Critical Scenario Identification for Testing of Autonomous Driving Systems

Song, Qunying

2022

Critical Scenario Identification for Testing of Autonomous Driving Systems

Qunying Song

Licentiate Thesis, 2022
Department of Computer Science
Lund University

ISBN: 978-91-8039-211-2 (print)
ISBN: 978-91-8039-212-9 (pdf)
Licentiate Thesis 2, 2022
ISSN: **1652-4691**

Department of Computer Science
Lund University
Box 118
SE-221 00 Lund
Sweden

Email: qunying.song@cs.lth.se
WWW: https://cs.lth.se/qunying-song/

Printed in Sweden by Tryckeriet i E-huset, Lund, 2022

ABSTRACT

Background: Autonomous systems have received considerable attention from academia and are adopted by various industrial domains, such as automotive, avionics, etc. As many of them are considered safety-critical, testing is indispensable to verify their reliability and safety. However, there is no common standard for testing autonomous systems efficiently and effectively. Thus new approaches for testing such systems must be developed.
Aim: The objective of this thesis is two-fold. First, we want to present an overview of software testing of autonomous systems, i.e., relevant concepts, challenges, and techniques available in academic research and industry practice. Second, we aim to establish a new approach for testing autonomous driving systems and demonstrate its effectiveness by using real autonomous driving systems from industry.
Research Methodology: We conducted the research in three steps using the design science paradigm. First, we explored the existing literature and industry practices to understand the state of the art for testing of autonomous systems. Second, we focused on a particular sub-domain – autonomous driving – and proposed a systematic approach for critical test scenario identification. Lastly, we validated our approach and employed it for testing real autonomous driving systems by collaborating with Volvo Cars.
Results: We present the results as four papers in this thesis. First, we conceptualized a definition of autonomous systems and classified challenges and approaches, techniques, and practices for testing autonomous systems in general. Second, we designed a systematic approach for critical test scenario identification. We employed the approach for testing two real autonomous driving systems from the industry and have effectively identified critical test scenarios. Lastly, we established a model for predicting the distribution of vehicle–pedestrian interactions for realistic test scenario generation for autonomous driving systems.
Conclusion: Critical scenario identification is a favorable approach to generate test scenarios and facilitate the testing of autonomous driving systems in an efficient way. Future improvement of the approach includes (1) evaluating the effectiveness of the generated critical scenarios for testing; (2) extending the subcomponents in this approach; (3) combining different testing approaches, and (4) exploring the application of the approach to test different autonomous systems.

ACKNOWLEDGEMENTS

I started my PhD study just before the outbreak of the pandemic in 2020. Since then, it has been over two years of remote working, and almost everything has happened online. During this time, doing my research and living here in Sweden under this circumstances made the entire experience even more special. I am happy to reach this milestone – write my licentiate thesis in time, and by this chance, I would like to express my greatest appreciation to some people in my life.

This work was funded by WASP, which is the largest individual research program ever in Sweden. With the investment from WASP and efforts by the administrators, professors, researchers, and PhD students thereof, I had the chance to meet many intelligent and supportive people, take many interesting and useful courses and listen to some inspiring conferences and seminars. It has been a great opportunity for learning and networking, and thank you all for making this happen.

I want to express sincere gratitude to my supervisors, Prof. Per Runeson and Dr. Emelie Engström. There is a Chinese saying which translated as even if someone is your teacher in a day, you should regard him as your father for the rest of your life. Thank you for having me here – for guiding me through the years and helping me in my study. With all the trust, encouragement, flexibility, and advice you gave me, I always believe you are the best to have in my PhD study.

I want to say thank you to the industrial partners I have – Yury Tarakanov from Viscando and Stefan Persson from Volvo Cars, for setting things up for me and supporting me in my research. I would also like to thank everyone in computer science department, and especially the software engineering research group. I want to mention Sergio Rico, Adha Hrusto, Daniel Helgeson, Idriss Riouak, and Monina Rizwan. Without you, the PhD study would not be this wonderful.

Last but not least, it is my greatest fortune to have my family be with me. I am so grateful to have my wife Xiaohong Xu, the stone of the family and the beloved of my life, and my daughter Elina Yan Song who always gives me a lovely smile when I get home. Besides, I will always appreciate my parents, brother, little sister, and friends who have supported me unconditionally all the time.

Qunying Song
Lund, February 2022

LIST OF PUBLICATIONS

This thesis consists of an introduction and a compilation of four papers. The introduction gives an overview of the research topic, and it describes the papers and contributions briefly. Paper I and II have been formatted in this thesis template without additional changes from the original publications.

Publications included in the thesis

I **Concepts in Testing of Autonomous Systems: Academic Literature and Industry Practice**
Qunying Song, Emelie Engström and Per Runeson
In proceedings of IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), 2021.
DOI: 10.1109/WAIN52551.2021.00018

II **An Industrial Workbench for Test Scenario Identification for Autonomous Driving Software**
Qunying Song, Kaige Tan, Per Runeson and Stefan Persson
In proceedings of IEEE 3rd International Conference on Artificial Intelligence Testing (AITest), 2021.
DOI: 10.1109/AITEST52744.2021.00024

III **Critical Scenario Identification for Realistic Testing of Autonomous Driving Systems**
Qunying Song, Kaige Tan, Per Runeson and Stefan Persson
In submission to a journal, 2022.
DOI:10.21203/rs.3.rs-1280095/v1

IV **A Vehicle–pedestrian Time-To-Collision Model for Testing of Autonomous Driving Systems**
Qunying Song, Per Runeson and Stefan Persson
DOI: Working manuscript

Related Publications

V **Exploring ML testing in practice – Lessons learned from an interactive rapid review with Axis Communications**
Qunying Song, Markus Borg, Emelie Engström, Håkan Ardö and Sergio Rico
To be presented at CAIN, 2022.

Contribution statement

All papers included in this thesis have been co-authored with other researchers. The authors' individual contributions to Papers I-IV are as follows:

Paper I
Qunying Song, Emelie Engström, and Per Runeson designed the study together. Qunying Song did the literature review and led the interviews. Emelie Engström and Per Ruenson coordinated the focus group discussion and participated in the interviews. Qunying Song synthesized the results and wrote the initial version of the paper. All authors have contributed to reviewing and revising the paper.

Paper II
Qunying Song and Per Runeson designed the study. Qunying Song investigated the literature and developed the approach proposed in this study. Stefan Persson supported in tools and environment and joined the discussion, and Kaige Tan provided input to the approach. Qunying Song and Per Runeson wrote the paper together. Kaige Tan and Stefan Persson helped in reviewing the manuscript.

Paper III
Qunying Song and Per Runeson designed the study. Two cases were conducted in this study. Kaige Tan conducted the first case and wrote the chapter (i.e., Section IV) for that in the paper. Qunying Song conducted the second case, and Stefan Persson supported with tools and environment and information about the case system. Qunying Song wrote the initial version of the paper except for the first case. All authors have contributed to reviewing and revising the paper.

Paper IV
Qunying Song, Per Runeson, and Stefan Persson designed the study. Qunying Song studied the field, established the model, and validated it using real driving data collected by Viscando. Per Runeson and Stefan Persson supported planning the study, providing advice and related resources. Qunying Song wrote the initial version of the paper. All authors have contributed to reviewing and revising the paper.

CONTENTS

INTRODUCTION

1 Background

Autonomous systems have received considerable interest in academic research and have been used in different application domains, such as automotive, avionic, and robotics [43]. A fully autonomous system, for example, a level-5 autonomous vehicle by SAE International[1], has to properly handle various situations in real road traffic without a human driver [26], including those hazard occasions where an immediate reaction is required to avoid colliding with other road users or infrastructures.

Despite the widespread attention and popular trend of using autonomous systems for different purposes, many autonomous systems are considered both safety-critical and mission-critical. Thus effective testing is essential for verifying their safety and reliability. Inadequate and ineffective testing may fail to discover the defects and misbehavior of the system, consequently leading to severe accidents or significant economic losses [99]. Examples of such failures are the crash of ExoMars Mars Lander in 2016, which was analysed to be an implementation error and an estimated 350 million US dollar in loss [48], and fatal accidents caused by Tesla and Ubers' autonomous vehicles where frontal objects on the road were not correctly identified and consequently hit by the vehicles [64, 110].

Fully autonomous systems need to operate in unanticipated environments without human supervision and adapt their behaviors accordingly to fulfill the tasks appropriately. Efficient testing of such systems is notoriously tricky, and exhaustive testing is impractical due to the uncountable number of situations that may occur [45]. Taking autonomous driving systems as an example, the number of scenarios in real-world traffic is potentially infinite, and identifying all possible scenarios as well as covering them in testing is impractical, if not impossible [35, 36].

The advancement of emerging technologies like machine learning have fostered the development of highly autonomous features, for example, real-time object detection based on the perceived environment and stepwise decision-making

[1] https://www.sae.org/

through reinforcement learning. Nevertheless, it also adds an order of magnitude of complexity and uncertainty for testing autonomous systems [62]. There is no definitive way for testing or certifying such systems yet [45,96]. Thus, new testing techniques and approaches are developed and need to be extended to evaluate the safety and reliability of autonomous systems, for example, using simulation and critical scenario-based testing approach.

2 Research Goals

The general goal of this thesis is to improve software testing of autonomous systems by exploring and developing new techniques and approaches for testing. Accordingly, we need to understand the current state of the area, design a new solution and validate its feasibility in a real setting. Based on that, we have defined three research goals in a sequential order. The first goal is to explore the challenges and available techniques for testing autonomous systems in general and identify research gaps. Subsequently, the second and third goals are to design an intervention to address the gaps for testing by focusing on a sub-domain of autonomous systems – autonomous driving – and validate the intervention in an industrial context by collaborating with the automaker Volvo Cars.

2.1 Goal I – Exploration of the Field

Given the open challenges of testing autonomous systems and that no broad survey in this field was found, the first goal is to explore this area and get an overview of the state of the art. On one hand, this goal aims to get a broad view of the field by surveying different types of autonomous systems. On the other hand, this goal explores both existing academic literature and industry practices to get different perspectives. Achieving this goal helps us to understand the field and identify the research gaps as well as to search for solution candidates.

Specifically, relevant concepts for testing autonomous systems need to be defined, challenges and available techniques, as well as approaches and practices, need to be understood and classified to set a basis for designing feasible solutions for testing autonomous systems.

2.2 Goal II – Designing the Solution

Testing is significant for verifying the reliability and safety of autonomous systems, yet no common and definitive way for testing such systems has been established. Thus there is a need for studying and developing new testing techniques and approaches. The second goal is to design a solution based on existing tools and techniques to support the testing of autonomous systems effectively and efficiently.

Since different autonomous systems employ different functional and safety requirements and operate in distinct operational environments, this goal focuses on one particular sub-domain of autonomous systems – autonomous driving – and provides a systematic approach to facilitate testing of different autonomous driving systems.

2.3 Goal III – Validating the Solution

The third goal is to validate the intervention in a real industrial context. We do this by applying the approach in an industrial setting and demonstrate its feasibility for testing real autonomous driving systems. By reaching this goal, we show that the solution we provide in the thesis is feasible to use in practice and can support testing effectively.

Addressing this goal relies on access to real industrial context and autonomous driving systems from the industry. We collaborate with the automaker Volvo Cars and verify that our approach is feasible and effective for testing real autonomous driving systems from them. In addition, the validation also gives insights on what can be extended or refined in this approach in future work.

3 Related Work

This section first presents the existing studies that report software testing of autonomous systems in general and autonomous driving systems in particular. The second part describes scenario-based testing approaches and how they are used for testing the autonomous driving systems based on the existing literature.

3.1 Testing of Autonomous Systems

Existing studies have explored the characteristics of autonomous systems and the challenges and approaches for testing such systems. Among them, Helle et al. present an overview of autonomous systems and claim that traditional testing approaches, that aim for fault prevention, detection, and removal, are insufficient for autonomous systems because of unknown situations that may happen during operation and their dynamically changing behaviors [45]. They also investigated testing techniques and report mostly model-based approaches for testing autonomous systems. Harel et al. explored challenges for testing autonomous systems in general. They call for a foundation for testing those systems to address the challenges of specifying and analysing system behaviors, as well as to combine the model-based and data-driven approaches [43]. In addition, Sifakis defined an architecture of autonomous systems and theories how such systems would be trustworthy [95, 96].

Koopman et al. and Knauss et al. have surveyed mainly the challenges and provide inputs for testing autonomous driving systems. Koopman et al. list the

major difficulties for testing as (1) infeasibility of exhaustive testing, (2) no human driver involved, (3) complex system requirements, and (4) non-deterministic algorithms used, such as machine learning techniques [62]. In contrast, Knauss et al. explored the challenges of testing autonomous driving systems based on existing literature, and focus groups and interviews with industry practitioners. Among the 13 challenges they extracted, how simulation tests can smoothly support realistic testing of autonomous driving systems, and the complexity and explosion of test scenarios, are ranked the two most prominent challenges for testing [61].

In order to address the said gaps, Rajabli et al. present a systematic literature review on software verification for autonomous driving [84]; Kang et al. conducted a comprehensive survey of 22 simulation platforms and 37 available data sets for testing autonomous driving systems [56]. Similarly, Rosique et al. did a systematic literature review of the simulators and perception systems for testing autonomous driving systems [86]. In addition, Bhat et al. present tools and methods for testing autonomous vehicles at different engineering stages [15]. Still, there is no universal definition of what constitutes an autonomous system and which challenges and techniques exist for testing autonomous systems in general. Exploring existing literature and industry practices, which is the first goal of this thesis, is critical to understand and improve the field testing of autonomous systems.

3.2 Scenario-based Testing Approaches

Common approaches for testing autonomous driving systems include substantial real-world testing that places the system in its real operational environment and continuously observes the system's performance under different situations, or collecting real driving data at a large scale to enable testing and analysis in simulation. Kalra et al. have modeled the distance of driving tests needed to prove the safety of autonomous vehicles compared to human accident rates [55]. They argue that millions up to billions of miles of driving tests are needed. However, it is implausible for automakers to conduct that amount of driving test in a cost- and time-efficient way, especially since some critical situations are rare in real road traffic and may still not be covered during tests [58, 81]. Similarly, collecting driving data from real traffic at scale is also expensive and time-consuming, yet the quality of the collected data and mechanisms for data aggregation still need to be studied.

Using simulation and scenario-based testing is considered a promising alternative to complement the approaches mentioned above and facilitate the testing of autonomous systems in an efficient way [84]. Simulation enables early verification of the autonomous driving systems without accessing the vehicle and real traffic and minimizing the risks of harming other road users. Although simulation has its limitations – an evident one is low fidelity and limited representation of the real-world complexities – simulation supports testing of the basic implementation and system behavior prior to deploying the vehicles in real road traffic [61].

The scenario-based testing approach plays a crucial role in testing autonomous driving systems, which aims to reduce the testing effort into a manageable number of scenarios [85]. Instead of spending testing resources on repetitive scenarios that expose very low safety risks and require no urgent reaction, critical scenario-based testing focuses on identifying and testing those most critical scenarios that might cause collision or near-collision situations or consequences. Ulbrich et al. define a *scenario* as a temporal sequence of scenes representing the world model that includes the road, road users, infrastructures, environment, and weather etc. [105]. Menzel et al. extended this definition into three different abstraction levels [72]. *Functional scenarios* are usually described in natural language, and *logical scenarios* are parameterization of functional scenarios by identifying the relevant parameters and parameter range and distribution. *Concrete scenarios* are instantiations of logical scenarios by assigning concrete values to the parameters.

Different techniques are used to identify or generate critical scenarios for testing autonomous driving systems. Riedmaier et al. [85] present studies on scenario-based approaches, and Zhang et al. [114] on critical scenario identification approaches, which both are systematic literature reviews for autonomous driving systems. Examples of the surveyed approaches include using deep learning for generating critical test scenarios and search-based algorithms to optimize the generation of critical scenarios. Scenarios are executed and evaluated as critical with different criteria, for example, using surrogate measurements like Time-to-Collision (TTC) or Post-Enchroament-Time (PET) [107]. Functional specifications and related industrial standards (e.g., ISO-26262, ISO/PAS-21448, Responsibility Sensitive Safety) can also be used to derive criticalities for the intended functions.

Although existing studies have reported critical scenario identification for testing autonomous driving systems using different techniques, very few studies provide a complete solution for it [41]. Existing studies have focused on solving parts of the critical scenario identification, such as scenario optimization or improving the scenario representation and simulation, thus not providing a complete approach for identifying critical scenarios for testing autonomous driving systems. Besides, existing studies do not validate their approaches in an industrial context with real autonomous driving systems. Therefore, it is significant to address such gaps and design a systematic approach for critical scenario identification that is generic for testing different autonomous driving systems. In addition, the approach should be validated in industrial context and be feasible to support the testing of real autonomous driving systems.

