains in state 2 if T1 is less than or equal to $tr2$, T2 is between $tr3$ and $tr4$, and the capacity is greater than EoL capacity. The model also remains in the same state if T1 is greater than $tr2$, T2 is less than or equal to $tr4$, and the capacity is greater than EoL capacity. State 3 is the end state (when EoL is reached) and does not transition to further

states. This topology and its conditions are a modification from [BMAS19]. In [BMAS19], a lifetime model is developed for the power management with optimal power split in hybrid electrical vehicles (HEVs). The model and topology introduced in [BMAS19], considers the temperature, charging/discharging current, and DoD as inputs for the estimation of degradation. Each state represent the SoH of the battery defined by the capacity of the battery at that particular state. The model is also used to predict the capacity in future based on specific lifetime points of the battery (at 20 %, 40 %, 60 %, and 80 %). For an example, when 20 % of the the battery's lifetime is known, the remaining 80 % of the lifetime is predicted (showing a drop in the reference and estimated capacities). A very similar technique for the estimation is considered in the proposed state machine model, by training different portions of data. The threshold values are defined using NSGA-II.

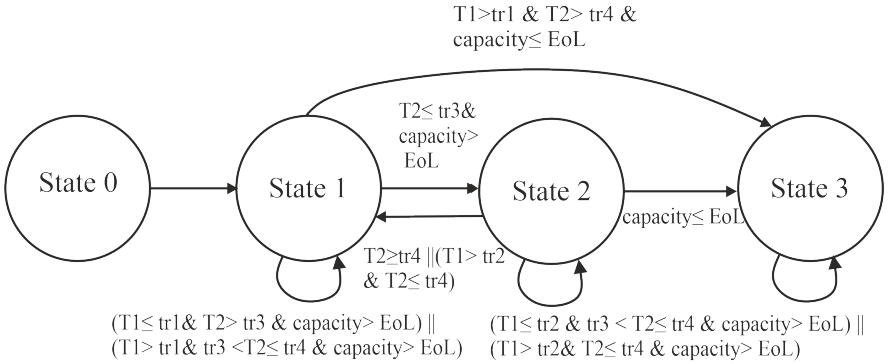

Figure 5.2: NARX-based state machine model II

Optimization

The same design parameters as the NARX-based state machine model I (thresholds associated with the temperatures, weights, and biases) are optimized here as well using NSGA-II. The total number of parameters optimized is 187. The similar objective function, generation, and population size for the NSGA-II is considered as well.

5.3 Summary

The application of the proposed state machine approach to develop a model for the capacity fade estimation is introduced in this chapter. Two models are developed by combining the NARX and state machine approach, such that each damage state is represented using states and the capacity is estimated using NARX (associated with each states). Temperature is considered as the model input, as it is one of main factors influencing the battery's capacity.

6 Experimental design, results, and validation

In this chapter, the experimental design and configurations for the driving behaviors/intentions recognition as well as for the capacity fade of LIBs estimations are explained. The different experiments designed are detailed in this chapter. In addition, data processing of the obtained data are explained. Next, the training and test procedure are detailed. Finally, the results based on the different models for the driving estimations are given. Similarly, the results based on two different NARX-state machine models are detailed. The data for the driving behaviors/intentions estimation are obtained using a driving simulator in the SRS simulator lab, while the data for the LIB degradation estimation are based on the battery data set from the Prognostics Center of Excellence Data Set Repository (PCoE) NASA and the battery test rig in SRS.

Parts of the contents, figures, and tables presented in this chapter are modified based on previous publications [DRS20] [DRS21] [DS22b] [DS23b] [DS22a] [DS23a] [DS23d].

6.1 Experimental design for the driving behavior recognition

In this thesis, a state machine-based model was introduced, with the extensions of the model. As mentioned previously, ENV and ET variables are considered as it is common to use these variables for driving behavior estimations. Most contributions consider ENV variables, while ET is only considered in a few contributions as these variables don't always improve the estimation performance with a few exceptions. As the evaluation on the effectiveness of these variables is the goal, hence, the need for the collection of the ET data.

A lane change can be defined differently based on the timing of different driving actions. In this thesis, two different experiments are considered based on the different definitions of a lane change: experiments A and B.

6.1.1 Laboratory setup

The driving data is collected using the driving simulator in the SRS chair (Figure 6.1). The software for the simulator is $SCANEeR^{TM}$. The driving simulator consist of 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 lane changing maneuvers. The scenarios in the experiment are based on a highway environment in two directions. The data is acquired at a frequency of 20 Hz. Using the $SCANEeR^{TM}$ software, the ENV features are collected. This software consist of several modules, such as acquisition, model-handler, traffic tools, and scenario modules which are programmed to record the ENV features during a drive. To explain the data acquiring process, the simulator has certain sensors such as radars, cameras, and GPS to collect the data when

simulation is running. For an example, the acquisition module is used to collect information about speed using the built-in sensors. In the real-world, these information can be collected using CAN-bus or cameras. The software is linked to Matlab simulink model (driving assistant system model), such that the information obtained is transferred to the simulink model which allows the data to be access easily. The simulation of the driving simulator process is shown in Figure 6.2.

Figure 6.1: Driving simulator [DWS18]

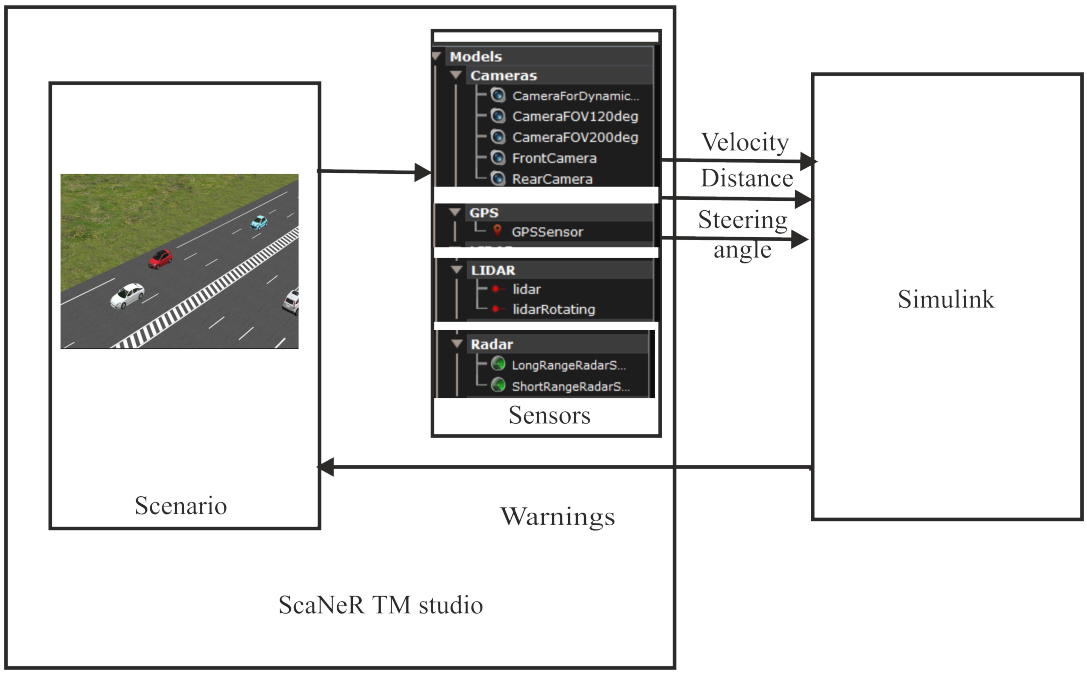

Figure 6.2: Process of acquiring data [Den20]

In addition to the driving simulator, an eye-tracker with the Facelab software (Figure 6.3) is used to collect data related to the eye movements of the drivers. The eye information collected include the saccadic information. Some of the features collected include gaze and head tracking features at 180 degrees. Examples of these features are the head position,

head rotation, head eyeball position, PERCLOS, blink frequency, and eye gaze. The eye-tracker is calibrated for each participant individually. Three main elements are calibrated: the cameras, head model, and world model. The calibration starts with calibrating the cameras. This process consists of holding a target in front of the participants head, taking multiple shots by moving the target around the participant's face at different orientations, and ensuring the target is accurately tracked. Next, the head model is calibrated. Here, reference points on the face and feature templates are chosen. The reference points are used to define the 3D head position structure and accurate eye position. It is also used to define mapping between multiple views. Feature templates are regions that are actually tracked by the eye-tracker, which are easily identifiable even when the head rotates in different positions. Finally, each object in the world model is calibrated. This is one of the most important part of the calibration for accurate tracking. The calibrated objects in the world model are:

- Main screen
- Right/left screen
- rear view mirror on the screen
- side view mirrors on the screen
- a control pad near the driver (only for autonomous driving)

Figure 6.3: Eye-tracker [DS22a]

The ET features are only used to test the effectiveness of the features as stated in section 4.3, for the rest only ENV features are considered. The simulator can also perform autonomous maneuvers however, in this thesis the drives are only limited to manual maneuvers.

6.1.2 Experiment A

In this experiment [DS18], each participant performed a 40 minute drive to obtain a training data set, another 10 minute drive for the test data. A traffic environment with other vehicles to simulate an actual driving environment is utilized. The scenario in this experiment is based on a highway in two directions. The drivers can perform different maneuvers while driving with other surrounding vehicles participating in the driving simulation. For 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 are obtained from participants ages between 25 to 30 years old,

all of which held a valid driving license. This experiment is used for the development of state machine model (section 4.1 [DRS20]), ANN-based state machine model (section 4.2.1 [DRS21]), HMM-based state machine model II (section 4.2.4 [DS23b]), and hyperparameter evaluations (section 4.4 [DS23a]). In the state machine and ANN-based state machine models, data from three participants were considered. For the hyperparameter evaluations, data from seven participants were considered.

In experiment A, the time a lane change occurs is defined as t_{lane} and the time of last significant change in the steering wheel angle is t_{angle}. Hence, the time interval between the t_{angle} (start of lane change) and t_{lane} is defined as the lane changing duration [DS18].

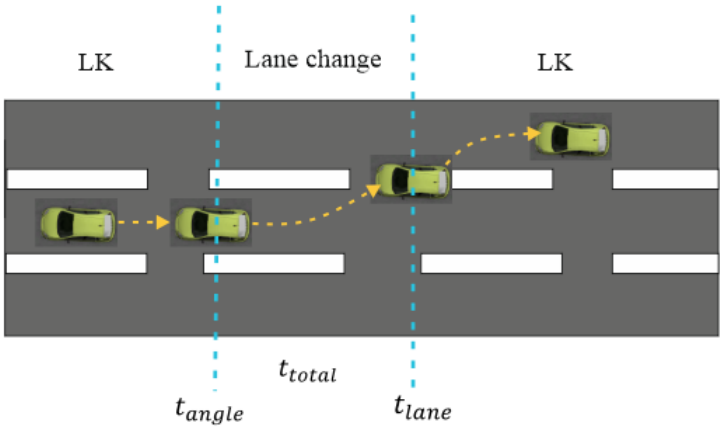

Figure 6.4: Driving scenario at highway (experiment I)

6.1.3 Experiment B

Experiment B [DWS18] [DHBS20] is employed for the evaluations of features (section 4.3 [DS22a]) and HMM-based state machine model I (section 4.2.6 [DS22b]). For the evaluation of features, each participant performed a 30 minute drive to generate training data, another 10 minute drive for the test data. For the HMM-based state machine model, each participant performed a 25 minute drive. 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. The scenario is based on a highway in two directions, whereby the driver can perform different maneuvers. The data are obtained from participants with valid driving license between the ages of 25 to 30 years old. Data from five participants are used for the evaluations of the features. In the HMM-based state machine model, data from nine participants are considered. In experiment B, the interval between the time the indicator is activated $t_{indicator}$ and t_{lane} is defined as the lane changing duration t_{total} [DWS18] [DHBS20] (Figure 6.5) .

6.1.4 Data processing

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

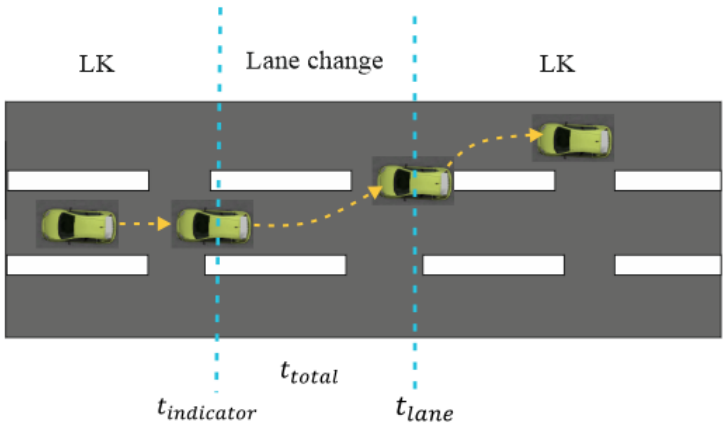

Figure 6.5: Driving scenario at highway (experiment II)

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 the experiments, the t_{total} is between 2 to 3 seconds (s), whereby the driver activates the indicator 2 to 3 s before performing a lane change. Different preset t_{total} values of 2 s, 2.5 s, and 3 s are therefore tested for labeling behaviors to evaluate the impact of t_{total} on the lane change recognition abilities. This is tested with models that utilized the data of experiment B (feature evaluations and HMM-based model I only). For the feature evaluations, the lane changing labels based on preset values of 2 s, 2.5 s, and 3 s are tested with the model. The labels based on the preset value that developed estimations closest to the actual behaviors (labels) are selected to the define the behaviors for the proposed model. Similarly, the labels based on the preset values are also tested using a single HMM developed in [DWS18] for the HMM-based state machine model I to define the accurate behaviors.

Inaccurate data are removed as part of the labeling processing for all experiments. For 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 [DWS18].

A summary of the experimental designs is presented in Table 6.1.

6.2 Experimental design for the LIB degradation estimation

For the degradation of LIBs, battery data sets based on a charging and discharging process from the PCoE NASA and SRS battery test rig are considered.

6.2.1 NASA battery data

The PCoE is mainly involved with prognostic development research, focusing on the gaps within various areas such as aeronautics, battery degradation, etc. The center employs lab

Table 6.1: Summary of experiments

Experiment	Participants	Experimental duration	Training and test	Data processing	Model application
A	3-7	Training: 40 minutes Test: 10 minutes	Training: 40 minutes Test: 10 minutes	$t_{total} = t_{lane} \text{-} t_{angle}$	State machine model, ANN-based state machine model, (including hyperparameter otpimization), HMM-based state machine model II
B	5-9	Training: 30 minutes, Test: 10 minutes (Feat. eval.), 25 minutes (HMM-state machine)	Training: 30 minutes, Test: 10 minutes (Feat. eval.) 70:30 (HMM-state machine)	$t_{total} = t_{lane} \text{-} t_{indicator}$	Evalaution of features, HHM-based state machine model I

facilities for testing, measuring, diagnosis, and prognosis of the health management. The data sets considered in this thesis are the charging and discharging at different temperatures set [SG07] and the randomized battery usage set [BKD14].

Six data sets based on three different experiments are defined as experiment I [SG07], experiment II [BKD14], and experiment III [BKD14]. Experiment I only simulates a constant current (CC)-constant voltage (CV) charging and discharging process, which is non-dynamic. On the other hand, experiment II and III simulate a dynamic charging and discharging operation using a random walk (RW) process (includes CC-CV process). To evaluate the health of the battery, the capacity based on only the discharge operation for all experiments are taken into account. The CC-CV process is shown in Figure 6.6 for the charging operation. To explain this process, the batteries are first charged at CC until a specific maximum voltage is reached. Then, the charging continues at a CV by maintaining the maximum voltage with the current decreasing until a specific cut off threshold rate. As for the discharging operation, it begins with a CC operation until the voltage decreases to a specific point, then the CV operation begins. For all experiments two data sets are considered, whereby one is used for training and the other for test to generate capacity fade estimation. These data sets are utilized for the evaluation of the NARX-based state machine model I.

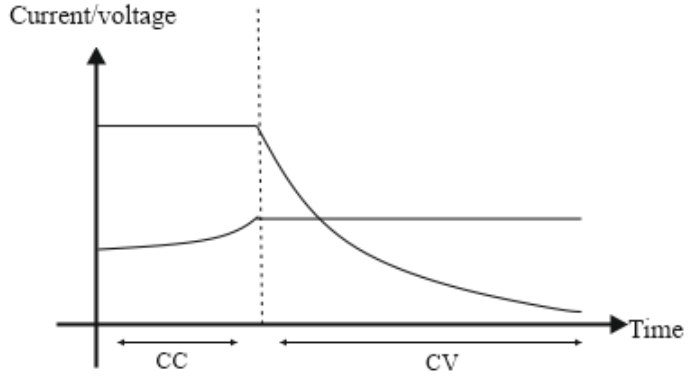

Figure 6.6: CC-CV process (charging process) [Mar15]

Experiment I

Experiment I utilizes two battery data sets (B0005 and B0006) [SG07]. The batteries are first charged using the CC mode at 1.5 A. When the battery voltage reaches 4.2 V, the charging switches to the CV mode until the current falls to 20 mA. Discharging process begins with the CC mode at 2 A until the voltage decreases to 2.7 V and 2.5 V for B0005 and B0006, respectively [SG07]. The nominal capacity of the batteries are 2 A. The EoL is reached when the capacity reaches 70 % of the nominal capacity (1.44 A). The charging and discharging process do not simulate dynamical load profiles. The ambient temperature is room temperature. A total of 168 discharge cycles are performed and are considered for the estimation for the capacity degradation estimation.

