THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Machine Learning-Based Prediction Models and Threat Detection for Lane-Keeping Assistance

John Dahl

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Machine Learning-Based Prediction Models and Threat Detection for Lane-Keeping Assistance

John Dahl
ISBN 978-91-7905-834-0

Technical Report No. 5300
ISSN 0346-718X

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, May 2023

To my family.

Abstract

Traffic accidents have been an ongoing problem for over a century and many efforts have been made to improve traffic safety. Historically, the focus has been on passive safety with innovations, such as crumpling zones, three-point seat belts, and airbags, that aim to mitigate the impact of collisions. As technology advanced, the focus shifted toward active safety, which aims to avoid accidents.

Advanced driver assistance systems are nowadays utilized in vehicles to support the driver in critical situations where the driver is likely to fail the driving task. The system uses sensor information to estimate the risk of a threatful event, such as an unintended lane departure, and decides whether an automatic avoidance maneuver should be activated. However, from a legal perspective, it is the driver who is responsible for the driving, and consequently, the driver must be able to override an erroneous maneuver. This is an important aspect, as it restricts the system to the use of low-intensity maneuvers. That implies that a maneuver needs to be activated sufficiently early in time to be able to avoid the threatful situation, i.e., a long prediction horizon is needed to detect the threat in time.

The decision to intervene with a supportive automatic avoidance maneuver is based on the output from the threat assessment, which uses a prediction model to estimate how the current traffic situation is evolving with time. Designing a well-functioning prediction model is challenging, as it must deal with multiple sources of uncertainties, such as sensor noise and drivers' intentions, and it becomes even harder as the prediction horizon increases.

This thesis focuses on how machine learning can be used to improve the performance of a lane-keeping assistance system. The goal has been to develop learning-based prediction models that are high-performing, robust, and efficient to compute in real time. The approach has been to evaluate the performance of linear and non-linear regression models using real-world data. The results show that both linear and non-linear prediction models are significantly better than a kinematic model. It also shows that linear prediction models are nearly as good as non-linear models, especially for shorter prediction horizons. However, the linear model is significantly easier to compute in real time and may therefore be a sufficient alternative for applications where computational power is restricted. Moreover, the robustness towards anomalies and samples that are out of the operational design domain can be improved by utilizing uncertainty-aware prediction models.

Keywords: Threat assessment, Threat detection, Machine learning, ADAS.

List of Publications

This thesis is based on the following publications:

[A] **John Dahl**, Gabriel Rodrigues de Campos, Claes Olsson Jonas Fredriksson, "Collision Avoidance: A Literature Review on Threat-Assessment Techniques". IEEE Transactions on Intelligent Vehicles (T-IV), 2019.

[B] **John Dahl**, Rasmus Jonsson, Anton Kollmats, Gabriel Rodrigues de Campos, Jonas Fredriksson, "Automotive Safety: a Neural Network Approach for Lane Departure Detection using Real World Driving Data". IEEE Intelligent Transportation Systems Conference (ITSC), 2019.

[C] **John Dahl**, Gabriel Rodrigues de Campos, Jonas Fredriksson, "A Path Prediction Model based on Multiple Time Series Analysis Tools used to Detect Unintended Lane Departures". IEEE Intelligent Transportation Systems Conference (ITSC), 2020.

[D] **John Dahl**, Gabriel Rodrigues de Campos, Jonas Fredriksson, "Performance and Efficiency Analysis of a Linear Learning-Based Prediction Model used for Unintended Lane-Departure Detection". IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2022.

[E] **John Dahl**, Gabriel Rodrigues de Campos, Jonas Fredriksson, "Prediction-Uncertainty-Aware Threat Detection for ADAS: A Case Study on Lane-Keeping Assistance". IEEE Transactions on Intelligent Vehicles (T-IV), 2023.

[F] **John Dahl**, Gabriel Rodrigues de Campos, Jonas Fredriksson, "Intention-Aware Lane Keeping Assist Using Driver-Gaze Information". Accepted to the IEEE Intelligent Vehicles Conference (IV), 2023.

Other publications by the author, not included in this thesis, are:

[G] Emil Klintberg, **John Dahl**, Jonas Fredriksson, Sebastien Gros, "An improved dual Newton strategy for scenario-tree MPC". IEEE Conference on Decision and Control (CDC), 2016.

Acknowledgments

Pursuing a PhD has in many aspects been like a trip on a rollercoaster. It begins in the dark, where it is completely unclear what is going to happen next. Suddenly, there is a loud bang and an insane acceleration, and a moment later, you find yourself in a position where everything is about teaching, studying, and writing. Once you think it has started to settle, you enter the first vertical drop and realize that your reviewers are not as amused as you about the precious first submitted paper. After some mild turns back and forth, you are happy to conclude that it could have been a lot worse and that the process contributed to a better paper. Then it strikes you, it is not over, this is only the introductory part of the track. The ride goes on and novel ideas crash and burn along what seems to be a spiral of inversions, but every turn in the loop makes you better prepared for the next and slowly you start to enjoy it. When the track straightens out, and you finally can breathe again, you realize a lot of important insights and results have been made. The feeling of an urgent need for pulling the emergency brake and going home is replaced by a proudness in what you have learned and achieved since the beginning. From nowhere, the ride slams the brakes and you realize it is over. It is with a good feeling you leave your seat, and even if it was fun, it is with a little hesitation that you would like to go for a second ride.

I am deeply thankful for the opportunity given by Volvo Cars, Zenuity, and Zenseact to let me take on the pursuit of an industrially sponsored PhD study. I am forever thankful to my industrial supervisor Claes Olsson, who initiated the research project and took me on board. His sense of technical details, expert knowledge of ADAS, and commitment to the project goals have contributed substantially to my personal development as a PhD student. I also want to express my gratitude to Gabriel de Campos, who joined the project as an industrial co-supervisor. He has spent endless hours discussing every small and big matter of being a PhD student and has motivated and prominently guided me to improve my writing and communication skills, regardless if it is during the day, night, weekend, or vacation time. It has been a pleasure to work with my academic supervisor Jonas Fredriksson, who always had new ideas to expand and improve my research. Thank you for being supportive and positive, regardless if we have faced a moment of success or failure. Being a PhD student would have been boring and lonesome without my AGP colleagues at Zenseact, and I would especially thank my team manager Mats Nordlund and team coach Carl Lindberg for supporting me in this journey. I also want to thank my former and current colleagues at Chalmers for all the interesting lunch-brake discussions, whatever the subjects have been.

I want to thank my mom for inspiring me already at an early age to critically thinking and to investigate how things actually work. It turned out that the big pile of "for dummies" books you brought to me as a kid was a good investment. I also want to thank my grandparents, who taught me the importance of hard work to reach goals, reflection, and enjoy the soft and small things in life.

I am extremely grateful for the endless support given by my family. This thesis would not have been possible without my beloved wife Anna, who is an endless source of inspiration, motivation, and positive thinking. Her ability to withstand my obsession with discussing comments from anonymous reviewers and non-converging training of neural networks is mind-blowing. I am also happy for my children that has brought joy and an extra essence of purpose to our lives. Elise and Alvin, you are the best thing that ever happened to me, and I am so proud of you!

Acronyms

ADAS:	Advanced driver assistance system
AEB:	Automatic emergency braking
ANN:	Artificial neural network
BNN:	Bayesian neural network
DM:	Decision making
DMS:	Driver monitoring system
LKA:	Lane-leeping assistance
MTT:	Mean triggering time
ODD:	Operational design domain
TA:	Threat assessment
TTC:	Time to collision

Contents

Part I

CHAPTER 1

Introduction

Transportation of passengers and goods on the road network is nowadays one of the core pillars of modern society. Unfortunately, transportation is also associated with severe accidents, which cause 1.35 million fatalities every year [1]. The situation is worst in low-income countries, where traffic accidents were the 7:th most common cause of death in the year 2019 [2]. However, road traffic fatalities are a global problem and there is a need for safer vehicles that can avoid traffic accidents.

Historically, the work on improving traffic safety has been focused on passive safety, aiming to minimize driver and passenger injuries in collisions. Noticeable innovations, among many others, are the three-point seat belt [3], crumpling zones [4], automatically inflated airbags [5], and whiplash protection system [3], which have been updated and improved continuously as the technology advances. As the electronics became more capable in terms of computational power [6], the focus shifted towards active mitigation systems, such as the anti-lock braking system [7], followed by the electronic stability control (ESC) system that aimed to maintain traction of the wheels to avoid skidding [8].

When cameras and radar became available for automotive applications, the systems evolved into predictive systems, i.e., active safety systems, that use automatic maneuvers or warn drivers to avoid collisions. One of the first predictive systems was Volvo's automatic emergency braking (AEB) system that was introduced in the year 2008 [3], which applies the brakes if a collision is likely to happen. Nowadays, there are numerous active safety systems in commercial vehicles, such as lane-keeping assistance (LKA), parking aid, blind spot information system, cyclist collision avoidance, etc. The mentioned systems are commonly denoted as advanced driver assistance systems (ADAS) in the automotive community.

As technology advances, with better affordable sensors and more computational onboard power, there are opportunities to make even more effective systems. The automotive industry is currently focusing on self-driving vehicles, motivated by making safer vehicles and improving the accessibility of transportation to society [9]. Despite the significant efforts, there are only a few fully automated vehicles on the market, e.g., Waymo One [10], GM's Cruise Ride [11] and Motional [12]. These vehicles are so-called robotaxis that only operate in a restricted area. The main limitation is the safety aspect, which includes that the manufacturer needs to ensure that the vehicle is safe for public use, i.e., that the vehicle can adequately handle driving under various conditions without the supervision of a human driver. The safety aspect has turned out to be a very challenging problem. For example, a recent empirical study showed that Tesla cars using the full self-driving (FSD) beta software, are causing a critical driving error every four minutes when the car is used in autonomous mode [13]. As of today's date, it is expected that fully autonomous vehicles will be introduced on the broader market in the year 2030-2040 [9]. Hence, large efforts are still needed before fully autonomous vehicles is a reality.

1.1 Challenges

There is an important difference in liability between self-driving vehicles and ADAS. A driver that uses ADAS is responsible for the safety of the vehicle, in contrast to self-driving vehicles where the manufacturer is responsible. Hence, it allows automotive manufacturers to deploy effective assistance and safety systems without ensuring the same level of functional safety as for self-driving vehicles. Nevertheless, even if the driver is responsible for the driving, it is still important that the active safety system is consistent and predictable, i.e., works well and according to the expectations, as it otherwise may scare or annoy the driver, or ultimately, set the driver and the vehicle at risk. This is especially important for collision avoidance systems, such as AEB and LKA systems, since these systems are typically using evasive automated maneuvers with relatively high intensity to avoid a potential accident. In particular, this is a problem for steering interventions, since erroneous steering maneuvers may rapidly force the driver into critical situations, at least if the velocity is high. It is therefore crucial that the driver can override any automatic steering maneuver, which is ensured by limiting the steering control effort such that it never exceeds 50 N[14]. This limitation comes with the consequence that avoidance maneuvers by steering have to be triggered early to have enough time to avoid the critical situation. It is also important that hardware and software are designed and implemented with care to make the system robust to signal anomalies, mechanical failure, software errors, etc.