4 Research Methodology

We used the design science paradigm [88] to guide the work gradually from the problem domain to the solution domain. The design science paradigm is described as a frame for depicting, analyzing, and communicating software engineering re-

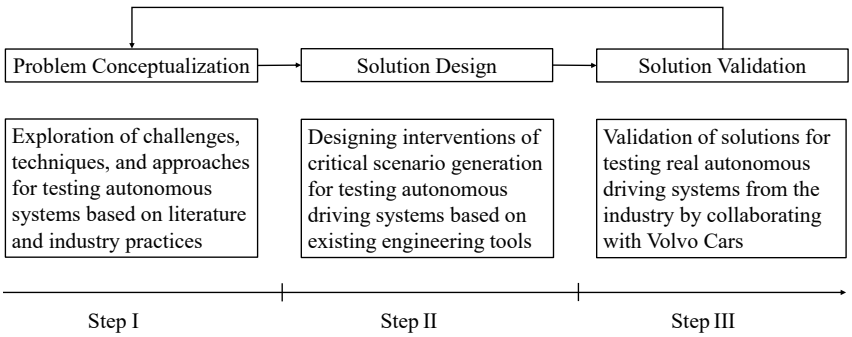


Figure 1: Mapping of the three research steps (i.e, bottom text boxes) and corresponding research activities (i.e., middle edged text boxes) into the design science elements (i.e., top text boxes connected with arrows). The arrows between different elements indicate how one activity can support or provide input to the next.

search and contains three major elements – problem conceptualization, solution design, and validation. A technological rule (TR) is often used in design science to describe the research contributions. Technological rules can be extracted at different level of abstractions and are usually presented in the form of *TO ACHIEVE <effect> IN <context> DO <intervention>* [88].

We formulated two technological rules in this thesis as listed below, including a general one (i.e., TR 1) with a broad scope and a high level of abstraction, and a concrete one (i.e., TR 2) with a more narrow scope and concrete intervention.

- *TR 1: To improve software testing of autonomous systems, explore and design new testing approaches.*
- *TR 2: To test autonomous driving systems effectively and efficiently, identify critical test scenarios in simulation.*

We have conducted the thesis in three research steps. We summarize the research activities for each step and how the activities are mapped under each element of the design science paradigm in Figure 1. We explored and conceptualized the problems for testing autonomous systems in the first step. Then we focused on the solution domain in the second step and designed a critical scenario identification approach to address the explosion of test scenarios. In the third step, the approach was employed to test real autonomous driving systems as validation and was extended to improve the sub-components further.

4.1 Step I: Problem Conceptualization

The first step of the thesis is essentially aligned with the first goal as described in Section 2 – to explore the field testing of autonomous systems, identify existing

problems, and investigate any technical solutions available. In this step, we have to understand:

1. How are autonomous systems defined, and which concepts exist?
2. Which challenges for testing autonomous systems remain?
3. What techniques, approaches, and practices for testing autonomous systems are available?

This step had a broad scope to include findings and insights from different sources as an exploratory stage. We first studied the existing literature on software testing of autonomous systems and then explored the industry practices by having a focus group discussion and interviews with the industry practitioners. Results of this step is Paper I which we summarize in Section 5.1, where concepts of autonomous systems were surveyed, challenges and available techniques, approaches, and practices for testing autonomous systems were classified.

4.2 Step II: Solution Design

Based on the previous step, we identified a significant challenge that impedes the testing of autonomous systems – the infinite number of test scenarios due to the unpredictable and complex environment that is unknown during design. The second step was to create an intervention to address that gap – generating critical test scenarios for autonomous systems, based on existing tools and techniques. As introduced earlier in Section 3, using simulation and scenario-based testing approaches is considered a promising alternative to reduce the testing effort into a manageable number of scenarios and focus on the most critical ones in the test.

In this step, we focused on a particular sub-domain of autonomous systems – autonomous driving – and set our study in an industrial context by collaborating with the automaker Volvo Cars. We studied different engineering tools based on their environment, such as SPAS simulation platform and modeFroniter – a process optimization tool. We integrated the tools and defined a workflow to form a complete approach for critical scenario identification for testing autonomous driving systems. The approach is reported in Paper II and summarized in Section 5.2. The tools employed are exchangeable, meaning that any tools involved can be replaced with other similar tools. Thus, the approach is generic for testing different autonomous driving and not subject to a particular function, tool, or technical environment that is intended.

4.3 Step III: Solution Validation

The third step validated the feasibility of the solution from the previous step in an industrial context and is reported in Paper III, which is summarized in Section 5.3. Specifically, our industrial partner Volvo Cars provided an early version of two

autonomous driving systems, namely an autonomous driving system and an autonomous parking system. The approach was then employed to generate critical scenarios which can be used to test those systems either in a simulated environment or in real road traffic.

Going beyond validating the approach's feasibility; the validation step also gave insights into what components were missing when implementing such an approach in practice, e.g., realistic distribution of parameters for scenario generation. Subsequently, we established a model that predicts the worst-case TTC distribution of vehicle–pedestrian interaction and demonstrated its potential use for testing, e.g., autonomous emergency braking systems. The model is presented in Paper IV, which is summarized in Section 5.4.

5 Results and Contributions

The research has resulted in four papers, two of which are published, and two are released as pre-print manuscripts. Paper I presents the results of the first step as described in Section 4, paper II the results of the second step, and paper III and IV are pre-prints that describe the work and results for the third step.

5.1 Paper I: Concepts in Testing of Autonomous Systems: Academic Literature and Industry Practice

In this paper, a pool of 45 papers on software testing of autonomous systems were synthesized with a focus group discussion of eight industrial practitioners and interviews with five experts in the autonomous domain. As a result, we conceptualized a definition of autonomous systems, and we classified the challenges, techniques, approaches, and practices for testing autonomous systems in general.

Our conceptualization defines autonomous systems, as systems that can fulfill specific tasks within an unstructured environment without human supervision. Four aspects of the systems are articulated – (1) self-aware of the environment and system states; (2) decision-making based on analysis of the situation; (3) adaptation based on goals and history; and (4) actuation of the plans derived from decision-making. The main challenges for testing autonomous systems are: the unpredictable environment, the complexity of the system requirements, design, and operation scenarios, data accessibility, and missing standard guidelines for testing. Examples of available techniques and approaches for testing autonomous systems include engineering recommendations, such as using simulation and the V-model paradigm, and techniques, such as model-based approaches, combinatorial testing, scenario-based approaches, and metamorphic testing.

As there is no universal definition that describes autonomous systems in general, one contribution in this paper is the conceptualization of what constitutes an autonomous system. We also provide a comprehensive list of techniques, ap-

proaches, and practices for testing autonomous systems. The results of this paper addressed the first goal of the thesis and helped us understanding the area and supported the design of our intervention in later steps. We noticed that testing of autonomous systems is intricate and the potential test scenarios are extensive, and using simulation and scenario-based testing is considered an efficient alternative to reduce overall testing effort and cost.

5.2 Paper II: An Industrial Workbench for Test Scenario Identification for Autonomous Driving Software

In this short paper, we established a workbench for critical scenario identification for testing autonomous driving systems based on our collaboration with Volvo Cars. The workbench integrates three existing engineering tools and a workflow for critical scenario identification. The tools involved are exchangeable, meaning that they can be substituted with any similar tools, so the workbench is, in principle, generic for testing any autonomous driving systems.

The three tools used include: (1) A requirement and verification management tool stores the system specifications, design documents, and testing artifacts; (2) A simulation platform – SPAS – simulates the scenario and records the simulation results; (3) An optimization tool – modeFrontier – optimizes the generation of scenarios based on the objective functions and simulation results.

The workflow starts by analyzing the system specifications and the operational environment in the requirement and verification management tool. Relevant parameters in the operational design domain are selected, and appropriate objective functions are defined to evaluate the criticality of a scenario, e.g., TTC. With the selected parameters and their value range and distribution, an initial suite of test scenarios can be generated based on a given sampling strategy and the intended size of the test suite. Next, scenarios in the initial test suite are executed in the simulation platform and the optimization tool generates new scenarios based on the completed simulation and the objective functions defined. In the end, the scenarios beyond the criticality thresholds are considered critical and can be used to substantiate test cases for testing autonomous driving systems.

The contribution of this paper is the implementation of a workbench that provides a systematic approach for critical test scenario identification for autonomous driving systems. The workbench identifies the most critical scenarios in simulation and supports the testing in an efficient way. Particularly, the workbench provides a complete tool chain and is generic for testing different autonomous driving systems. In addition, this paper also addresses the second goal of the thesis – to design a solution to tackle the existing gaps of testing autonomous driving systems.

5.3 Paper III: Critical Scenario Identification for Realistic Testing of Autonomous Driving Systems

In this paper, we applied the workbench for critical scenario identification for testing two real autonomous driving systems from Volvo Cars, namely an autonomous driving function and an autonomous parking function. The two functions provided were still in the early version of their development.

The autonomous driving function mainly provides driving in lane and speed adaptation according to the maneuver of the surrounding vehicles. Relevant parameters were selected, such as the number of vehicles in a scenario, and the initial position, velocity, and acceleration of vehicles involved. Two objective functions – TTC and jerk (i.e., acceleration rate) were used to measure the scenario's criticality in terms of collision probability.

The autonomous parking function scans the empty parking slot using the ultrasonic sensors installed. The host vehicle (a.k.a., ego-vehicle) is then parked into the slot by controlling the steering wheel, throttle, proposition, brake pedal, etc. The two parameters selected for this function are the parking slot length and yaw angle of the stationary vehicles near the target slot. According to the ISO-16787 standard, a minimum of 30 cm's distance to stationary vehicles and a yaw angle within 3 degrees to the central line of the parking slot defines the required capability of autonomous parking functions.

We created the optimization models for these two functions in the modeFrontier optimization tool using an optimization algorithm pilOPT, and have effectively identified critical test scenarios for both functions mentioned. In addition, we replicated the optimization models using another algorithm MOSA to compare the two different optimization algorithms. While no significant differences are consistently observed, pilOPT performed slightly better than MOSA in both cases.

The contribution of this paper is validation of the approach we proposed in Paper II for critical test scenario generation and the application of this approach for testing real autonomous driving systems. The results indicate that our approach is feasible and effective in identifying critical test scenarios. We also observed some future improvements to extend this approach, one is to use realistic distribution of parameter when generating critical scenarios, which is presented in the following section.

5.4 Paper IV: A Vehicle–pedestrian Time-To-Collision Model for Testing of Autonomous Driving Systems

Realistic distribution of parameters is required in scenario-based testing of autonomous vehicles, since the distribution of parameters decides the exposure of a scenario in reality. A different distribution of a parameter can change the probability of a scenario, thus would affect the potential criticality of it. Yet, realistic distribution of parameters, e.g., TTC distribution of vehicle–pedestrian interac-

tions in road traffic, are not provided in many cases. We established a model of the TTC distribution for vehicle–pedestrian interactions using the Poisson distribution and demonstrated its potential use for testing autonomous driving systems.

The model takes the mean arrival rate of vehicles and pedestrians as input and predicts the worst-case distribution of TTC based on the Poisson distribution of vehicles and pedestrians. By worst-case, we mean careless drivers and pedestrians that are not paying enough attention to the ongoing traffic due to fatigue, distraction, or drunk driving. A real-life example of such a situation is the fatal accident from Uber's autonomous vehicle, where the vehicle failed to detect the cyclist in front and the cyclist crossed the road without carefully evaluating the risks of the oncoming vehicle. We demonstrated the use of this model to test an autonomous emergency braking function from Volvo Cars where TTC to the frontal objects is one of the parameters for activating this function. The model provides the worst-case TTC distribution in a given traffic and enforces realistic distribution of parameters for generating or sampling scenarios for testing purposes.

We validated the model with real driving data from Viscando AB, and the model consistently dominated the real distribution of critical TTC (i.e., TTC < 3 seconds). The validation results indicated that the model could serve a worst-case distribution in real road traffic. Given that pedestrians have become one of the most vulnerable user groups on the road, the safety assessment of autonomous vehicles has to pay proper attention to the vehicle–pedestrian interactions. Modeling the realistic distribution of vehicles–pedestrian TTC can thus be a valuable input for testing autonomous driving functions like emergency braking.

6 Limitations and Discussion

We started the thesis with a broad scope of testing of autonomous systems in general and then focused on critical scenario identification for testing autonomous driving systems in solution design. Different types of autonomous systems employ different techniques and components; they provide different functionalities and operate in distinct operational environments and follow separate safety regulations. The proposed approach for critical test scenario identification has not yet been applied to other applications within the autonomous domain. Nevertheless, given that the explosion of test scenarios is a common challenge for fully autonomous systems, we believe the critical scenario identification approach is a meritorious alternative to address the testing of such systems.

The critical test scenario identification approach we provide involves a simple scenario representation, where initial values of the parameters are selected, and a static driver behavior model is implicitly assumed. The driver behavior model can be diverse and adaptive in real road traffic based on the traffic dynamics and change of the weather, road geometry, etc. Without a realistic behavior model, the optimization might generate scenarios that do not represent real-world situations,

e.g., suicide scenarios with unavoidable collisions at the beginning of the scenarios. Thus, our approach can be extended by incorporating realistic driver behavior models. Nonetheless, the thesis has demonstrated the feasibility of the approach and provides a basis for future extension of it.

One may also argue that different optimization algorithms can differ in their effectiveness and efficiency for finding the parameter sets of the most critical scenarios. Even though it is not a goal in this thesis to find the best optimization algorithms, different algorithms can be compared and evaluated for best fit in critical scenario identification for different autonomous driving systems. In addition, a clear gap for most existing studies is that no evaluation of the generated critical scenarios is provided. In other words, the proposed approaches have not been integrated into the actual testing process of the autonomous driving systems, thus no evidence is provided on how effective they are. Even though numerous critical scenarios were identified for the given systems using the approach we designed in this thesis, we need to better understand how the critical scenarios are used in practice and how the approach supports the quality assurance and safety assessment of autonomous vehicles. However, this is not an easy task and calls for deep collaboration with industrial players as well as getting into their engineering environment and processes of development of autonomous driving systems.

7 Future Work

The goals of the thesis are to explore the field testing of autonomous systems, design the solutions to address the current gaps, and validate its feasibility in real industrial context. We followed the design science paradigm and conducted the work in a series of incremental steps. The main outcomes are a comprehensive overview of software testing of autonomous systems and a systematic approach of critical scenario generation for testing of autonomous driving systems. As described in the previous section about the general limitations of the thesis and the approach thereof, future work can be derived from multiple perspectives.

Several research items can be defined for continuation on the second goal of the thesis – solution design – and improving the realism of the generated test scenarios. One is to extend the scenario representation to include more realistic driver behaviors to foster realistic interactions between vehicles in a complete driving scenario. A possible way to tackle that is to derive driver behavior models based on real driving data collected, and include such models for vehicles when executing the scenarios in simulation. By doing so, we could evaluate how realistic driver behavior models can affect or improve the generation of critical test scenarios.

Second, comparing different algorithms and identify which best fits a particular autonomous driving system for critical scenario generation. This is not a goal in the current thesis as already mentioned earlier, but a good input to our approach when selecting optimization algorithms. Particularly, a certain algorithm may out-

perform the others in efficiency and effectiveness to optimize generation of critical scenarios for a specific autonomous driving system. Thus, a comparative study to evaluate different algorithms would help us to observe the differences between them and identify the appropriate algorithm to use.

Thirdly, the critical scenario identification may be complemented by other approaches, such as the coverage-based approach, so the testing covers not only the critical test scenarios, but also the general types of scenarios. Efficient testing of different types of scenarios is a systematic way of verifying the safety and reliability of autonomous driving systems in various situations. One possible direction is to explore the effect of combining different testing approaches to maximize the scenario coverage for the safety assessment of autonomous driving systems.

Lastly, one more research item that is significant to address, is the third goal of the thesis – solution validation, which addresses the effectiveness of the approach in practice and the generated critical scenarios for testing purposes. We aim to integrate the critical scenario identification approach into the actual testing process of autonomous driving systems and see how the approach can support the quality assurance for such systems in actual engineering practice.

INCLUDED PAPERS

PAPER I

CONCEPTS IN TESTING OF AUTONOMOUS SYSTEMS: ACADEMIC LITERATURE AND INDUSTRY PRACTICE

Qunying Song, Emelie Engström and Per Runeson, In proceedings of IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), 2021.
DOI: 10.1109/WAIN52551.2021.00018

Abstract

Testing of autonomous systems is extremely important as many of them are both safety-critical and security-critical. The architecture and mechanism of such systems are fundamentally different from traditional control software, which appears to operate in more structured environments and are explicitly instructed according to the system design and implementation. To gain a better understanding of autonomous systems practice and facilitate research on testing of such systems, we conducted an exploratory study by synthesizing academic literature with a focus group discussion and interviews with industry practitioners. Based on thematic analysis of the data, we provide a conceptualization of autonomous systems, classifications of challenges and current practices as well as of available techniques and approaches for testing of autonomous systems. Our findings also indicate that more research efforts are required for testing of autonomous systems to improve both the quality and safety aspects of such systems.