Experiment II

Experiment II is based on a RW process. Two LIB data sets (RW 9 and RW 10) from the experiment are utilized in this contribution [BKD14]. The RW process is used to charge and discharge the batteries between -4.5 A and 4.5 A. After 1500 RW step cycles, a reference charge and discharge operation is done to evaluate the capacities. The EoL is defined when the capacity reaches 1.68 A for both batteries (80 % of the nominal capacity, as defined by battery manufacturers). The capacities are calculated based on the current and relative time. The number of discharge cycles for RW 9 and RW 10 are 80 and 77, respectively.

Experiment III

The batteries in Experiment III also undergo the RW operation. Two battery data sets (RW 1 and RW 7) are used [BKD14]. The RW is performed by charging the batteries for a randomly selected period between 0.5 and 3 hours. The batteries are then discharged using a randomly selected current between 0.5 A and 3 A. Following 50 RW cycles, a reference operation is done to evaluate the capacities. The EoL capacities are 1.60 A and 1.59 A (80 % of the nominal capacity) for RW 1 and RW 7, respectively. The capacities are calculated based on the current and relative time. Here, 48 discharge cycles were performed for RW 1, while 49 discharge cycles for RW 7.

For experiment I, the normalized batteries' temperature during the discharge operation is considered as the model's input for estimations. For experiments II and III, the normalized batteries' temperature during the reference discharge operation (after RW) is considered as input. The battery temperatures are in degree Celsius ($°C$). Through out the discharge phase, capacity decrease is in observed for all experiments. In Table 6.2, the data sets used for training and test as well as the EoL capacities for the different experiments are given.

Table 6.2: Summary of the training and test data

Training data sets	**Test data sets**
B0005 (EoL: 1.44 Ah)	B0006 (EoL: 1.44 Ah)
RW10 (EoL: 1.68 Ah)	RW9 (EoL: 1.68 Ah)
RW7 (EoL: 1.59 Ah)	RW1 (EoL: 1.60 Ah)

6.2.2 SRS battery test rig data

The battery test rig (Figure 6.7) at the SRS chair is also used to perform charging and discharging operations of a 18650 type cylindrical LFP cell with 1.5 Ah nominal capacity. The aim is to collect both ambient and cell temperatures to study their effects on battery aging related to capacity fade. The experiment conducted simulates realistic dynamic electric vehicle (EV) load profiles using the LFP cell. The test rig components are given in Table 6.3. The data obtained from the test rig is utilized for the evaluation of the NARX-based state machine II.

A CC-CV charging and discharging method is performed in this thesis. The voltage and current regulation are used to simulate the current profiles. During the charging and discharging operation different variables such as temperature, current, voltage, SoC are

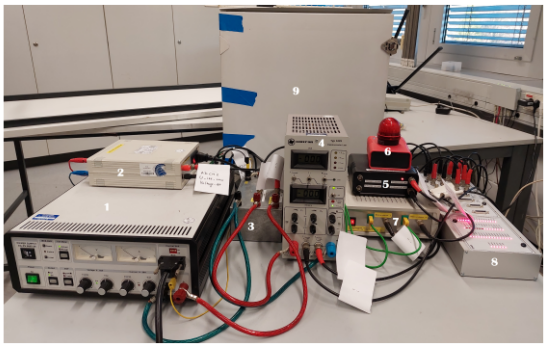

Figure 6.7: Battery test rig [Tha22]

Table 6.3: Components of the battery test right

Number	Components
1	Laboratory power supply unit EA-PS 9065-20
2	Isolation amplifier for terminal voltage
3	Relay box
4	Electronic load STATRON Type 3229
5	Isolation amplifier for discharge current
6	Signal lamp
7	Transmitter for temperature measurements
8	Input output module DS1104
9	Refrigerator to maintain the temperature

recorded. A dSPACE real time computer module DS1104 (through the use of control desk 7.2) is used to control the sequence of the operation and enables the communication between the test rig and the simulated cell model designed in simulink. This communication enables the real time recording of terminal voltage, charging current, discharging current, ambient temperature, and surface temperature. The simulink model is also used to affect the parameters and current profiles to simulate EV driving behaviors. The laboratory power supply unit EA-PS 9065-20 is used to regulate the current and voltage within ranges of 0 A to 20 A and 0 V to 65 V. The electronic load can regulate current flowing through between 7 mA and 50 mA and load voltage of 2.5 V to 80 V. The red signal lamp flashes if there is an error in the measurements. The temperature of the batteries and the ambient temperatures are measured using a thermocouples. The operating conditions for the cell and test rig in the experiment are given Table 6.4.

The power supply current regulation is from 0 A to 7.5 A, while the voltage is up to 3.6 V. As part of the CC-CV operation, the cell is charged at CC of 1.5 A until the maximum voltage of 3.6 V is reached (EoC). Then, the cell continues charging at CV of 3.6 V, while the current decreases to 0.3 A (as a cut off threshold rate). During the discharge phase, a fully charged cell is discharged until the voltage reached 2.5 V (EoD). The ambient temperature range is within 7 $°C$-12 $°C$, while the range for the battery's surface temperature is 12 $°C$-19 $°C$.

During charging and discharging operation, three parameter tests are performed:

- Impulse test: to determine the internal resistance

Table 6.4: Operating conditions

Operating variables	Values/Ranges
Ambient temperature [$°C$]	7-12
Surface temperature [$°C$]	12-19
Voltage regulation [V]	0-3.6
Current regulation [A]	0-7.5
Nominal charge current [A]	1.24
Nominal discharge current [A]	1.5
Nominal capacity [Ah]	1.5
End-of-charge voltage (EoC) [V]	3.6
End-of-discharge voltage (EoD) [V]	2.5

- Terminal voltage test: to determine the open circuit voltage (OCV) characteristics as well as the parametization of cell and observer design
- Capacity test: to determine the capacity (for battery's health evaluation)

These parameter tests are performed in regular spans of cycles given in Table 6.5. The parameter tests are performed at the beginning and at specific intervals, such as at cycle 100, 175, etc. to determine internal resistance, OCV characteristics, and capacity. As the health of the battery is evaluated using the battery's capacity, the capacity test is of importance to observe the capacity degradation based on the discharge process only.

Table 6.5: Frequency of parameters [Hol22]

Intervals	Cycles
1	0
2	25
3	100
4	175
5	250
6	375
7	400

Impulse test

Based on Figure 6.8, the test starts with a 600 s pause with the open circuit terminal voltage U_1 developed during this time. The voltage drops to U_2 when the circuit is closed and drops further to U_3. When the circuit is opened again, the voltage rises to U_4 and further to U_5. Using the terminal voltages, the internal resistance, R_i can be calculated as

$$R_i = U_1 - U_2/I_{diss}. \tag{6.1}$$

Terminal voltage

Here, a fully charged cell is discharged at a constant discharge rate in steps of 5 % discharge SOC until 0 % SOC is reached. This step-by-step discharge is needed to determined the

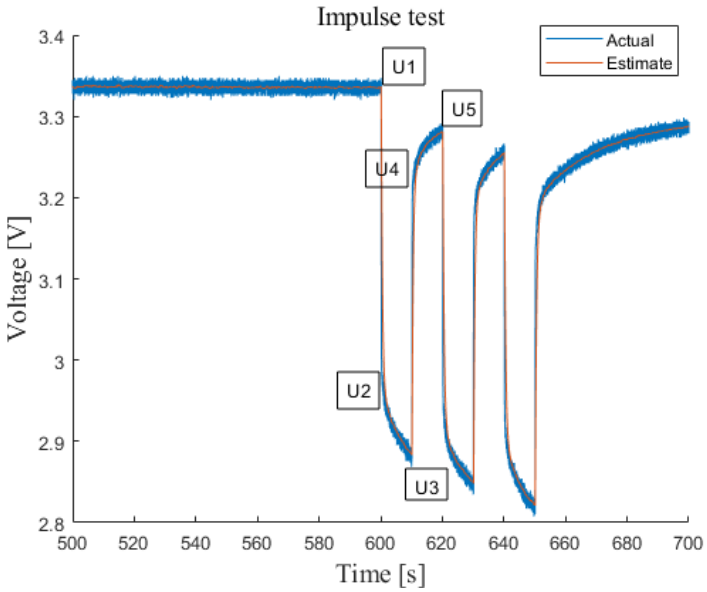

Figure 6.8: Impulse test process

OCV characteristics for the respective SOC. The OCV characteristics are important for the cell parmaterization and SOC observer design. The process is shown in Figure 6.9.

Capacity test

As only the discharge cycles are considered for capacity evaluations, the capacity is calculated as

$$C_i = I_{dis} * t_{dis}, \tag{6.2}$$

whereby C_i is capacity at specific time point i, I_{dis} is the discharge current, and t_{dis} is the time to discharge.

A total of 3550 discharge cycles are performed. The EoL is reached at 70 % of the nominal capacity. Hence, the EoL capacity is 0.88 achieved at cycle 2725 (Figure 6.11).

6.3 Training and test procedures

The training and test processes are based on the input variables and the actual lane changing states for the estimation of lane changing behaviors/intentions. For the capacity fade estimation, the training and test are based on the input variables and the actual capacity fade of the battery.

Most of the models for lane changing behavior/intention recognition are trained using one data from a diver and tested using another data from the same driver. Therefore, generally a driver has to perform two drives individually. The only exception is for the data sets used in the HMM-based state machine I development [DS22b]. Here, each driver only performed a single 25 minute drive. The data are split into 70:30 for training and test as stated previously.

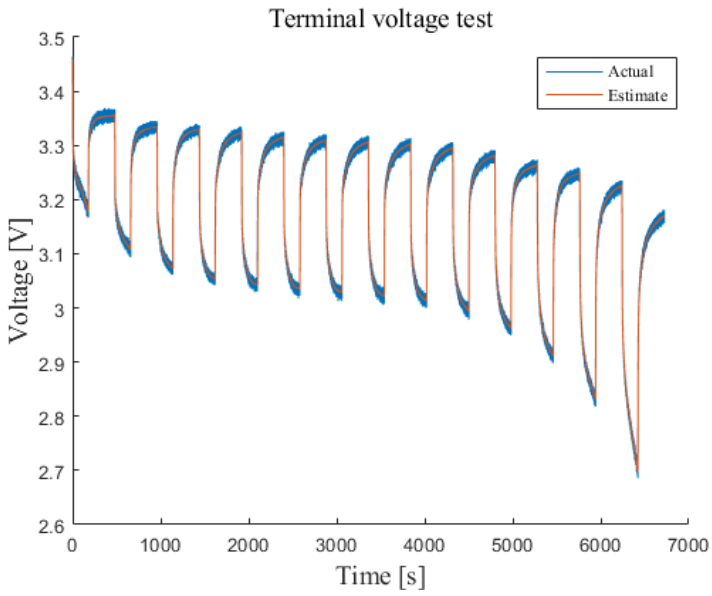

Figure 6.9: Terminal voltage

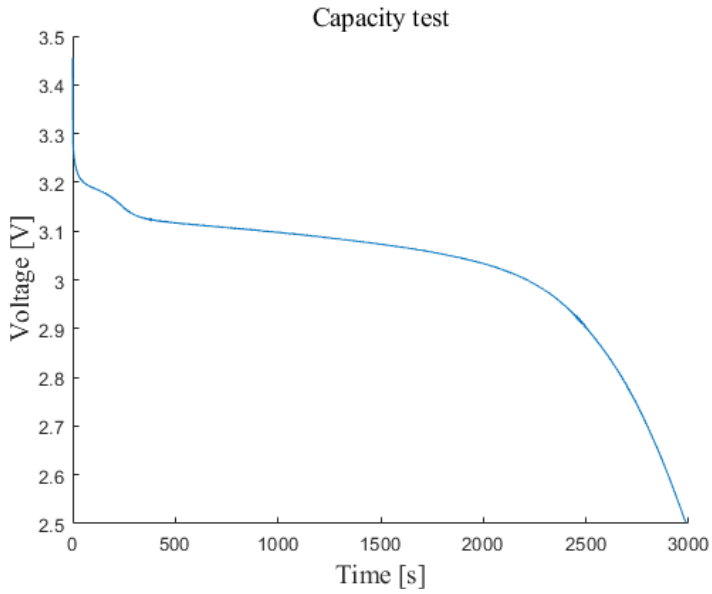

Figure 6.10: Capacity test

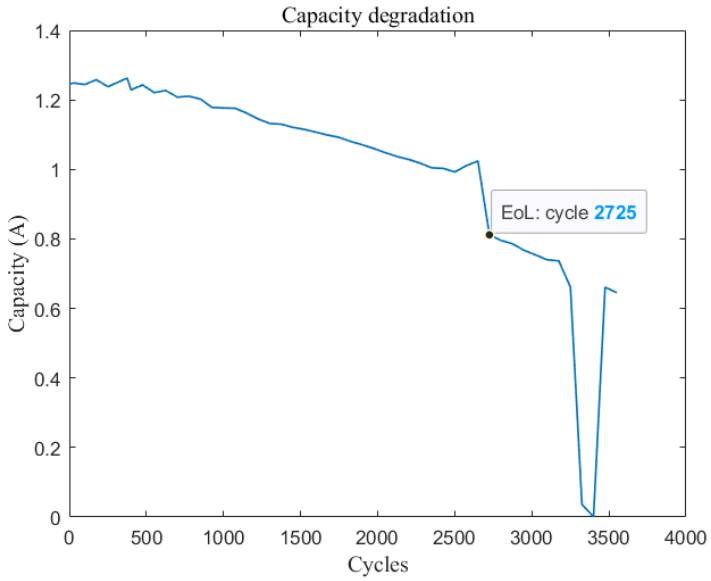

Figure 6.11: Capacity degradation

In the evaluations of ET and ENV features (section 4.3) [DS22a], a 10 fold corss-validation is first performed using two models (model I (ENV) and model II (ENV and ET), introduced in section 4.3) with the training data to evaluate the feature sets that developed the closest estimation to the actual intentions [ND15]. Here, the training data from a participant are divided into ten sub data sets, whereby nine will be combined for training and the remaining one will be used for validation. This process is repeated by ensuring each sub data set is used once for validation [ND15]. The model's performance based on a participant's data is given by the average performance of the ten sub data sets. The feature set based on the model that generated the best performance in the 10 fold corss-validation is then used as inputs for the training and test process.

As stated, certain models are first evaluated using lane changing labels based on specific preset values of the lane change duration to study the impact the duration on the estimation, subsequently defining behaviors accurately. For the evaluations of the different preset values to define lane changing behaviors, the similar training and test procedure applied for that specific model is utilized.

For the estimations of the capacity using the NARX-basesd state machine model I, one specific data is used for training, while the other battery's data is used for testing the model in each experiment. For the NARX-based state machine model II, capacity degradation is monitored by estimating the capacity starting at different time points/cycles [BMAS19]. The same operating conditions are considered at these different time points. Different portions are trained, while the remaining parts are used for test (the estimation of the capacity), starting from the end of training data set. The model is trained with 30 %, 50 %, and 70 % of the entire data, while the estimations begin from the time point the training data stops (test) to evaluate the effectiveness of the model. For an example, when 30 % of the data is trained, the rest of the 70 % are used as the test data for estimations.

6.3.1 Training

Driving behavior

The models are trained as follows:

1. Input variables and labels (actual driving behaviors/ intentions) are given into the models.
2. Using NSGA-II, a set of model parameters is generated (prefilter thresholds, weights, bias).
    - State machine : based on the optimal thresholds generated, the state machine model either switches states or remains in the same state to estimate the lane changing behavior.
    - ANN-based state machine: based on the optimal weights and biases generated, the model generates the predicted probabilities for the three different behaviors, in order to generate the estimations of ANN. The same holds for the evaluations of the ENV and ET features and hyperparameter optimization using this model.
    - HMM-based state machine I: the prefilter thresholds are determined by NSGA-II to define the observation sequence. Using these observation sequences, the behaviors are estimated. Using the TTC features as an example, two 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 three segments (as the threshold values divide the feature into three segments). Based on the selected segment of the different TTC variables, the observation sequences are calculated (as input sequences).
    - HMM-based state machine II: the prefilter thresholds of the input variables for the sub-HMMs are optimized using NSGA-II to develop the observation sequences.
    - Using the observation sequences, the probability for each of the sub-HMMs are calculated. The probabilities are fused using the optimal weights to develop the final probability for each state. The hidden state with the highest probability is selected as the final HMM estimation.
    - Based on the estimations of the ANN or HMM, the proposed model's estimations are determined using the state machine topology.
3. 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.
4. The process is repeated until convergence and the optimal model is obtained.

Capacity degradation of batteries

The models are trained as follows:

1. The temperatures (ambient, surface) and capacity values of a battery data set are given as input and target values respectively, for training using NSGA-II to develop optimal parameters (weights, bias, and temperature thresholds).

2. Using the optimized temperature thresholds and capacity at each time point, the state machine either switches from one state to another or remains in the same sate.

3. Based on the selected state, the estimated capacity at that time point is calculated using the NARX model with the optimized weight and bias values.