Disregarding the safety aspect mentioned above, there are many similarities between ADAS and self-driving vehicles. Both have to deal with sensor noise and limitations and make predictions on how the traffic situation will evolve in time. A major difference is that the self-driving vehicle has insight into the logic that affects the motion planning and decision-making for its own vehicle, i.e., ego-vehicle, while an ADAS vehicle has to deal with the ego-driver's intent, which is unknown to a great extent.

Statistics show that ADAS such as LKA, forward collision warning (FCW), and AEB can have a huge safety impact [15]. It is estimated that ADAS technologies have the potential to prevent about 62% of total traffic deaths in the US, and LKA accounts for 71% of these savings alone. Another analysis of head-on and single-vehicle crashes in Finland concluded that LKA systems have the potential of avoiding 27% of the crashes [16]. However, statistics also show that 35% of the vehicle users in the US either disable or do not use the LKA system [17]. A reason for disabling LKA is the high rate of false positive interventions or warnings which annoys the driver and makes them turn the system off [18].

A recent study, [19], showed that current ADAS account for 13% of the total industry problems (problems per 100 vehicles) and concluded that an important step towards self-driving technologies is to ensure that the existing ADAS is functioning to the highest degree. Moreover, estimates show that low-speed AEB reduces rear-end crashes by 38% [20] and that Lane departure warnings/LKA systems have an effectiveness of 30% [21] for Volvo vehicles. While being a great achievement, it is troublesome that so many cases are not covered. Hence, it is motivated to understand what are the limiting factors for ADAS and what can be done to solve them.

1.2 Approach

The research project was initiated to investigate how the availability, effectiveness, and robustness of steering-based ADAS could be improved. The project originated from the insight that the current steering-based ADAS was working well, but the utilization was low, i.e., it was disabled/unavailable for a large fraction of the time. To address the problem and ensure that the existing ADAS is functioning to the highest degree, it is important to understand the restricting factors and how they can be limited. An exploratory research approach was used to understand the existing problems. The project's first step was a case study on system limitations, i.e., finding the root cause of the low utilization. A small group of stakeholders, technical experts, and developers within the company were interviewed to share their views on the issue. The study revealed two prominent limitations: 1), the driver's intention was not included in the threat assessment or in the decision-making, and 2), the system was non-robust to noise and anomalies in the sensor signals and there was a strong dependency on the most recent signals from the sensors. Those limitations made the system prone to activate unwanted steering interventions, which caused the drivers to turn off/deactivate the ADAS.

The second step was to study how these limitations had been addressed within the literature. The main conclusion was that including the driver's intention in the threat assessment and decision-making is challenging. For example, some works used a rule-based approach, while others tried to include the driver's intention using a probabilistic approach. However, such approaches rely on simplistic assumptions and tailored models for specific scenarios and are therefore sensitive to anomalies and noise. Another observation was the rapid development of machine learning and its potential to learn complex relationships directly from data. Based on the literature research findings, it was hypothesized that an approach based on machine learning would demonstrate an improved ability to incorporate the driver's intention in the threat assessment and decision-making process and effectively contribute to a system that is resilient to anomalies and noise. However, there was also a concern that a learning-based approach would be challenging to run in real-time, due to the size and complexity of the models. The hypothesis was subsequently verified through the implementation of machine learning-based models utilizing real-world data and rigorous evaluation of various factors such as signal selection, model architecture selection, and robustness.

The evaluation was based on an experimental approach and especially three research questions were in focus:

RQ1: *Can the challenges related to driver intention and sensor issues be addressed by supervised learning in the context of a lane-keeping assistance system?*

RQ2: *What input data is needed to achieve a high system performance?*

RQ3: *How could real-time computation-limitations be considered?*

1.3 Limitations

The objective of this thesis is to investigate threat assessment and decision-making techniques for ADAS applications, with a specific emphasis on LKA. The proposed prediction models rely on machine learning algorithms that utilize time series input data. To evaluate the efficiency of the proposed models and algorithms, an experimental research approach was employed, which leveraged large real-world datasets including numerous relevant scenarios. It is important to note that this study solely focuses on the ego-vehicle driver and does not incorporate other road users in the threat assessment or decision-making process. Additionally, this thesis does not explore any functional safety implications, i.e., ensuring that the system operates safely according to ISO26262 [22], by utilizing machine learning in ADAS. It is also not focusing on sensing, sensor fusion, planning, or actuation.

1.4 Outline

The thesis is organized into Part I and Part II. Part I gives an introduction and starts with a motivation for the research in Chapter 1, followed by Chapter 2 which provides the general concept of ADAS, mainly focusing on the important aspects of threat assessment and decision-making. Chapter 3 presents the principles for the system performance evaluation, including the used data sets and performance aspects. A brief overview of regression models is found in Chapter 4. A summary of the appended papers in Part II is found in Chapter 5, followed by a discussion in Chapter 6 on the main contributions. Finally, Part I is concluded in Chapter 7 with final remarks and proposals for future research directions.

CHAPTER 2

Advanced driver assistance systems

An advanced driver assistance system uses sensor information and algorithms to get an understanding of the traffic situation, which is used to decide whether an intervention is needed to support the driver. Conceptually, the system can be abstracted into a scheme of four modules, as seen in figure 2.1. The sensing and estimation module uses sensors to monitor the vehicle's and the driver's states. This information is then used in the threat assessment module, which uses a prediction model and threat metrics to assess the threat level of the situation. The decision-making module determines when an automatic maneuver is needed, based on the predicted threat level. The decision-making consists mainly of two parts: threat detection and maneuver planning. The actuation module is responsible for the vehicle following the planned trajectory.

In the following, the four modules are introduced to give more insights into the challenges that need to be considered in the design process. However, the main focus is on threat assessment and decision-making, as those are of particular interest in finding answers to the research questions of this thesis. It is followed by a subchapter introducing the LKA system used as an application in this thesis.

2.1 Sensing and estimation

The sensing and estimation module is responsible for measuring the ego-vehicle's motion, contextual information, such as lane markers, and driver behavior. Information from the exterior of the vehicle is measured by the sensor platform, which might consist of an array of various sensors, such as cameras, radars, lidars, and ultrasonic distance sensors [23]–

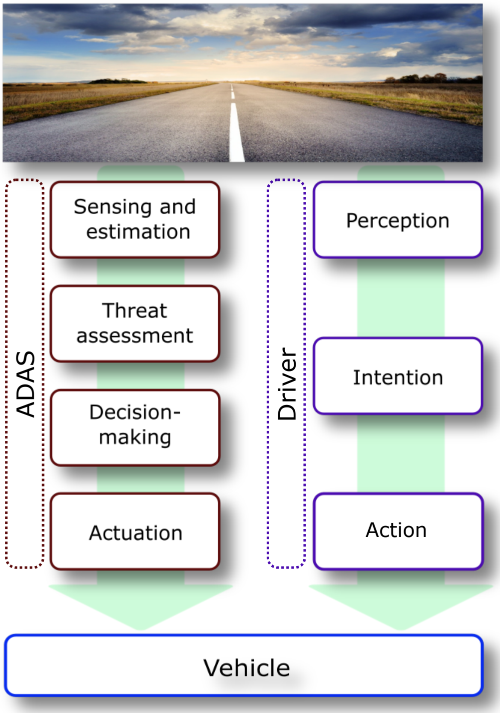

Figure 2.1: An ADAS operates side-by-side with the driver and uses sensors and algorithms to detect when a supportive avoidance maneuver is needed.

[25]. The kinematics of the ego-vehicle, e.g., the speed and the yaw rate, are sampled from interior sensors [26], [27]. The driver state can be estimated using a driver monitoring system that is typically camera-based [28]–[32] but might also rely on the principles of rate variability measurements by electrocardiograph (ECG) [33], brain activity estimation using electroencephalogram (EEG) [34], IR body temperature sensors, etc [35].

The measurements are always affected by noise, to some degree, either due to the design limitations of the sensors, e.g., the pixel density of the camera is too low, or the sensors are affected by inaccuracies in the hardware, e.g., drift in the readings from the gyroscope, or problems with signal integrity caused by non-stationary magnetic fields, for example. Therefore, sensor fusion is used to merge the information from the different sensors to reduce the noise level in the estimated signals [23], [36], [37].

2.2 Threat assessment

The main objective of the threat assessment is to predict how the traffic situation will evolve from the current time instant and to determine the risk or threat level. As one can imagine, there are many ways to end up in a risky situation, and it is therefore hard to anticipate them all. Hence, the threat level is typically abstracted into threat metrics, where each metric assesses the threat level for a specific event, e.g., an unintended lane departure or a collision with a pedestrian. What they have in common is that they rely on the future motion of the ego-vehicle and most often other road users, which is predicted by the prediction model.

Prediction models

Prediction models for threat assessment have been a research topic for several decades and have evolved from simple kinematic models to uncertainty-aware machine learning-based models [38]. As mentioned above, a prediction model is used to predict the traffic situation in time and space. There are many ways to design a prediction model, and several considerations must be made. The general predictive performance depends on the versatility of the model, i.e., a complex model that can deal with uncertainties and outliers tend to work better than a basic model designed for strict requirements. A closely related aspect is the real-time performance of the deployed model, which is determined by the computational complexity of the model. The goal of any ADAS is to use a prediction model with a high prediction performance and good real-time capabilities. However, in practice, those two are typically hard to achieve simultaneously.

The deterministic kinematic motion of an object is determined by its initial condition, e.g., position, heading, speed, etc, and Newton's laws of motion [39]. A kinematic model resembles the relationship between the current and future state, which tends to give simple mathematical formulas that are extremely efficient to compute in real time. A *kinematic prediction model* can be defined in several ways, depending on the assumptions regarding the future motion. The simplest model for longitudinal and lateral motion is the constant velocity (CV) model [40], which assumes that the longitudinal v_t^{lon} and lateral v_t^{lat} velocities are constant over the prediction horizon h,

$$x_{t+h} = x_t + v_t^{lon} h \tag{2.1}$$

$$y_{t+h} = y_t + v_t^{lat} h, \tag{2.2}$$

causing the vehicle's motion to follow a straight line with a fixed heading. This is often a reasonable assumption in situations where the road is fairly straight. Similarly, the constant turn rate and velocity (CTRV) model assumes that the vehicle's motion is following an arc with constant radius and velocity [40]. Naturally, this is a convenient assumption for curvy roads. Another popular version is the constant turn rate and acceleration (CTRA) model [41], [42], where the velocity is varying with time, based on the assumption of constant acceleration. Moreover, the model can be tailored for certain objects, such as the kinematic bicycle model [43].