1 Introduction

Autonomous systems are expected to replace humans in carrying out a variety of functions [43], and can be central and crucial for different industry domains such as automotive, robotics, and aviation. Advances in machine learning and artificial intelligence have enabled an overwhelming progress for such systems. There are already prototypes of autonomous vehicles that are tested on the road, and autonomous systems have replaced humans to a significant extent for decision-making in investment markets, particularly for asset management [96].

While autonomous systems are becoming prevalent and have enormous potential for the society, how to test these systems is not resolved yet due to the unpredicted environment they operate in, and their adaptive behaviour [45]. One problem is that no industrial standards or common approaches have been settled for testing of such systems [5, 61]. Further, research conducted on autonomous systems tend to be conducted in isolation from industry practice, for example, purely in simulation environments [23].

To better understand the essence of autonomous systems and the current status of testing of such systems, we conducted an exploratory study by synthesizing academic literature, focus group discussions and interviews with industry practitioners. *Our contribution is a synthesis of autonomous systems concepts, their characteristics and functionalities, empirically grounded in research and practice. We also classify challenges, approaches, techniques, and practices available for testing of autonomous systems.* The results indicate that the current state of testing such systems is far from being desirable and it is in need of major improvements. Our synthesis aims at providing tools for industry and academia to align communication on the topic, and to jointly meet the need for more knowledge.

While similar studies have been conducted in related areas they are either not focusing specifically on autonomous systems, nor on their testing. Most of the existing studies we found were either focusing on only one autonomous domain, for example, self-driving cars [61], or a particular aspect of the system, like safety [17]. Our results are comprehensive with respect to the testing of autonomous systems in general, and are inclusive towards both academic research findings and industrial practices. We believe they can serve as a good framework both for future research and industrial development.

2 Related Work

Helle et al. present an overview of autonomous systems and testing approaches for such systems, mostly from an avionic perspective [45]. In their paper, the authors introduced the concepts and characteristics of autonomous systems as well as the challenges for testing them. They conclude that, due to the dynamically changing environment and system behaviour, conventional testing approaches that aim for

fault avoidance, removal, and tolerance are infeasible to ensure the quality of autonomous systems. In addition, they surveyed existing approaches and presented mainly model-based testing approaches and related tools. We extend their insights by including also industry practices and by exploring numerous techniques, approaches, and engineering practices for testing of such systems beyond the model-based approaches.

Knauss et al. conducted an empirical study aiming to collect testing challenges for autonomous vehicles [61]. Similar to our study, they combined a literature review with focus groups, and interviews with both researchers and practitioners. Our study is broader, focusing on autonomous systems in general and have an explicit goal to extract existing techniques and approaches in addition to the challenges.

Borg et al. [17] present challenges and approaches for testing of deep learning based automotive applications. Their results point to safety cages as a promising solution to be investigated further. The study contains a systematic literature review of 64 papers on safety analysis or verification and validation of machine learning based autonomous cyber-physical systems. Six workshops were conducted with practitioners from the automotive domain, in order to bridge the gap in understanding the state-of-the-art and obstacles on the way forward.

Zhang et al. report a systematic literature review on testing and verification of neural-network-based safety-critical cyber-physical systems [112]. Their study includes 83 papers from 2011 to 2019. The authors present an overview of different neural networks and a summary of existing approaches for verification of such systems as well as their pros and cons. Another similar study was conducted by Zhang et al. [111] with focus on testing of machine learning systems. In this study, the authors surveyed 144 papers between 2014 and 2019 on testing and verification of machine learning systems, and provided an overview and classifications of techniques and approaches that are employed in research. As a comparison, our study focuses on autonomous systems, independently if they are driven by machine learning technologies or not.

3 Research Methods

We launched a multi-method study, consisting of a semi-systematic literature review, a focus group discussion, and four interviews, as shown in Figure 1, conducted in the given order. We analyzed the collected data qualitatively and established one thematic model per activity [90]. Then the outcomes from all the three activities were compared and aggregated into a coherent outcome by the end. Our study is guided by three research questions, aimed to build on and complement related work, as defined in Section 2:

RQ1 How is the concept of autonomous systems defined?

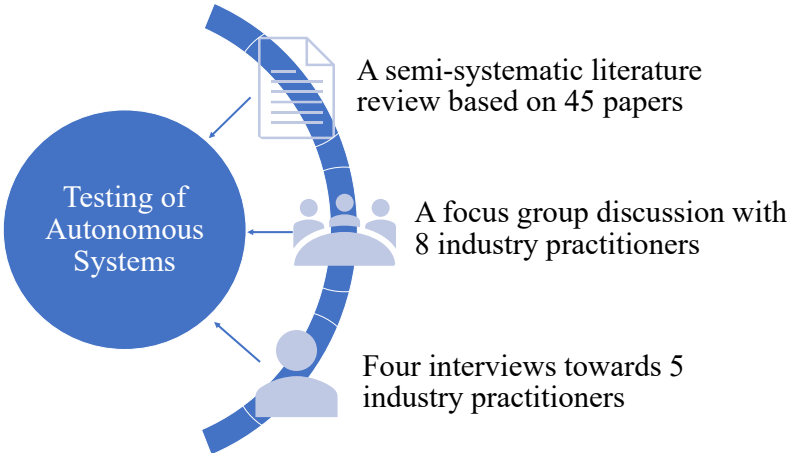


Figure 1: Overview of Research Methods

RQ2 Which are the principal challenges related to testing of autonomous systems?

RQ3 What approaches and practices for testing of autonomous systems are used or proposed?

3.1 Literature Review

Paper Selection

We conducted a semi-systematic literature review [97], where the research questions are broad and the searching and selection process is flexible. Due to the relative immaturity of the research field, the purpose was to retrieve an overview rather than to aggregate specific evidence. Thus peer-reviewed research papers as well as book chapters, were included and referred to as "papers" below.

We used IEEE Xplore, ACM, Scopus, Wiley, and Web of Science as the indexing services for finding the literature. Our search criteria "testing of autonomous systems" was applied to titles, abstracts and keywords. Titles and abstracts were examined for inclusion/exclusion based on relevance. We also used backward snowballing to track more relevant papers. In total, we identified 45 papers as listed in the complementary material [3] in which the majority of them were acquired through the initial search results.

Table 1: Overview of participants in the focus group

#	Position	Domain	Experience
1	Manager	Health Care	25-30 years
2	Test Specialist	Software Engineering	20-25 years
3	Researcher & Engineer	Software Engineering	10-15 years
4	Researcher & Engineer	Logistics	0-5 years
5	PhD Cand. & Engineer	Software Engineering	0-5 years
6	Professor	Artificial Intelligence	20-25 years
7	Senior Researcher	Software Engineering	5-10 years
8	PhD Candidate	Robotics	0-5 years

Analysis and Collation

The first author saved all selected papers into Zotero, and studied the full-texts. For each of the papers, important segments discussing the features under study were highlighted, then short labels and, detailed notes if necessary, were created in Zotero. The labels were then refined and sorted to be coherent, consistent, and distinctive codes as unclear and duplicate entries were removed. Lastly, they were imported in XMind for thematic synthesis based on the guidelines by Cruzes et al. [27]. A thematic model was created with 84 codes that organized around the three research questions and reviewed by the second and the third authors. Details of the thematic model can be found in the complementary material [3].

3.2 Focus Group

Participants Selection

We arranged a focus group [28] discussion in April 2020 to get insights from industry practitioners. Participants were selected using convenient sampling [38] based on an invitation towards a network of testers in Southern Sweden. In total, 8 participants joined the focus group, as summarized in Table 1.

Implementation

Due to the pandemic, the focus group was conducted via Zoom. We started with a general introduction about this study and testing of autonomous systems; then the three research questions were discussed, each devoted about 30 minutes. For each question, first, participants discussed the question in breakout rooms with 2-3 persons each and wrote answers on Padlet; second, all participants were brought back to the main session and the moderator led a discussion to elaborate and expand the answers on Padlet. The focus group was video recorded with consensus from all participants and the Padlet notes were saved by the end.

Result Analysis and Synthesis

We adopted the inductive thematic synthesis approach proposed by Cruzes et al. [27] for coding. The first author conducted the primary coding. First, the recorded videos were reviewed, and notes were taken; Padlet answers were then combined to generate the codes. Second, the codes were imported and anlyzed in XMind for thematic synthesis. A thematic model was created with 37 codes that organized around the three research questions. The resulting model was reviewed by the second and the third authors, and can be found in the complementary material [3].

3.3 Interviews

Participants Selection

To validate and complement the findings from the previous activities, we conducted four interviews with industry practitioners that were not involved in the focus group. Now, we specifically approached experts in our network who had worked with autonomous systems. Five interviewees from industry accepted the invitation as shown in Table 2, where #3 and #4 were interviewed together.

Implementation

The interviews were conducted on Zoom by two of the authors each. Two of the interviews were 60 minutes in length and the other two lasted for 45 minutes. The interview schema was semi-structured, guided by Runeson et al. [90], and Rowley et al. [87]. The interview questions, as listed in the complementary material [3], were derived from the literature review and focus group discussion. The interviews were video recorded with consent from the interviewees.

Result Analysis and Synthesis

We used the same synthesis approach [27] as above. Recorded videos were first transcribed in Nvivo, and important segments of the text were highlighted and coded, resulting in 62 codes. They were imported in XMind for thematic synthesis, where, duplicate and unclear entries were removed, and a thematic model with the codes was created and organized around the three research questions. Lastly, the resulting model was reviewed by the second and third authors, and details of the model can be found in the complementary material [3].

3.4 Final Thematic Model Synthesis

The thematic models generated from the three activities were reviewed, refined, and further compared to eliminate any potential conflicts across these models in XMind. Then they were synthesized into a final thematic model. First, the distinctive themes and codes from the three models were identified and moved to the

Table 2: Overview of participants of interviews

#	Position	Domain	Experience
1	Industrial PhD Candidate	Automotive	5-10 years
2	Solution Architect	Automotive	5-10 years
3	Lead Software Architect	Mobility	20-25 years
4	Software Architect	Mobility	20-25 years
5	Technical Manager	Manufacturing	20-25 years

final model, then the rest of the themes and codes, which share the same or similar purpose, were analysed further to either be added to the final model or merged with other codes. Second, the final thematic model was reviewed to ensure that it integrates the themes and the codes from all three models, and resulting codes remain being coherent, consistent and distinctive.

3.5 Validity

As reported above, we have used systematic research methods to improve the validity of the synthesized conceptual model. Being an exploratory study, aiming to understand concepts and practices, we value *construct validity* highest. To strengthen the construct validity, we have asked open questions, not relying on predefined terms and concepts. The *reliability* or *trustworthiness* of the study is addressed through rigorous data collection and analysis procedures. The analysis of the data took place in three steps, each focusing on one source of empirical evidence (literature, focus group, and interviews) to ensure that the concepts may emerge from the empirical source, which thereafter were unified into one conceptual model. The *external validity* is related to the scope of the model. We have extended the scope beyond the most prevalent automotive domain, by searching the literature broadly, interviewing people from other domains, and explicitly asking for autonomy concepts in virtual only domains, like stock markets. However, we don't make any claims with respect to completeness of industry domains.

4 Results

The resulting conceptual model presents three different contributions of our study: 1) A conceptualization of autonomous systems, 2) A classification of challenges for testing of autonomous systems, and 3) A classification of available techniques and approaches as well as current practices for testing of autonomous systems. The following sub-sections explain the findings in a more detailed manner, including excerpts from our analysis model in Figures 2–5.

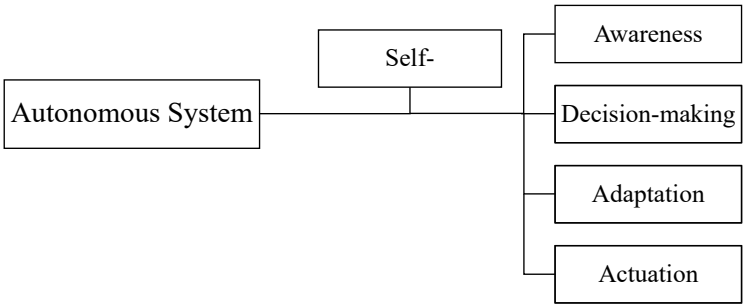


Figure 2: Concepts of autonomous systems

4.1 Conceptualization of Autonomous Systems

As there is no universal definition of what constitutes an autonomous system, we synthesized the variety of notions used in literature and practice for different application domains. The result was a taxonomy of aspects defining an autonomous system, as presented in Figure 2. Both the literature and industry practitioners similarly described that, autonomous systems *are capable of performing certain tasks in an unstructured environment without human supervision* [45]. More specifically, the system must be *self-aware*, meaning that it can analyse the situation and do the *decision-making* on its own, given the environmental conditions and its own states. Furthermore, the system must be able to *actuate* the plans to fulfil the desired tasks, and *adapt* its behaviour to optimize the goals by learning from the past.

A recent study by Sifakis [95] confirms our conceptualization by formalizing an architecture of autonomous systems, in which, the architecture defines five compulsory modules of an autonomous system: perception, reflection, planning, goal management, and self-adaptation.

The mechanism of adaptation (rule-based or self-evolving) and appearance (physical or virtual) of the autonomous systems are two characteristics discussed. While autonomous systems are expected to adapt their behaviour subject to conflicting goals and dynamically changing environments [45, 95], some interviewees emphasized the *self-evolving capability*, without being explicitly implemented, as an essential nature of these systems. However, the others insisted on that rule-based systems can still hold some level of autonomy, in which, they are able to handle certain situations by themselves in a more limited way. As said by the interviewees #3 and #4: *We believe the rule-based systems can still be seen as autonomous systems as long as they offer the intelligence and autonomy in handling the tasks.* The dispute on mechanism of adaptation becomes essentially a matter of the level of autonomy, as described in the automotive domain by the SAE 6-level

of autonomous driving [17].

As for the appearance of the systems, the most intuitive cognition of autonomous systems involves the physical components such as sensors and electronics, as what has been integrated in vehicles and robotics. However, they can also appear in the digital form, e.g. smart software applications in the investment market [96], where they offer some intelligence, and react autonomously and virtually over non-physical media. This is a common view from the interviews and focus group discussion, as interviewee #2 stressed: *A pure software system can also be instance of autonomous systems, since it is still the software control units, which lie in the heart of the autonomous systems, that actually enable the system autonomy.*

Given the broad interpretation of autonomous systems in different industry contexts, ranging from automotive, robotics, aviation, healthcare, cyber-security, to smart software systems, a *definition of autonomous system should incorporate systems both in a physical and digital form*. Besides, the definition should also be inclusive for different mechanisms of adaptation. A rule-based system can still generate some level of autonomy and handle unforeseen situations without human involvement, and self-evolution can be viewed as a feature that enables full autonomy where the system has the intelligence to reason, analyse and learn from both the surroundings and experiences, without being explicitly programmed during design.

4.2 Challenges for Testing

The challenges for testing of autonomous systems, as defined by our conceptualization model, come from two primary concerns, namely quality and safety. The quality aspects are committed to assuring the correctness of the design, the code, and the behaviour, while the safety aspects are about ensuring that potential incidents are within an acceptable threshold. Sifikas et al. [96] articulated that the machines must cope with the human order and should not expose any risk or danger to human society. Also, according to Helle et al. [45]: *Humans usually have high expectations for autonomous systems but low tolerance on their faults.*

Unfortunately, our results indicate that the challenges on quality and safety of autonomous systems are far from being resolved due to the *unpredictable environment*, the *complexity*, *data accessibility*, and *no standards or guidelines* that are settled for testing, see Figure 3.

Unpredictable Environment

The unpredictable environment is one of the major impediments that add uncertainties to testing as the systems can run into any environmental conditions that were unknown during design. Further, the same input can lead to different results since the system will learn and adapt its behaviour after deployment [45]. For example, a satellite has to respond to all risks in space for years and human inter-

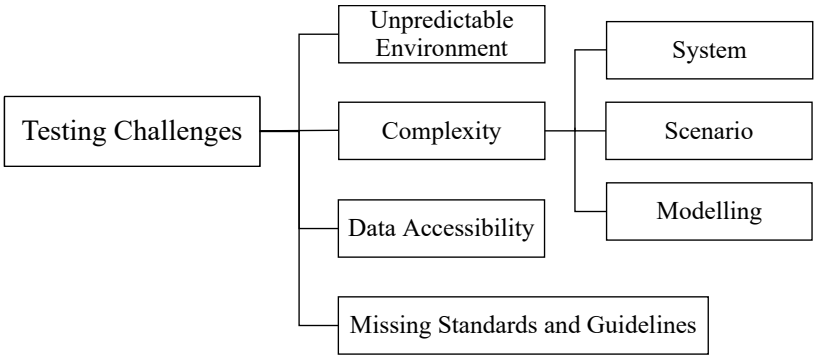


Figure 3: Challenges for testing of autonomous systems

vention is impractical after launch. This leaves a large explosion of parameters to be explored and it is infeasible to cover all possible scenarios [45, 62]. As reported in research on autonomous cars [62, 69, 103] and also argued by the interviewees, a number of million kilometers' driving test for the vehicle would guarantee the safety based on statistic prediction and the level of ambition. However, in reality, it is too expensive to conduct that distance of driving test in the traffic within years and there might still be corner cases, low-frequency errors, that are not covered.