4. By calculating the deviation between the estimated and actual capacity, the objective function is evaluated.

6.3.2 Test

The test is performed as follows :

1. The trained model is applied to the test data set for state estimations and capacity estimation.

2. Driving behaviors: the actual driving states (from test data set) and estimated driving states are compared using ACC, DR, and FAR.

3. Batteries: the estimated capacity is calculated using the NARX based on the estimated state. Then, the actual and estimated capacity are compared using MSE and RMSE.

6.4 Results of the driving behavior recognition

Here, the results based on the different state machine models, feature evaluations, and hyperparameter evaluations are presented here.

6.4.1 State machine-based approach

Here, results based on three data sets from three participants are shown [DRS20]. As mentioned previously, the training data set is based on a 40 minute drive, while the test data set is based on a 10 minute drive. The results in Figures 6.12 to 6.14 are based on test data sets 1,2, and 3. In the ordinate, the y-axis represents the three different states, while the x-axis represents time, in seconds (s). The blue line represents the estimated driving behavior (or calculated states) and the red dotted line represents driving behavior from the driving simulator.

All figures show different lane changing behaviors. The driver makes the choice of staying on the same lane or making a lane change by assessing the traffic situation. Besides lane changing behaviors, input variables changes throughout the drive for each driver. In all figures, a close fit between the actual and measured driving behaviors is observed.

The ACC, DR, and FAR values for each test data based on the corresponding trained models are shown in Tables 6.6 to 6.8. In addition, a trained model is also tested using data sets from other drivers to show the generalization and transferability.

The trained model using training data set 1 is tested with test data set 1, data set 2 (a combination of training and test data set 2), and data set 3 (a combination of training and test data set 3) in Table 6.6. Table 6.7 shows the performance when the trained model using training data set 2 is tested with test data set 2, data set 1 (a combination of training and test data set 1), and data set 3 (a combination of training and test data set 3). In Table 6.8, it shows when the trained model using data set 3, the model is tested

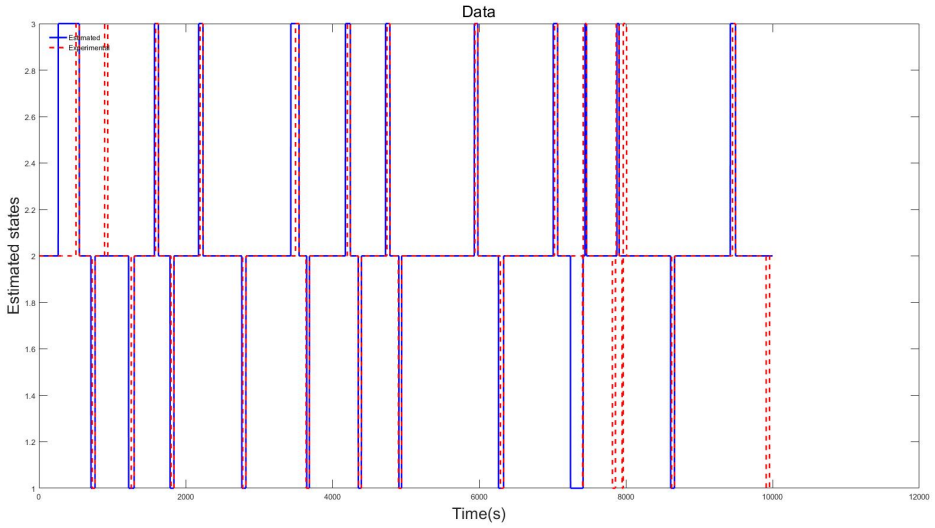

Figure 6.12: Calculated and measured states (test data set 1) [DRS20]

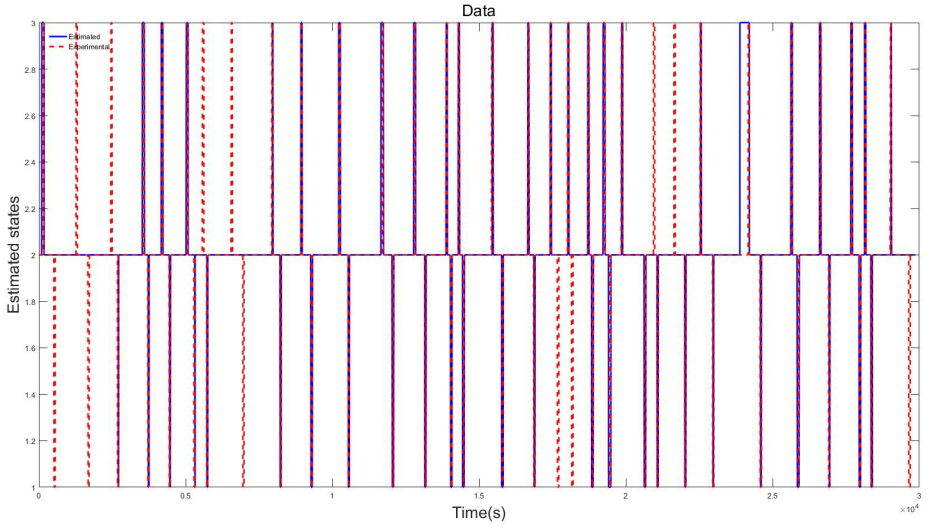

Figure 6.13: Calculated and measured states (test data set 2) [DRS20]

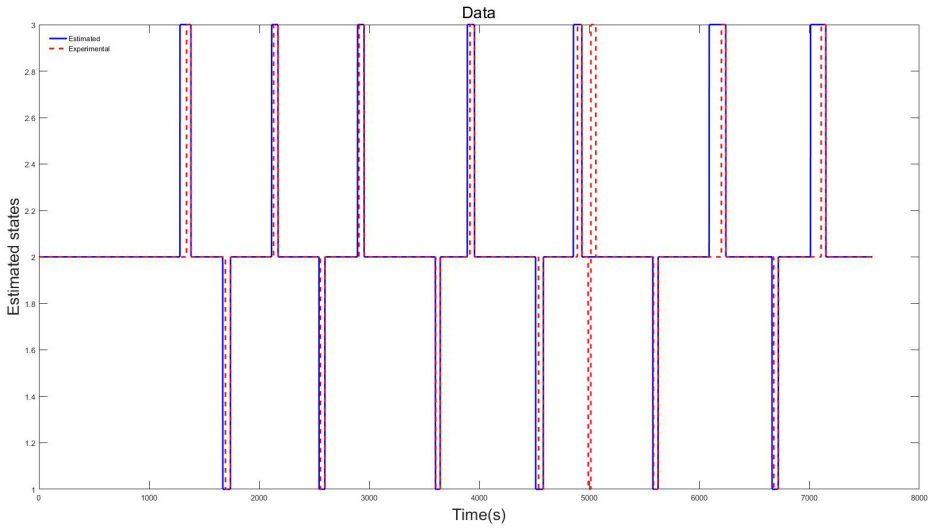

Figure 6.14: Calculated and measured states (test data set 3) [DRS20]

with test data set 3, data set 1 (a combination of training and test data set 1), and data set 2 (a combination of training and test data set 2).

Table 6.6: Recognition results (training data set 1, test with data sets 1-3) [DRS20]

Metrics	**Test data set 1 (%)**	**Data set 2 (%)**	**Data set 3 (%)**
$ACC_{overall}$	92.90	95.30	91.69
ACC_{right}	96.02	97.70	98.61
DR_{right}	88.94	73.27	87.07
FAR_{right}	3.32	1.41	0.82
ACC_{keep}	93.11	95.37	92.08
DR_{keep}	93.32	97.37	91.99
FAR_{keep}	8.88	22.31	7.11
ACC_{left}	96.66	97.53	93.23
DR_{left}	88.76	80.75	95.92
FAR_{left}	2.94	1.58	6.92

The results generally show a good fit between the actual and estimated behaviors for the data sets used. This method produces high ACC, DR, and low FAR for most of the states. For an example, in Table 6.6, the overall ACC for test data set 1 is 92.90 %, with the highest left maneuver ACC of 96.66 % and low false alarm rates for all maneuvers. The overall ACC for test data set 2 (Table 6.7) is the highest in comparison to other data sets with an ACC of 95.77 %. A high ACC of the left maneuver at 98.08 %, which is not only the highest within the test data set 2, but also the highest value when compared with other test data sets. However, the FAR for keep in test data set 2 is higher than the rest of the maneuvers within this data set. The same can be said regarding the FAR for keep in test data set 1 and 3. On the other hand, the FAR for right and left for test data set 2 is low at 1.41 % and 1.42 % respectively. The detection rates are also acceptable for the different maneuvers in all the data sets, with values larger than 73 %. The results from this

Table 6.7: Recognition results (training data set 2, test with data sets 1-3) [DRS20]

Metrics	**Data set 1 (%)**	**Test data set 2 (%)**	**Data set 3 (%)**
$ACC_{overall}$	92.89	95.77	91.69
ACC_{right}	96.22	97.48	98.56
DR_{right}	86.93	79.31	86.01
FAR_{right}	3.32	1.41	0.82
ACC_{keep}	93.05	95.97	91.93
DR_{keep}	93.40	97.22	91.74
FAR_{keep}	10.23	13.74	6.30
ACC_{left}	96.45	98.08	92.88
DR_{left}	88.66	89.78	95.92
FAR_{left}	3.15	1.42	7.28

Table 6.8: Recognition results (training data set 3, test with data sets 1-3) [DRS20]

Metrics	**Data set 1 (%)**	**Data set 2 (%)**	**Test data set 3 (%)**
$ACC_{overall}$	92.69	95.30	93.35
ACC_{right}	96.22	97.70	96.22
DR_{right}	86.93	73.27	91.55
FAR_{right}	3.32	0.98	1.12
ACC_{keep}	92.97	95.37	93.35
DR_{keep}	93.40	97.37	93.74
FAR_{keep}	12.11	22.31	11.04
ACC_{left}	96.30	97.53	94.75
DR_{left}	85.28	80.75	86.80
FAR_{left}	3.14	1.58	4.88

contribution generally show close resemblance to the results from previous works [DS18] [DWS18]. The newly introduced approach is therefore easier to understand and from the machine learning perspective interpretable. Furthermore, when the trained models from a specific driver are tested with data from other drivers, high ACC, DR, and low FAR are generally achieved proving the model's generability. The values show close resemblance to the corresponding test data of the specific driver.

To verify the effectiveness of this method, the results developed in this paper are compared with results developed from using other techniques. In [DS18] which uses a Fuzzy logic (FL)-based HMM approach on same data set as this work, average values of ACC, DR, and (1-FAR) are higher 80 % are achieved. Here, the values of ACC, DR and (1-FAR) from the data sets are also generally higher than 80 %, with some exceptions, for an example when training data set 2, the DR for right in test data set 2 is 79.31 %. In contrast to [DS18], the main advantage of the approach introduced in this contribution is that the approach is interpretable, which is not the case for all the approaches applied in [DS18]. The average performance of the three drivers based on the method developed in [DS18], SVM (optimized parameters), and the proposed approach is given in Table 6.9 for comparisons. The results show that the proposed state machine approach outperforms in most metrics the other approaches when tested using the same three data sets.

Table 6.9: Performance comparison between different approaches

Metrics	**Proposed approach (%)**	**FL-HMM(%)**	**SVM(%)**
$ACC_{overall}$	94.06	87.19	90.79
ACC_{right}	96.57	94.12	96.87
DR_{right}	86.62	88.77	74.42
FAR_{right}	1.95	5.65	2.01
ACC_{keep}	94.14	87.21	90.81
DR_{keep}	94.76	87.27	93.67
FAR_{keep}	11.22	1.36	36.76
ACC_{left}	96.50	93.07	93.91
DR_{left}	88.44	83.65	53.31
FAR_{left}	3.08	6.44	4.09

6.4.2 ANN-based state machine approach

Two models are evaluated here: approach I (one ANN) and II (three ANNs) [DRS21]. Only evaluations of ACC, DR, and FAR based on approach II using different data sets are presented here as approach II generated a better performance than approach I. Nevertheless, the results based on approach I and II were close. In Figures 6.15 to 6.17, 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.

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 was also achieved when approach I is applied.

In Tables 6.10 to 6.12 the ACC, DR, and FAR results of different test data sets based on the model using approach II are shown. When a model is trained using a data set

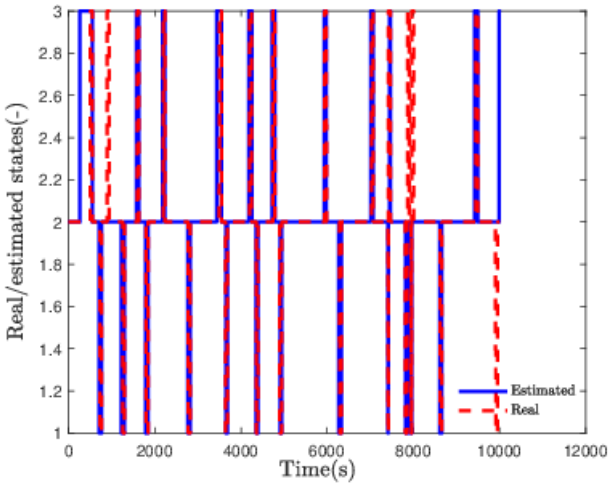

Figure 6.15: Actual and estimated states (test data 1) [DRS21]

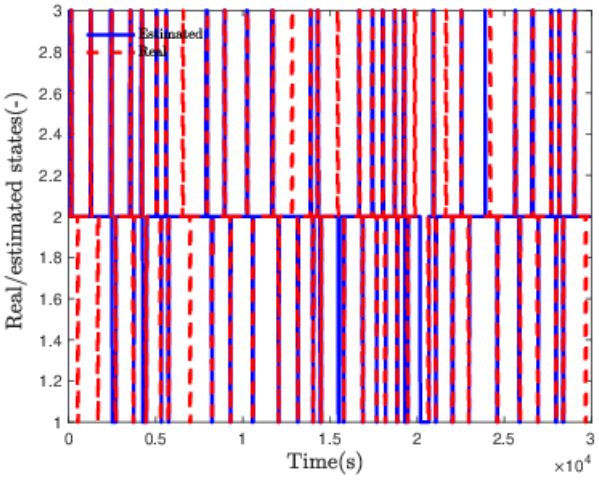

Figure 6.16: Actual and estimated states (test data 2) [DRS21]

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.

In Table 6.10, the ACC, DR, and FAR for test data set 1, whole data set 2 (combined training and test data set 2), and whole data set 3 (combined training and test data set 3) when the model is trained using data set 1 are shown. Based on Table 6.10, 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 different test data sets, high ACC, DR, and low FAR are achieved for the different states.

In Table 6.11, ACC, DR, and FAR for test data set 2, whole data set 1, and whole data set 3 when using trained data set 2 are presented. Here, the overall accuracy for test

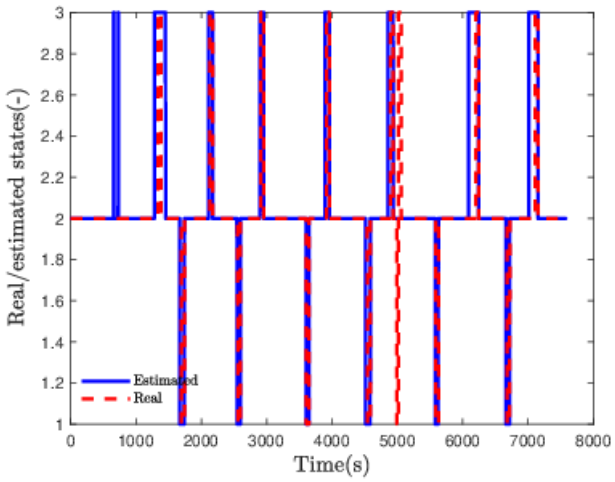

Figure 6.17: Actual and estimated states (test data 3) [DRS21]

Table 6.10: Evaluation of metrics (data set 1) [DRS21]

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

Table 6.11: Evaluation of metrics (data set 2) [DRS21]

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

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 %.

Table 6.12: Evaluation of metrics (data set 3) [DRS21]

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

Next, in Table 6.12 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 are given. 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 (optimized model parameters with NGSA-II and one hidden layer with ten neurons) are presented using the same three test data sets in Table 6.13. Based on mean performances shown in Table 6.13, 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).

6.4.3 HMM-based state machine approach I

Data sets from nine drivers are considered for the application of the proposed model [DS22b]. The first goal is to study the effects of the lane changing duration (t_{change}) on lane changing recognition abilities by testing different preset lane changing duration (2 s, 2.5 s, and 3 s) for labeling a lane change. To do so, the HMM developed in [DWS18] is

Table 6.13: Comparisons between different approaches [DRS21]

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

used to estimate the lane changing behaviors based on the variables of models I (distances, velocity, lane) and II (TTC). In Table 6.14, the average performance values based on nine 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 best values. Hence, t_{change} of 2.5 s is used to define a lane changing behavior for the evaluations of the proposed HMM-based state machine approach I.

Table 6.14: Average performance based on different lane changing duration [DS22b]

States	Metrics	Model I [%]			Model II [%]		
Duration [s]		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

The average values of ACC, DR, and FAR based on the test data for both proposed models are given in Table 6.15 and Figure 6.18. Based on the results in Table 6.15, both models generate high ACC, DR, and low FAR, with the exception of FAR_{keep}. The obtained results show 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. As an example, the actual and estimated lane changing states corresponding to test data set of driver 2 from both models are plotted in Figure 6.19 and Figure 6.20. 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. The data are recorded every 0.05 s. The figures show the proximity between the actual and estimated states.