In practice, kinematic models are very accurate, as long as the fundamental assumptions hold. However, the assumptions are typically only fulfilled for short prediction horizons, as the short-term trajectory is determined by the inertia of the moving vehicle rather than the driver's actions. However, for longer prediction horizons, i.e., several seconds of prediction, the discrepancy between the assumed action and the actual action might significantly reduce the accuracy of the prediction model. Moreover, since the predictions are solely based on the current state, they tend to be very sensitive to sensor noise, which is not accounted for in the model.

The kinematic approach can be expanded by including knowledge of the uncertainties in the model, instead of neglecting them, as is the case of deterministic kinematic modeling. The most straightforward way is to assume that some variables in the kinematic model are random, i.e., the velocity or acceleration, with given probabilistic distributions. Similarly, the model's inputs, e.g., the driver's actions, can be modeled as probabilistic variables, with probabilities assigned to each action [44]–[46]. The *probabilistic prediction models* can be used to estimate the likelihood of a future event, i.e., a collision or lane departure, or to predict the future trajectory with corresponding variance. The parameters of the probability distributions can either be estimated from data or be manually designed based on engineering principles and in-depth knowledge.

A generalization of the above-mentioned approach is to assume that the set of actions and other circumstances leading to a critical event can be modeled. For example, the probability of a collision may depend on the observability of the threat, the driver's action, and the vehicle state, i.e., the probability of a collision depends on factors that are to some degree uncertain. This idea can be extended further to also consider temporal dependencies, which means that the probability of collision may depend also on factors from the past. These dependencies can be structured using Bayesian networks [47], [48], which are based on directed acyclic graphs, where nodes represent random variables and edge the conditional dependencies between random variables. However, calculating the posterior distribution over the output of the Bayesian network is not trivial in the general case. It is therefore common to use approximate numerical solutions, such as Markov chain Monte Carlo (MCMC) sampling and importance sampling. In ADAS/AD applications, the network structure is manually deduced from analytic reasoning. The corresponding conditional dependencies are typically assumed to be known, but the distribution parameters need to be learned from data or assigned manually. Hence, the framework of Bayesian networks provides a way to include known information and gives a clear overview of the dependencies in the model. On the other hand, as with every handcrafted model, the underlying assumptions of the system must hold.

The drawback of kinematic and probabilistic models, i.e., that they rely on a set of assumptions, can be overcome by instead using a *learning-based approach*, that can learn the properties directly from the driving data [49]–[53]. For example, a learning-based prediction model can be trained to predict the future trajectory of the ego-vehicle without relying on any modeling assumptions regarding the driver's intention. The learning is typically supervised, i.e., the target output of the model is annotated before the training starts. The annotation process might be time-consuming, especially for computer vision tasks, as it may involve manual annotation from humans. However, the annotation is self-supervised for time-series data, as the target of the predicted output, e.g., the future trajectory, can be accessed directly by sampling forward values in the series. The predictive performance is largely determined by the quality and quantity of the training data and the complexity of the model architecture [54]. However, these models often have poor real-time performance due to their high computational demands, and therefore it is important to implement such prediction models with great care to avoid unnecessary computations. One major shortcoming with learning-based models is that it can be difficult to encode common knowledge into the models, meaning that even simple concepts may need to be learned from the training data. As a result, both data and computations are required to learn what may already be known. Additionally, training large models can pose challenges such as vanishing gradients, slow convergence, and overfitting.

Threat metrics

The prediction model only predicts the future motion of the road participants, but it does not relate it to risk level. This is done using a threat metric, which is typically expressed in a certain domain, e.g., time, spatial, or acceleration.

One of the most common threat metrics is the time-to-collision (TTC) [55]–[57], which computes the time until a collision between the ego-vehicle and another object occurs, i.e., a low TTC translates into an imminent collision in the future. The time-to metric is generic and can easily be redefined for a specific threat, such as the time-to-lane-change (TLC) [58]. Other time-based metrics are time-to-brake (TTB) and time-to-steer (TTS) [59] or time-to-react (TTR) [59], [60].

The spatial domain metrics determine the treat level as a function of the relative distance towards other objects. For example, minimal safe distance (MSD) [61] reflects the spatial distance between the ego and a surrounding vehicle. A similar metric is the distance-to-lane-marker (DTLM), also referred to as the distance-to-line-crossing (DTLC) by Euro-NCAP [62], which determines the future distance between the side of the ego-vehicle and the lane marker, where the threat level increases as the distance approaches zero.

Acceleration domain metrics determine the acceleration required to avoid a threat. Common metrics are steer-threat-number (STN) and brake-threat-number (BTN) [63], which compute the acceleration needed in the lateral and longitudinal direction, respectively, to avoid an object.

The above-mentioned threat metrics are continuous, in the sense that the risk level is reflected in the metric as a real number. There exists also Boolean threat metrics, which are true if a certain event is happening within a given time in the future, or false otherwise, see e.g. [64], [65].

Naturally, different metrics, regardless of their domain, can be used in combination to cover several threats simultaneously.

2.3 Decision-making

The decision-making for ADAS is event-driven, in contrast to comfort systems that typically decide a low-level control action at every time instance. This means that the system updates the threat metric continuously, but only decides to intervene if a threat-full event is imminent.

As introduced in the previous subchapter, the threat assessment tries to identify predefined events that are non-safe, based on sensor information up to the current time instant. The outcome of the threat assessment module is a threat level that reflects how close the vehicle is to end up in a specific event. The decision to intervene is determined by a threat-detection logic function, which has a Boolean output, where a decision to intervene corresponds to a value of 1 (True), or 0 (False) otherwise. The threat-detection logic function can be implemented in many ways, depending on what threat metrics are used.

For Boolean threat metrics, the detection logic function is given by the threat metric directly. However, most threat metrics are, as mentioned earlier, expressed in real numbers, which makes it reasonable to intervene when the threat metric is violating a predefined threshold. The threshold-based detection logic can easily be extended by adding more conditions that need to be fulfilled, e.g., by adding more threat metrics or/and quality metrics (e.g., confidence in the predictions), which makes it possible to be more certain before an intervention is triggered. The downside of adding conditions is that the number of thresholding constants also increases, which increases the computational burden in the system performance optimization, where proper values for the thresholds need to be found.

The output of the decision-making module, besides the request for activation of a steering intervention from the threat detection, is a reference trajectory. The reference trajectory can be planned using, for example, a potential field approach combined with model predictive control (MPC) [66], reinforcement learning [67]–[69], graph optimization [70], or spline optimization [71]. See [72] for a comprehensive review of planning algorithms.

2.4 Actuation

The last module is responsible for the low-level control of the vehicle, which involves ensuring that the reference trajectory is followed if there is a request for an intervention from the decision-making module. The control can be based on, for example, optimal control, such as model predictive control [66], [73], [74] or MPC with fuzzy logic [75]. There are also examples of end-to-end learning approaches [76], [77]. The steering/braking is typically actuated using an electric motor in a closed-loop operation. It is of great importance from a safety perspective that the motor and electronics are designed to be tolerant towards faults and failures to ensure that the driver has enough time and force to counteract any erroneous intervention.

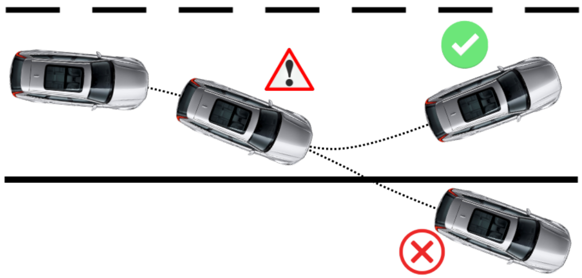

Figure 2.2: An LKA system supports the driver with a corrective steering maneuver if the driver is at risk of unintentionally departing from the lane.

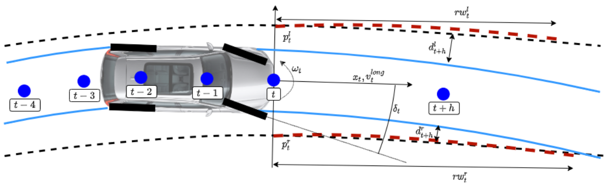

Figure 2.3: Overview of the used signals.

2.5 Lane-keeping assistance

A lane-keeping assistance system aims to ensure that the driver is not unintentionally departing from the lane by supporting the driver with an automatic steering intervention, see figure 2.2 for an illustration. As stated earlier, it is crucial that a supportive steering intervention is activated early enough to allow an effective avoidance maneuver. As this motivates long prediction horizons (typically a few seconds long), this implies that the prediction model should take the driver's intention into consideration to maintain a high predictive performance. It is also important that the system is robust against anomalies and sensor noise to avoid falsely triggered interventions, as it might otherwise compromise the driving experience, and ultimately, make the driver deactivate the system.

The sensor platform used in this thesis consists of a front-looking camera and in-vehicle sensors. The used signals are illustrated in figure 2.3, where v is the longitudinal velocity, δ is the steering angle of the front wheel, and ω is the yaw rate. The software of the front-looking camera is estimating the road geometry and lane markers as two third-order polynomials, one for each side:

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \tag{2.3}$$

which are valid for a range of view rw for the respective side.

Additionally, there is also a camera-based driver monitoring system, which tracks the

Figure 2.4: Driver-gaze target areas of the interior of the vehicle.

driver's eye-gaze direction. For the sake of convenience, the gaze direction is contextualized by abstracting it into gaze target (gt) areas of the interior of the vehicle, as seen in figure 2.4. Finally, each gaze target is assigned a probability by the monitoring system, that reflects the likelihood that the driver is looking into the corresponding area.

CHAPTER 3

Performance evaluation

The thesis relies on an experimental approach and it is therefore important to understand the underlying principles of the performance evaluation. This chapter introduces the data sets and important aspects of system evaluation.

3.1 Data sets

This thesis is based on two field test data sets: A), a data set that was collected in the year 2013 by Volvo Cars (3150 hours of driving), and B), a data set from the year 2018 by Zenuity (5300 hours of driving). The data was collected using fleets of Volvo vehicles, with professional drivers (contracted by the respective company to drive the cars), mainly in the EU, but also in the United States of America and parts of Asia. The drivers were to a high degree young men, that had received driving training before participating in the field test. The vehicles sampled information using their respective sensor platform, including, among others, a front-looking camera and internal sensors used to measure the kinematics of the motion. All ADAS and AD functionality was deactivated during the entire field test.

3.2 Evaluation of system performance

The interpretability of the performance results is closely related to the fairness of the implementation and evaluation of the systems. This subchapter introduces insights on how to set up a fair comparison of methods.

Operational design domain

The first step is to determine an operational design domain (ODD) that defines the scenario for which the system is intended to work in, in this case the LKA system. This includes contextual requirements that need to be fulfilled, such as visible lane markers, minimum and maximum velocities, road types, road curvature, etc. The purpose of the ODD is to limit the scope of the functionality, to make it easier to make proper design choices in the implementation of the system, e.g., choose threat metrics and extract relevant training data. It is also useful in the evaluation of the system's nominal performance, to ensure that the test cases reflect the intended functionality. Furthermore, it can be used for robustness evaluation, where the system's ability to deal with out-of-ODD samples is tested.