System and Scenario Complexity

The complexity of testing autonomous systems lies in all artifacts involved in the operational environment and the system itself. The system is typically built as a system of systems, which involves many software control units and hardware electronic components. The emergence of AI technologies has increased the system complexity due to the limitations of existing techniques in addressing their test-ability, interpret-ability, and visualize-ability [17].

The performance of these AI-enabled systems depends largely on the data, where the implementation provides only the pre-trained model and leaving the actual behaviour non-deterministic and subject to data acquired during operation. Therefore, conventional testing approaches such as unit testing, component testing, and code review are inadequate to ensure the quality and prevent, identify or remove undesired consequences [8, 17, 78]. The testing has to ensure not only the correctness of the code and algorithms, but also the behaviour of the system that is determined by the actual input. This requires a large quantity of data, that are reliable and based on a real-world distribution, and a thorough understanding of how the systems are evolving.

The black-box nature of deep learning algorithms makes it even harder to understand or visualize the process of the decision-making [17, 112] as it can go through hundreds of layers before reaching a certain decision, and millions of pa-

rameters can be involved in this process. Besides, the AI components are exposed to adversarial attacks that can mislead the data and the communications in between [61]. A recent example is that a man with 99 mobile phones on a kid's cart flawed Google Maps as a traffic congestion [106].

Scenario complexity is yet another challenge at the core of the testing of autonomous systems [5]. It is hard to track, record, and replicate failures, particularly for fatal crashes or near-crashing cases [78]. It is unclear how engineers can define a scenario that includes all artifacts, either in a real environment or a simulation environment [61, 79]. Thus a better understanding is needed, of which factors, or objects, in the testing scenario led to the existing consequence, and whether the scenarios used for testing reflect the actual situations on how human operators react. In addition, established terminology and tools are imminently demanded [5].

As a result, the testing, which to a great extent is relying on the *modelling* of the system, the environment, and the scenarios, is getting intricate due to the complexity of all of them. It leaves academia and industry to improve and invent tools and approaches on how to model the environment, the system, and scenarios to keep the simulation environment align with the real-world [61, 78].

Data Accessibility

To be able to analyse, model, and test the systems, more data of good quality are required. However, the data becomes extremely costly when it comes to the collection, labelling, interpretation, validation, and generation of testing data [9, 61, 62, 104]. The developers and testers must not only collect the data, but also understand the significance, dimensions, and distribution of the data in different formats, label and validate the data to not under-fit or over-fit the performance. On some occasions, the data engineers must generate reliable data to compensate lack of data for testing purposes. In addition, one of our interviewees also expressed that they were struggling with data ownership issues to acquire data access between organizations.

Missing Standards and Guidelines

One problem that aggravates the complexity for testing of autonomous systems is that no standards and guidelines are settled [45, 62, 78]. Thus a new foundation must be established for autonomous systems [43] to understand, e.g. how to specify the requirements with suitable terminology, what quality criteria and safety performance to adopt, what oracle to pass or fail the test cases, how to conduct the regression test if revisions are made during tests, and what policies, regulations and ethical standards to apply. New tools and approaches must be invented to guide and automate the testing in an efficient and effective way. Quotes from two interviewees state that *#2: The industry is not prepared yet to address these issues into standards and set guidelines on how to do it. #5: More education and*

research are required to get the industry ready to mitigate the challenges and for the society to get along with the autonomous technologies and products.

4.3 Techniques, Approaches and Practices

Existing techniques, approaches, and practices for testing autonomous systems are insufficient to address the testing in an efficient and effective way, whereas they still address some important testing perspectives. Most of our findings, as described in the following sub-sections, are extracted from the literature study since the industrial practitioners usually focused on one or two of the approaches.

Practices – Available Industry Standards

Some industry standards are mentioned by the literature and industry practitioners from the focus group as well as the interviews, even though none of them specifies what is required for testing the fully autonomous systems. Among them, ISO-26262 addresses the functional safety for road vehicles [62, 108], which impacts on how automotive software is designed, developed, and tested. However, as more autonomous functionalities are involved and enabled by AI technologies, such as deep learning neural networks, the techniques within this standard such as code review and coverage-based testing are no longer applicable [17]. The ISO-16787 specifies test procedures and performance requirements for assisted parking systems [71]. It serves more as a suggestion and is up to each nation to implement. IEC-61508 introduces the fundamentals of functional safety for the electrical/electronic/programmable safety-related systems and focuses on the hazards caused by malfunctioning rather than any external environmental related factors [17, 112]. A newly published standard, which aims for the safety of the intended functionalities (SOTIF) for automotive, is described in ISO/PAS-21448 [17, 112]. This standard provides guidance and measures needed for the applicable design, verification, and validation to achieve the SOTIF.

Practices – Engineering Recommendations

Several engineering recommendations, as shown in Figure 4, were articulated both in the literature and by the industry practitioners, such as, *using simulation* for testing of autonomous systems to reduce the cost of accessing expensive hardware facilities and the risk of generating safety issues and economic losses. At the same time, simulation can considerably improve the testing efficiency by monitoring the test, recording the data, and generating the failure or test report. Besides, simulation also benefits test analysis by visualizing the process of the test.

The *V-model paradigm* is a common practice for test decomposition [47, 62]. In V-model engineering, software testing on different levels require different techniques and approaches, starting with unit testing, component testing, and integration testing, and then moving the entire system with both software and hardware

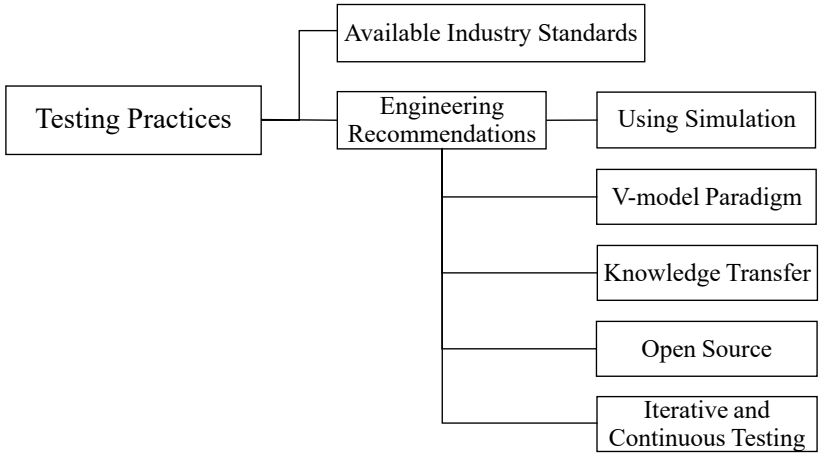


Figure 4: Practices for Testing of Autonomous Systems

parts into a simulation environment, to test ground and into the real-world to ensure that both the functional and non-functional requirements are satisfied.

Our interviewees and focus group participants also emphasized that *knowledge transfer* is essential. By learning from other industry domains and collaborating with both academia and industry, the entire industry should set up standards and regulations jointly, *open source* the data and platforms for reusing instead of having every player creating its own. As said by the interview #2: *We must learn from the other industry domains and transfer knowledge across. We must also initiate the collaboration among the industry for reusing the data and the tools instead of creating your own.*

Kang et al. presented 37 datasets and 22 virtual testing environments that are publicly available for closed-loop testing for autonomous vehicles [56]. Academic researchers can well contribute to explore possible alternatives, with the industry providing the test data, test-beds, and test results. Thus, industry and academia should move forward hand in hand. One of the most important and practical strategies for testing of autonomous systems is to go from requirement-driven engineering, to aim for *iterative and continuous engineering*, where it may initially start with limited testing data and an incomplete testing model, as expressed by interviewee #2: *We do not expect the testing can be solved with everything known beforehand, but rather taking it continuously in step-wise. The point is, when we start testing, we will get the data and we know better what is the problem.*

Techniques and Approaches (Conventional)

Conventional testing approaches are deemed as inadequate for addressing the autonomous nature of the systems [45], but they still dominate the testing efforts and

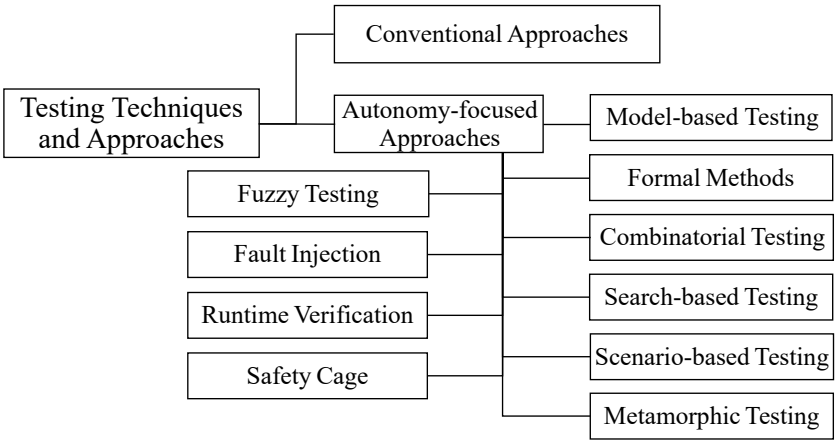


Figure 5: Techniques and Approaches for Testing of Autonomous Systems

resources for the time being. Regardless of the complexity of the system architecture, the software is still enabling the system intelligence and autonomy. The conventional software testing process, in brief words, can start with approaches such as unit testing, component testing, integration testing and non-functional testing. Depending on the testing and integration plans, the other components and sub-systems are then included and tested in simulation with Software-in-the-loop, Hardware-in-the-loop, and Vehicle-in-the-loop for the automotive [47]. In the later stage of testing, it involves ground testing or test in production environment, such as, test track and real road testing for vehicles and mobile robots before deployment [5, 47].

Techniques and Approaches (Autonomy-focused)

There are testing techniques and approaches used for solving autonomy-derived issues, as proposed by existing academic research and industrial practices, and shown in Figure 5. *Model-based testing* approaches [7, 45] are used to model the system properties, constraints, and behaviour, and thus can be further utilized for automatic generation and execution of the test cases. *Formal methods* [7, 17, 47] is another similar way to analyse and represent the system design, inputs and outputs using domain terminologies, and validate the system against a formal specification. These techniques and approaches are commonly used for rule-based systems.

Several other techniques are developed with promising results in reducing the number of test cases and total test efforts. Among these, *combinatorial testing* [103, 108] combines and adjusts multiple parameters in one test scenario instead of having one parameter being updated with the rest remaining unchanged; *search-based testing* [37] and *scenario-based testing* [82] are used to explore and

identify critical scenarios through statistical learning, e.g. using probabilistic models or genetic searching algorithms, where it analyses the previous testing results, studies the differences of them and approaches to the criticality objectives. Also, as highlighted by our interviewees #2, that *We think scenario-based testing is a very promising approach for testing of autonomous vehicles. Since it will be too expensive and impractical for us to cover all possible test cases, we should identify different scenarios instead, especially the worst-case scenarios, and put most of our test efforts in.*

Fuzzy testing [111] and *fault injection* [48, 62] are approaches that have been utilized to improve the test coverage and identify the corner cases, particularly for machine learning based applications. In detail, fuzzy-testing requires a large quantity of randomly selected data and validates the system performance based on the distribution and coverage of the test input. Fault injection is another variant where a set of special and faulty values are prepared to stimulate the systems and finding the corner cases. Another approach like DeepTest [104] was developed, where the researchers used image transformation to represent different real-world driving conditions and activate more neurons in the neural network as an indication for testing the autonomous driving algorithms.

Runtime verification include strategies, such as run time monitoring [68] and actuator-monitor architecture [8], which refers to that the system constantly monitors the behaviours during operation and report any anomalous situation as well as collecting data for reusing and optimization purposes. In addition, *safety cage* [17] is another approach that signals the anomalous inputs during operation by setting a confidence threshold and involving another control algorithm for situations below the threshold.

In order to address the test oracle issue, *metamorphic testing* [65, 108] was applied to define the metamorphic relations instead of specifying a certain value for asserting the test output. It is effective for many complex systems where the output of the test scenarios are hard to quantify but are consistent according to the inputs and certain principals, the metamorphic relations then act as the test oracle and expose a fault if the result fails to comply with them.

5 Discussion

With the advent of autonomous systems, testing of such systems has become a challenge to practice as classical test approaches are not sufficient. Also the concept of *autonomous systems* raises questions – what do we mean by autonomy, and what kinds of systems may be labeled autonomous? Further, as this is an emerging field, with research and development spent in both industry and academia, it is of certain importance that the concepts are aligned to allow joint efforts and reduce the gap between industry and academia.

We therefore studied both academic literature and industry practice; the literature through a classical literature review, and industry practice through focus group discussions and interviews with practitioners. By synthesising a thematic model of the findings from different sources, we have presented an inclusive conceptualization of autonomous systems (RQ1). We found a reasonable agreement on the *autonomy* concept to include aspects of performing tasks in unstructured environments without human supervision. However, whether or not the autonomy includes *self-evolution* is not agreed upon. Further, while a lot of research and broader discussions on autonomous systems relate to physical systems, like robots and cars, both literature and practice confirm that autonomous systems may be non-physical as well, for example, in banking and trade.

As a consequence, we propose that research be conducted on autonomous systems across domains, with physical as well as non-physical systems. Thereby general properties and techniques for autonomous systems may be developed rather than techniques for specific domains.

Given the characteristics of autonomous systems, we identified several *challenges for testing* (RQ2). First and foremost, autonomous systems are expected to be able to meet *unpredictable* situations and contexts, which by definition makes it impossible to test for a subset of such situations. Even when trying to specify example scenarios, they become very *complex* or are not resembling reality. *Access to data* is a key challenge, both to train and test autonomous systems, and since the field is emerging, *standards and guidelines* for testing are not yet established.

Implications for research and practice are that brute force traditional testing will never scale for autonomous systems. Rather, new approaches to modeling and simulation are needed, which align well with operational environments. Further, access to realistic data is a key for both training and testing. Most probably, domains have to collaborate on deriving and curating data.

Even though the foundation for testing of autonomous systems seems weak, there are techniques and approaches used both in research and practice (RQ3). However, the academic contributions are mostly adaptations of approaches for conventional systems, and fewer novel approaches. Testing of autonomous systems do require novel approaches, but these may be well designed adaptations and combinations of elements already used for conventional systems.

We do not claim that all kinds of autonomous systems will encounter the same challenges, nor that we have covered all kinds of systems. Nevertheless, we have explored the field of testing of autonomous systems and provided many insights based on both the academic publications and industry practices. Our results clearly indicate that the testing of autonomous systems is encountering a variety of challenges and must be improved aggressively.

In the near future, we would like to extend the insights from more autonomous contexts, other than automotive, robotics, and manufacturing. We would also like to experiment with and compare the pros and cons of different techniques in real industrial contexts. As articulated by Harel et al. [43], a foundation for the next-

generation autonomous system must be established, and according to Sifakis [96]: *No power of decision to autonomous systems should ever be granted without rigorous and strictly grounded guarantees under the pressure of economic interests and on the grounds of ill-understood performance benefit.*

6 Conclusion

We have conducted an exploratory study on concepts related to testing of autonomous systems, through a semi-systematic literature review, a focus group and interviews with industry practitioners. We contribute to synthesizing 1) Concepts of autonomous systems, 2) Challenges for testing of autonomous systems, and 3) Techniques and approaches as well as practices for testing of autonomous systems. The findings are based on insights from both a literature review and industry practices, and can serve as a frame to facilitate, both academia and industry, on testing of autonomous systems.

The conceptualization defines autonomous systems to be capable of performing specific tasks in an unstructured environment without human supervision. Our study also suggests that limited techniques and approaches have been reported so far by the academia and industry, and testing of autonomous systems must be substantially improved. The industry is trying different approaches without having common guidelines and standards, which is difficult and to advance the field, industry and academia must join forces in collaboration.

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

PAPER II

AN INDUSTRIAL WORKBENCH FOR TEST SCENARIO IDENTIFICATION FOR AUTONOMOUS DRIVING SOFTWARE

Qunying Song, Kaige Tan, Per Runeson and Stefan Persson, In proceedings of IEEE International Conference on Artificial Intelligence Testing (AITest), 2021.
DOI: 10.1109/AITEST52744.2021.00024

Abstract

Testing of autonomous vehicles involves enormous challenges for the automotive industry. The number of real-world driving scenarios is extremely large, and choosing effective test scenarios is essential, as well as combining simulated and real world testing. We present an industrial workbench of tools and workflows to generate efficient and effective test scenarios for active safety and autonomous driving functions. The workbench is based on existing engineering tools, and helps smoothly integrate simulated testing, with real vehicle parameters and software. We aim to validate the workbench with real cases and further refine the input model parameters and distributions.