Table 6.15: Average performance based on the test data [DS22b]

States	Metrics	Model I [%]	Model II [%]
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

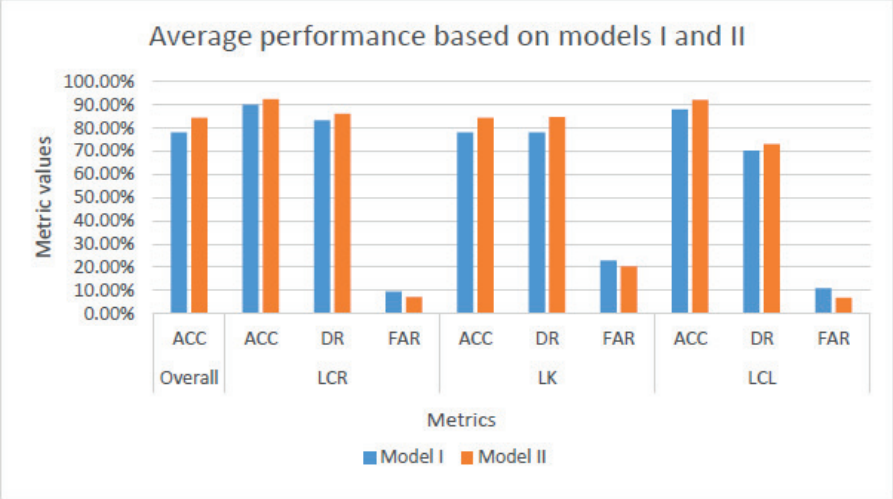

Figure 6.18: Test results of models I and II [DS22b]

A generability test is performed as well in which the average performances are presented in Table 6.16 and Figure 6.21 for models I and II. In the generability 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 6.15. A trained model based on specific driver is tested with other drivers and the average based on each trained model is computed. The generability test results in Table 6.16 show that model II also outperforms model I except for ACC_{right}, FAR_{right}, and FAR_{keep}. However, the metric values based on Table 6.15 are higher. The values in Table 6.16 are also close to the values obtained in Table 6.15, which establishes the generalibility of the proposed models. Using drivers 1 and 2 as examples for the generability test, the estimated and actual lane changing states based on test data of driver 2 (using trained model of driver 1) are plotted in Figure 6.22 and Figure 6.23.

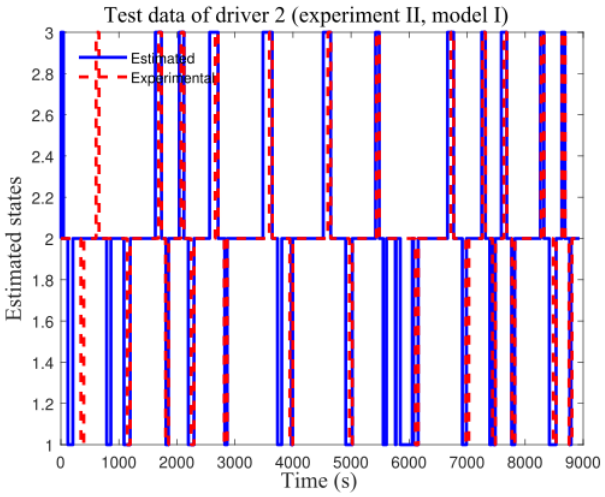

Figure 6.19: Test data of driver 2 (model I) [DS22b]

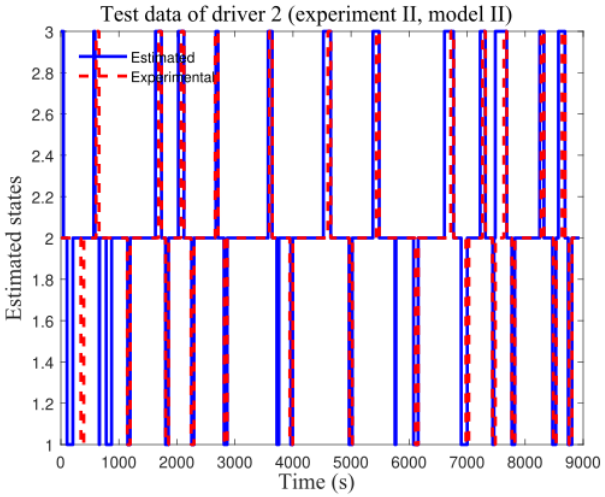

Figure 6.20: Test data of driver 2 (model II) [DS22b]

Table 6.16: Average performance based on generability test [DS22b]

States	**Metrics**	**Model I** [%]	**Model II** [%]
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

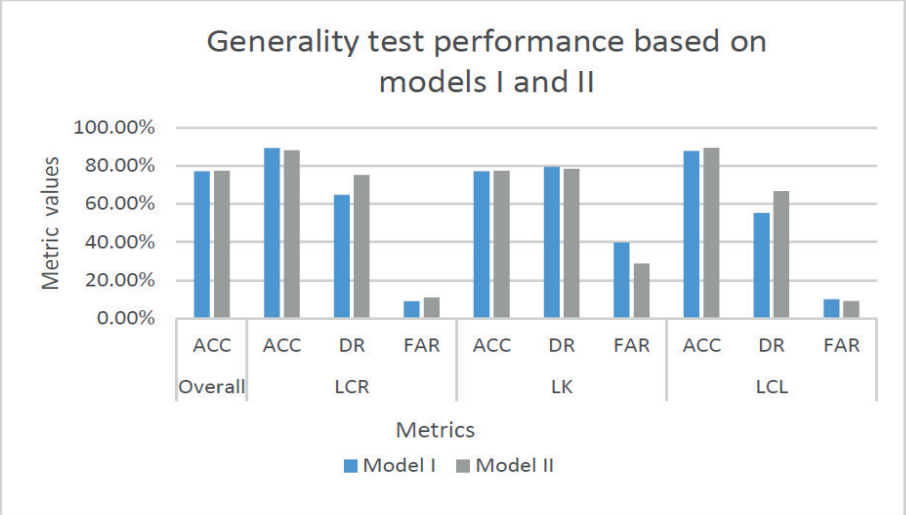

Figure 6.21: Generability test results of models I and II [DS22b]

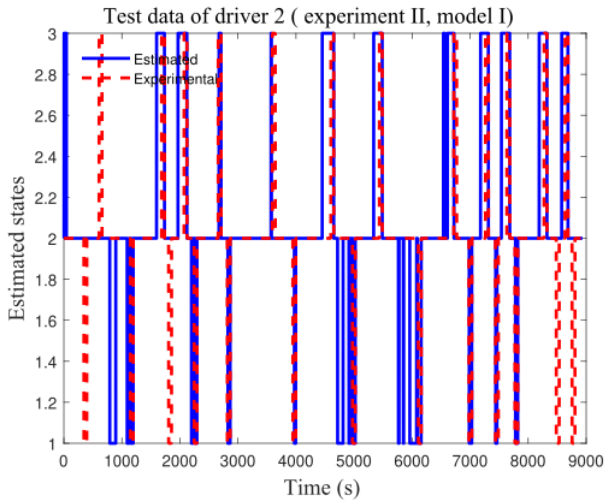

Figure 6.22: Generability test based on driver 2 (model I) [DS22b]

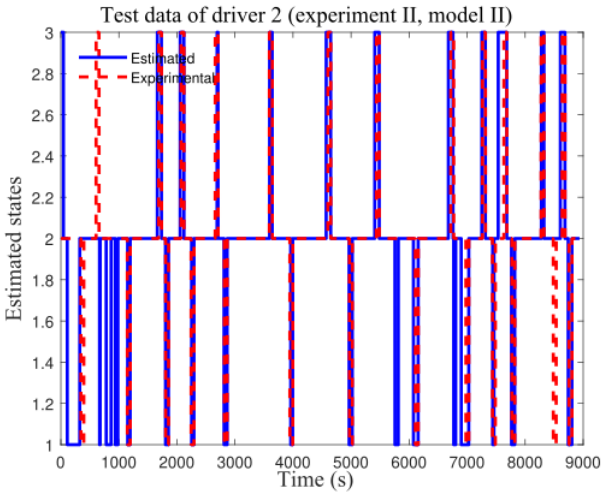

Figure 6.23: Generability test based on driver 2 (model II) [DS22b]

6.4.4 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 [DS22b]. The ANN-based state machine approach is developed in [DRS21], while the HMM is based on [DWS18]. The prefilter thresholds are optimized as well in the improved HMM, while the ANN-based state machine use biases and weights defined by the NSGA-II optimization to develop estimations. Model II is used for comparisons as it has a better performance than model I. All approaches also uses the same input variables as model II.

In Table 6.17, the average metric values based on all 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 Figure 6.24 to Figure 6.26. The area under curve (AUC) values of each method based on the different behaviors are also presented in Table 6.18.

Table 6.17: Comparisons between different approaches [DS22b]

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

From the results, the HMM-based state machine approach has a better performance

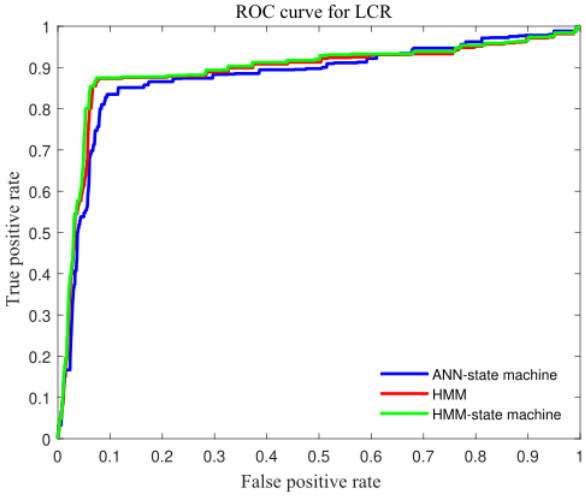

Figure 6.24: LCR ROC curve [DS22b]

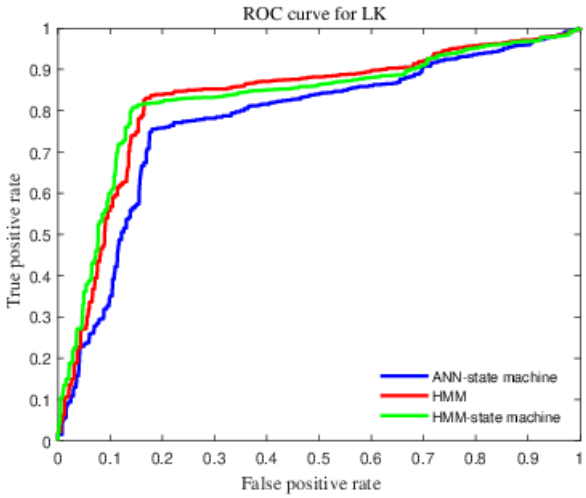

Figure 6.25: LK ROC curve [DS22b]

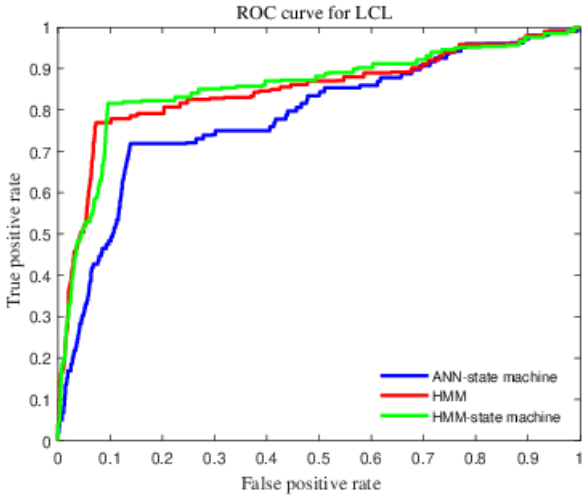

Figure 6.26: LCL ROC curve [DS22b]

Table 6.18: AUC values of different approaches [DS22b]

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

than the other two approaches in most metrics. On the other hand, the individual HMM approach outperforms the ANN-based approach, except for 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. On the other hand, the improvements based on the HMM-based state machine approach are not significantly better than other other approaches. This could be due to the scarce data or imbalanced data. While the other approaches like LSTM can be considered, but this method has lower interpretability and is prone to overfitting.

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 6.18), 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.

6.4.5 HMM-based state machine II

In this section, the evaluation of the HMM-based state machine model II is presented [DS23b]. To verify the sub-HMM combinations that 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 eleven sub-HMM combinations (as part of the proposed approach) is presented in Table 6.19.

Table 6.19: Average metric values of different models based on six test data sets [DS23b]

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

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:

- 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.

- On the other hand, HMM III, VIII, X, and XI generated balanced performances throughout the metrics in contrast to the rest. For 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.

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

- 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 [DS19a]) 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. For the conventional HMM model, a prefilter with five thresholds is applied to data variables to quantize each variable into segments.The conventional HMM uses default weights and prefilter threshold values, instead of optimized values. The average performance based on six drivers are evaluated.

Table 6.20: Comparisons based on HMM III [DS23b]

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

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.

6.4.6 Evaluation of Features

Data from five participants are considered for evaluations, such that two different drives were performed by each participant for training and test data [DS22a]. As the 10 fold cross-validation is performed first, this validation is based on the different preset lane changing duration (t_{total}). The ACC, DR, and FAR are calculated for the evaluations of models I (ENV) and II (ENV and ET) uisng only the training data collected, as part of

Table 6.21: Comparisons based on HMM V [DS23b]

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 6.22: Comparisons based on HMM VIII [DS23b]

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 6.23: Comparisons based on HMM X [DS23b]

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 6.24: Comparisons based on HMM XI [DS23b]

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

the cross-validation (Table 6.25). From the performance of both models, the metric values are the best when using 2 s as the preset t_{total}. Thus, the performance based on the preset t_{total} of 2 s will be used for comparison of the models. In Table 6.25, the average metric values based on all drivers are given.

Table 6.25: Average metric values of models I and II [DS22a]

States/Intentions	**Metrics**	**Models**	
		Model I (ENV) [%]	**Model II (ENV and ET)** [%]
Overall	*ACC*	80.01	71.28
Right	ACC	87.83	81.51
	DR	80.40	76.94
	FAR	11.10	17.77
Keep	ACC	80.35	72.14
	DR	79.75	68.99
	FAR	17.86	18.44
Left	ACC	91.84	88.90
	DR	81.21	79.43
	FAR	6.59	9.72

From the results, model I generally generates better ACC, DR, and FAR average values than model II for all states. Also, high ACC, DR, and low FAR are achieved in model I. The overall ACC in model I is higher than model II. For most data sets, the ACC and DR values based on model I are higher than 80 %. The ACC_{left} of model I is the highest average accuracy achieved with a value of 91.84 %. Compared with model II, FAR values of model I are lower, whereby the lowest value achieved is 6.59 % in FAR_{left}. In addition, the metric values of the left are better than right and keep for both models. The training of model I (about 600 to 711 s) is completed faster than model II (about 2013 to 2100 s) for all data sets. As a conclusion, it can be stated that the additional use of eye-tracking data does not provide any additional information to effectively recognize the intention. Thus on the individual's level, no consistent pattern arises based on the information collected in the time leading up to the actual decision. A possible reason is it is assumed driving decisions are mainly based on environmental conditions, such as the driver's relationship with surrounding vehicles. Another reason is due to the combination

of high and low accuracy features, as shown in fusion research [RS16]. Therefore, model I will be used for the training and test.

The ACC, DR, and FAR values of the different states based on the different drivers' data are presented in Table 6.26 using only ENV variables. Using driver 1 as an example, test data from the driver as well as test data from other drivers are used for test based on the trained model of driver 1.

In Table 6.26, the test data show the performance values based on the test data corresponding to the specific driver, while average denotes the average performance values when other drivers' test data are used. The results show most ACC and DR values are higher than 80 % with some exceptions. For most data sets, the values of ACC_{right} and DR_{left} are higher than 90 %. Low FAR values are achieved with a few exceptions like FAR_{keep} based on trained model of driver 4. Generally, DR_{right} and FAR_{keep} are worse in comparison with DR and FAR of other states, particularly when the trained models of drivers 1, 4, and 5 are used for test. From the results, it can be concluded that high ACC, DR, and low FAR are mainly achieved when the model is tested using different data sets. This not only proves the effectiveness of the model's recognition ability, but also its ability for generalization. In Figure 6.27, the estimated and actual states are shown for test data 3.

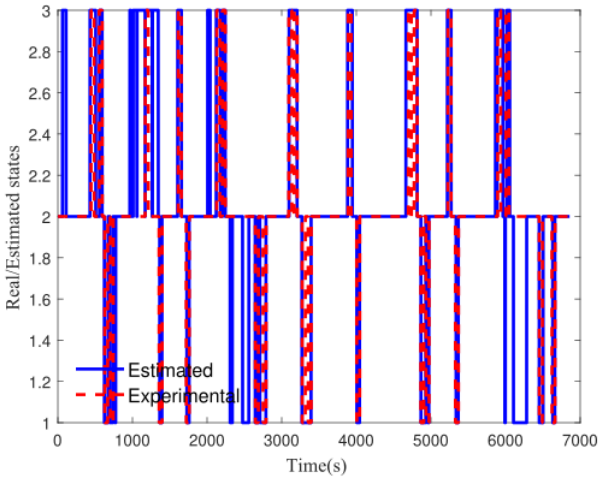

Figure 6.27: Real and estimated states (test data 3) [DS22a]

Further comparisons are done between the recognition performance of the proposed model and other machine learning-based models. Here, a standard CNN, HMM, and SVM are used for comparisons based on ENV features [DHBS20]. In Table 6.27, the average ACC, DR, and FAR based on the same five test data sets for the different models are presented.