Data preparation

The training and performance evaluation is directly dependent on the used data set. There are many ways to split the data into training and evaluation data. A common approach is to use bootstrapping, especially for small data sets, where data sets are created by sampling with replacement from the original data set [78]. There is also the hold-out validation, where the original data set is divided randomly into a training and an evaluation data set with a given ratio [54]. This method is straightforward to use and is preferably used when the number of samples is large in the original data set. The same principle is used in k-fold cross-validation, but here is the data divided into k-groups of the same size. The model is evaluated using one group while being trained on the others. The groups are iterated until all groups have been evaluated. The final evaluation is the average results of all groups. This method is widely used, but it may suffer from increased computational demand and increased variance in the predictions [79].

In terms of fairness, regardless of the splitting technique that is in use, it is important that all compared models are treated the same, i.e., they are trained and evaluated on the same data sets.

Performance metrics

As described in subchapter 2.3, the event-driven threat detection logic is binary; either the system decides to intervene or waits and continues with the threat assessment, where the correct decision is determined by whether the situation is threat-full, or not. As a consequence, in terms of performance, there are four possible outcomes of a decision, which are the elements of the confusion matrix. A true positive (TP) corresponds to that the system decides to trigger an intervention and that the intervention is truly needed. A too trigger-happy system is likely to make decisions that are false positive (FP), which means that the system is triggering an unwanted intervention. Moreover, a true negative (TN) means that the system decides not to intervene in a situation where intervention is truly not wanted. A too-conservative threat-detection logic is prone to make decisions that are false negative (FN), i.e., the system chooses not to intervene in a situation where an intervention is needed.

The performance of the LKA system is evaluated using a validation data set, that contains both positive cases (events), and negative cases (no events). For each decision, one of the four elements in the confusion matrix is updated, depending on the correctness and situation. However, it is challenging to draw conclusions directly from the matrix, since they are not normalized by the size of the evaluation data set. Instead, it is common to use performance metrics that are compounds of the confusion matrix elements, where the true positive rate (TPR) (also referred to as *recall*):

$$TPR = \frac{TP}{TP + FN} \tag{3.1}$$

and false positive rate (FPR):

$$FPR = \frac{FP}{FP + TN} \tag{3.2}$$

are among the most widely used. They benefit from their simplistic formulation, which makes them easy to interpret, at least individually, where a high-performing system is expected to have a high TPR and a low FPR. However, it is not obvious how to compare the performances of two systems using the TPR and FPR metrics. For example, it is not straightforward to conclude whether a system with $TPR = 0.9$ and $FPR = 0.1$ is equally good as a system with $TPR = 0.95$ and $FPR = 0.15$, as it depends on how the metrics are weighted. One way to ease the analysis is to use the positive likelihood ratio:

$$PLR = \frac{TPR}{FPR}, \tag{3.3}$$

which is high (unbounded) for a good system performance or low (not lower than 0) for a poor system performance. Another approach is to consider the F1-score [80]:

$$F1 = \frac{2TP}{2TP + TN + FP}, \tag{3.4}$$

which is a weighted combination of TP, FP, and TN. However, the F1-score is sensitive towards imbalances in the validation data set, i.e., the results, for a given system, may vary depending on the ratio of the number of positive and negative samples in the data set. Therefore, it is also common to use the Matthews correlation coefficient:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{3.5}$$

which considers all the confusion matrix elements. This leads to a metric that is less sensitive to imbalances in the data [81].

Acceptance time window

If the driver support system is well designed and correctly implemented, the majority of the decisions will result in an average activation timing that is close to the target, i.e., the prediction horizon. However, there is always a presence of variations, due to uncertainties, such as sensor noise and underlying shortcomings in the threat assessment, that will give rise to a spread of the predictions. It is therefore reasonable to accept system activations as a TP if the timing of the decision is fairly close to the nominal target. Hence, it is convenient to define an acceptance time window that has lower and upper limits on how much the timing of the intervention can deviate from the nominal performance, and still be classified as correct.

Performance calibration

A fair comparison between LKA systems can only be achieved if they are properly calibrated, as there might be a bias in the prediction models that affects the timing of the intervention. For example, the training of a learning-based model is likely to end up in local minima, which can cause one prediction model to be slightly more trigger-happy than an identical model that has another set of trained weights. It can also be a bias in the data itself, e.g., from a poorly calibrated sensor. Hence, even systems based on kinematic prediction models need to be calibrated.

The calibration process optimizes the thresholds of the threat-detection logic using calibration data (that is mutually exclusive from the training and validation data), such that the mean triggering time of the system is equal to the target.

CHAPTER 4

Brief introduction on supervised learning

The goal of supervised learning is to model the relationship between inputs and outputs from labeled data points that are annotated by a supervisor, such as a human expert or a supervisory algorithm. The learning process (also referred to as training, estimating, or fitting depending on the context) is typically formulated as an optimization problem, with the objective of finding optimal values for the model parameters (coefficients, weights) given a training dataset. The problem is historically referred to as a regression problem, and many modeling approaches are available to solve it. This chapter briefly introduces common methods for regression, ranging from simple linear regression to uncertainty-aware regression based on non-linear artificial neural network models.

4.1 Regression analysis

Informally, regression analysis is a statistical approach to finding the "least bad" model that estimates the relationship between the outputs (dependent variables) and the inputs (independent variables), where the true relationship is typically determined by an underlying stochastic process with unknown properties. The general approach is to collect observations from the process into a training data set, consisting of sample pairs, i.e., input-output pairs, which are used to estimate the parameters of the model, i.e, train or fit the model. There are mainly two applications for regression models. Firstly, it can be used for system identification, where a model is trained to learn a dynamical system's properties, which can then be used to analyze aspects of the dynamical system. For example, the behavior and stability of a system can be analyzed by studying the poles and zeros of an autoregressive moving average model with exogenous input signals (ARMAX) [82]. The second usage is predictions, where the model is used with new input samples, i.e., that are not part of the training data, to predict corresponding outputs. The model is expected to work well, i.e, to generalize, for new samples, as long as they have similar statistical characteristics as the training samples.

Regression analysis can be used for linear and nonlinear relationships between the inputs and outputs and involve one or several inputs. However, most problems involve several inputs and non-linearities.

4.2 Model types

Regression models can be divided into parametric and non-parametric models. Parametric modeling assumes a specific functional form for the relationship between the inputs and outputs of the model. The model architecture is fixed in advance, i.e., before the model is fitted to the training data, and the number of parameters is typically finite and fixed. Examples of parametric models include linear regression and logistic regression. Non-parametric modeling, on the other hand, does not make any assumptions about the functional form of the relationship between the inputs and outputs of the model. Instead, it learns the underlying relationship directly from the data. Non-parametric models can have an infinite number of parameters, and the complexity of the model may increase with the size of the data. Examples of non-parametric models include decision trees, k-nearest neighbors, and Gaussian processes.

The main advantage of parametric models is that they are often simpler and faster to train, and they can be more interpretable since the relationship between the inputs and outputs is explicitly defined. Additionally, the fixed structure makes it easy to implement in limited computational hardware. However, they may not be flexible enough to capture complex relationships in the data. The benefit of using non-parametric models is their flexibility in capturing complex relationships in the data without making any assumptions about the underlying functional form [54]. However, they can be computationally expensive and may suffer from overfitting.

Note that artificial neural networks typically have a fixed model architecture, i.e., the number of parameters, i.e., weights, is defined beforehand. Hence, it is per definition a parametric model. However, the number of parameters is typically large, and the usage of the model structure, i.e., how signals are propagated through the network, is determined by the data-driven training. The training is typically computationally demanding, and the model is in general very hard to interpret afterward. Hence, neural networks are parametric but share the most pros and cons with non-parametric models.

4.3 Linear regression models

The goal of linear regression is to find the best-fitting line (or hyperplane in higher dimensions) that describes the relationship between the inputs and the outputs [83]. The input and output are scalar-valued in one-dimensional linear regression:

$$y = mx + b \tag{4.1}$$

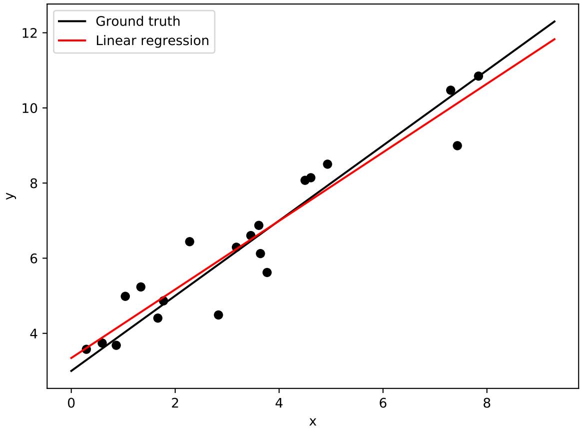

Figure 4.1: A line (red) is fitted to the noisy data (black dots).

where m (slope) and b (intercept) are the regression coefficients, y is the output, and x is the input. The aim is to find the values of m and b that minimize the residual sum of the squared errors between the predicted values $\hat{y}$ and the actual values of y:

$$RSS = \sum (y - \hat{y})^2. \tag{4.2}$$

An example is shown in figure 4.1, where training data points are created using a linear model $y = 2x + 3 + v$, where v is Gaussian zero-mean noise with standard deviation $\sigma = 0.8$. The linear regression model is fitted to the training samples ($0 \leq x \leq 8$) and extrapolates to unseen samples ($8 < x \leq 9.3$).

In multiple linear regression, there are multiple inputs, and the relationship between the inputs and output is described by a hyperplane. The equation of the hyperplane is typically represented as:

$$y = b + m_1 x_1 + m_2 x_2 + ... + m_k x_k \tag{4.3}$$

where y is the output, x_1, x_2, ..., x_k are the inputs, m_1, m_2, ..., m_k are the coefficients of the hyperplane, and b is the intercept.

The main advantage of linear models is that a simple closed-form solution exist to the optimization problem and that the model is very easy to compute in real-time [83].

4.4 Linear Bayesian regression

Linear Bayesian regression is a method that combines linear regression with Bayesian inference. It allows for the estimation of a probability distribution over the regression coefficients rather than just point estimates and provides a framework for uncertainty estimation [84].

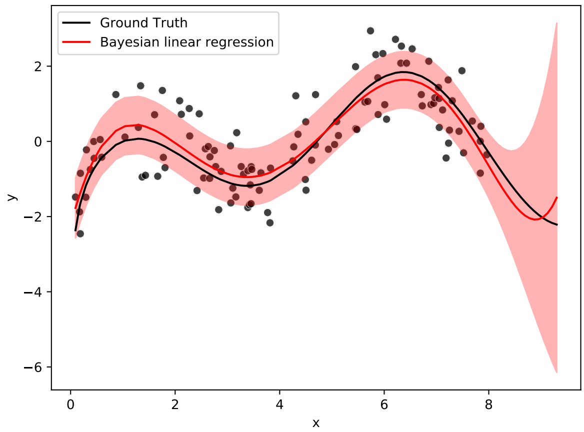

Figure 4.2: A 6^{th} order polynomial is fitted to noisy input data.