1 Introduction

Testing of unsupervised autonomous driving (AD) features is a grand challenge for the automotive industry, since the variation of scenarios to test is extremely large and it might therefore be difficult to get sufficient scenario coverage within

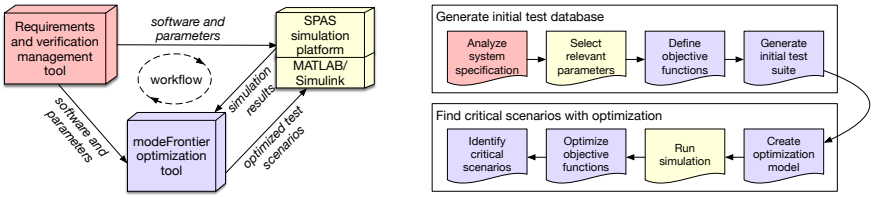


Figure 1: Overview the test scenario identification workbench, consisting of interconnected tools (left) which are used in the workflow (right)

available testing budget when relying on live testing only [85]. To utilize available resources, virtual testing and simulations is a primary industry concern, according to a recent survey [61]. However, simulation only is of limited assistance for auto makers; they must smoothly bridge the virtual software model-in-the-loop to the final testing with the real vehicle-in-the-loop. Industrial workbenches have been presented [41, 98] but since they are not fully open nor standardized, there is a need for further development of autonomous driving test benches.

We therefore designed a workbench for efficient test scenario identification for autonomous driving software, integrated with the industrial development process and products.

We have set up an industry–academia collaboration project between Volvo Cars and Swedish universities to design a workbench consisting of three interconnected tools, as shown in Figure 1 (left): a Requirement and Verification Management Tool for embedded systems; SPAS, a proprietary simulation platform for Active Safety (AS); and modeFrontier, a multidisciplinary design optimization platform. Further, we define a workflow, see Figure 1 (right), first generating an initial test database from the system specifications and standards, second optimizing the model to identify the most critical scenarios.

2 Related Work

A number of simulation platforms are available for autonomous driving, either open source or commercial. Despite that many of them are similar from a functional perspective, one may differ from the others regarding the simulation engine, scripting language used, or the capability to support specific sensor types, operating systems, and the X-in-the-loop integration [86]. CARLA, an open-source simulator, enables 3D visualized urban environments for autonomous driving, where the road, weather, vehicle dynamics, pedestrians, and sensor suites can be modeled [31]. Airsim, another open source simulator, that simulates both vehicles and drones, enables both software-in-the-loop and hardware-in-the-loop for physically and visually realistic simulations [94].

To make the tools more integrated, main automotive manufacturers are developing their own products for validation, verification, testing and simulation [86]. For example, Opel developed their toolchain for simulation-based identification of critical scenarios [41]. Analogously, Volvo Cars developed their SPAS platform, to serve the development and testing of different AS/AD systems. The primary advantages of proprietary toolchains are, that they maintain full control of the design and implementation of them, and can flexibly adapt as well as to deploy a new iteration for any specific feature integration. Further, it enables tight integration of their product software in their simulation environments. However, as development of tools and test rigs are costly, cross-company initiatives exist, for example presented by Solmaz and Holzinger [98].

3 Tools and Workflow

3.1 Requirement and Verification Management Tool

The Requirement and Verification Management Tool is a development platform for embedded systems with a strong foundation in the automotive industry. The tool is used for specifying and maintaining the requirements, architecture designs, test solutions and other type of system specifications for vehicle features, as well as the test results that measure the coverage and fulfilment of the requirements.

3.2 SPAS Platform

SPAS is a simulation platform for integration and testing, based on MATLAB/Simulink and developed by Volvo Cars. The platform is used as the model-in-the-loop testing platform for early verification of Active Safety (AS) and Autonomous Driving (AD) functions at Volvo Cars.

There are two main parts in the SPAS environment, namely the SPAS basic model and the AS/AD software. The SPAS basic model includes models of the environment, driver, powertrain, transmission, driveline, chassis, brakes, steering, electrical system, and vehicle system control modules. The AS/AD software is the implementation of the driving function, which will eventually be deployed in production vehicles.

3.3 modeFrontier

modeFrontier is a tool for process automation and optimization in the engineering design process. It offers a graphical approach to build an optimization model with a variety of built-in applications and external programs for MATLAB, Java and Python etc. modeFrontier also provides visualization and statistical analysis tools to visualize the optimization process and interpret the optimization results.

3.4 Scenario Identification Workflow

The test scenario identification workflow has two main phases, *Generate initial test database*, and *Find critical scenarios with optimization*. An overview of the workflow is presented in Figure 1 (right).

The first task in the workflow is an initial *system specification analysis* in the Requirement and Verification Management Tool, where the system requirements, design and test artefacts are created and maintained. Next, *relevant parameters and objective functions* for SPAS are selected based on systematic research of the system specifications and industrial standards, like ISO-16787 [49]. *Parameters* are quantifiable properties of the driving scenarios and are crucial for the performance of the system functionalities. *Objective functions* are the measurements that can be tracked or computed during the autonomous drive maneuver, to indicate the performance of the system functionalities in regards to efficiency, accuracy and safety etc. Last in the first phase, an *initial test suite* is generated in modeFrontier, based on the selected parameters.

In the second phase of the workflow (lower right part of Figure 1), the *modeFrontier optimization model* is created to automate the rest of the process including, the SPAS *simulation* of the initial test scenarios, recording the simulation results, and *optimizing the objective functions* using search algorithms in modeFrontier. The solutions evolve through optimization of the objective functions by analyzing the existing results and exploring the parameter space. The last task in the workbench is to *identify critical scenarios* from optimization results where a criticality threshold is defined for the objective functions. Thus, any scenarios that are beyond the threshold are considered critical.

In contrast to other scenario-based test benches for autonomous vehicles [41, 85], our workbench is generic, which means the driving function, simulation platform, parameters and objective functions are exchangeable. Thus, the workbench can, in principle, be used for critical scenario identification for any driving functions, and is not subject to a particular technique or simulator that is intended. In addition, our workbench automates the critical scenario identification process, which does not require expert involvement to develop, or deploy the workbench as well as to extract scenarios, and thus outperforms the others in simplicity and efficiency. Our further work includes validating the workbench with real cases and refining the input model parameters and distributions.

Acknowledgment

This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP). Thanks to our colleagues in SERG and Volvo Cars for their review of earlier versions of the manuscript.

PAPER III

CRITICAL SCENARIO IDENTIFICATION FOR REALISTIC TESTING OF AUTONOMOUS DRIVING SYSTEMS

Qunying Song, Kaige Tan, Per Runeson and Stefan Persson, In submission to a journal, 2022.
DOI:10.21203/rs.3.rs-1280095/v1

Abstract

Autonomous driving has become an important research area for road traffic, whereas testing of autonomous driving systems to ensure a safe and reliable operation remains an open challenge. Substantial real-world testing or massive driving data collection does not scale since the potential test scenarios in real-world traffic are infinite, and covering large shares of them in the test is impractical. Thus, critical ones have to be prioritized. In this study, we establish a systematic approach for critical test scenario identification with integrated tools and a workflow to explore the most critical test scenarios and facilitate testing of the autonomous driving functions. We also demonstrate the effectiveness of our approach by using two real autonomous driving systems from the industry by collaborating with Volvo Cars. Our main contribution in this work is a feasible and complete tool-chain for critical test scenario identification that is general for testing different autonomous driving systems.

1 Introduction

While autonomous driving is expected to improve traffic capacity and reduce road accidents, testing of the autonomous driving systems is a prerequisite to ensure the reliability and safety of such systems [99]. Inadequate or ineffective testing could fail to discover potential defects and misbehavior in the system and lead to severe accidents in the road traffic [36]. The fatal accident caused by Uber's autonomous vehicle is a typical example of such failure where a cyclist in front was not detected and subsequently hit by the vehicle at an intersection in Arizona, US, in 2018 [110].

Current approaches for testing autonomous driving systems that rely on substantial real-world testing, or collecting real driving data at scale, are considered both inefficient and ineffective since they take an unpractical amount of time to complete and may still not cover rare traffic situations [81]. The regular road traffic, most of the time is considered non-critical [58]. Therefore, new approaches for testing autonomous driving systems based on critical scenario identification are increasingly demanded [35, 36]. Critical scenarios are referred to as scenarios that can lead to a collision or near-collision consequence or situation here and are of interest for testing autonomous vehicles.

Nevertheless, existing studies mostly present parts of a complete solution for critical test scenario identification, for example, focusing on either simulation or optimization of driving scenarios [11, 60]. Also, the reported studies are in many cases function-specific, for example, by proposing interventions based on a particular function module, like motion-planning for the highway scenario [80]. Therefore, the feasibility of such approaches for testing different autonomous driving functions is unclear. In addition, previous studies tend not to validate their approaches on real driving functions from industry, but instead on some basic implementations based on existing platforms like MATLAB Simulink [50], or using publicly available driving components like DeepDriving [35, 36]. The effectiveness of those approaches for testing real autonomous driving systems under real traffic conditions is not demonstrated.

To tackle the challenges mentioned above and facilitate testing of different autonomous driving systems, we have proposed a critical test scenario identification approach in our previous short paper [100]. We extend the approach in the current work and generate critical test scenarios for two industrial autonomous driving functions by partnering with the automaker Volvo Cars, including a parking function in the low-speed maneuvering domain and a driving function in the high-speed maneuvering domain. Our approach utilizes three existing engineering tools (requirement and verification management tool, SPAS simulation platform, and modeFrontier–process optimization and automation tool). It presents a systematic approach for identifying critical driving scenarios through co-simulation with system implementations. The results of applying our approach on the two autonomous driving functions have demonstrated the effectiveness of this approach

for identifying the most critical scenarios for testing. Consequently, the identified scenarios can be used to substantiate test cases for autonomous vehicles in both simulated and real-world testing. To clarify our scope, the work does not aim to find the best optimization algorithm but to present a systematic approach for critical test scenario identification and demonstrate the effectiveness of this approach for testing real autonomous driving functions.

Our work provides a complete solution by integrating different components into a feasible tool-chain as a whole for critical test scenario identification for autonomous driving. It enables an end-to-end workflow from the initial analysis of the system specifications until generating a set of critical scenarios that can be used for testing. The approach is generic as the tools involved are exchangeable and is not subject to any particular driving function, development technique, simulator or application tools that might be intended. Thus, the approach can, in principle, be used for critical scenario identification for testing any autonomous driving systems. In addition, we provide evidence showing that the approach is effective in identifying critical scenarios for testing realistic autonomous driving functions. We also want to highlight that, due to industrial confidentiality concerns, only partial data and result analysis are presented, where sensitive information is removed, still demonstrating the principal outcomes.

The rest of the paper is organized as follows. Section 2 describes concepts based on the literature on critical scenario identification for testing of autonomous driving systems. Section 3 explains the research method and context we use for conducting the study. Section 4 and section 5 detail the case studies that use the proposed approach for identifying critical test scenarios for two industrial driving functions. We present discussions and limitations of the study in Section 7, and conclude the paper in Section 8.

2 Terms and Related Work

In this section, we first present the terms and concepts used in this study; then, we summarize the literature we surveyed on critical test scenario identification for autonomous driving and compare our work.

2.1 Terms and Concepts

In this part, terms like scenario and critical scenario are introduced, and their composition and differentiation to other similar terms are presented. Besides, relevant concepts predicated on those terms are described to form the basis of this study. Our interpretation and discussion of these terms and concepts are still based on the context of autonomous driving, with a particular focus on testing.

Scenario

According to Ulbrich et al., a *scenario* is defined as a temporal sequence of scenes, with actions and events of the elements that are involved within this sequence [105]. By actions and events, they mean, for example, maneuvers like cut-out and avoid colliding with a vehicle ahead. Given this definition, a scenario consists of at least one scene with corresponding actions and events, and a scene, in this context, is embodied as the geo-spatially stationary environment, dynamic elements, and a self-representation of all actors and observers.

Based on the definition proposed by Ulbrich et al. [105], Menzel et al. further refined the definition of scenario into three different abstraction levels – *functional*, *logical*, and *concrete* scenarios [72]. Specifically, functional scenarios usually describe the involved entities and their behaviors using a natural language. Logical scenarios specify the state space of the functional scenarios with the relevant parameters, parameter range and distribution. Concrete scenarios are instantiations of the logical scenarios by assigning concrete values for the parameters within the desired value range and distribution. The relevant parameters are selected for logical scenarios to describe the environmental constitution of the function scenarios, the behavior of the elements involved, and the physical capabilities and constraints of the autonomous vehicle. Bagschik et al. [10], have proposed a five-layer model which defines the required parameters for the scenarios, including road-level, traffic infrastructure, temporary manipulation of the road and traffic infrastructure, objects, and environment. Yet, the value range and distribution of the parameters and the possible relations between the parameters have to be further investigated to instantiate realistic concrete scenarios.

We adopt the definition of scenario proposed by Menzel et al. [72] in our work, where functional scenarios are retrieved from the system specifications and analysed to derive the parameters for logical scenarios. Subsequently, concrete scenarios are simulated and optimized to identify the most critical ones for testing autonomous driving systems. We have also observed that similar terms such as elements, entities, objects, and traffic participants are often used in the literature to refer to the different road users in the traffic, such as pedestrians, cyclists, vehicles of different types etc. We stick with the framework from Menzel et al. [72] to use entities within the definition of scenarios.

Critical Scenario

There is no universal agreement as yet on what constitutes a *critical scenario*, although different interpretations in the literature share a high similarity. Zhang et al. describe a critical scenario as a dangerous road situation that may lead to an unsafe decision for the autonomous vehicle, and appropriate countermeasures must be taken immediately to avoid collision [109]. In contrast, Kluck et al. focus more on the concrete scenarios level and consider a scenario to be critical if underlying parameter values cause a malfunction of the autonomous driving system [60].

Hallerbach et al. propose critical scenarios as the scenarios that need to be tested, which can be derived from both functional and non-functional requirements (e.g., traffic efficiency, driver comfort etc.) [41]. Herein we take the interpretation from Kluck et al., where critical scenarios are defined as the scenarios with a parameter set that has a high probability of revealing unintended and unsafe behavior of the systems, which may cause a collision or near-collision situations of the vehicle and other entities on the road traffic [60].

An integral part entailed in critical scenarios is how we quantify and evaluate a scenario to be critical or not, thus the indication of criticality of a scenario must be represented in a quantifiable way. Different surrogate measurements for safety evaluation of traffic conflicts are used, for example, Time-to-Collision (TTC), Post-Enchroachment Time (PET), Time-to-Brake etc [67]. Among these surrogate indicators, TTC is used the most, according to a review study by Aliaksei et al. [63]. Safety metrics can also be extracted from industrial standards and used as the criticality indicators, for example, ISO-15622 for adaptive cruise control [85], ISO-26262 for general automotive development and test [33], and Responsibility-Sensitive Safety (RSS) for autonomous vehicles [57]. Several performance metrics, including safety, functionality, mobility, and drivers' comfort, are used for generating test scenarios for autonomous vehicles by Feng et al. [33], and they use a combination of the maneuver challenge and exposure frequency as the indicator of critical scenarios. Eventually, selecting the criticality indicators must be specific to a particular driving system and the system specifications.

Furthermore, we must differentiate critical scenarios from similar terms to avoid misunderstanding. Gambi et al. use accident scenarios from police reports as critical scenarios for testing autonomous vehicles [35], whereas Klischat et al. argue that an accident by a human driver may not necessarily be a critical scenario and can be avoidable by others or autonomous vehicles [58]. *Challenging* scenarios and *complex* scenarios are often used alternately, and one may consider they are the same as critical scenarios. Riedmaier et al. [85] claim that a scenario is critical if the behavior of the system is evaluated after the scenario has been executed either in real-world or in simulation and the criticality being measured. Scenarios are challenging or complex only if the scenario itself is evaluated somehow and classified as challenging or complex. Ponn et al. point out that challenging scenarios are not always necessarily critical ones but more often lead to critical ones when executed [80]. Lastly, we also differentiate the concept of critical scenarios from corner-case scenarios (also referred to as edge cases). Karunakaran et al. define an edge case as an unknown and unsafe scenario that is hard to predict during the test and can lead to severe results for the autonomous vehicle [57]. Since critical scenarios can either be known or unknown, we believe the edge cases are a subset of the critical scenarios that are of high interest for its identification and testing the autonomous driving systems.