The values related to CNN, HMM, and SVM are generated based on the results from [DHBS20]. The results show the developed model performs better than CNN in most metrics with the exception of DR_{right} and FAR_{left}. Particularly, significant improvements in the average FAR_{right}, DR_{keep}, and DR_{left} are observed with improvements from 75.21% to 6.61 %, 20.81 % to 82.10 %, and 0 % to 91.67 %, respectively. Compared with HMM, the model's performance is also better in most metrics, with the exception of average ACC_{right},

Table 6.26: Metric values of test data sets [DS22a]

States/Intentions	**Metrics**	**Test data** [%]	**Average** [%]
Driver 1			
Overall	ACC	80.50	81.91
Right	ACC	89.25	90.30
	DR	72.47	74.61
	FAR	9.07	8.20
Keep	ACC	81.12	82.36
	DR	80.14	81.92
	FAR	14.46	15.51
Left	ACC	90.63	91.16
	DR	91.81	89.24
	FAR	9.49	8.65
Driver 2			
Overall	ACC	80.71	76.79
Right	ACC	90.50	89.02
	DR	75.61	74.87
	FAR	8.10	9.59
Keep	ACC	81.16	77.20
	DR	79.72	75.03
	FAR	11.93	12.67
Left	ACC	89.76	87.35
	DR	95.30	95.17
	FAR	10.76	13.41
Driver 3			
Overall	ACC	82.84	81.37
Right	ACC	92.33	92.34
	DR	86.24	74.70
	FAR	7.11	5.94
Keep	ACC	82.84	81.69
	DR	81.29	80.88
	FAR	9.49	14.57
Left	ACC	90.51	88.72
	DR	94.77	92.51
	FAR	9.88	11.66
Driver 4			
Overall	ACC	83.67	82.44
Right	ACC	93.49	92.74
	DR	42.33	41.68
	FAR	1.73	2.30
Keep	ACC	84.22	83.21
	DR	88.13	86.68
	FAR	34.76	32.86
Left	ACC	89.63	88.93
	DR	81.71	83.84
	FAR	9.63	10.57
Driver 5			
Overall	ACC	81.62	81.95
Right	ACC	91.05	91.82
	DR	72.30	75.26
	FAR	7.04	6.61
Keep	ACC	82.28	82.42
	DR	81.19	81.16
	FAR	12.89	11.56
Left	ACC	89.91	89.67
	DR	94.77	96.25
	FAR	10.58	10.96

Table 6.27: Average metric values of different machine learning-based models [DS22a]

States/Intentions	Metrics	Models			
		Proposed model [%]	**CNN** [%]	**HMM** [%]	**SVM** [%]
Right	ACC	91.32	66.85	92.00	93.50
	DR	69.79	80.41	69.71	57.73
	FAR	6.61	75.21	5.04	1.47
Keep	ACC	82.32	71.82	81.41	79.42
	DR	82.19	20.81	83.77	97.65
	FAR	16.71	20.59	25.21	69.14
Left	ACC	90.09	87.43	89.41	85.92
	DR	91.67	0.00	78.77	9.25
	FAR	10.07	0.45	8.67	0.49

FAR_{right}, DR_{keep}, and FAR_{left}. Overall, the metric values between the HMM-based model and the proposed model are close for all data sets. The model also performs better than SVM for most metrics in terms of DR_{right}, ACC_{keep}, FAR_{keep}, ACC_{left}, and DR_{left}. Significant improvements are observed in FAR_{keep} and DR_{left}. Thus, the ANN-based state machine model performs better than CNN, HMM, and SVM in most metrics. The comparisons show that the model's performance is comparable to other ML-based models.

6.4.7 Hyperparmater optimization

In this section, the results based on the BO and GA are presented [DS23a]. The optimal hyperparameter values for each individual ANN model for the different data sets are given. Subsequently, the lane changing behavior recognition performances of the ANN-based state machine models (based on hyperparameters optimized using BO and GA) are given. Then, comparisons between both models as well as with the model without hyperparameter optimization [DRS21] are presented.

Hyperparameter Optimization Results

Based on the results of the BO and GA, the hyperparameter values obtained for each data sets are given in Table 6.28 and Table 6.29. As mentioned, the data sets used are the training data set of each driver.

Table 6.28: Optimal hyperparameter values (BO) [DS23a]

Data sets	Hidden layer neurons	Activation functions (first layer)	Learning rates	Epochs
1	18	tansig	0.0211	2812
2	18	tansig	0.0039	2961
3	14	logsig	0.0020	9971
4	13	tansig	0.2422	3167
5	20	logsig	0.0014	2028
6	18	tansig	0.0148	4797
7	20	logsig	0.0272	852

Table 6.29: Optimal hyperparameter values (GA) [DS23a]

Data sets	Hidden layer neurons	Activation functions (first layer)	Learning rates	Epochs
1	18	logsig	0.2609	7192
2	18	tansig	0.3298	2718
3	19	tansig	0.8791	1725
4	15	tansig	0.1390	8136
5	17	logsig	0.0531	4402
6	11	tansig	0.6680	6776
7	18	tansig	0.2562	5260

The number of hidden neurons are higher than or equal to 18 based on the BO for most of the data sets, while GA selects values between 11 to 19. The tansig is generally selected as the optimal activation function in both methods, with a few exceptions. High (higher than 1) learning rates averts convergence as the objective function displays divergent behavior (faster training) or it converges faster to a sub optimal solution. Conversely, low learning rates (lower 0.001) cause slow convergence (slow training) [WLB+19]. The obtained results show that the learning rates are generally optimal for all data sets in both methods. Nevertheless, the learning rates are much larger based on GA than Bayesian optimization. The optimal number of epochs required are higher than 800 for both methods. In addition, the computational times to run the BO and GA do not differ by much, whereby the former requires less than 400 seconds, while the latter requires more than 470 seconds (on a standard office PC (2.6 GHz)). A possible reason the BO is slightly faster is because the BO only updates the posterior causing it to be less computationally expensive.

Lane Changing Recognition Evaluations

In Table 6.30, the ACC, DR, and FAR of the recognition models based on both methods and the original model (without hyperparameter optimization) [DRS21] are compared. The original model consists of a hidden layer of 10 neurons, uses the tansig activation function (first layer), 100-200 epochs, and a learning rate of 0.01. The metrics based on three different maneuvers are used to evaluate the models. Here, the average performance based on the test data of seven drivers are presented. For each metric, the model that achieved the highest value and lowest value are highlighted in green and red, respectively.

The performance based on both methods show moderate (60 %-69 %) to high (higher than 80 %) ACC, DR, and low FAR for all maneuvers. The model based on BO generates rather high ACC and DR except for DR_{keep}, with a value of 68.89 %. High ACC and DR are generated based on GA as well, with the exception of $ACC_{overall}$, ACC_{keep}, and DR_{keep}. The results show that the model with optimized hyperparameters based on BO outperforms the model based on the GA in most metrics, with the exception of ACC_{right}, FAR_{right}, FAR_{keep}, and DR_{left}. A possible reason for this is because the BO takes past evaluations of hyperparameters into consideration, which enables the method to focus on a specific search space to develop the next set of hyperparameter values accurately. Nevertheless, the metric values obtained based on both techniques do not differ significantly. The original model produces high ACC, DR, and low FAR in all the maneuvers as well [DRS21]. Furthermore, the original model outperforms the model based on BO and GA in most metrics, with the exception of DR_{right}, FAR_{keep}, and DR_{left} (for both). This

Table 6.30: Average metric values of different models based on seven test data sets [DS23a]

States	Metrics	Developed models		
		Bayesian [%]	GA [%]	Original [%]
Overall	ACC	71.27	68.38	84.55
Right	ACC	80.18	82.30	90.08
	DR	89.48	85.71	78.63
	FAR	20.49	17.75	8.88
Keep	ACC	71.59	68.62	82.88
	DR	68.89	65.76	87.11
	FAR	16.74	15.47	36.99
Left	ACC	90.77	85.84	92.38
	DR	74.78	81.79	42.58
	FAR	8.44	13.82	2.88

could be due to the non-optimal search region for both optimization techniques. However, the BO generates a much balanced performance compared to the original performance. For example, no outliers are observed in performance based on BO, unlike the original's FAR_{keep} and DR_{left} (with poor performance). For both these metrics, the BO shows a significant improvement. The original model has the most green highlighted values (highest values), while the GA-based model has the most red highlighted values (lowest values).

6.5 Results of the LIB degradation estimation

In this section, the estimated capacity fade based on the different experiments are presented. In addition, the RMSE and MSE values are given for comparisons.

6.5.1 NARX-based state machine approach I

The results are based on the NASA data [DS23d]. For further insights, the temperature threshold values obtained for each experiment as well as the estimated and actual capacity when the battery reaches EoL (changes to state 2) are given.

Experiment I results

In Figure 6.28, the estimated capacity (blue line) and actual capacity (red dotted line) of test data B0006 are shown. The yellow line indicates the EoL capacity (1.44 Ah). Despite some off trends at the end, the estimated capacity fade curve shows a close proximity to the actual capacity fade. Low RMSE and MSE values of 0.0656 Ah and 0.0043 Ah are achieved based on the proposed approach, showing the model performs well. The state progression is shown in Figure 6.29. The tr_1 and tr_2 defined by the optimizer are -0.4408 and -1.3387, respectively. Here, it can be observed that the estimation begins with state 1, changes to state 2 (EoL) at cycle 100 (estimated capacity: 1.4335 Ah) and changes to state 3 at the end. The actual EoL cycle is also 100 (actual capacity: 1.4312 Ah). The estimated state remains in state 1 until cycle 100 as the threshold conditions for a transition are not met (based on the temperature and capacity at the each specific time point). Once the conditions are met, a change to state 2 is observed. The estimated state progression is

same as the actual state progression of the battery, proving the accuracy of the model [DS23d].

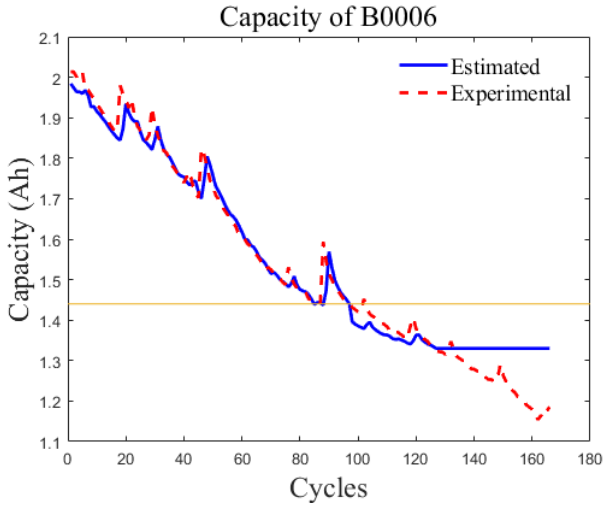

Figure 6.28: Actual and estimated discharge capacity of B0006 [DS23d]

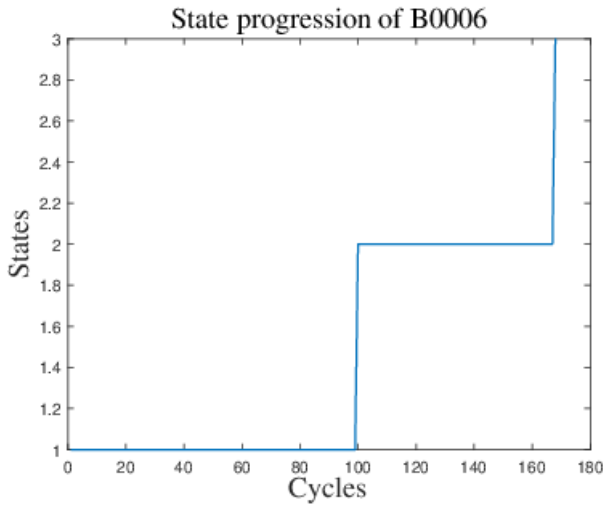

Figure 6.29: State progression of B0006 [DS23d]

Experiment II results

In Figure 6.30, the estimated capacity of RW9 is close to the actual capacity. The model has a good performance with low RMSE and MSE values (0.0361 Ah and 0,0013 Ah). The actual and estimated state profiles are close to each other (Figure 6.31). The tr_1 and tr_2 defined here are -0.4515 and -1.2338 respectively. The actual EoL is reached at cycle 15 (actual capacity: 1.6491 Ah). The estimated state also switches to state 2 (EoL reached) at cycle 15 (estimated capacity: 1.6402 Ah). Hence, the model correctly predicts the EoL cycle. The performance of the model also shows that this method is applicable for data with dynamic profiles.

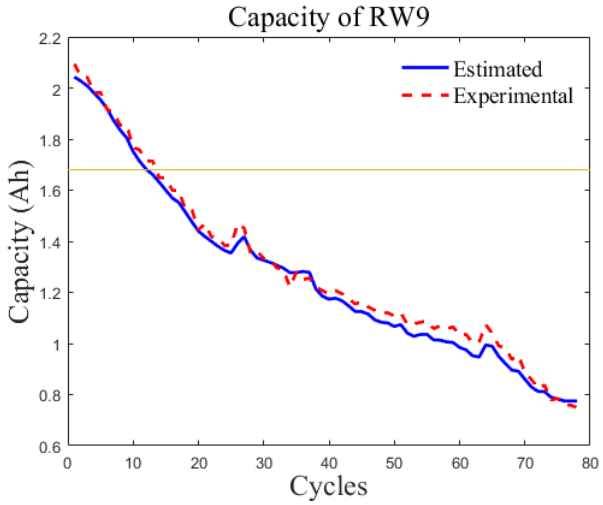

Figure 6.30: Actual and estimated discharge capacity of RW9 [DS23d]

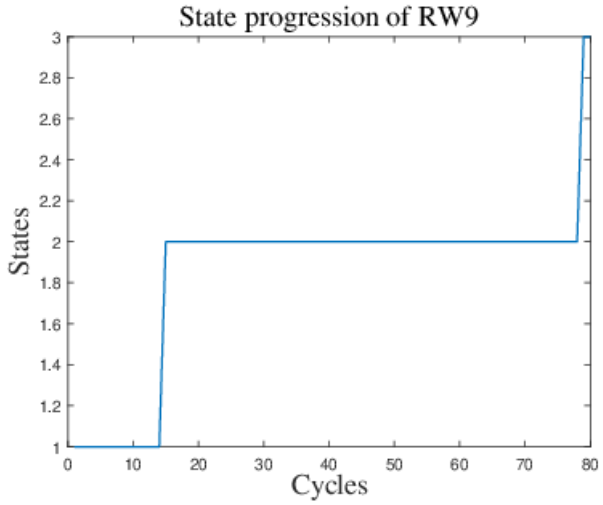

Figure 6.31: State progression of RW9 [DS23d]

Experiment III results

The RMSE and MSE values based on RW1 are 0.0337 Ah and 0.0011 Ah. The low error rates show that the actual capacity values are close to the estimated values. A deviation is observed in the estimated capacity towards the end, nevertheless the estimation is mostly accurate through the capacity fade (Figure 6.32). The state progression also shows that the model estimates state one in the beginning and switches to state 2 at cycle 25 (estimated capacity: 1.5732 Ah), close to the actual EoL cycle (actual capacity: 1.5964 Ah), which is cycle 23 (Figure 6.33). The model switches to the final state towards the end of the discharge capacity. The tr_1 and tr_2 developed here are 0.4103 and -1.882, respectively. The model also performs well when applied to this dynamical data.

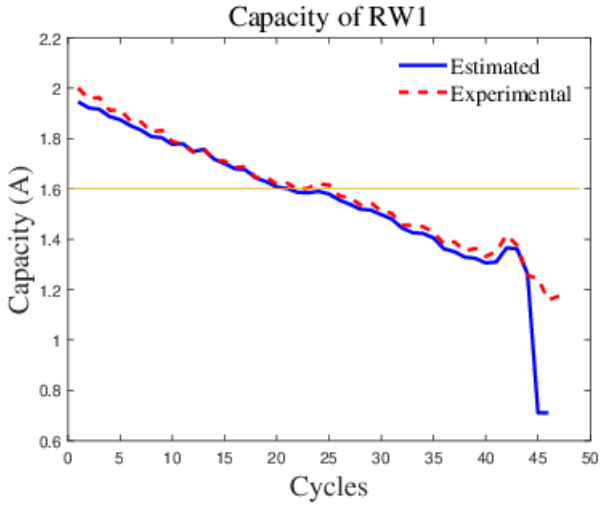

Figure 6.32: Actual and estimated discharge capacity of RW1 [DS23d]

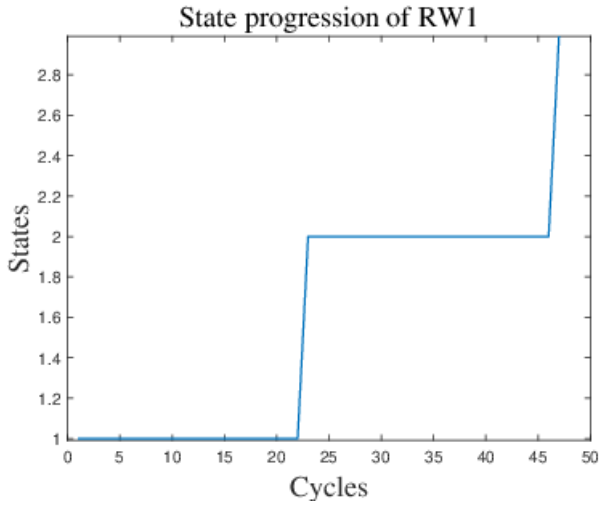

Figure 6.33: State progression of RW1 [DS23d]

The results based on the different test data are summarized in Table 6.31, which shows the RMSE, MSE, actual, and estimated cycle of EoL state. Based on the generated results,

it can be concluded that the proposed model can track the capacity fade effectively. In addition, the model is able to estimate the state progression closely. The results also show the necessity to select optimal parameters for the estimation process.