Compared to linear regression, a major difference is that prior beliefs about the distribution of the regression coefficients are incorporated into the model and updated based on the data using Bayes' theorem. The posterior distribution of the regression coefficients is then used to make predictions and estimate the prediction uncertainty.

An example is illustrated in figure 4.2 using data points created by a non-linear function $y = \log x + \cos x + v$, with v being zero-mean Gaussian noise with $\sigma = 0.75$. The regression model is designed as a 6^{th} order polynomial $y = a_0 + a_1 x + a_2 x + \ \dots\ + a_6 x^6$, which is linear in the coefficients. As one can see, the fitted model (red) has a good fit in the interval $0 \leq x \leq 8$, which is also indicated by the low standard deviation (shaded red). However, when the model is extrapolating ($8 < x \leq 9.3$), the fit rapidly becomes worse, which is indicated by the growing standard deviation.

The key advantage of linear Bayesian regression is that it allows for the incorporation of prior knowledge and beliefs about the data into the model, and provides a framework for updating these beliefs as new data becomes available. This is particularly useful in cases where the amount of data is limited, or when prior knowledge is available that can help constrain the possible values of the regression coefficients. The downside is that finding good prior distributions is sometimes non-trivial. A poor choice of prior can be compensated for by adding more data, but it might be problematic if there is no easy way of getting more data.

4.5 Autoregressive models

An autoregressive (AR) model is a statistical model that is commonly employed to model stochastic processes, where the properties of the process are unknown [85]. The model has a structure similar to that of linear regression, with the exception that the inputs are old (lagged) values of the output. The AR model can be expressed as:

$$y(t+1) = a_1 y(t) + a_2 y(t-1) + \cdots + a_k y(t-k) + w(t), \tag{4.4}$$

where a and k denote the model parameters and the order of the model, respectively, and $w(t)$ represents independent, zero-mean process noise.

The AR model can be used to forecast the next sample of the process using the predictor:

$$\hat{y}(t+1|t) = a_1 y(t) + a_2 y(t-1) + \cdots + a_k y(t-k), \tag{4.5}$$

which can be used recursively to predict multiple steps into the future.

The AR model can also be extended to multidimensional processes, where $y(t)$ is a vector instead of a scalar. This model is referred to as the vector autoregressive (VAR) model [83], which has the same structure as the AR model, but the parameters of the model are matrices. Both the scalar and vector versions have closed-form solutions that are very efficient to compute. Moreover, the AR model can also be implemented for non-linear processes, by considering, for example, artificial neural networks [86].

The AR model can be combined with a moving average (MA) model to form an autoregressive moving average (ARMA) model, which also incorporates the prediction errors into the model. The ARMA model is most suitable for stationary processes, while the autoregressive integrated moving average (ARIMA) model is designed for non-stationary processes. The ARMAX model is an extension of the ARMA model that also models external (exogenous) inputs to the process, i.e., time series data that is assumed to be fully known at all time instances.

4.6 Artificial neural networks

Artificial neural networks are well known for their ability to learn complex non-linear relationships between inputs and outputs directly from the training data and have increased in popularity for regression problems over the last decades. The network architectures are in this thesis divided into feed-forward and recurrent models.

Feed forward models

A feedforward neural network (FFNN) is a type of artificial neural network that is designed to process inputs in a forward direction without loops or cycles, i.e., the input layer receives the input data, which is then processed by one or more hidden layers, and finally produces an output at the output layer. Each layer is composed of a set of neurons, which are connected to the neurons in the previous layer by weighted connections. Each neuron in the hidden layers applies a non-linear activation function, e.g., a sigmoid or rectified linear function, to the weighted sum of the inputs, which allows the network to learn complex non-linear relationships between the inputs and the outputs. The output layer typically applies a linear activation function for regression problems and a softmax function for logistic regression problems. FFNNs are commonly trained using backpropagation, which involves computing the gradients of the loss function with respect to the weights of the network and updating the weights using an optimization algorithm such as stochastic gradient descent [87].

There are many variants of FFNNs. One of the most common is the multilayer perceptron (MLP) [88], consisting of a sequence of hidden layers with fully connected neurons. Convolutional neural networks (CNN) have the ability to automatically learn important features from the data and are commonly used in computer vision and natural language processes (NLP) applications [89], but can also be used for regression tasks. Autoencoders use encoding and decoding layers to reconstruct the input data [90], which can be used to detect anomalies in the input data, for example. Moreover, transformer networks utilize attention functionality to perform sequence-to-sequence tasks, which is common in NLP [91].

Recurrent models

A recurrent neural network (RNN) is a type of network architecture that is designed to handle sequential data, such as time series or natural language processing. Unlike feedforward neural networks that process inputs in a single pass, RNNs have loops in their architecture that allow them to maintain an internal state or memory of past inputs. At each time step, an RNN takes an input and its internal state from the previous time step and computes the output. The internal state is updated and passed on to the next time step, allowing the network to capture information about the sequence of inputs over time.

RNNs can be trained using backpropagation through time (BPTT) [92], which is an extension of backpropagation [93] used for training feedforward neural networks. BPTT calculates gradients of the loss function with respect to the weights of the network and updates them using an optimization algorithm such as stochastic gradient descent [87].

One of the most popular types of RNN is the Long Short-Term Memory (LSTM) network [94], which includes specialized memory cells that can selectively remember or forget information over time. LSTMs are particularly good at capturing long-term dependencies in sequential data. Another popular variant is the Gated Recurrent Unit (GRU) [95], which has a simpler architecture than LSTM but is still capable of capturing long-term dependencies in sequential data.

Bidirectional RNN processes the input sequence in both forward and backward directions, using two separate hidden states. Bidirectional RNNs are particularly useful for tasks such as speech recognition and named entity recognition, where the context of the input is important [96], [97].

Note that it is common to mix feed-forward models with recursive models to specialize the model's architecture to the problem, e.g., the model's input layer is fully connected to a layer of LSTM-cells, that propagates its output to a fully connected MLP layer, followed by a linear output layer.

4.7 Uncertainty-aware neural networks

An uncertainty-aware neural network is a type of model that is designed to make robust predictions and estimate the uncertainty of the predictions. It is of particularly importance in scenarios where the model needs to make decisions based on incomplete or noisy data, or where the cost to make a wrong prediction is high. Uncertainty-aware ML models are typically based on Bayesian principles, i.e., the network is stochastic and relies on prior information, or the ensembling technique, which is more a frequentist's approach.

Bayesian neural networks

In a Bayesian neural network, the weights of the network are treated as random variables with a prior distribution and the goal is to estimate the posterior distribution of the weights, given the training data [98]. This allows the network to represent the uncertainty in its predictions as a predictive probability distribution. However, there is typically no closed-form expression for the posterior distribution over the weights and neither for the predictive distribution over the output. Hence, they need to be approximated by numerical methods, such as Markov chain Monte Carlo (MCMC) sampling. Another Bayesian approach is the Monte Carlo dropout network (MCDO), which uses a regularization technique that randomly drops out some of the neurons in the network during training, which forces the network to be more robust to variations in the inputs [99]. The dropout is also used in the forward pass to numerically estimate the uncertainty of the network by generating multiple predictions for the same input.

Ensemble neural networks

Ensemble methods combine multiple instances of a neural network to improve performance and robustness. This can be done by using techniques such as bagging, deep-ensemble, boosting, stacking, or ensemble selection [100]. Bagging involves training multiple instances of the same neural network on different subsets of training data and averaging the output predictions [101]. The deep ensemble approach is similar, but the ensemble networks are trained on the same data, where diversity within the ensemble is enforced by randomizing the initialization of the model's parameters before the training is started [102], [103]. Boosting trains multiple instances of the same neural network sequentially, with each model correcting the errors of the previous model [104]. Stacking trains multiple neural network models with different architectures and combines their predictions using a meta-model [105]. Ensemble selection selects a subset of models most likely to perform well on test data using techniques such as Bayesian model averaging or cross-validation [106].

As stated above, the ensembling technique improves the accuracy and robustness of neural networks, but the method can also be used to estimate the uncertainty of the network predictions, i.e., the prediction uncertainty can be estimated by the sample variance of the predictions across the ensemble.

CHAPTER 5

Summary of included papers

This chapter provides a summary of the included papers.

5.1 Paper A

John Dahl, Gabriel Rodrigues de Campos, Claes Olsson Jonas Fredriksson
Collision Avoidance: A Literature Review on Threat-Assessment Techniques
IEEE Transactions on Intelligent Vehicles, Volume: 4, Issue: 1, pp. 101 - 113, 2019
©—IEEE DOI:10.1109/TIV.2018.2886682 .

This paper presents a review of threat assessment methods used for collision avoidance. It contributes with an overview of the field and contributes with analysis and discussion on robustness, computational complexity, and best suitable applications. It concludes that driver intention is one of the main challenges in the threat assessment problem, which is hard to model using engineering. It also suggests that machine learning is a promising technique for threat assessment, but it suffers from high computational demand.

The authors' contribution: The thesis author was responsible for collecting and reviewing papers, with support from the second author. The majority of the writing, analysis, and discussion was made by the thesis author. The second author mainly wrote the last part of Section 2.3 dealing with formal verification and supervisory control. The three last authors contributed with support to the writing process and also contributed to the structuring of the reviewed papers.

5.2 Paper B

John Dahl, Rasmus Jonsson, Anton Kollmats, Gabriel Rodrigues de Campos, Jonas Fredriksson
Automotive Safety: a Neural Network Approach for Lane Departure Detection using Real World Driving Data
IEEE Intelligent Transportation Systems Conference (ITSC)
pp. 3669–3674, 2019
©IEEE DOI: 10.1109/ITSC.2019.8917288 .

This paper uses a supervised machine learning approach, based on a feed-forward artificial neural network, to detect unintended lane departures for multidimensional time series data. The problem is formulated as a classification problem, where the output of the prediction model is binary, i.e., it should predict 1 if the current situation is likely to evolve into an unintended lane departure within a time period equal to the prediction horizon, or 0 otherwise. The system is evaluated using a real-world data set and a kinematic benchmarking model. It also introduces the concept of down-sampling in real-time, a technique aimed to reduce the computational burden.

The authors' contribution: The author of this thesis was responsible for the problem formulation and the approach to solving it. The thesis author also supervised the second and third authors in their master thesis project, where they developed the code base that was partly used in this work. The paper was mainly written by the thesis author, and the experiments were run by the thesis author and the second author. The last two authors contributed with support in the analysis of the results and in the writing process.

5.3 Paper C

John Dahl, Gabriel Rodrigues de Campos, Jonas Fredriksson
A Path Prediction Model based on Multiple Time Series Analysis Tools used to Detect Unintended Lane Departures
IEEE Intelligent Transportation Systems Conference (ITSC), pp. 1-7
2020
©IEEE DOI:10.1109/ITSC45102.2020.9294510 .