Scenario-based Testing

Scenarios are commonly used to substantiate test cases for autonomous driving systems [32]. As stated by Kluck et al., a test case is the value assignments of all relevant parameters of the scenario, which essentially aligns with the definition of the concrete scenarios [60]. However, a test case should entail not only a scenario but also a pass-fail criterion to evaluate the resulting behavior of the system [36, 72]. An example test case for an autonomous lane-keeping function, as given by Gambi et al. [36], is that the vehicle must follow a navigation path on a generated road map. The test fails if the vehicle cannot get to the destination or drives out of the lane.

Scenario-based testing is highly accepted and plays a key role in the validation of the safety of autonomous driving systems [85]. It is inherently connected to the concept of scenario as we have presented in the previous subsections, and it examines the resulting behavior of the autonomous vehicle in terms of interactions with the road infrastructure, with other road entities, and compliance with the functional specifications as well as the safety regulations [11]. The scenario-based testing approach aims to reduce the test effort to a manageable number of scenarios by limiting the testing to meaningful scenarios based on the testing budget [81]. The number of concrete scenarios can be infinite due to the combinatorial explosion of parameter values [11], and identifying all possible scenarios is difficult regardless of which approach is used [41]. According to Batsch et al., scenario-based testing usually runs in simulation with Software-in-the-Loop (SIL). Still, it can also be carried out with Hardware-in-the-Loop (HIL), or in the real world with proving ground (also known as test tracks in some studies) or regular road traffic [11].

Despite the remarkable benefits of using scenario-based approaches for testing autonomous vehicles, identifying relevant scenarios for the system under test remain the prerequisite, especially those critical scenarios that violate the desired safety requirements [85]. Open questions still challenge us in regards to what constitutes good test scenarios and how to generate them systematically [35]; how to define and collect realistic test scenarios [32]; and how to identify the critical scenarios for testing [109]. Menzel et al. propose many different sources that can be used for deriving test scenarios, which include, but are not limited to, functional specifications, system boundaries, the operational environment, legal requirements, and real driving data collected [72]. While common and safe scenarios without significant actions can be easily identified and reduced, the success of a scenario-based testing approach is highly reliant on its ability to find more critical scenarios within a given testing budget [82]. That is the core of the current study – to establish a complete tool-chain for critical test scenario identification and to employ it to test real autonomous driving systems.

2.2 Critical Test Scenario Identification

The general idea of critical test scenario identification, as described by Ponn et al., is that a concrete scenario is selected, executed, and evaluated with the criticality metrics [81]. As reported in the literature, there are different approaches for critical scenario identification, ranging from using search-based algorithms, deep learning techniques, expert opinions, etc. We categorize the literature we surveyed based on our interpretation and compare it with our work in the following subsections.

For a complete literature overview, we refer to the systematic literature reviews by Zhang et al. [114] for critical scenario identification, and Rajabli et al. [84] for software verification and validation, as well as the survey by Riedmaier et al. [85] for scenario-based approaches, all for assessment of safety of autonomous vehicles.

Knowledge-based Approaches

The knowledge-based approaches leverage expert knowledge to generate, extract, or select scenarios for testing. This approach is not frequently reported in the literature due to its evident constraints. As an example, Ponn et al. [80] involve experts from the autonomous driving domain for selecting parameters of scenarios and assessing the weight of the parameters as well as evaluating resulting critical scenarios for testing the autonomous driving systems.

The advantages of using this approach include the quick creation of an initial catalogue of test scenarios [85], yet the drawbacks are non-negligible. It requires expert involvement and is labour-intensive, and may lack the diversity and complexity of real-world scenarios, especially those accidents that impose complicated situations and rarely happen [109]. In addition, the generation and selection of scenarios might be subjective, where simple scenarios are ignored but can still cause severe consequences. As a result, the derived scenarios are often considered lacking evidence for proof of safety in real traffic [81].

Compared to our approach, we do not require any expert involvement, and identification of the critical scenarios is automated by integrating the existing engineering tools. Specifically, the selection of scenarios is based on optimizing the parameter space and simulation of the scenarios. Thus, it is not limited or biased by subjective knowledge acquired.

Data-driven Approaches

The data-driven approaches extract critical scenarios based on available data sets that have been collected beforehand. The data can be presented in many different forms, for example, scenario libraries, police accident reports, or sensor data collected by test vehicles. Scenario extraction and selection techniques and tools are then used for identifying critical scenarios from the data.

Among the published studies, Gambi et al. [35] generate effective and critical test scenarios for autonomous driving by reconstructing crash accidents from police reports in simulation, using natural language processing. Zhang et al. [109] introduce a toolkit for extracting critical scenarios based on real traffic accident videos and reproducing the scenarios in simulation. The extracted scenarios are then used for the safety assessment of autonomous vehicles. Erdogan et al. [32] propose an architecture to enable test scenario generation, where test scenarios are first extracted from a video stream that contains real-world sensor data and then is stored in a structured database cluster with scenario definitions and the corresponding measurements. A user interface is implemented and included in this architecture to customize and adapt the conditions for test scenario generation based on the aforementioned scenario database.

Deep learning has been actively used for critical test scenario identification through the studies that we surveyed. Ding et al. [30] train a generative model for generating safety-critical scenarios by sampling through the parameters and rewarding the risky scenarios. The generative model gets a higher reward when a riskier scenario is generated. Another study that uses reinforcement learning is reported by Karunakaran et al. [57] for automatically generating scenarios and optimizing the learning towards the worst-case scenarios with respect to the RSS safety metrics. A few other studies that employ deep learning techniques include Batsch et al. [11] using Gaussian Processes to train and optimize the parameter selection towards the most critical scenarios on the performance boundary, and Jenkins et al. [52] using a recurrent neural network to generate accident scenarios for testing the autonomous driving systems based on the in-vehicle and vehicle-to-infrastructure data generated from simulators. In a related application domain, Porres et al. [82] use online supervised learning to train a generative model for searching and selecting critical scenarios for testing the autonomous maritime collision avoidance systems through the operation.

Even though diverse techniques for extracting or generating critical scenarios based on real driving data have been studied, limitations can be observed and described in these studies. A prerequisite of using such techniques is a data set that is comprehensive [85], whereas it is well known that collecting real driving data at scale is both time-consuming and expensive but still does not guarantee to include all corner cases [57, 83]. As highlighted by Hallerbach et al. [41], the major drawback of using recorded data is the incompleteness of the data set, thus we have to understand how the data is acquired and how representative it is. The quality of the data can be affected by various factors such as the type of sensors used and how they are installed [81], the location where the data is collected, and the fact that rare-occurring situations are difficult to collect [30]. After all, we still have to understand how to extract and select scenarios given massive data collected [59].

In contrast, our work does not rely on collecting data from different sources, and thus is not subject to the quantity or diversity of the data set. Instead, we first analyse the system's function and operational design domain (ODD) based on the

system specifications. ODD is a concept that defines the operational environment of autonomous vehicles and is used to derive test scenarios and safety assessment of autonomous driving systems [40]. Then we create an optimization model using the existing engineering tools to integrate the parameters, the objective functions, and a simulation platform that runs the scenarios and records the results. The optimization model optimizes scenario generation towards the objective functions for critical scenarios using the design of experimentation (DoE) [85]. DoE is a systematic approach for analysing the relationship between input parameters and output values and how the effect (output) changes over variation of the conditions (parameter sets). The DoE generation in our work represents the selection of parameter values and the creation of new test scenarios.

Search-based Approaches

The search-based approaches employ search algorithms to optimize critical scenarios from the operational design domain of the autonomous driving system. This approach typically requires the execution of the scenarios in simulation and an objective function that measures the criticality of the scenarios. The search process evolves based on the parameter space and the objective function value of the executed scenarios. Also, it usually limits the search to a certain number of iterations based on the testing budget and computational resources available. Our approach falls into this category.

Klischat et al. [58] use evolutionary algorithms to optimize the drivable area of the vehicle to generate complex scenarios for testing the motion planning of the autonomous vehicles. Similar work is reported by Althoff et al. [6] to generate safety-critical scenarios for collision avoidance of autonomous vehicles by optimizing the drivable area as well. Besides, Buehler et al. [19] also employ evolutionary algorithms for generating critical scenarios for functional testing of an autonomous parking system. Specifically, genetic algorithms are a class of evolutionary algorithms commonly used for search-and-optimization problems. Gambi et al. [36] use genetic algorithms to evolve the generation of virtual road networks for testing the lane-keeping function. Kluck et al. [60] propose an approach for test parameter optimization using genetic algorithms and have employed it for testing an autonomous emergency braking function.

The advantages of using a search-based approach for solving optimization of critical scenarios for testing of autonomous driving systems are prominent, since the selection of parameter values is rather difficult before the test and covering the entire parameter space is costly [81]. In addition, this approach does not rely on collecting substantial driving data and is easy to implement. However, some limitations are also stated in the existing studies. For example, generated scenarios may not be realistic in the real-world traffic, simulation of the scenarios are often computationally expensive, and only low-dimensional scenarios can be handled effectively in optimization [30]. To complement the said limitations, Beglerovic

et al. [12] simulate and optimize test scenarios based on a light-weighted surrogate model instead of the real system, Feng et al. [33] establish a sophisticated model of relevant parameters, metrics, and searching process for critical scenario generation, and Hallerbach et al. [41] create a complete tool-chain for critical test scenario identification for autonomous driving systems.

We believe that the search-based approach can well compensate for the scarcity of sensor data and generate critical scenarios that can be used to substantiate test cases for autonomous driving systems. While most of the existing studies use either a simple implementation of the autonomous driving function based on engineering tools like MATLAB Simulink (e.g. Ponn et al. [80]), or publicly available driving components such as DeepDriving and Beam.AI by Gambi et al. [35, 36], for validating the approaches, their effectiveness on realistic autonomous driving functions is not demonstrated. Besides, many of them are also function-specific, which is relevant to a particular function or operational domain for, e.g. parking system [19], motion-planning [58], or highway scenario [12], and use only a limited set of scenarios for validation from e.g. NHTAS [109] and Euro NCAP [60]. Our approach is generic in that the tools involved are exchangeable and are not determined by the driving functions, so it can, in principle, be used for critical test scenario generation for any autonomous driving system. We also demonstrate the effectiveness of this approach by using two real autonomous driving systems from the industry. As articulated by Hallerbach et al. [41] and Ding et al. [30], there exist very few studies that provide a complete solution for critical test scenario identification which are generic to different autonomous driving systems. The major contribution of our work is to address such a gap and facilitate the testing of autonomous driving systems.

3 Research Context and Method

In this section, we describe the research context and method we use for the current study. We conducted the work on critical test scenario identification in the autonomous driving domain using the design science paradigm [89]. We *formulate the problem* of critical test scenario identification by looking into the existing literature and industrial practices. Then we *design the solution* by integrating the existing engineering tools and a workflow, and *validate* it in the industrial context using two real autonomous driving functions by collaborating with Volvo Cars. Lastly, we *infer the potential usage* of our approach for testing autonomous driving systems in general.

3.1 Research Context

We base the current study on the critical test scenario identification approach that we presented briefly in our previous work [100] for testing autonomous driving

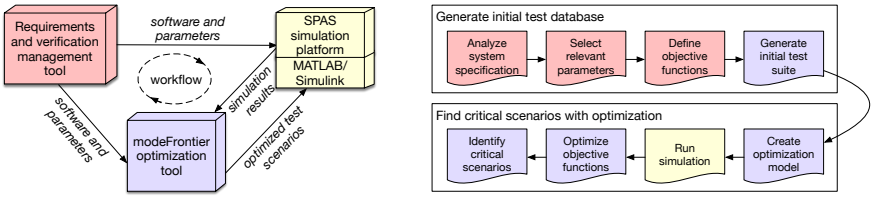


Figure 1: Overview of the critical test scenario identification approach, consisting of interconnected tools (left) which are used in the workflow (right). The figure is an adaptation from our previous work, and we refer to Song et al. [100] for more details of the approach.

systems. As shown in Figure 1, the approach integrates three different engineering tools and a workflow for the identification of critical test scenarios. The identified scenarios can then be used to substantiate test cases for testing autonomous driving systems.

The three engineering tools are, (1) a requirement and verification management tool that is used for storing and analysing the system specifications; (2) an internal simulation platform – SPAS, that is developed by Volvo Cars and used for early verification of the active safety and autonomous driving functions; and (3) a process automation and optimization tool – modeFrontier, which is a commercial tool for process and design optimization. The tools are exchangeable, meaning that we can substitute them with other similar tools to cope with different technical environments or autonomous driving functions under test. For example, we can use a different simulator like Carla [31] or AirSim [94] to simulate the scenario execution. Other simulators for autonomous driving are presented by Kang et al. [56] and Rosique et al. [86]. Bhat et al. [15] discuss tools as well as methodologies for autonomous driving systems during different engineering stages.

The workflow includes two main phases, see Figure 1 (right). In the first phase, we start by analyzing the system specifications to understand the functionalities and the operational design domain of the system. The system specifications can be in different forms such as functional specifications, design documents, related standards or regulations etc. Based on that, we select relevant parameters that constitute a driving scenario and the value range and distribution of the parameters. Also, we define objective functions that measure the criticality of the executed scenarios. Lastly, we generate an initial suite of scenarios by sampling through the parameter space based on the intended distribution and size of the initial test suite.

In the second phase, we create an optimization model in modeFrontier to integrate the selected parameters, objective functions, and a simulator. The optimization model optimizes the scenario generation with respect to the objective functions and identifies the most critical ones. It executes the scenarios in the initial test suites in simulation, continuously exploring the parameter space and

evolving through the completed simulation. We also configure the number of iterations for the optimization model in modeFrontier based on the testing budget and computational resources available.

We also base our study on collaboration with our industry partner – Volvo Cars, where they support us by providing access to the tools above and two autonomous driving functions, namely autonomous driving function and autonomous parking function. We replicate our approach on these two functions for identifying critical scenarios for testing such systems. By having access to real industrial systems, we set up our approach and demonstrate its effectiveness for testing using realistic autonomous driving functions.

3.2 Research Method

We conducted the study under the design science paradigm [89] and have mainly conducted four steps as enumerated below. The problems of critical test scenario generation challenges are conceptualized in the industrial context [100]. We report our design of the critical test scenario identification approach (steps 1 and 2 below), and validate the approach using two autonomous driving functions from Volvo Cars (steps 3 and 4) and expand a potential usage context (step 4).

1. For each autonomous driving function, we analyze the system specifications and implementation through the requirement and verification management tool to understand the functionality and the operational design domain of the system. We then identify the relevant parameters with the value range and distribution of each parameter and define objective functions and the criticality thresholds for the autonomous driving function.

2. We explore the tool modeFrontier and create the optimization model by integrating the parameters, objective functions, and the SPAS simulation platform. Besides, we also configure the optimization algorithm, size of the initial test suite, and the number of iterations for the optimization model.

3. We replicate the optimization model from the previous step by selecting a different algorithm to compare two different algorithms and show our approach's generality to different optimization approaches. To clarify here, the contribution of the work is not to find the best algorithm but to provide a complete approach for critical test scenario identification and generate critical test scenarios for real autonomous driving functions from the industry.

4. We start the optimization process in modeFrontier, and debug the errors if the process fails or suspends. After the optimization processes finish, we perform further analysis on the results in modeFrontier and export the findings in tables and figures, which we present in section IV and V. The identified critical scenarios are provided to the related engineering teams to test

and investigate potential flaws in the specification, design, or implementation of the system.

4 CASE I: Autonomous Driving Function

This section describes the work and results for the first case that uses the proposed approach we present in Section 3. We aim to generate critical test scenarios for an early version of an autonomous driving function from Volvo Cars, which in this paper is referred to as the AD function (ADF).

4.1 Analyse System Specifications

ADF offers unsupervised in-lane driving in queue situations up to a specific speed limit $v_{\max}$, and enables the host vehicle to keep a safe distance to the preceding vehicle within the lane. The cardinal functionalities of ADF can be summarized as (1) driving in a lane and (2) proactively adapting speed. These requirements specify that the host vehicle shall stay in lane and maintain a safe longitudinal and lateral distance to infrastructure, other vehicles and entities on the road. In addition, the host vehicle shall comfortably control speed to comply with the current speed limit.

Figure (2) shows snapshots in a scenario at different time steps. It is simulated in the SPAS platform and demonstrates the functionality of ADF. The host vehicle equipped with ADF is marked red, while others are visualized in blue. At the beginning of the scenario, the host vehicle drives at a relatively high speed compared to other vehicles. When driving around the bend, the host vehicle detects the front vehicle in the same lane, thus ADF drives the host vehicle to decrease the speed gradually. At the end of the scenario, the host vehicle manages to adapt its speed to follow the front vehicle.