Table 6.31: Performance of the model based on different data sets [DS23d]

Test data	**RMSE**	**MSE**	**Actual EoL cycles**	**Estimated EoL cycles**
B0006	0.0656	0.0043	100	100
RW9	0.0361	0.0013	15	15
RW1	0.0337	0.0011	25	23

Comparisons between different methods

Comparisons between the proposed approach and a standard ANN are performed as well to validate the model's performance (Table 6.32) using the same input variables. The ANN consists a hidden layer of ten neurons. While the standard ANN generates low RMSE and MSE values, the proposed approach achieves a lower RMSE and MSE than the standard ANN for all data sets showing the model's effectiveness. A possible reason for this is due to the NARX abilities to handle time series data well.

Table 6.32: Comparisons between ANN and proposed approach [DS23d]

Test data	**RMSE (*Ah*)**		**MSE (*Ah*)**	
	Proposed approach	**ANN**	**Proposed approach**	**ANN**
B0006	0.0656	0.1021	0.0043	0.0104
RW9	0.0361	0.1807	0.0013	0.0326
RW1	0.0337	0.2421	0.0011	0.0586

6.5.2 NARX-based state machine approach II

The results for different portions of the test rig data (test data) are given in Table 6.33. As shown in the first row of the table, the trained model based on the first 30 % of the data is used to test the remaining 70 % to evaluate the effectiveness of the model.

Table 6.33: Performance based on different data portions

Test data	**RMSE(*Ah*)**	**MSE(*Ah*)**
70 % (30 % trained)	0.3223	0.1039
50 % (50 % trained)	0.2726	0.0743
30 % (70 % trained)	0.2313	0.0535

Low errors are generally achieved for the different instances showing its ability to estimate over different time spans. The lowest RMSE and MSE achieved are 0.2313 *Ah*

and 0.0535 *Ah* respectively, when the model is trained with 70 % of data, while the other 30 % is used for test. The highest RMSE and MSE are 0.3223 *Ah* and 0.1039 *Ah* based on the trained model of only 30 % of data, which is expected. In Figures 6.34-6.35, the estimated (blue line) and actual capacity (red dotted line) are presented for the different instances.

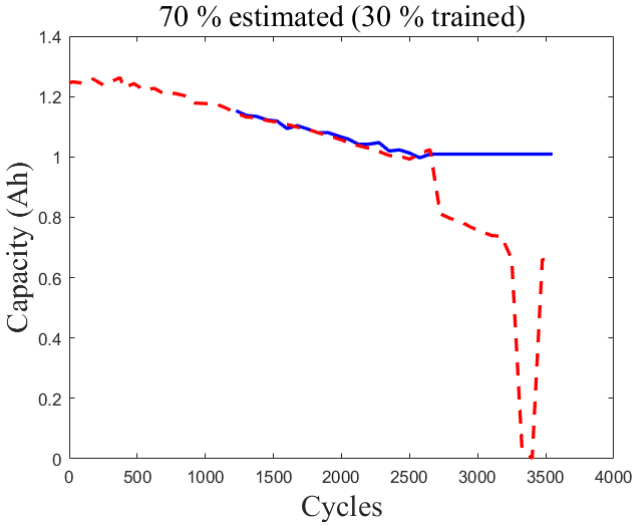

Figure 6.34: Actual and estimated discharge capacity based on 70 % of data (30 % trained)

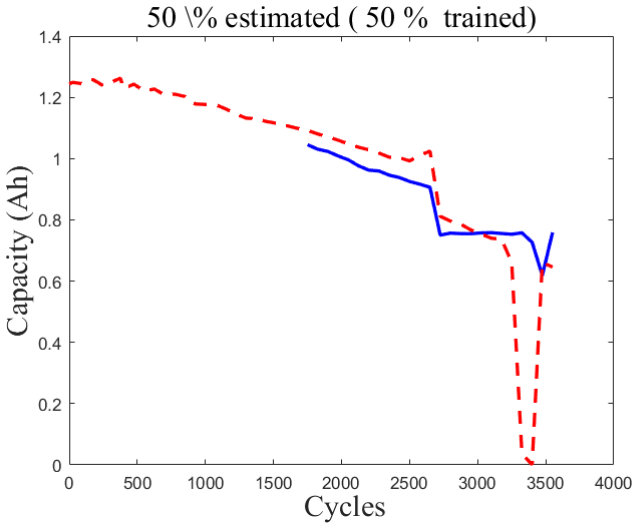

Figure 6.35: Actual and estimated discharge capacity based on 50 % of data (50 % trained)

In addition, the switching states for the different instances using the whole data are presented in Figures 6.36-6.38. The different portions of the data are trained, but the estimation is based on the whole data set to show the switching from the start. Switching states can be observed throughout the operation of the battery, particularly between states

1 and 2. In general, the state begins with initial state 0 and switches to state 1. State 0 is not observed in the figures as the NARX time step begins with time step 3 as mentioned previously. Switching behaviors between 1 and 2 are observed continuously and finally a switch to state 3. However, the state 3 is estimated at cycle 3500, which is not close not the actual EoL cycle of 2725.

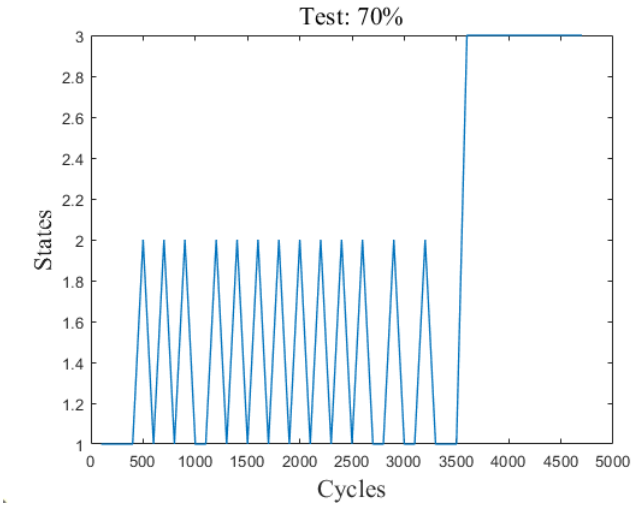

Figure 6.36: State progression (30 % trained)

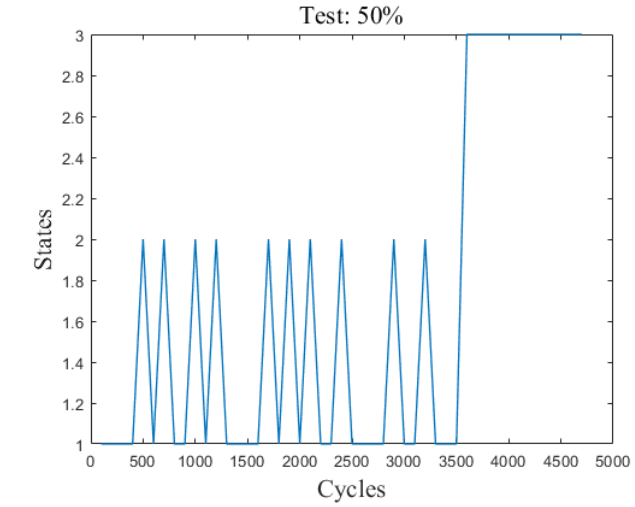

Figure 6.37: State progression (50 % trained)

6.6 Summary

In general, the application of the state machine approach for both the driving behaviors and capacity degradation estimation (particularly the neural network-based state machine model) shows its adaptability and its accurate estimation performance. The selection of appropriate parameter values through optimization aides the estimation process.

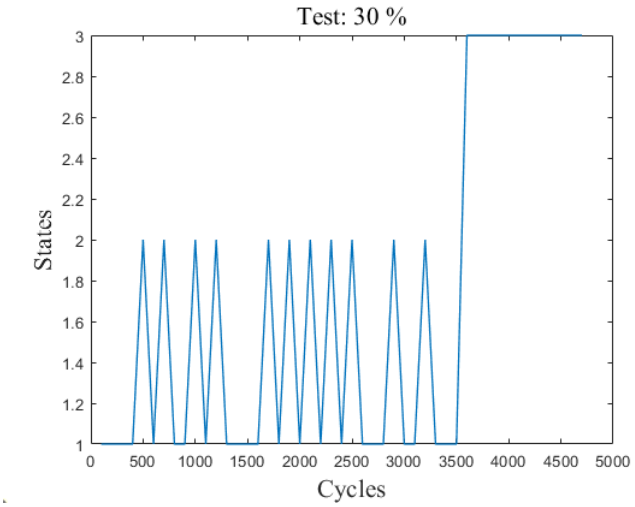

Figure 6.38: State progression (70 % trained)

7 Summary, conclusions, and outlook

In this chapter, the summary and conclusion about the state machine-based approach for the development of estimation models in two similarly structured system are given. In this context, the recognition of lane changing behaviors and the capacity degradation behavior estimation are discussed. In addition, the outlook based on the developed approach and the results are listed.

7.1 Summary and conclusions

The application of ML-based approaches has commonly been used for the behavior estimation of systems with structural changes. These approaches tend to learn from patterns to estimate the changing behaviors. Systems with structural changes studied in this thesis included the lane changing driving behaviors and degradation behavior of LIBs (capacity fade). Here, the focus is on development of estimation models for lane changing behaviors and capacity fade of LIBs. While ML-approaches are often used to develop the estimation models, there is still a lack of approaches that can develop an optimal model, prompting the need for the development of a new model. In this thesis, a state machine-based model is developed for the both domains.

The varying human driving behaviors makes it possible for the state machine to model the lane changing behaviors using states and control the transitions between them. It also defines behavioral changes (multi-state switching problems) using state transitions defined by transition conditions. Three different states are defined: LCR, LK, and LCL. For the initial model, the transition conditions are defined by the threshold values associated with different driving features. The threshold values are model parameters defined using NSGA-II. Here, the state transitions generate the estimation of the model.

The ANN and HMM are integrated into the initial state machine approach to develop a combined model. In these models, the estimations of ANN and HMM are used as the transition conditions following the same topology of the initial model. Two types of ANN-based state machine models are developed: combining the state machine with one common ANN and with three different ANNs (representing the different behaviors). In this thesis, two types of improved HMMs are developed as well. The first HMM includes the application of prefilter on the different input variables (distances and TTCs), while the second HMM uses both the prefilter application and different sub-HMMs for the different input variables (to develop the improved HMM model). Hence, combining the improved HMMs with the state machine model develops combined models, with the improved HMMs as derived HMMs.

As features associated with the driver's environment and characteristics tend to affect the performance of estimation models, the effects of ENV and ET variables are studied in this thesis using the ANN-based state machine approach.

As mentioned, selecting optimal parameters and hyperparameters of the estimation model can be a challenging task. Model parameters of the initial state machine model are the threshold values, while for the combined models, the parameters are weights, biases, and/or threshold prefilter values. The values developed aid in the selection of the next state (for state machines), a particular neuron (for ANNs) or the particular observation variable (for HMMs) which ultimately affects the lane changing estimation. Hence, the model parameters are optimized using NSGA-II. Generalization of the approach is also studied by using the same parameter values to test the model with different data sets. The aim is to study if a set of same parameter values is able to generate optimal estimations when testing the model using different data sets. This avoids the need for different models with different parameter values for different data sets. As for the hyperparameter optimization, BO and GA are employed. The optimization methods select the optimal values from a specified range. First, BO and GA are applied to the ANN models to develop the optimal individual ANN models. As the hyperparameters are set prior to training, the ANN models with optimized hyperparameters are then combined with the state machine for training the proposed model.

Based on the open questions posed in section 2.3 for driving behavior estimations, several conclusions can be drawn:

- The results generally show that the models are able to achieve high ACC, DR, and low FAR for the state machine approach and combined approaches. This shows the optimality of the state machine approach through the development of the optimal model parameters. Furthermore, the model can be generalized based on its ability to develop accurate estimations when a trained model with optimized parameters is tested using different test data.

- The ANN-based state machine model outperforms the conventional ANN's performance. For the HMM-based state machine models, it can be concluded that a HMM-derived model can be incorporated into the state machine model. As for the HMM-based state machine model I, the model with prefilter application on the TTC variables developed better results than the prefitler application on distances. This suggests the significance of the TTC features for the model's estimations. The proposed approach outperforms the conventional HMM and ANN-based state machine model. However, the improvement is small, which could be due to scarce or imbalanced data. On the other hand, it could also be due to the design of the experiment (the way the lane change is defined). For the HMM-based state machine II model, sub-HMM combinations III, VIII, X, and XI developed a balanced performance throughout and also improved results compared to the conventional HMM. These sub-combinations consist of operational features, which shows the importance of these features for the estimations.

- In addition, using only ENV variables tend to result in a better performance than using a combination of ENV and ET. Therefore, the addition of ET variables are not significant for the estimation of lane changing behaviors. As mentioned previously, this is either due to the fact the driving decisions are mainly based on ENV features or combination a lower accuracy features with higher accuracy features.

- One of the issues faced in existing literature is the development of optimal model parameters. The NSGA-II is able to generate optimal parameters based on the

estimation performance of the various models. For hyperparameter optimization of the state machine model, the model with optimized hyperparameters generated using BO resulted in better lane changing estimation performance than GA. A limitation that exists is the model with no hyperparameter optimization (only model parameters are optimized) outperformed the models based on BO and GA for most metrics. Nevertheless, the original still performs poorly in some of the metrics, whereas the model with BO-optimized hyperparameters has a more balanced performance.

For the capacity fade estimations, existing approaches do not consider different degradation states, which is tackled in this thesis through the use of the NARX-based state machine model. Two models are developed here. Different degradation states are developed to define the progression of degradation using the state machine. The NARX calculates the capacity at a given time point depending on the degradation state of the battery using normalized temperatures. A transition from one state to another is based on the threshold conditions associated with the temperatures. From the results, several conclusions can be drawn in relations to the questions posed in section 2.3:

- The results show that the first model is able to estimate the capacity fade with very low MSE and RMSE for different battery data sets (both non-dynamic and dynamic operation). In addition, the model outperforms a conventional ANN model, showing the effectiveness of the model. The estimated state progression and actual state progression are close, such that EoL state is estimated at a cycle close to the actual EoL cycle. For the first model, the state switches from state 1,2 and 3 consecutively at accurate time points.

- The second model is able to estimate the capacity over various time spans with acceptable range of error rates. Here, the model switches from the initial point of state 0 to state 1. The model then switches between states 1 and 2 repeatedly, and finally transitions to state 3 towards the end. Switching states are observed for this model. However, the estimation of state 3 (EoL state) cycle is not close to the actual cycle for all cases.

- Degradation states can be modeled using a state machine model, defined using the capacity and the discharge cycles of the batteries. Existing models have used the Arrhenius equation to calculate the capacity. By using the NARX instead, effective estimations with low errors rate can still be achieved. In addition, electrochemical information is not needed to develop this model, which reduces the complexity of the model with the ability to generate effective estimations.

- As the parameters of the model are also optimized using NSGA-II, the optimal parameters enable the model to estimate with low error rates.

A major limitation with the state machine model for the application in both domains is it does not always generate very high estimation performance. For the driving behavior estimations, certain models tend to have better performance when specific ENV features are used, while others do not develop optimal estimations using the same features. Also, the simple structure of the model may have a negative effect on the model.

7.2 Outlook

While significant results have been achieved through the application of the state machine for the estimation of driving behaviors, future work can include the use of automated methods to select the relevant driving features as inputs, such as the use of Deep Neural Network. In addition, using other features can be tested as well, for both ENV and ET features. As only saccadic ET features are considered in this work, fixation ET features can be incorporated as well to test the efficiency of the model in future. In addition, vehicle dynamics are not particular considered in this work, which should be studied further. Another constraint faced is the time required for training, hence modifying the model's structure or using another optimization technique should be researched. The HMM-based models are also not generalizable due to the properties of HMM, thus requiring different models and parameters for different driving data. Methods to develop a generalized model should be researched. Furthermore, changes to the transition conditions in the state machine model (whether the threshold-based or estimation of another method) should be considered for performance improvement. In this thesis, only lane changing behaviors are considered for the application of the state machine model. In future, the state machine approach can be extended for the estimation of driving styles (aggressive/non-aggressive), drunk driving, and fatigue driving.

As for the capacity fade estimation, changes to topology and threshold conditions defining the different states can be studied in future. The current models only consider the temperatures as input. However, a combination of other stress factors should be considered.