The neural network approach in Paper B showed good performance in comparison to the kinematic model, but, as mentioned in Paper A, the network-based model is significantly harder to compute in real-time. To overcome this limitation, a linear model was developed based on multiple linear regression and with inspiration from autoregressive modeling. This model can be trained (estimated) in closed form and it is efficient to compute in real-time. Moreover, the problem formulation is changed in this work, from a classification problem in Paper B, to a path prediction problem. The intuition for this change was that a path is easier to interpret than a binary classifier. The linear model's performance was benchmarked towards a kinematic model and showed superior performance.

The authors' contribution: The thesis author was responsible for developing the problem formulation in collaboration with the co-authors, planning and implementing the algorithms, and authoring the main core of the paper. The last two authors contributed with support in the analysis of the results and in the writing process.

5.4 Paper D

John Dahl, Gabriel Rodrigues de Campos, Jonas Fredriksson
Performance and Efficiency Analysis of a Linear Learning-Based Prediction Model used for Unintended Lane-Departure Detection
IEEE Transactions on Intelligent Transportation Systems, Volume: 23, Issue: 7, pp. 9115 - 9125,
2022
©IEEE DOI:10.1109/TITS.2021.3090941 .

This paper analyses how the linear prediction model from Paper C performs relatively a non-linear model based on neural networks. The results showed that the linear model is as good as the non-linear model, for prediction horizons up to 1.25 s, while having superior real-time performance. Moreover, the paper provides a thorough analysis of how to select the model's inputs. It explores which input signals contribute the most to the predictive performance and provides a methodology for downsampling the signals without losing valuable information.

The authors' contribution: The thesis author was responsible for developing the problem formulation in collaboration with the co-authors, planning and implementing the algorithms, and authoring the main core of the paper. The last two authors contributed with support in the analysis of the results and in the writing process.

5.5 Paper E

John Dahl, Gabriel Rodrigues de Campos, Jonas Fredriksson
Prediction-Uncertainty-Aware Threat Detection for ADAS: A Case Study on Lane-Keeping Assistance
IEEE Transactions on Intelligent Vehicles, Early access, Mars,
2023
©—–IEEE DOI:10.1109/TIV.2023.3253555 .

This paper uses uncertainty-aware prediction models to detect untrustworthy predictions in real-time. The paper presents four threat-detection logics that use the uncertainty information to improve the robustness of the system when it is used outside of the operational design domain or when there are anomalies in the inputs. The performance was evaluated using a real-world data set and is including a detailed analysis of how the robustness of the threat detection is affected by different sources of anomalies. The results showed that a robust transient sensitive threat-detection logic worked the best for 1 s prediction horizon, while a robust disjoint transient sensitive logic was the preferred choice for the 2 horizon.

The authors' contribution: The thesis author identified the problem and developed the idea to address it. The thesis author also planned and conducted the experiments and wrote the core of the paper. The third author and thesis author came up with the focus and the layout of the paper. The two last authors contributed to the analysis and the writing.

5.6 Paper F

John Dahl, Gabriel Rodrigues de Campos, Jonas Fredriksson
Intention-Aware Lane Keeping Assist Using Driver-Gaze Information
Accepted to the IEEE Intelligent Vehicles Conference (IV),
2023 .

This paper analyzes how a camera-based driver monitoring system improves the performance of LKA system. The goal of this work is to distinguish unintentional lane departures from intended lane departures where no turning indicators are used. This is used to ensure that an automatic intervention is only triggered when the departure is unintentional. The system performance is evaluated by testing with and without gaze-tracking information as an input to the neural networks-based prediction model. The results show that the information from the gaze tracking improves the performance of the system, especially for the 1 s prediction horizon.

The authors' contribution: The thesis author was responsible for the problem formulation, preparing the data, implementing the algorithms, analyzing the results, and writing the paper. The second and third authors contributed with helpful discussions and support in the writing.

CHAPTER 6

Discussion

The focus of this thesis has been to develop threat assessment and decision-making methods that are able to include driver intention and sensor noise. As concluded from the review in Paper A, and also from later works [64], [107], there are many approaches to these problems. However, the approaches rely to a big extent on expert knowledge of how to parameterize the models. Additionally, the sensor noises could have been partly solved by additional filtering, but filters contribute to delays (phase-sifts in the frequency domain) in the signals and can distort the signal too much. Hence, it was hypothesized in this project that a learning-based approach could help to solve these issues. The aim was to design a prediction model that yielded a good predictive performance while being efficient to compute. The attention has mainly been on network architectures, the selection of input signals, and implementation aspects in general. It also deals with robust threat detection under uncertainty, where the goal is to suppress system activations in cases when the predictions are untrustworthy, without sacrificing the overall system performance.

6.1 Contribution

Based on the insights of the appended papers, the research questions are discussed as follows:

RQ1: *Can the challenges related to driver intention and sensor issues be addressed by supervised learning in the context of a lane-keeping assistance system?*

The RQ1 is addressed by Paper B, D, E and F. Paper B evaluates a neural network-based prediction model, which is used to predict unintentional lane departures. The problem is formalized as a logistic regression problem, i.e., it should predict a value equal to 1 if a lane departure is imminent within h s prediction horizon, or zero otherwise. The prediction model was based on a feed-forward multiple perceptron network. Its performance was evaluated by comparing the results with a kinematic constant velocity model and the results showed that the neural network-based prediction model had superior performance, which is aligned with a related analysis based on simulated data [108]. There were also unpublished experiments with other kinematic models, such as the constant turn-rate and velocity model, but they showed to perform worse than the constant velocity model, especially for the longer prediction horizons. This showed to stem from that drivers were oscillating back and forth in the lane, which was exaggerated by the CRTV model, and it was, therefore, prone to falsely predict lane departures.

Robustness towards anomalies in the input data was handled in Paper E, where uncertainty-aware prediction models were used to detect erroneous predictions. The focus of the paper was on event-detection logic used for decision-making, where four logics were introduced and evaluated. In summary, it was shown that prediction uncertainty derived from the uncertainty-aware prediction model had a significant impact on the performance in situations with anomalies in the inputs. Moreover, it is also evident from the results that MLP architectures can be as good as recurrent models, i.e., LSTM, and Bayesian models if the input signals are selected with great care to the details. This is also coherent with the results presented in [109] and [110], where no significant differences between an MLP and an LSTM could be seen. The merits of the MLP, as compared to the more complex alternatives, are the stable training, easiness to deploy it in the vehicle, and that it is relatively lightweight in terms of computational demand.

In conclusion, based on the results, it is evident that the machine learning approach contributes to increased predictive performance. Unfortunately, it is not obvious why it works, as it is a black box model that is very difficult to interpret. Hence, it is not known in Papers B-E to what extent the filtering of the input signals contributes to the performance, nor to which extent the driver intention is captured by the model. However, Paper F shows that the machine learning-based prediction model is able to better capture the driver's intention if a camera-based driver monitoring system is used, relatively not having it.

RQ2: *What input data is needed to achieve a high system performance?*

The prediction model's in this thesis used lagged signals as inputs, i.e., it used a sequence of old data points. It was discovered in Paper B that 3 consecutive samples (the sampling frequency is 40 Hz) were not enough to achieve high predictive performance. The authors, therefore, elaborated on various lengths of the input sequences and came up with a hypothesis that the most recent signals reflect the current state, while older ones contribute to the trend. Interestingly, the results from Paper B showed that the sampling pattern, i.e., the way the old samples were picked, did not change the predictive performance much. It seemed to be more important to pick old enough samples than how densely they are sampled. However, the oldest sample was only 0.35 s old. This aspect was further analyzed in Paper C, where up to 2 s old samples were tested. The results showed that the best performance was achieved by using 1 s old samples. Even more interesting was that a logarithmic sampling pattern of the old sample gave the same or better performance as using all consecutive samples. As shown in Paper D, the sampling pattern is a key to achieving a good real-time capability, i.e., the fewer inputs, the fewer computations are needed to calculate the output of the model. The best performing pattern was determined in Paper D by systematically testing sample rates between 1.25 and 40 Hz, where 5 Hz turned out to be the lowest sample rate that still could maintain the predictive performance. It was also investigated which kinematic and road geometry sensor signals contributed the most to the predictive performance, where it was shown that the first two coefficients of the 3:rd order lane-marker polynomials and the front wheel angle were the most important. Paper F showed that signals from a camera-based driver monitoring system improved the system performance in cases where intentional lane departures should be distinguished from intentional lane departures.

RQ3: *How could real-time computation-limitations be considered?*

Paper C contributed with a linear prediction model formulation, which aimed to be more efficient to train and compute in real-time, compared to the MLP used in Paper B and the other non-linear network architectures in Paper E and F. In this work, the problem formulation was changed to a path prediction problem. The intuition was that a predicted path should be easier to interpret than the logistic output in Paper B, which would make eventual debugging easier. The results revealed that the linear model was superior to the constant velocity model. This motivated a deeper analysis in Paper D, where the performance of the linear model was compared to an MLP model. Interestingly, the performance evaluation showed that a linear model makes predictions that are as good as a non-linear model, for prediction horizons up to 1.25 s, while the latter model architecture is superior for longer prediction horizons. However, a comparison in time complexity, i.e., the computational burden of a forward pass in the model, showed that the linear model is orders of magnitude easier to compute. Hence, the results show that high predictive performance can be achieved even for relatively lightweight solutions that are efficient to compute in real-time. This finding is also in line with the results in [52], where it is concluded that a lightweight recursive model can be as good as more complex LSTM and GRU models. These findings are essential in the work of developing new active safety systems since the computational burden of the threat assessment and decision-making algorithms determines the requirements of the computational hardware platform.

6.2 Validity

Internal validity is closely related to the research design, where the goal is to reduce factors that might lower the trustworthiness of the research. For example, the sensor hardware platform used to collect the field test data is fixed for the entire test, which ensures that the raw data is consistent. However, the software of the sensors, and even the sensor fusion algorithms, are updated continuously to fix discovered issues. Hence, there is a risk of a mismatch between data that are sampled on different dates. This issue is avoided by using an offline postprocessing tool that updates all collected data such that it is aligned with the latest software version. Moreover, the Papers B-E are based on data set A and Paper F on data set B. The same signals were collected in the two sets, except for the gaze-tracking signals that were only included in set B. However, the sensor platforms were not the same, and the software was not the same either. It is therefore interesting that the results from, e.g., Paper D and Paper F are similar for prediction horizons up to 1.5 s. This may indicate that small variations and differences in the hardware/software do not significantly affect the system's performance. Finally, the drivers that participated in the data collection were trained drivers, which may reduce the risk of including bias in terms of maturation in the drivers' skill level.

External validity refers to the extent to which the study's findings can be generalized to other contexts or populations. The experiments were performed using large real-world datasets with varying driving from several continents, which suggests that the results can be generalized to real implementation in an LKA system used by the average driver. Furthermore, it is likely that other active safety features, such as oncoming collision avoidance and AEB, also would benefit from using machine learning-based threat assessment, as they also suffer from the uncertainty in the drivers' intention and sensor noise. However, the level of improvement in terms of performance cannot be anticipated based on this research.