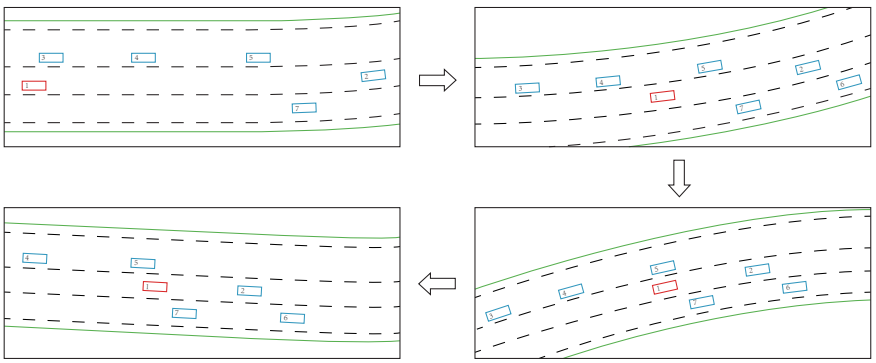


Figure 2: A series of visualized scenes of the autonomous driving function (ADF)

4.2 Select Relevant Parameters

With the guidance of the requirement and verification management tool, the operational environment for ADF is characterized, where it covers all kinds of possible influential factors on the road, including traffic, vehicle status, environment, infrastructure, other road users, and driver behavior.

We select parameters from two domains: (1) movable entities, including dynamic behaviors of the host vehicle and other road users, and (2) road topology, including highway infrastructure and traffic conditions. In this section, we elaborate on the parameter selection of movable entities as an example.

Road users include all kinds of vehicles, pedestrians and animals. Since the operational environment for ADF is on the highway, pedestrians and animals rarely appear. At the current stage, only vehicle models are taken into consideration. The vehicles in the ADF function can be divided into three categories: host vehicle, lead vehicle, and other vehicles. We denote the vehicle set by $\mathcal{V} = \{V_h, V_l, V_{o1}, ..., V_{oN}\}, N \in \mathbb{Z}$, where V_h and V_l represent the host and lead vehicle, while $V_{o1}, ..., V_{oN}$ are other vehicles on the road. The simulation period of each scenario is set to T, unless a collision happens to trigger an early termination. In addition, parameters for each movable object are analyzed in four aspects to define a scenario: initial position, velocity, acceleration profile, and the number of vehicles. In what follows, we denote the position, velocity and acceleration of vehicle-i at time step-t by $\mathbf{p}_i(t) = [p_i^x(t), p_i^y(t)], v_i(t)$ and $a_i(t)$. Specifically, the velocity and acceleration are expressed as scalars since only the longitudinal information is of interest.

Initial position

The initial position of vehicles is selected, including the longitudinal and lateral positions along the road. Several constraints are defined to limit value selections. Each vehicle should keep a safe distance to others. The two-second rule, which a rule of thumb estimating safe distance at any speed for vehicles, is set as the baseline to deduce minimal initial longitudinal distance $|p_i^x(t_0) - p_j^x(t_0)| \geq d_{\text{min}}^x, \forall i, j \in \mathcal{V}$. Also, to limit the scope of a scenario, we define a distance range between head-most vehicle and back-most vehicle in a simulation scenario, and its upper limit is denoted by D_{max}. Regarding the lateral distance, the vehicle must leave a $d_{\text{min}}^y = 1.5m$ space when considering regular road width for a freeway of $3.5m$ [74], i.e., $|p_i^y(t) - p_j^y(t)| \geq d_{\text{min}}^y, \forall i, j \in \mathcal{V}$.

Velocity

ADF provides the nominal function only in situations when the host vehicle's velocity is lower than a specified level, i.e., $v_{V_h}(t) \leq v_{\text{max}}$. To evaluate ADF's performance, the host vehicle should be able to detect, catch up and follow the

lead vehicle. For this reason, the host vehicle should be in the right level of proximity to the lead vehicle, which allows the lead vehicle to be detected at the initial stage of a scenario. In addition, the initial velocity of the lead vehicle should follow $v_{V_l}(t_0) \leq v_{V_h}(t_0)$, or otherwise, the ADF will be deactivated and switched to human maneuver mode. Moreover, to keep the traffic smooth on the highway, the minimum speed for all vehicles is set to a specified level v_{min}. According to Abuelenin et al. [4], the traffic velocity on the road approximately complies with a normal distribution, and thus normal distribution is used for speed in scenario generation.

Acceleration

We restrict the acceleration of all vehicles on the road with $|a_i(t)| \leq 3m/s^2, i \in \mathcal{V}$ at each time step during the simulation, based on the real-world traffic data [16]. Besides, the longest acceleration period is restricted to 3 seconds. Acceleration values are sampled from a uniform distribution.

Number of vehicles

The number of vehicles on the road is jointly decided by d^x_{min}, D_{max} and the length of a vehicle. Each scenario has $|\mathcal{V}| = N + 2$ vehicles. Considering the special scenario when there is only one lane in the road, to respect the safe distance, the vehicles on the simulated segment of the road cannot approximately exceed 10. The upper limit on vehicle number also reduces the design space and accelerates the scenario optimization process. Moreover, to ensure there are enough vehicles on the road to formulate a critical scenario, the minimum number of vehicles is set to 5. Thus, the number of vehicles defined in a scenario is given as $5 \leq |\mathcal{V}| \leq 10$.

4.3 Define Objective Functions

We define the objective function from two perspectives to extract critical scenarios: vehicle behavior and driver reaction. For vehicle behavior, criticality is defined as the closeness to an accident, of which TTC is used as an indicator. TTC is considered an objective function that should be minimized. A threshold time (ΔT_{thres}) is set to distinguish critical scenarios from non-critical ones.

Regarding the driver reaction, ADF should ensure the driving comfort as much as possible [33]. For this reason, jerk, measured as the rate of change in acceleration, is selected as another objective function to evaluate performance. When the jerk value is larger than $\pm 4m/s^3$, it would be not acceptable for most vehicles [14]. Thus, we try to maximize the absolute value of jerk to find the critical scenarios and set $|\dot{a}_{\text{thres}}| = 4m/s^3$ as a threshold for the corresponding scenarios to be considered as critical.

Both TTC and jerk are evaluated at each scene and are updated by time frames. We select the extreme values of TTC and jerk within a simulation period to repre-

sent the criticality of a scenario. For this reason, the simulation will not be terminated prematurely if the value of TTC or jerk has exceeded the threshold unless a collision is detected.

4.4 Generate Initial Test Suite

To generate an initial test suite, we use MATLAB to translate the specifications in the requirement and verification management tool, and send the outputs to modeFrontier for DoE generation. ModeFrontier has different approaches for design space exploration, and in this study, the Space filler DoE is leveraged to guide the test scenario generation. This approach gives the most uniform filling of the design space, where the risk of missing corner cases can be mitigated. Latin Hypercube Sampling (LHS) is applied to generate random design configurations. In addition, an initial test set ((i.e., with 50 scenarios, according to the rule of thumb for DoE)) is generated as the input for the initial evaluation.

4.5 Create Optimization Models

The optimization process in modeFrontier follows the scheme as discussed in Figure 1. First, test scenarios are initialized in MATLAB with concrete values for each selected parameter and used for simulation in SPAS. After the simulation finishes, results of objective functions are recorded and saved in modeFrontier, where the data is parsed and analyzed. Subsequently, a new test scenario is generated with distinct parameter values by the optimization model and executed in simulation. The entire optimization process ends after the specified number of iterations, and critical scenarios can then be extracted and analyzed. Figure 3 shows the modeFrontier optimization model for ADF. The block on top is parameters for generating new scenarios. The sub-blocks inside represent sub-parameters that need to be specified for a scenario. The parameter values are transferred to the middle block, where the SPAS simulation is performed. Lastly, the blocks at the bottom are used to define the objective functions. The optimization is then based on the objective functions of the scenarios in SPAS simulation.

Two optimization algorithms, namely Multi-Objective Simulated Annealing (MOSA) and pilOPT, are used for optimization purposes. The optimization models that use each algorithm are created separately, and Figure 3 is an example of the model how it looks. pilOPT is an in-house developed algorithm in modeFrontier, which can effectively handle the multi-strategy searching problem and minimize the amount of time and computational resources required[1]. It combines the advantages of local and global search algorithms to get the optimum solutions. In contrast, MOSA is a heuristic searching algorithm, which is regarded as the benchmark algorithm to be compared with pilOPT.

[1] https://engineering.esteco.com/modefrontier/

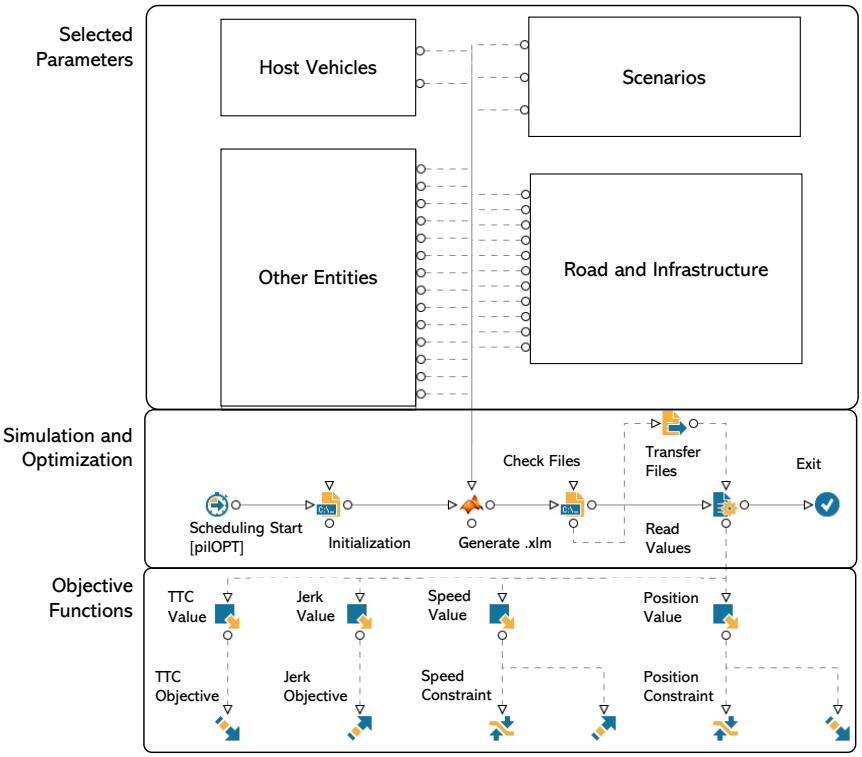


Figure 3: modeFrontier optimization model for the ADF

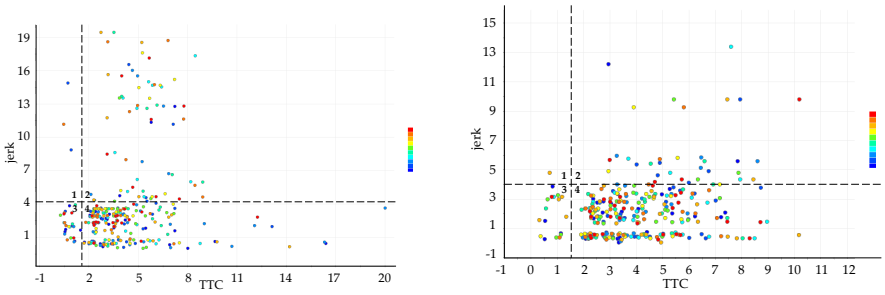
4.6 Run Simulation and Optimization

After creating the optimization model and setting up the simulation environment, the optimization process is started in modeFrontier. The number of optimization iterations is determined mainly based on computational resources available and is set to 300 in this case. Higher intensive grid search can be performed with more powerful computing resources, although the number of available software licenses of commercial tools may also be limiting. After running the simulation and optimization, the results are saved to analyse the critical scenarios further.

4.7 Identify Critical Scenarios

Figure 4 shows the simulation results for each test scenario during optimization by MOSA and pilOPT, and the relationship between TTC and jerk value is plotted. In the figure, sequence information of optimization iterations is represented by the

change of colors (i.e. blue the earlier iterations, and towards red means the later iterations).



(a) MOSA (b) pilOPT

Figure 4: modeFrontier optimization results from the autonomous driving function (ADF) with (a) MOSA and (b) pilOPT algorithms. Each dot represents a test scenario. The dash lines are the criticality thresholds for TTC and jerk, respectively, while the scale of the two axis is left out for confidentiality reasons. Colors indicate the sequence number of the iterated simulations.

For the MOSA algorithm, we observed a clear boundary among test scenarios with very low jerk values. In addition, another boundary exists to partition test scenarios whether their TTC values exceed ΔT_{thres} or not. For test scenarios with low jerk values, the TTC values are mostly over ΔT_{thres}, indicating that in those test scenarios, the host vehicle does not experience the sharp acceleration, thus being obvious safe scenarios. According to the definition of critical scenarios, if a test scenario has $|\dot{a}_i(t)| > |\dot{a}_{\text{thres}}|, i \in \mathcal{V}$, it is considered a critical scenario. Therefore, quadrants 1, 2 and 3 in Figure 4 (a) are critical scenarios. As scenarios are randomly distributed as in the figure, no distinct region feature and difference emerge with the optimization process.

In Figure 4 (b), there is no obvious boundary on either axis, but the figure is divided into two groups. Test scenarios in the first group, located on the upper part of the figure, have remarkably high jerk values. The number of critical scenarios is summarized in Table 1 to compare the difference between MOSA and pilOPT. The number of scenarios caused by violating TTC and jerk constraints is not summing up the total amount since there are some scenarios where both criteria are critical. We conclude that, in this case study, pilOPT has a better performance in finding critical scenarios than MOSA, especially with respect to jerk. This is, however, not our primary focus here, and optimization algorithms have to be further explored.

Table 1: number of critical test scenarios with respect to the objective functions

	MOSA	pilOPT
jerk	33	87
TTC	18	28
total	45	95

5 CASE II: Autonomous Parking Function

In this section, we describe the work and result for the second case that uses the proposed approach we present in Section 3 to generate critical test scenarios for an early version of an autonomous parking function from Volvo Cars.

5.1 Analyse System Specifications

The Autonomous Parking Function (APF) aims to detect and park the vehicle into a feasible parking slot between two stationary vehicles autonomously, where a driver is not required. The function should park the vehicle in both parallel and perpendicular slots, either reversely or forwardly.

The case study APF version supports only the rearward parking in parallel slots (i.e. parking slots that are parallel to the road direction) where the parking manoeuvre is performed mainly in three steps. First, the vehicle drives at a low speed and passively scans the empty slots using the ultrasonic sensors that are deployed on the front side of the vehicle. Second, the vehicle identifies the target slot and performs motion planning to park the vehicle in it without colliding the vehicles around. Lastly, the vehicle starts to actuate the rearward parking manoeuvre by controlling the steering wheel, propulsion, shifting gear and braking, and follows the trajectory that is computed in the previous step. When the vehicle reaches the final position that has been planned, it deactivates the parking function and sets a brake torque to stop the vehicle.

Figure 5 illustrates the function and operational scenes of APF. Specifically, five vehicles (numbered 2–6 and in blue) are parked parallel to the road direction and remain stationary. The host vehicle (i.e. numbered with 1 and in red – the vehicle with APF installed, also known as ego vehicle) first drives from the left and passes the stationary vehicles, it scans and identifies an empty slot between the rear vehicle (4, referred as V_r) and the front vehicle (5, referred as V_f). Then APF reversely parks the host vehicle into the slot without colliding with other vehicles and stops at a feasible position subject to physical constraints such as the slot length and the maximum steering angle the vehicle can complete. In an optimal situation, the host vehicle should stop at the centre of the parking slot with a sufficient distance to both V_r and V_f, and the vehicle stands parallel to the parking slot.

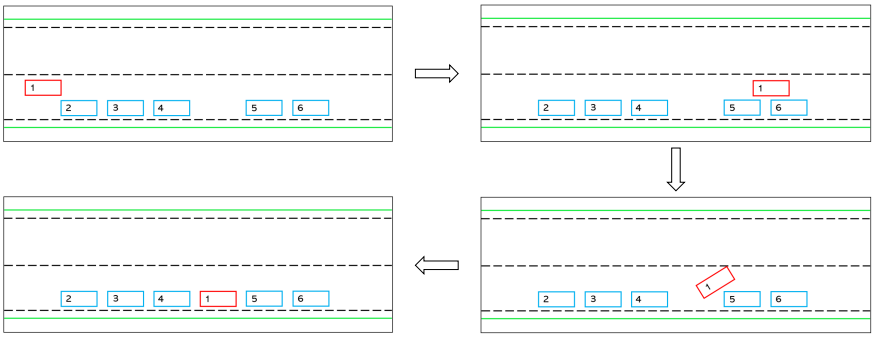


Figure 5: A series of visualized scenes of the autonomous parking function (APF)

5.2 Select Relevant Parameters

After analysing the system specifications and current design of APF by using the requirement and verification management tool, we identify two relevant parameters for constituting a test scenario for APF, namely slot length and angle of the stationary vehicle.