Bibliography

[ACH+09] Androutsopoulos, K. ; Clark, D. ; Harman, M. ; Li, Z. ; Tratt, L.: Control Dependence for Extended Finite State Machines. In: *Fundamental Approaches to Software Engineering*. Berlin, Heidelberg : Springer, 2009, pp. 216–230

[ACW16] Ashwin, T.R. ; Chung, Y.M. ; Wang, J.: Capacity fade modelling of lithium-ion battery under cyclic loading conditions. In: *Journal of Power Sources* 328 (2016), pp. 586–598

[BB12] Bergstra, J. ; Bengio, Y.: Random Search for Hyper-Parameter Optimization. In: *J. Mach. Learn. Res.* 13 (2012), pp. 281–305

[BD11] Bonchek-Dokow, E.: *Cognitive modeling of human intention recognition, PhD Thesis*. Interdisciplinary Studies Unit, Bar Ilan University, Gonda Multidisciplinary Brain Research Center, Diss., 2011

[BED08] Berndt, H. ; Emmert, J. ; Dietmayer, K.: Continuous Driver Intention Recognition with Hidden Markov Models. In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 1189–1194

[BKD14] Bole, B. ; Kulkarni, C. ; Daigle, M.: Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use. In: *Annual Conference of the PHM Society*, 2014

[BMAS19] Beganovic, N. ; Moulik, B. ; Ali, A. ; Söffker, D.: Lifetime Model Development for Integration in Power Management of HEVs By Terms of Minimizing Fuel Consumption and Battery Degradation. In: *Annual Conference of the PHM Society* Bd. 11, 2019

[BS17] Beganovic, N. ; Söffker, D.: Remaining lifetime modeling using State-of-Health estimation. In: *Mechanical Systems and Signal Processing* 92 (2017), pp. 107–123

[Bö10] Börger, E.: *The Abstract State Machines Method for High-Level System Design and Analysis*. London : Springer, 2010. – 79–116 S

[Car07] Carruthers, P.: The Illusion of Conscious Will. In: *Synthese* 159 (2007), pp. 197–213

[CCF+21] Catelani, M. ; Ciani, L. ; Fantacci, R. ; Patrizi, G. ; Picano, B.: Remaining Useful Life Estimation for Prognostics of Lithium-Ion Batteries Based

on Recurrent Neural Network. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11

[CJ15] Carbonne, Y. ; Jacob, C.: Genetic Algorithm as Machine Learning for profiles recognition. In: *2015 7th International Joint Conference on Computational Intelligence (IJCCI)* Bd. 1, 2015, pp. 157–166

[CN03] Christensen, J. ; Newman, J.: Effect of Anode Film Resistance on the Charge/Discharge Capacity of a Lithium-Ion Battery. In: *Journal of The Electrochemical Society* 150 (2003), pp. A1416

[CP12] Chen, C. ; Pecht, M.: Prognostics of lithium-ion batteries using model-based and data-driven methods. In: *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference*, 2012, pp. 1–6

[CRKK15] Craye, C. ; Rashwan, A. ; Kamel, M. S. ; Karray, F.: A Multi-Modal Driver Fatigue and Distraction Assessment System. In: *International Journal of Intelligent Transportation Systems Research* 14 (2015)

[CV04] Cortes, C. ; Vapnik, V.: Support-Vector Networks. In: *Machine Learning* 20 (2004), pp. 273–297

[CYLA15] Chan, R.W. K. ; Yuen, J. K. K. ; Lee, E. W. M. ; Arashpour, M.: Application of Nonlinear-Autoregressive-Exogenous model to predict the hysteretic behaviour of passive control systems. In: *Engineering Structures* 85 (2015), pp. 1–10

[Deb01] Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. USA : John Wiley & Sons, Inc., 2001

[Den20] Deng, Q.: *Improved machine learning approaches for individualized human assistance, supervision, and behavior prediction, PhD Thesis*. Chair of Dynamics and Control, University of Duisburg-Essen, Diss., 2020

[DF16] Dou, F. ; Feng, D.: Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In: *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016, pp. 901–906

[DFBH17] Dang, H. Q. ; Fürnkranz, J. ; Biedermann, A. ; Hoepfl, M.: Time-to-lane-change prediction with deep learning. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–7

[DHBS20] Deng, J. ; Hillebrand, K. ; Benjamin, C. R. ; Söffker, D.: Prediction Performance of Lane Changing Behaviors: A Study of Combining Environmental and Eye-Tracking Data in a Driving Simulator. In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), Nr. 8, pp. 3561–3570

[DK13] Daigle, M. ; Kulkarni, C. S.: Electrochemistry-based battery modeling for prognostics. In: *Annual Conference of the PHM Society*, 2013

[DLH+19] Downey, A. ; Lui, Y.H. ; Hu, C. ; Laflamme, S. ; Hu, S.: Physics-based prognostics of lithium-ion battery using non-linear least squares with dynamic bounds. In: *Reliability Engineering System Safety* 182 (2019), pp. 1–12

[DPAM02] Deb, K. ; Pratap, A. ; Agarwal, S. ; Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), pp. 182–197

[DRS20] David, R. ; Rothe, S. ; Söffker, D.: State Machine Approach for Lane Changing Driving Behavior Recognition. In: *Automation* 1 (2020), pp. 68–79

[DRS21] David, R. ; Rothe, S. ; Söffker, D.: Lane changing behavior recognition based on Artificial Neural Network-based State Machine approach. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 3444–3449

[DS18] Deng, Q. ; Söffker, D.: Improved Driving Behaviors Prediction Based on Fuzzy Logic-Hidden Markov Model (FL-HMM). In: *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 2003–2008

[DS19a] Deng, Q. ; Söffker, D.: Classifying Human Behaviors: Improving Training of Conventional Algorithms. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1060–1065

[DS19b] Deng, Q. ; Söffker, D.: Modeling and Prediction of Human Behaviors based on Driving Data using Multi-Layer HMMs. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2014–2019

[DS22a] David, R. ; Söffker, D.: Effect of environmental and eye-tracking information: An Artificial Neural Network-based state machine approach for human driver intention recognition. In: *2022 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2022, pp. 16–22

[DS22b] David, R. ; Söffker, D.: A Study on a HMM-Based State Machine Approach for Lane Changing Behavior Recognition. In: *IEEE Access* 10 (2022), pp. 122954–122964

[DS22c] Deng, Q. ; Söffker, D.: A Review of HMM-Based Approaches of Driving Behaviors Recognition and Prediction. In: *IEEE Transactions on Intelligent Vehicles* 7 (2022), pp. 21–31

[DS23a] David, R. ; Söffker, D.: Comparison of different hyperparameter optimization methods on driving behavior recognition. In: *2023 Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2023. – accepted

[DS23b] David, R. ; Söffker, D.: A modified Hidden Markov Model (HMM)-based state machine model for driving behavior recognition: Effectiveness of features using different sub-HMMs. In: *2023 Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2023. – accepted

[DS23c] David, R. ; Söffker, D.: A review on machine learning-based models for lane-changing behavior prediction and recognition. In: *Frontiers in Future Transportation* 4 (2023)

[DS23d] David, R. ; Söffker, D.: A State Machine-Based Approach for Estimating the Capacity Loss of Lithium-Ion Batteries. In: *Annual Conference of the PHM Society*, 2023

[DSLL15] D.Tran ; Sheng, W. ; Liu, L. ; Liu, M.: A Hidden Markov Model based driver intention prediction system. In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (2015), pp. 115–120

[DT09] Doshi, A. ; Trivedi, M. M.: On the Roles of Eye Gaze and Head Dynamics in Predicting Driver's Intent to Change Lanes. In: *IEEE Transactions on Intelligent Transportation Systems* 10 (2009), pp. 453–462

[DTB+10] Dai, J. ; Teng, J. ; Bai, X. ; Shen, Z. ; Xuan, D.: Mobile phone based drunk driving detection. In: *Pervasive Health*, IEEE, 2010, pp. 1–8

[DWS18] Deng, Q. ; Wang, J. ; Söffker, D.: Prediction of human driver behaviors based on an improved HMM approach. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 2066–2071

[DJ+18] Díaz-Álvarez, A. ; M.Clavijo ; Jiménez, F. ; Talavera, E. ; Serradilla, F.: Modelling the human lane-change execution behaviour through Multilayer Perceptrons and Convolutional Neural Networks. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 56 (2018), pp. 134–148

[EDSD96] Eberhart, R. C. ; Dobbins, R. C. ; Simpson, P. K. ; Dobbins, R. W.: *Computational Intelligence PC Tools.* AP Professional, 1996

[Eur23] European Commision: Road deaths per million inhabitants – preliminary data for 2022. In: *Mobility & Transport -Road Safety: CARE database*, 2023

[FS98] Fine, S. ; Singer, N.: The Hierarchical Hidden Markov Model: Analysis and Applications. In: *Machine Learning, Springer* 32 (1998)

[GBH21] Griesbach, K. ; Beggiato, M. ; Hoffmann, K. H.: Lane Change Prediction With an Echo State Network and Recurrent Neural Network in the Urban Area. In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–7

[GHHW19] Gu, J. ; Han, Y. ; Han, M. ; Wei, L.: Vehicle Lane Change Decision Model Based on Random Forest. In: *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2019, pp. 115–120

[Gil62] Gill, A.: *Introduction to the theory of finite-state machines.* McGraw-Hill, 1962

[Gol93] Golson, S.: One-hot state machine design for FPGAs. In: *Proc. 3rd Annual PLD Design Conference & Exhibit* Bd. 1, 1993

[HHW20] Hasenjäger, M. ; Heckmann, M. ; Wersing, H.: A Survey of Personalization for Advanced Driver Assistance Systems. In: *IEEE Transactions on Intelligent Vehicles* 5 (2020), pp. 335–344

[HJ9] Han, T. ; Jing, J. ; Özgüner, Ü.: Driving Intention Recognition and Lane Change Prediction on the Highway. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 957–962

[HK21] Hema, D. ; Kumar, K. A.: Hyperparameter optimization of LSTM based Driver's Aggressive Behavior Prediction Model. In: *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 752–757

[Hol73] Holland, J. H.: Genetic Algorithms and the Optimal Allocation of Trials. In: *SIAM J. Comput.* 2 (1973), pp. 88–105

[Hol22] Holderbaum, M.: *Implementierung, Auswertung und Vergleich realit¨atsnaher Alterungsversuche an LiFePO4-Akkumulatoren mittels eines DSP-basierten HIL-Systems, Master Thesis.* Chair of Dynamics and Control, University of Duisburg-Essen, 2022

[Hue97] Hue, X.: Genetic algorithms for optimization: Background and applications. In: *Edinburgh Parallel Computing Centre, Univ. Edinburgh, Edinburgh, Scotland, Ver* 1 (1997)

[HWMP11] He, W. ; Williard, N. ; M.Osterman ; Pecht, M.: Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. In: *Journal of Power Sources* 196 (2011), pp. 10314–10321

[HZW12] He, L. ; Zong, C. ; Wang, C.: Driving intention recognition and behaviour prediction based on a double-layer Hidden Markov Model. In: *Journal of Zhejiang University SCIENCE C* 13 (2012), pp. 208–217

[HZZL17] Hu, X. ; Zou, C. ; Zhang, C. ; Li, Y.: Technological Developments in Batteries: A Survey of Principal Roles, Types, and Management Needs. In: *IEEE Power and Energy Magazine* 15 (2017), pp. 20–31

[IPMB+17] Izquierdo, R. ; Parra, I. ; Muñoz-Bulnes, J. ; Fernández-Llorca, D. ; Sotelo, M. A.: Vehicle trajectory and lane change prediction using ANN and SVM classifiers. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6

[IQP+19] Izquierdo, R. ; Quintanar, A. ; Parra, I. ; Fernández-Llorca, D. ; Sotelo, M. A.: Experimental validation of lane-change intention prediction methodologies based on CNN and LSTM. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3657–3662

[JF15] Jiang, B. ; Fei, Y.: Traffic and vehicle speed prediction with neural network and Hidden Markov Model in vehicular networks. In: *2015 IEEE Intelligent Vehicles Symposium (IV)* (2015), pp. 1082–1087

[JF17] Jiang, B. ; Fei, Y.: Vehicle Speed Prediction by Two-Level Data Driven Models in Vehicular Networks. In: *IEEE Transactions on Intelligent Transportation Systems* 18 (2017), pp. 1793–1801

[JLL20] Jin, C. ; Liu, Y. ; Lu, P.: Gauss mixture hidden Markov model to characterise and model discretionary lane-change behaviours for autonomous vehicles. In: *IET Intelligent Transport Systems* 14 (2020), pp. 401–411

[JRGV16] Joy, T. T. ; Rana, S. ; Gupta, S. ; Venkatesh, S.: Hyperparameter tuning for big data using Bayesian optimisation. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 2574–2579

[JXP22] Jia, H. ; Xiao, Z. ; P.Ji: Real-time fatigue driving detection system based on multi-module fusion. In: *Computers Graphics* 108 (2022), pp. 22–33

[Kan13] Kang, H. B.: Various Approaches for Driver and Driving Behavior Monitoring: A Review. In: *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 616–623

[KJ17] Keil, P. ; Jossen, A.: Impact of Dynamic Driving Loads and Regenerative Braking on the Aging of Lithium-Ion Batteries in Electric Vehicles. In: *Journal of The Electrochemical Society* 164 (2017), pp. A3081–A3092

[KKK20] Klitzke, L. ; Koch, C. ; Köster, F.: Identification of Lane-Change Maneuvers in Real-World Drivings With Hidden Markov Model and Dynamic Time Warping. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7

[Koz03] Kozlowski, J. D.: Electrochemical cell prognostics using online impedance measurements and model-based data fusion techniques. In: *2003 IEEE Aerospace Conference Proceedings* Bd. 7 IEEE, 2003, pp. 3257–3270

[KPLL13] Kumar, P. ; Perrollaz, M. ; Lefèvre, S. ; Laugier, C.: Learning-based approach for online lane change intention prediction. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 797–802

[KRH17] Koenig, A. ; Rehder, T. ; Hohmann, S.: Exact inference and learning in hybrid Bayesian Networks for lane change intention classification. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1535–1540

[KS20] Kögler, F. ; Söffker, D.: State-based open-loop control of plant growth by means of water stress training. In: *Agricultural Water Management* 230 (2020), pp. 105963

[KWB+11] Kasper, D. ; Weidl, T. ; Breuel, G. ; Tamke, A. ; Rosenstiel, W.: Object-oriented Bayesian networks for detection of lane change maneuvers. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 673–678

[LC12] Lee, B. G. ; Chung, W.Y.: Driver Alertness Monitoring Using Fusion of Facial Features and Bio-Signals. In: *IEEE Sensors Journal* 12 (2012), pp. 2416–2422

[LMH17] Lee, Y. P. ; McMains, S. ; Hedrick, J. K.: Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6

[LPZ+13] Liu, D. ; Pang, J. ; Zhou, J. ; Peng, Y. ; Pecht, M.: Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. In: *Microelectronics Reliability* 53 (2013), pp. 832–839

[LRL07] Liang, Y. ; Reyes, M. L. ; Lee, J. D.: Real-time detection of driver cognitive distraction using support vector machines. In: *IEEE transactions on intelligent transportation systems* 8 (2007), pp. 340–350

[LRW16] Li, X. ; Roetting, M. ; Wang, W.: Bayesian Network-based Identification of Driver Lane-changing Intents Using Eye Tracking and Vehicle-based Data, 2016

[LSB+17] Liu, Z. ; Sun, G. ; Bu, S. ; Han, J. ; Tang, X. ; Pecht, M.: Particle Learning Framework for Estimating the Remaining Useful Life of Lithium-Ion Batteries. In: *IEEE Transactions on Instrumentation and Measurement* 66 (2017), pp. 280–293

[LW17] Leonhardt, V. ; Wanielik, G.: 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

[LW18] Leonhardt, V. ; Wanielik, G.: Recognition of Lane Change Intentions Fusing Features of Driving Situation, Driver Behavior, and Vehicle Movement by Means of Neural Networks. In: *Advanced Microsystems for Automotive Applications 2017*, Springer International Publishing, 2018, pp. 59–69

[LWWX19] Li, Y. ; Wang, H. ; Wang, L. ; Xu, C.: A Driver's Physiology Sensor-Based Driving Risk Prediction Method for Lane-Changing Process Using Hidden Markov Model. In: *Sensors* 19 (2019)

[LWXW16] Li, K. ; Wang, X. ; Xu, Y. ; Wang, J.: Lane changing intention recognition based on speech recognition models. In: *Transportation Research Part C: Emerging Technologies* 69 (2016), pp. 497–514

[LZT+14] Lin, N. ; Zong, C. ; Tomizuka, M.i ; Song, P. ; Zhang, Z. ; Li, G.: An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control. In: *Mathematical Problems in Engineering* 2014 (2014), pp. 1–15

[LZX+20] Li, P. ; Zhang, Z. ; Xiong, Q. ; Ding, B. ; Hou, J. ; Luo, D. ; Rong, Y. ; Li, S.: State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network. In: *Journal of Power Sources* 459 (2020), pp. 228069

[LZZF20] Liu, S. ; Zheng, K. ; Zhao, L. ; Fan, P.: A driving intention prediction method based on Hidden Markov Model for autonomous driving. In: *Computer Communications* 157 (2020), pp. 143–149

[Mar15] Maranda, W.: Capacity Degradation of Lead-acid Batteries Under Variable-depth Cycling Operation in Photovoltaic System. In: *2015 22nd International Conference Mixed Design of Integrated Circuits Systems (MIXDES)*, 2015