On the other hand, there are also potential weaknesses. For example, professional drivers might not represent the average population of drivers, and the drivers in the field test might be more conservative in their driving, knowing that the test vehicles are expensive and damaging them is troublesome as the organization is waiting for the results. Additionally, the used definition of an unintentional lane departure, see Section 2.2 in Paper B, might be too general in certain situations. It relies on that a lane departure is unintentional if the vehicle is not crossing the lane marker by more than the half width of the vehicle, and that the vehicle is back in lane within 4 seconds. However, being a reasonable assumption, it does not take into account that the driver might avoid something else on the road and therefore temporarily depart from the lane. This is a matter of definition since avoiding something on the road might also occur without an intention to leave the ego lane. Unfortunately, the time series signals do not track objects other than vehicles and road users, so it is not easy to determine what causes an unintended lane departure per our definition. However, from a safety perspective, it is better to detect those cases as unintended lane departures, than risking to miss a case because the prediction model is uncertain whether it is a driver avoidance maneuver or not.

6.3 Prediction horizon

An LKA system requires the driver to be responsible for driving at all times, even when the ADAS is supporting the driver. Hence, there must be a way for the driver to override the system's interventions. For steering interventions, that means that the steering torque applied by the system must not exceed the achievable counter torque of the driver. Moreover, it has to be applied in a such way that the driver has enough time to understand what is happening, to give the driver a chance to counteract the intervention. These system limitations force the automatic avoidance maneuvers to be slow, yielding a long maneuver time. Hence, the decision to intervene with an automatic maneuver must be taken early enough, such that the system can avoid the threat successfully. This implies that the prediction model must have a long enough prediction horizon.

Unfortunately, the longer into the future one tries to predict, the harder it is. This means that early interventions (using a long prediction horizon) are prone to be faulty, i.e., the LKA system supports the driver in situations where the driver already has full control of the situation. However, from a strict functional safety perspective, this might not be a problem, as both the driver and the system try to maintain safety, but the driver might perceive the system's faulty interventions as intrusive and disturbing. Hence, there is a risk that the driver eventually shuts the system off if the driver does not comply with the intervention. It is therefore desirable to make the decisions as close as possible in time to a critical event while maintaining enough time for an effective avoidance maneuver. As concluded in Paper D, a sufficient prediction horizon for LKA is in the interval $[1 - 1.75]$ s, which is aligned with the suggestion of between 1.5-2.5 s in [111].

6.4 Real-world data

The implementation and analysis of the proposed approaches are based on large proprietary data sets that are not open to the common. The data sets are wast in size and reflect regular driving in urban and rural road environments, under various weather conditions and times of day, in the US, EU, and Asia. The data sets contain time-series data derived from an automotive-grade camera and sensor system, and the vehicles are driven by trained drivers. Hence, the data sets are comprehensive and representative of real-world driving and are thereby ideal for system evaluation. Indeed, it would have been beneficial to use open data instead, but to the best of the author of this thesis's knowledge, there exist only a few data sets with time-series data from regular driving, which are too small to be used for a quantitative system evaluation. For example, the PREVENTION data set consists of 6 hours of driving, containing only 12 unintended lane departures. It is the general case that open data sets are not suitable for real-time threat assessment and decision-making based on vehicle-mounted sensor information. For example, the KITTI [112], ApolloScape [113], A2D2 [114] and BDD100K [115] data sets are sampled locally on the vehicle but is aimed towards computer vision and contains no annotations for road geometry. The Cityscapes [116] data set is sampled every 20 s, or 20 m, whatever comes first, which makes it unsuitable for threat assessment due to too low time resolution. Other popular data sets are the Next Generation Simulation HW101 (NGSIM) [117], HighD [118] and INTERACTION [119] data sets that use either cameras mounted to tall buildings and civil infrastructure or by flying drones. They provide a bird-view perspective of the traffic in certain locations, which is unrealistic for real-world evaluation of active safety systems.

CHAPTER 7

Concluding Remarks and Future Work

The main challenge for the next generation ADAS is to include the driver's intention in the threat assessment, keep the computational complexity low, and make the algorithms robust to anomalies and noise in the sensor data. This thesis is focusing on how to use learning-based prediction models in threat assessment to improve the performance of a lane-keeping assistance system. The learning-based models have been benchmarked against a kinematic model, and the results show that linear and non-linear models are superior in performance, especially for longer prediction horizons. The results indicate that learning-based prediction models are better at dealing with noise in the input data and learning the driver's intention. However, it is not known which of the two factors contributes the most to the increased performance. It is evident, though, that adding the gaze information of the driver to the inputs improves the ability of the learning-based prediction model to capture the driver's intention. Moreover, the results show that a linear prediction model is nearly as good as a non-linear model based on artificial neural networks, at least for shorter prediction horizons. This is convenient, as the time complexity of the linear model is magnitudes lower than the time complexity of non-linear models, which makes the former easier to run in real-time. It has also been shown that the time complexity can be reduced by carefully selecting the input signals to avoid redundancy and by downsampling the input information in real-time to avoid signal samples that are not contributing to improved prediction performance. Finally, uncertainty-aware prediction models can be used to improve the system's robustness, by utilizing the prediction uncertainty in the decision-making to avoid false detections of unintentional lane departures.

There are several possible directions for further research. It is not unlikely that the driver's intention is affected by surrounding vehicles and objects on the road, e.g., the ego-vehicle driver might intend to keep a safe distance from an overtaking vehicle to the left, which causes a lane departure on the right side of the ego-vehicle. Information on the surroundings might help the prediction model and the detection algorithm to correctly classify the event as an intentional lane departure. The results presented in this thesis are based on a large field test data set with professional drivers. However, it would be interesting to compare the performance using a data set that is created to represent the average driver, with respect to driving skills, age, driving styles, etc. It would also be interesting to evaluate the performance for slower velocities and smaller radius of curvature of the roads, as it is likely that the driving behavior is substantially different for those types of roads. In terms of real-time performance, it would be interesting to investigate techniques for knowledge distillation, i.e., reduce the size of the model, to improve the computational burden of using uncertainty-aware prediction models. Finally, a major concern using learning-based methods is the system safety aspect, i.e., to ensure that the model works as intended within the operational design domain.

References

[1] World Health Organization, “Global status report on road safety 2018,” 2018.

[2] (2023). “The top 10 causes of death. fact sheet. geneva: World health organization,” [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (visited on 03/03/2023).

[3] (2022). “Volvo safety heritage,” [Online]. Available: https://www.volvocars.com/intl/v/safety/heritage (visited on 12/21/2022).

[4] (1959). “The crumple zone man,” [Online]. Available: https://www.autospeed.com/cms/a_113292/article (visited on 12/21/2022).

[5] (1974). “Oldsmobile air cushion folder,” [Online]. Available: http://www.oldcarbrochures.com/static/NA/Oldsmobile/1974_Oldsmobile/1974_Oldsmobile_Air_Cushion_Folder/1974%20%5C%%2020Oldsmobile%20%5C%%2020Air%20%5C%%2020Cushion%20%5C%%2020Folder-06-07.html (visited on 12/21/2022).

[6] G. E. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, no. 8, 1965.

[7] (1971). “Four wheel sure brake,” [Online]. Available: https://www.web.imperialclub.info/Yr/1971/SureBrake/SBPage01.jpg (visited on 12/21/2022).

[8] (1983). "Toyota technical development - chassis," [Online]. Available: `http://www.toyota-global.com/company/history_of_toyota/75years/data/automotive_business/products_technology/technology_development/chassis/index.html` (visited on 12/21/2022).

[9] T. Litman, "Autonomous vehicle implementation predictions: Implications for transport planning," 2022.

[10] (2023). "Waymo one," [Online]. Available: `https://waymo.com/waymo-one/` (visited on 04/01/2023).

[11] (2023). "Gm cruise rides," [Online]. Available: `https://getcruise.com/rides/` (visited on 04/01/2023).

[12] (2023). "Motional," [Online]. Available: `https://motional.com/las-vegas` (visited on 04/01/2023).

[13] (2022). "Tesla full self-driving safety analysis," [Online]. Available: `https://dawnproject.com/wp-content/uploads/2022/01/Tesla-FSD-Safety-Analysis.pdf` (visited on 12/19/2022).

[14] "Addendum 78: Un regulation no. 79," United Nations, Tech. Rep., 2017.

[15] National Highway Traffic Safety Administration, *Target crash population for crash avoidance technologies in passenger vehicles*, Mar. 2019.

[16] R. Utriainen, M. Pöllänen, and H. Liimatainen, "The safety potential of lane keeping assistance and possible actions to improve the potential," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 556–564, 2020.

[17] Consumer Reports, *Consumer perceptions of adas*, Nov. 2019.

[18] ——, *Consumer reports' guide to adas usability: Consumer insights on understanding, use, and satisfaction of adas*, Dec. 2022.

[19] G. Effler. (2022). "Satisfaction issues with advanced driver assistance systems leads to new j.d. power study," [Online]. Available: `www.jdpower.com/business/press-releases/2022-adas-advanced-driver-assistance-systems-quality-and-satisfaction-study` (visited on 12/19/2022).

[20] B. Fildes, M. Keall, N. Bos, A. Lie, Y. Page, C. Pastor, L. Pennisi, M. Rizzi, P. Thomas, and C. Tingvall, "Effectiveness of low speed autonomous emergency braking in real-world rear-end crashes," *Accident Analysis & Prevention*, vol. 81, pp. 24–29, 2015.

[21] S. Sternlund, J. Strandroth, M. Rizzi, A. Lie, and C. Tingvall, "The effectiveness of lane departure warning systems—a reduction in real-world passenger car injury crashes," *Traffic Injury Prevention*, vol. 18, no. 2, pp. 225–229, 2017.

[22] (2018). "Iso 26262-1:2018 - road vehicles, functional safety," [Online]. Available: `https://www.iso.org/%5C%5Cstandard/68383.html` (visited on 04/01/2023).

[23] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, 2021.

[24] T. Meng, J. Huang, C.-M. Chew, D. Yang, and Z. Zhong, "Configuration and design schemes of environmental sensing and vehicle computing systems for automated driving: A review," *IEEE Sensors Journal*, 2023, Early access.

[25] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 5, pp. 694–711, 2006.

[26] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai, "A review of visual slam methods for autonomous driving vehicles," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 104 992, 2022.

[27] L. Xiong, X. Xia, Y. Lu, W. Liu, L. Gao, S. Song, and Z. Yu, "Imu-based automated vehicle body sideslip angle and attitude estimation aided by gnss using parallel adaptive kalman filters," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10 668–10 680, 2020.

[28] A. Halin, J. G. Verly, and M. Van Droogenbroeck, "Survey and synthesis of state of the art in driver monitoring," *Sensors*, vol. 21, no. 16, p. 5558, 2021.