[MD20] Mohanty, M. ; Dey, P. P.: Modeling the lane changing behavior of major stream traffic due to U-turns. In: *Transportation Engineering* 2 (2020), pp. 100012

[MDPB09] Meyer-Delius, D. ; Plagemann, C. ; Burgard, W.: Probabilistic situation recognition for vehicular traffic scenarios. In: *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 459–464

[Mea55] Mealy, G. H.: A Method for Synthesizing Sequential Circuits. In: *Bell System Technical Journal* 34 (1955), pp. 1045–1079

[Mic85] Michon, J. A.: *A Critical View of Driver Behavior Models: What Do We Know, What Should We Do?* Boston, MA : Springer US, 1985. – 485–524 S

[MKW+15] Murphey, Y. L. ; Kochhar, D. S. ; Watta, P. ; Wang, X. ; Wang, T.: Driver Lane Change Prediction Using Physiological Measures. In: *SAE International Journal of Transportation Safety* 3 (2015), pp. 118–125

[Moo56] Moore, E. F.: Gedanken-experiments on sequential machines. In: *Automata studies* 34 (1956), pp. 129–153

[MS05] Mandalia, H. M. ; Salvucci, M. D. D.: Using Support Vector Machines for Lane-Change Detection. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 49 (2005), pp. 1965–1969

[MT16] Miyajima, C. ; Takeda, K.: Driver-Behavior Modeling Using On-Road Driving Data: A new application for behavior signal processing. In: *IEEE Signal Processing Magazine* 33 (2016), Nr. 6, pp. 14–21

[MTM05] Mori, N. ; Takeda, M. ; Matsumoto, K.: A Comparison Study between Genetic Algorithms and Bayesian Optimize Algorithms by Novel Indices. In: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA : Association for Computing Machinery, 2005, pp. 1485–1492

[MTWR05] McCall, J.C. ; Trivedi, M.M. ; Wipf, D. ; Rao, B.: Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, 2005, pp. 59–59

[MTZ78] Mockus, J. ; Tiesis, V. ; Zilinskas, A.: The Application of Bayesian Methods for Seeking the Extremum. In: *Towards Global Optimization* 2 (1978), pp. 117–129

[Nat23] National Center for Statistics and Analysis, National Highway Traffic Safety Administration: Early estimate of motor vehicle traffic fatalities in 2022 (Crash Stats Brief Statistical Summary. Report No. DOT HS 813 428)8. In: *NHTSA's National Center for Statistics and Analysis*, 2023

[ND15] Neto, U. M. B. ; Dougherty, E. R.: *Error estimation for pattern recognition.* John Wiley & Sons, 2015

[NTTB14] Nagasaka, S. ; Taniguchi, K. ; Takenaka, K. ; Bando, T.: Prediction of Next Contextual Changing Point of Driving Behavior Using Unsupervised Bayesian Double Articulation Analyzer. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 924–931

[Ple04] Plett, G. L.: Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. In: *Journal of Power sources* 134 (2004), pp. 277–292

[PLKR11] Patel, M. ; Lal, S.K.L. ; Kavanagh, D. ; Rossiter, P.: Applying neural network analysis on heart rate variability data to assess driver fatigue. In: *Expert Systems with Applications* 38 (2011), pp. 7235–7242

[Pow11] Powers, D. M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. In: *Journal of Machine Learning Technologies* 2 (2011), Nr. 1, pp. 37–63

[PTH+15] Patil, M. A. ; Tagade, P. ; Hariharan, K. S. ; Kolake, S. M. ; Song, T. ; Yeo, T. ; Doo, S.: A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation. In: *Applied Energy* 159 (2015), pp. 285–297

[QLH12] Qin, H. ; Liu, J. ; Hong, T.: An eye state identification method based on the Embedded Hidden Markov Model. In: *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, 2012

[Rab89] Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. In: *Proceedings of the IEEE* 77 (1989), pp. 257–286

[Ras83] Rasmussen, J.: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13 (1983), pp. 257–266

[RBEL91] Rumbaugh, J. ; Blaha, W. ; Eddy, F. ; Lorensen, W. E.: *Object-oriented modeling and design.* Bd. 199. Prentice-hall Englewood Cliffs, NJ, 1991

[RL23] Rossander, M. ; Lideskog, H.: Design and Implementation of a Control System for an Autonomous Reforestation Machine Using Finite State Machines. In: *Forests, MDPI* 14 (2023)

[RS16] Rothe, S. ; Söffker, D.: Comparison of different information fusion methods using ensemble selection considering benchmark data. In: *2016 19th International Conference on Information Fusion (FUSION)* IEEE, 2016, pp. 73–78

[SDBD12] Smith, A.J. ; Dahn, H. M. ; Burns, J.C. ; Dahn, J.R.: Long-term low-rate cycling of LiCoO2/graphite Li-ion cells at 55 C. In: *Journal of The Electrochemical Society* 159 (2012), pp. A705

[SG07] Saha, B. ; Goebel, K.: Battery Data Set. In: *NASA Prognostics Data Repository*, 2007

[SG09] Saha, B. ; Goebel, K.: Modeling Li-Ion battery capacity depletion in a particle filtering framework. In: *Annual Conference of the PHM Society*, 2009

[SL02] Salvucci, D. D. ; Liu, A.: The time course of a lane change: Driver control and eye-movement behavior. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 5 (2002), pp. 123–132

[SLY+17] Sun, W. ; Lin, A. ; Yu, H. ; Liang, Qi. ; Wu, G.: All-dimension neighborhood based particle swarm optimization with randomly selected neighbors. In: *Information Sciences* 405 (2017), pp. 141–156

[SLZ+11] Shiwu, L. ; Linhong, W. ; Zhifa, Y. ; Bingkui, J. ; Feiyan, Q. ; Zhongkai, Y.: An active driver fatigue identification technique using multiple physiological features. In: *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 2011, pp. 733–737

[SMD+18] Su, S. ; Mülling, K. ; Dolan, J. ; Palanisamy, P. ; Mudalige, P.: Learning Vehicle Cooperative Lane-changing Behavior from Observed Trajectories in the NGSIM Dataset, 2018

[SR17] Söffker, D. ; Rothe, S.: New Approaches for Supervision of Systems with Sliding Wear: Fundamental Problems and Experimental Results Using Different Approaches. In: *Applied Sciences* 7 (2017)

[Sta22] Statistische Bundesamt: Driver-related causes of accidents involving personal injury in road traffic. In: *GENESIS-Online database*, 2022

[SWWB15] Schlechtriemen, J. ; Wirthmueller, F. ; Wedel, A. ; Breuel, K. D.: When will it change the lane? A probabilistic regression approach for rarely occurring events. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1373–1379

[SZJZ18] Sun, J. ; Zuo, K. ; Jiang, S. ; Zheng, Z.: Modeling and Predicting Stochastic Merging Behaviors at Freeway On-Ramp Bottlenecks. In: *Journal of Advanced Transportation* 2018 (2018)

[Tha22] Thakur, P. R.: *Optimization of a dSPACE/Simulink powered Test Rig Design evaluating Battery Parameters for accelerated Aging, Masters Thesis.* Chair of Dynamics and Control, University of Duisburg-Essen, 2022

[TSLL15] Tran, D. ; Sheng, W. ; Liu, L. ; Liu, M.: A Hidden Markov Model based driver intention prediction system. In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2015, pp. 115–120

[TTK08] Thiemann, C. ; Treiber, M. ; Kesting, A.: Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data. In: *Transportation Research Record* 2088 (2008), pp. 90–101

[TVT11] Tomar, R. S. ; Verma, S. ; Tomar, G. S.: SVM Based Trajectory Predictions of Lane Changing Vehicles. In: *2011 International Conference on Computational Intelligence and Communication Networks*, 2011, pp. 716–721

[UM15] Ulbrich, S. ; Maurer, M.: Situation Assessment in Tactical Lane Change Behavior Planning for Automated Vehicles. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 975–981

[Ven05] Venturini, M.: Simulation of Compressor Transient Behavior Through Recurrent Neural Network Models. In: *Journal of Turbomachinery* 128 (2005), Nr. 3, pp. 444–454

[VLMT13] Van Ly, M. ; Martin, S. ; Trivedi, M. M.: Driver classification and driving style recognition using inertial sensors. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1040–1045

[VTMB20] Vetturi, D. ; Tiboni, M. ; Maternini, G. ; Bonera, M.: Use of eye tracking device to evaluate the driver's behaviour and the infrastructures quality in relation to road safety. In: *Transportation Research Procedia* 45 (2020), pp. 587–595

[WCZ+19] Wu, J. ; Chen, X. Y. ; Zhang, Hao ; Xiong, L. D. ; Lei, H. ; Deng, S. H.: Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. In: *Journal of Electronic Science and Technology* 17 (2019), pp. 26–40

[WKC+22] Wu, J. ; Kong, L. ; Cheng, Z. ; Yang, Y. ; Zuo, H.: RuL Prediction for Lithium Batteries Using a Novel Ensemble Learning Method. In: *Energy Reports* 8 (2022), pp. 313–326. – 2022 International Conference on the Energy Internet and Energy Interactive Technology

[WLB+19] Wu, Y. ; Liu, L. ; Bae, J. ; Chow, K. H. ; Iyengar, A. ; Pu, C. ; Wei, W. ; Yu, L. ; Zhang, Q.: Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks. In: *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1971–1980

[WLHG+11] Wang, J. ; Liu, P. ; Hicks-Garner, J. ; Sherman, E. ; Soukiazian, S. ; Verbrugge, M. ; Tataria, H. ; Musser, J. ; Finamore, P.: Cycle-life model for graphite-LiFePO 4 cells. In: *Lancet* 196 (2011), pp. 3942–3948

[WM97] Wolpert, D.H. ; Macready, W.G.: No free lunch theorems for optimization. In: *IEEE Transactions on Evolutionary Computation* 1 (1997), pp. 67–82

[WM13] Wilson, P. ; Mantooth, H. A.: *Chapter 6 - Block Diagram Modeling and System Analysis*. Oxford : Newnes, 2013. – 169–196 S

[WMK16] Wang, X. ; Murphey, Y. L. ; Kochhar, D. S.: MTS-DeepNet for lane change prediction. In: *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 4571–4578

[Wor18] World Health Organization: Global Status Report on Road Safety 2018. In: *WHO Library Cataloguing-in-Publication Data*, 2018

[WTG+17] Wissing, C. ; T.Nattermann ; Glander, K. H. ; Hass, C. ; Bertram, T.: Lane Change Prediction by Combining Movement and Situation based Probabilities. In: *IFAC-PapersOnLine* 50 (2017), pp. 3554–3559

[WTZK16] Weidl, A. L. ; Tereshchenko, V ; Zhang, S. ; Kasper, D.: Situation awareness and early recognition of traffic maneuvers. In: *EUROSIM Congress on Modelling and Simulation*, 2016

[WWH+22] Wang, H. ; Wang, X. ; Han, J. ; Xiang, H. ; Li, H. ; Zhang, Y. ; Li, S.: A Recognition Method of Aggressive Driving Behavior Based on Ensemble Learning. In: *Sensors* 22 (2022)

[WWK+14] Waldmann, T. ; Wilka, M. ; Kasper, M. ; Fleischhammer, M. ; Wohlfahrt-Mehrens, M.: Temperature dependent ageing mechanisms in Lithium-ion batteries – A Post-Mortem study. In: *Journal of Power Sources* 262 (2014), pp. 129–135

[WWZ19] Wu, W. ; Wang, Y. ; Zhang, K.: Remaining Useful Life Prediction of Lithium-Ion Batteries Using Neural Network and Bat-Based Particle Filter. In: *IEEE Access* 7 (2019), pp. 54843–54854

[WY10] Wu, Q. ; Yu, W.: A driver abnormality recondition model based on dynamic Bayesian network for ubiquitous computing. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)* Bd. 1, 2010, pp. 320–324

[WYT+16] Wang, D. ; Yang, F. ; Tsui, K. L. ; Zhou, Q. ; Bae, S. J.: Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Spherical Cubature Particle Filter. In: *IEEE Transactions on Instrumentation and Measurement* 65 (2016), pp. 1282–1291

[WYZT17] Wang, D. ; Yang, F. ; Zhao, Y. ; Tsui, K. L.: Battery remaining useful life prediction at different discharge rates. In: *Microelectronics Reliability* 78 (2017), pp. 212–219

[WZHX18] Wang, W. ; Zhao, D. ; Han, W. ; Xi, J.: A Learning-Based Approach for Lane Departure Warning Systems With a Personalized Driver Model. In: *IEEE Transactions on Vehicular Technology* 67 (2018), pp. 9145–9157

[XC17] Xu, X. ; Chen, N.: A state-space-based prognostics model for lithium-ion battery degradation. In: *Reliability Engineering System Safety* 159 (2017), pp. 47–57

[Xia15] Xiao, Y.: Model-Based Virtual Thermal Sensors for Lithium-Ion Battery in EV Applications. In: *IEEE Transactions on Industrial Electronics* 62 (2015), pp. 3112–3122

[YLS+17] You, F. ; Li, Y. ; Schroeter, R. ; Friedrich, J. ; Wang, J.: Using Eye-Tracking to Help Design HUD-Based Safety Indicators for Lane Changes. New York, NY, USA : Association for Computing Machinery, 2017

[YLW18] Yuan, W. ; Li, Z. ; Wang, C.: Lane-change prediction method for adaptive cruise control system with Hidden Markov Model. In: *Advances in Mechanical Engineering* 10 (2018), pp. 1687814018802932

[YS20] Yang, L. ; Shami, A.: On hyperparameter optimization of machine learning algorithms: Theory and practice. In: *Neurocomputing* 415 (2020), pp. 295–316

[ZWL15] Zhang, L. ; Wu, X. ; Luo, D.: Human activity recognition with HMM-DNN model. In: *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)* (2015), pp. 192–197

[ZWWC21] Zhang, H. ; Wu, D. ; Wang, Z. ; Chen, Y.: An Ensemble Method for the Heterogeneous Neural Network to Predict the Remaining Useful Life of Lithium-ion Battery. In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* IEEE, 2021, pp. 2433–2438

[ZXX21] Zhang, Q. ; Xiao, J. ; Xi, X.: Estimation of Vehicle Longitudinal Speed Based on Improved Kalman Filter. In: *Journal of Physics: Conference Series* 2113 (2021), pp. 012011

[ZZW22] Zhao, S. ; Zhang, C. ; Wang, Y.: Lithium-ion battery capacity and remaining useful life prediction using board learning system and long short-term memory neural network. In: *Journal of Energy Storage* 52 (2022), pp. 104901

[ZZZ+16] Zheng, L. ; Zhang, L. ; Zhu, J. ; Wang, G. ; Jiang, J.: Co-estimation of state-of-charge, capacity and resistance for lithium-Ion batteries based on a high-fidelity electrochemical model. In: *Applied Energy* 180 (2016), pp. 424–434

This thesis is based on the results and development steps presented in the following previous publications.

Journal articles

[DRS20] David, R.; Rothe, S.; Söffker, D.: State Machine approach for lane changing driving behavior recognition. Automation, MDPI, Vol. 1, 2020, pp. 68-79.

[DS22b] David, R.; Söffker, D.: A study on a HMM-based state machine approach for lane changing behavior recognition. IEEE Access, Vol. 10, 2022, pp 122954-122964.

[DS23c] David, R.; Söffker, D.: A review on Machine Learning-based models for lane changing behavior prediction and recognition. Frontiers in Future Transportation, Vol. 4, 2023.

Conference papers

[DRS21a] David, R.; Rothe, S.; Söffker, D.: Lane changing behavior recognition based on Artificial Neural Network-based State Machine approach. 2021 IEEE International Conference on Intelligent Transportation (ITSC), Indianapolis, USA, 2021, pp. 3444-3449.

[DS22a] David, R.; Söffker, D.: Effect of environmental and eye-tracking information: An Artificial Neural Network-based state machine approach for human driver intention recognition. The 12th IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), Salerno, Italy, 2022, pp. 16-22.

[DS23a] David, R.; Söffker, D.: Comparison of Different Hyperparameter Optimization Methods on Driving Behavior Recognition. Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), 2023, Philadelphia, USA, 2023, accepted.

[DS23b] David, R.; Söffker, D.: A modified Hidden Markov Model (HMM)-based state machine model for driving behavior recognition: Effectiveness of features using different sub-HMMs. Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), 2023, Philadelphia, USA, 2023, accepted.

[DS23d] David, R.; Söffker, D.: A State Machine-Based Approach for Estimating the Capacity Loss of Lithium-Ion Batteries. Annual Conference of the PHM Society, Salt lake city, USA, vol 15, 2023 .

The following student theses have been supervised by Ruth David and Univ.-Prof. Dr.-Ing. Dirk Söffker, which are not included in this thesis:

[NI21] Ni, C., Implementation, experimental test, and improvement of a professional research driving simulator including human integration and data acquisition, Master Thesis, January 2021

[AND21] Andoni, D., Time complexity of different machine learning algorithms used for driving behaviour prediction/recognition, Bachelor Thesis, April 2021

[RAJ22] Raj.R, Fundamentals of different driving styles and the associated existing algorithms, Bachelor Thesis, March 2022

[KLU22] Kluge, A., Fundamentals of different driving styles and the associated existing algorithms, Bachelor Thesis, April 2022

[LU22] Lu, Y., Adjustment and testing of a software for a 3 DOF motion platform of a driving simulator, Master Thesis, November 2022