[29] M.-H. Sigari, M.-R. Pourshahabi, M. Soryani, and M. Fathy, "A review on driver face monitoring systems for fatigue and distraction detection," *International Journal of Advanced Science and Technology*, vol. 64, pp. 73–100, 2014.

[30] M.-H. Sigari, M. Fathy, and M. Soryani, "A driver face monitoring system for fatigue and distraction detection," *International journal of vehicular technology*, vol. 2013, 2013.

[31] A. Kashevnik, I. Lashkov, A. Ponomarev, N. Teslya, and A. Gurtov, "Cloud-based driver monitoring system using a smartphone," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6701–6715, 2020.

[32] A. Kashevnik, I. Lashkov, and A. Gurtov, "Methodology and mobile application for driver behavior analysis and accident prevention," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2427–2436, 2020.

[33] R. Buendia, F. Forcolin, J. Karlsson, B. Arne Sjöqvist, A. Anund, and S. Candefjord, "Deriving heart rate variability indices from cardiac monitoring—an indicator of driver sleepiness," *Traffic injury prevention*, vol. 20, no. 3, pp. 249–254, 2019.

[34] Y. Lu and L. Bi, "Combined lateral and longitudinal control of eeg signals-based brain-controlled vehicles," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 9, pp. 1732–1742, 2019.

[35] S. F. A. Razak, S. Yogarayan, A. A. Aziz, M. F. A. Abdullah, and N. H. Kamis, "Physiological-based driver monitoring systems: A scoping review," *Civil Engineering Journal*, vol. 8, no. 12, pp. 3952–3967, 2022.

[36] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *Ieee Access*, vol. 8, pp. 2847–2868, 2019.

[37] J. Kocić, N. Jovičić, and V. Drndarević, "Sensors and sensor fusion in autonomous vehicles," in *Telecommunications Forum (TELFOR)*, 2018, pp. 420–425.

[38] J. Dahl, G. Rodrigues de Campos, C. Olsson, and J. Fredriksson, "Collision avoidance: A literature review on threat-assessment techniques," *IEEE Trans. on Intelligent Vehicles*, vol. 4, no. 1, pp. 101–113, 2019.

[39] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, p. 1, 2014.

[40] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," *International Conference on Information Fusion*, pp. 1–6, 2008.

[41] D. Svensson, "Derivation of the discrete-time constant turn rate and acceleration motion model," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2019, pp. 1–5.

[42] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," *2013 IEEE/RSJ international conference on intelligent robots and systems*, pp. 4363–4369, 2013.

[43] P. Polack, F. Altché, B. d'Andréa-Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" In *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 812–818.

[44] A. Broadhurst, S. Baker, and T. Kanade, "Monte carlo road safety reasoning," in *IEEE Intelligent Vehicles Symposium.*, IEEE, 2005, pp. 319–324.

[45] A. Eidehall and L. Petersson, "Threat assessment for general road scenes using monte carlo sampling," in *IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1173–1178.

[46] G. Rodrigues de Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative receding horizon conflict resolution at traffic intersections," *IEEE Conference on Decision and Control*, pp. 2932–2937, 2014.

[47] S. Lefèvre, "Risk Estimation at Road Intersections for Connected Vehicle Safety Applications," Ph.D. dissertation, Universitè de Grenoble, 2012.

[48] A. Armand, D. Filliat, and J. Ibañez-Guzman, "A Bayesian Framework for Preventive Assistance at Road Intersections," *IEEE Intelligent Vehicles Symposium*, 2016.

[49] W. Wang, D. Zhao, W. Han, and J. Xi, "A learning-based approach for lane departure warning systems with a personalized driver model," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 10, pp. 9145–9157, 2018.

[50] X. Mo, Y. Xing, and C. Lv, "Interaction-aware trajectory prediction of connected vehicles using cnn-lstm networks," *IEEE Industrial Electronics Society Conference*, pp. 5057–5062, 2020.

[51] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2020.

[52] K. Min, D. Kim, J. Park, and K. Huh, "RNN-based path prediction of obstacle vehicles with deep ensemble," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 252–10 256, 2019.

[53] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, *et al.*, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," *IEEE International Conference on Robotics and Automation*, pp. 10 563–10 569, 2020.

[54] I. Goodfellow, Y. Bengio, *et al.*, *Deep learning*. MIT press, 2016.

[55] A. Van der Horst and G. R. Brown, "Time-to-collision and driver decision making in braking," 1989.

[56] R. van der Horst and J. Hogema, "Time-to-collision and driver decision making in braking," in *Proceedings of the International Co-operation on Theories and Concepts in Traffic safety*, 1994.

[57] J. C. Hayward, "Near-miss determination through use of a scale of danger," in *Annual Meeting of the Highway Research Board*, Highway Research Board, 1972, pp. 24–34.

[58] P. Falcone, M. Ali, and J. Sjöberg, "Predictive threat assessment via reachability analysis and set invariance theory," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1352–1361, 2011.

[59] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 528–540, 2006.

[60] S. Sontges, M. Koschi, and M. Althoff, "Worst-case analysis of the time-to-react using reachable sets," in *IEEE Intelligent Vehicles Symposium*, 2018.

[61] S. Noh and W. Y. Han, "Collision avoidance in on-road environment for autonomous driving," *International Conference on Control, Automation and Systems*, 2014.

[62] European new car assessment programme, *Test protocol – lane support systems*, Nov. 2015.

[63] J. Nilsson, A. Ödblom, and J. Fredriksson, "Worst-case analysis of automotive collision avoidance systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 1899–1911, 2016, ISSN: 0018-9545.

[64] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Neurohr, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 1–35, 2023.

[65] J. M. Ambarak, H. Ying, F. Syed, and D. Filev, "A neural network for predicting unintentional lane departures," *IEEE International Conference on Industrial Technology*, 2017.

[66] X. Tang, K. Yang, H. Wang, J. Wu, Y. Qin, W. Yu, and D. Cao, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, 2022.

[67] L. Zhang, W. Ding, J. Chen, and S. Shen, "Efficient uncertainty-aware decision-making for automated driving using guided branching," in *International Conference on Robotics and Automation*, IEEE, 2020, pp. 3291–3297.

[68] L. Li, W. Zhao, C. Wang, and Z. Luan, "Pomdp motion planning algorithm based on multi-modal driving intention," *IEEE Transactions on Intelligent Vehicles*, pp. 1–10, 2022.

[69] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 294–305, 2020.

[70] J. Y. Hwang, J. S. Kim, S. S. Lim, and K. H. Park, "A fast path planning by path graph optimization," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 1, pp. 121–129, 2003.

[71] D. Madås, M. Nosratinia, M. Keshavarz, P. Sundström, R. Philippsen, A. Eidehall, and K.-M. Dahlén, "On path planning methods for automotive collision avoidance," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 931–937.

[72] S. Teng, P. Deng, Y. Li, B. Li, X. Hu, Z. Xuanyuan, L. Chen, Y. Ai, L. Li, and F.-Y. Wang, "Path planning for autonomous driving: The state of the art and perspectives," *arXiv preprint arXiv:2303.09824*, 2023.

[73] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario mpc approach for uncertain multi-modal obstacles," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 947–952, 2021.

[74] P. Typaldos, M. Papageorgiou, and I. Papamichail, "Optimization-based path-planning for connected and non-connected automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103 487, 2022.

[75] N. Awad, A. Lasheen, M. Elnaggar, and A. Kamel, "Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles," *ISA transactions*, vol. 129, pp. 193–205, 2022.

[76] S. Hecker, D. Dai, and L. Van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 435–453.

[77] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," *arXiv:1604*, pp. 1–9, 2016.

[78] U. Michelucci and F. Venturini, "Estimating neural network's performance with bootstrap: A tutorial," *Machine Learning and Knowledge Extraction*, vol. 3, no. 2, pp. 357–373, Mar. 2021.

[79] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[80] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2020.

[81] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BioMed Central Genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[82] R. Johansson, *System Modelling and Identification.* Prentice Hall, 1993.

[83] H. Lütkepohl, *New introduction to multiple time series analysis.* Berlin: Springer Science & Business Media, 2005.

[84] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis.* Chapman and Hall/CRC, 1995.

[85] L. Ljung, "System identification: Theory for the user," *Prentice Hall PTR*, 1999.

[86] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the arma, arima, and the autoregressive artificial neural network models in forecasting the monthly inflow of dez dam reservoir," *Journal of Hydrology*, vol. 476, pp. 433–441, 2013.

[87] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[88] "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5, pp. 183–197, 1991.

[89] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.

[90] G. E. Hinton and R. Zemel, "Autoencoders, minimum description length and helmholtz free energy," *Advances in neural information processing systems*, vol. 6, 1993.

[91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[92] M. C. Mozer, "A focused backpropagation algorithm for temporal," *Backpropagation: Theory, architectures, and applications*, vol. 137, 1995.

[93] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[94] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[95] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[96] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[97] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," *arXiv preprint arXiv:1801.02143*, 2018.

[98] E. Goan and C. Fookes, "Bayesian neural networks: An introduction and survey," in *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, K. L. Mengersen, P. Pudlo, and C. P. Robert, Eds. Cham: Springer International Publishing, 2020, pp. 45–87.

[99] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, PMLR, 2016, pp. 1050–1059.

[100] P. Bühlmann, *Bagging, boosting and ensemble methods.* Springer, 2012.

[101] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.

[102] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[103] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," *arXiv preprint arXiv:1912.02757*, 2019.

[104] Y. Freund, R. E. Schapire, *et al.*, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, vol. 96, 1996, pp. 148–156.

[105] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[106] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *International conference on Machine learning*, 2004, p. 18.

[107] Y. Li, K. Li, Y. Zheng, B. Morys, S. Pan, and J. Wang, "Threat assessment techniques in intelligent vehicles: A comparative survey," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 4, pp. 71–91, 2021.

[108] J. M. Ambarak, "Predicting unintentional traffic lane departures using neural networks," Ph.D. dissertation, Wayne State University, 2018.

[109] Y. Xing, C. Lv, H. Wang, D. Cao, and E. Velenis, "An ensemble deep learning approach for driver lane change intention inference," *Transportation Research Part C: Emerging Technologies*, vol. 115, p. 102 615, 2020.

[110] V. Ilić, D. Kukolj, M. Marijan, and N. Teslić, "Predicting positions and velocities of surrounding vehicles using deep neural networks," *IEEE Zooming Innovation in Consumer Technologies Conference*, pp. 126–129, 2019.

[111] M. Roth, J. Stapel, R. Happee, and D. M. Gavrila, "Driver and pedestrian mutual awareness for path prediction and collision risk estimation," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 896–907, 2022.

[112] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[113] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2702–2719, 2020.

[114] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi Autonomous Driving Dataset," *arXiv preprint arXiv:2004.06320*, 2020.

[115] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *IEEE conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[116] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[117] (2020). "Next generation simulation hw101," [Online]. Available: `https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm` (visited on 10/06/2022).

[118] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2018, pp. 2118–2125.

[119] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.