Slot length describes the actual length of the parking slot and is the primary parameter that determines whether a parking slot is feasible or not. Buehler et al. [19] adopted both the slot length and slot width as the two parameters that depict the parking space and used them to explore critical test scenarios for an autonomous parking system. Given the current design and the operational design domain of APF, we presume a sufficient slot width in the current study and thus select slot length as a relevant parameter for scenarios.

Based on the setup in Figure 5, slot length can be quantified and adjusted by changing the position of either V_r or V_f on the coordinate system of the simulation platform. Herein we select the position of V_f (referred as PoV_f) as the derived parameter for slot length. The value range of slot length contains both a lower bound – the minimum slot length APF should handle without colliding the stationary vehicles, and an upper bound – an adequate slot length that APF manages while keeping a sufficient distance to the stationary vehicles and a considerable yaw angle to the parking slot. Due to confidentiality concerns, we do not report the specific values here.

The angle of the stationary vehicle represents the yaw angle rate of the stationary vehicles (i.e. V_r and V_f), and is a parameter that determines the shape of the parking slot as well as the motion planning of APF. Since we here focus on rearward parking, and the slot length is generally larger than the standard parking slot length, we consider the yaw angle of V_f having the most impact (referred as

AnV_f). The value range for this parameter is set to $[-3°, 3°]$ according to the ISO-16787 standard [49] which is a standard specification for testing autonomous parking functions and is up to each nation to implement. According to this standard, a vehicle should remain within $[-3°, 3°]$ to the central line of the parking slot after completing the parking maneuver. Thereby, we take this standard specification as a reference for setting AnV_f.

5.3 Define Objective Functions

The basic acceptance criterion for a parking scenario, according to ISO-16787 standard [49], includes that the host vehicle should keep a minimal $0.3\ m$ distance to other vehicles around and standstill with a yaw angle within $\pm 3°$ to the central line of the parking slot. We consider scenarios beyond these two criteria critical and should be identified as critical test scenarios for APF. Based on these two criteria and the setup shown in Figure 5, the distance to V_r (referred as DtV_r) and V_f (referred as DtV_f) should be minimized through optimization to identify the scenarios with less than $0.3\ m$ distance to either of them. In addition, the yaw angle of the host vehicle (referred to as AnV_h) needs to be optimized to identify the scenarios that end with an angle beyond $\pm 3°$.

Nevertheless, we cannot have all the aforementioned objective functions in one optimization model due to the natural conflicts among them. For example, minimizing DtV_r is essentially maximizing DtV_f since these two vehicles are located on the two end sides of the parking slot. Thus, these two objective functions must be separated into two different optimization models. In addition, we cannot maximize and minimize AnV_h at the same time to identify critical test scenarios that are greater than $3°$ and those lower than $-3°$. Thus, these two objective functions have to be separated into two different optimization models as well. The resulting set of objectives is four, hence leading to four optimization models with two objective functions each as shown in Table 2.

Table 2: modeFrontier optimization models and corresponding objective functions for APF

Model	Objective function 1	Objective function 2
1	minimize DtV_r	maximize AnV_h
2	minimize DtV_r	minimize AnV_h
3	minimize DtV_f	maximize AnV_h
4	minimize DtV_f	minimize AnV_h

5.4 Generate Initial Test Suite

We generate an initial set of test scenarios in modeFrontier to enable further optimization of the parameters towards the most critical scenarios. Based on the two

parameters we select (i.e. PoV_f and AnV_f) and the objective functions we define, we first compute the size of the initial test suite using the rule of thumb for DoE [102], as shown in Equation 1. N_{par} is the number of parameters and N_{obj} is the number of objective functions. As for APF, the size of the initial test suite is eight, given two parameters are selected, and two objective functions are defined for each optimization model.

$$Initial\ suite\ size = 2 * N_{par} * N_{obj} \tag{1}$$

Next, an initial suite of test scenarios can be generated by sampling the parameters based on the intended distribution. However, the realistic distribution for both DtV_r and AnV_h are unclear and are difficult to model or predict. Hence, we generate the initial test scenarios with the Latin Hypercube Sampling (LHS) strategy and uniform distribution. In LHS, the parameter space is divided into equal parts with respect to the target sampling size (i.e. the size of the initial test suite) and the sampling position is randomly chosen according to the parameter distribution [11]. LHS is considered superior to other sampling approaches like random sampling and ensures that the entire parameter space is covered as evenly as possible [11]. As there is no such real distribution for the selected parameters provided, we also use the uniform distribution to assure every parameter value interval is equally likely.

5.5 Create Optimization Models

We create the optimization models in modeFrontier by integrating the selected parameters, the objective functions, and the SPAS simulation platform. Similar to what has been presented in Figure 3, the parameters are defined as inputs to the optimization model and are used to generate scenarios for simulation. An initial set of values for the parameters are sampled preliminary with LHS and are considered the initial test suite to enable further optimization of critical test scenarios. The objective functions are the output of the optimization model and are optimized based on the parameter space and the completed scenario simulation.

We configure the number of optimization iterations to 80 based on the testing budget and computational resources available. In other words, the optimization model first runs the initial test scenarios (i.e. 8 scenarios) in the SPAS simulation platform and tracks the objective functions' value. Then the optimization model optimizes the selection of parameters for another 72 iterations based on the completed simulation results. Parallelization of optimization is possible in modeFrontier, given enough computational resources are available. Lastly, we select the optimization algorithm in modeFrontier based on our previous experience (i.e., Section 4) where pilOPT was used. In addition, we also replicate two optimization models (1 and 3 in Table 2) using MOSA to compare two different optimization algorithms and demonstrate the generality of our approach in using different optimization strategies. Thus, we create six optimization models in total, as shown

in Table 3. To clarify again, we do not aim to find the best optimization algorithm in this study but to integrate the entire tool-chain and a workflow for critical test scenario identification.

Table 3: modeFrontier optimization models and results for APF. By results, we mean the number of critical test scenarios identified with respect to the objective functions.

Model	Objective function 1	Objective function 2	Algorithm	Iteration	Result
1	minimize DtV_r	maximize AnV_h	pilOPT	80	41
2	minimize DtV_r	maximize AnV_h	MOSA	80	40
3	minimize DtV_r	minimize AnV_h	pilOPT	80	35
4	minimize DtV_f	maximize AnV_h	pilOPT	80	40
5	minimize DtV_f	maximize AnV_h	MOSA	80	29
6	minimize DtV_f	minimize AnV_h	pilOPT	80	30

5.6 Run Simulation and Optimization

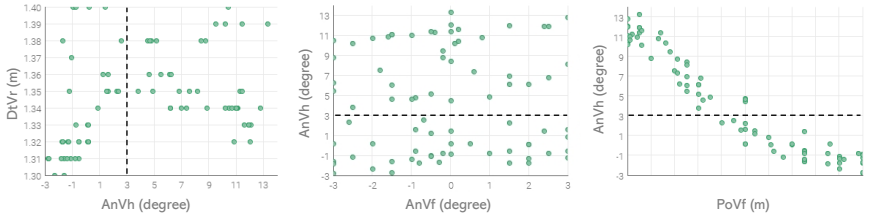
We start the optimization models in modeFrontier, and the optimization process runs automatically. For each optimization iteration, the simulation result is recorded and optimized with respect to the objective functions. After all iterations are completed, the optimization process terminates, and full results are saved. Since scenarios are simulated in the SPAS simulation platform and are triggered from modeFrontier, we have set a maximum time for a single simulation session to avoid suspending the entire optimization process.

5.7 Identify Critical Scenarios

The result of the optimization models can be visualized in modeFroniter using different charts or statistical analysis tools and be exported in many different formats. As mentioned earlier, we created six optimization models for APF, and each model consists of 80 evaluation iterations. By filtering the results with the criticality thresholds we define, the optimization models have identified 29 to 41 critical scenarios, as indicated by the last column (i.e. Result) in Table 3.

The critical scenarios are identified exclusively on one of the objective functions – AnV_h – and no critical scenarios identified for both DtV_r and DtV_f. As shown in Figure 6 (a) – the result of optimization model 1 from Table 3 for minimizing DtV_r and maximizing AnV_h using pilOPT, no critical scenario (i.e. $< 0.3\ m$) is identified in the DtV_r dimension as all scenarios resulted in a sufficiently large distance for DtV_r, which is considered as safe according to the industrial standard. That indicates the early implementation of the function we used is conservative on the distance to other vehicles. In contrast, 41 critical scenarios are identified based on the AnV_h which are greater than 3° to the central line of the parking slot.

Furthermore, Figure 6 (b) shows the correlation between parameter AnV_f and the objective AnV_h. The result indicates that AnV_f does not have a general effect on AnV_h and it is randomly distributed regardless the value of AnV_f. In contrast, an explicit pattern is drawn on PoV_f and AnV_h in Figure 6 (c), in which AnV_h keeps increasing when PoV_f decreases. When PoV_f is lower than a specific value, AnV_h is over the criticality threshold 3° and scenarios are identified as critical scenarios.



(a) Optimization result of critical scenarios for both AnVh and DtVr

(b) Critical scenarios optimized on AnVf (i.e., angle of front vehicle)

(c) Critical scenarios optimized on PoVf (i.e., position of front vehicle)

Figure 6: Result of minimizing DtV_r and maximizing AnV_h using pilOPT. The dash line in the sub-figures is the criticality threshold for AnV_h and the dots are the scenarios executed in the simulation. Scenarios on the right side of the dash line in sub-figure (a) and above the dash line in sub-figure (b) and (c) are the critical scenarios identified on with AnV_h larger than 3°. The scale of PoV_f in sub-figure (c) is removed for confidentiality reasons.

The results are consistent when using other optimization models with different combinations of objective functions. We identify critical scenarios on AnV_h only, and the visualized results indicate that AnV_h gets larger and exceeds the criticality threshold when PoV_f declines. The observations suggest that adapting the slot length and angle of the stationary vehicle does not generate critical test scenarios for APF with respect to the distance to the stationary vehicles. However, both of them lead to critical test scenarios where the angle of the host vehicle exceeds 3°. A clear trend is observed on the slot length that smaller slot length generally increases the angle of the host vehicle, which means a bad orientation to the parking slot after the parking maneuver is done.

Lastly, pilOPT generally identify more critical scenarios than MOSA for APF in this case, although there are no significant differences between them consistently. For the optimization models that minimize DtV_r and maximize AnV_h, pilOPT identifies 41 critical scenarios, and MOSA identifies 40. As for the models that minimize DtV_f and maximize AnV_h, pilOPT identifies 40 critical scenarios where MOSA identifies 29. According to the results, pilOPT performs better than MOSA, while further comparison between these two algorithms is required. Since

we do not aim to address the best optimization algorithm in the current study, we have demonstrated that our approach effectively identifies critical test scenarios and is general to different optimization algorithms or strategies.

6 Discussion

In this paper, we extend an approach for critical test scenario identification for autonomous driving and have used it for testing real autonomous driving systems. We argue that testing all possible driving scenarios in real road traffic is impractical, since it is expensive, time-consuming, and may still not cover all the rare-occurring traffic situations [57, 109]. In contrast to Kalra et al., who claim that millions or even billions of miles of driving tests are required to demonstrate the reliability of an autonomous vehicle [55], testing of autonomous driving functions must be based on a feasible number of test scenarios and focus on the most critical ones [60, 81]. Using critical scenario identification and simulation is considered a good alternative to address the gaps as mentioned above and enable testing of autonomous driving functions in a more efficient way [60, 68, 84].

In our approach, we integrate the existing engineering tools and a workflow as a complete solution for critical test scenario identification. In contrast, existing studies mostly present a partial solution for critical scenario identification and barely provide a complete tool-chain [41]. Applying a partial solution in practice may require additional work to integrate such an approach with the missing components or even not compatible with the used technical environment. We integrate different tools and a workflow into a systematic approach, which is complete and easy to use. The proposed approach relies on optimizing the parameter selection and simulation of the scenarios. As the tools involved are exchangeable, the approach is flexible and generic for testing different autonomous driving functions that are not subject to specific tools, techniques, or sensors employed in the function or simulation.

We demonstrate the effectiveness of our approach for critical test scenario identification, using real autonomous driving functions in both high-speed and low-speed maneuvering domains. This is different from the most common approach for validating proposed solutions for critical scenario identification in existing studies, which use a simple implementation of the autonomous driving function or publicly available driving components like DeepDriving [36]. Besides, many studies demonstrate the effectiveness of their approaches based on limited settings, such as a pedestrian step-out scenario [11] and certain scenarios from Carla Scenario Runner Library [30]. Even though the potential of such approaches might be extended, the connection to real autonomous driving functions and to find critical scenarios in general is not explicitly provided.

The two cases we present in Sections 4 and 5 include the actual work we implement and the results achieved on real autonomous driving functions using the

proposed approach. While the results are generally effective in finding the critical test scenarios for the given autonomous driving systems, we would like to stress that they are merely an early version of the autonomous driving systems. Thus, the results are subject to the current design and specifications of the systems when conducting the study. The main purpose is to demonstrate the industrial relevance and applicability of the approach in practice.

Future improvement and extension of our approach regarding its design and implementation are multi-fold, including, e.g. scenario composition, parameter distribution, and optimization algorithms. First, the composition and representation of scenarios can be improved to include different driver behavior models and enable the definition of complex spatio-temporal interactions between different entities within the driving maneuver. As highlighted by Feng et al. [33], existing studies mostly handle only low-dimensional scenarios, whereas the actual operational design domain for the autonomous driving functions is much more complicated. OpenDrive and OpenScenario, as used by Zhang et al. [109] and Erdogan et al. [32], to define static and dynamic elements in a full driving scenario in a structured way are good references to explore.

Second, realistic distribution of the relevant parameters selected should be investigated to improve the realism of the scenarios and real occurrence of the scenarios. As articulated by Batsch et al. [11], scenario-based testing sampling requires a true distribution of the parameters. A shift in the distribution may impact the relevance and potential damage of the scenarios [85], thus the distribution of parameters are important and need to be identified [30]. Different sampling approaches such as adaptive sampling [75], importance sampling [33], or modelling the distribution are a few candidates to be further studied.

Thirdly, we also propose to evaluate different optimization algorithms to best fit the generation of critical test scenarios for different autonomous driving systems and use parallelization to improve the efficiency of the simulation and optimization [82]. They are good directions to be sorted out in future research yet not the goals in the current study. Especially that parallelization is already a feasible option in optimization tools like modeFrontier; it's more about the computational resources that can be allocated count. Our primary focus in this work is to establish a complete approach for critical test scenario identification for autonomous driving and demonstrate the effectiveness of such an approach for the realistic testing of autonomous driving systems. As a preliminary step, tools and a workflow are integrated, and critical test scenarios are generated for real autonomous driving systems from the industry. Thus, it constitutes a basis for further exploration and refinement of the approach in practice.

Given the enormous challenges of testing autonomous driving systems, we face [61, 62], the importance of using simulation and critical test scenario generation increases steeply [52]. Further, as stated by Beglerovic et al., selection of relevant parameters, objective functions, and appropriate evaluation criteria is a non-trivial task since each of them comes with its own challenges, and the qual-

ity of critical test scenario generation is highly dependent on them [12]. Despite that sub-components within our approach can be further expanded and improved, we believe our work is worth the efforts and has a huge potential in the future in ensuring the safety and reliability of autonomous vehicles. Particularly since very few studies have been reported for presenting a complete solution for critical test scenario identification that is general for different autonomous driving systems, according to Hallerbach et al. [41].

7 Conclusion

Safety and reliability are indispensable properties for autonomous vehicles, yet there is no common standard way to test autonomous driving functions systematically and efficiently. Conventional requirements-driven testing approaches are impeded due to uncertainty of the operational environment and the complexity of the driving scenarios. Thereby, identifying the most critical scenarios for testing the autonomous driving systems is developed.

We establish a complete approach with integrated tools and a workflow in this study to explore critical test scenarios and facilitate the testing of autonomous driving systems. As a pilot study, we implement the approach on two autonomous driving systems from the industry by partnering with Volvo Cars. The results suggest that our approach effectively identifies critical test scenarios. The identified scenarios can be used to substantiate test cases for autonomous driving systems either in simulation or in the real world.

Future extension of the approach aims to improve the scenario representation, incorporate the realistic distribution of the parameters, and compare the effectiveness of different optimization algorithms. Eventually, the study provides a feasible and complete tool-chain for critical test scenario identification for autonomous driving and a basis for building sub-components further upon. Given the widespread attention on autonomous driving and the corresponding challenges for testing the enabling functions, we shed light on testing different autonomous driving systems efficiently and effectively.

8 Acknowledgement

This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP). Thanks to our colleagues in SERG and Volvo Cars for their review of earlier versions of the manuscript.

9 Statements and Declarations

There is no financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

PAPER IV

A VEHICLE–PEDESTRIAN TIME-TO-COLLISION MODEL FOR TESTING OF AUTONOMOUS DRIVING SYSTEMS

Qunying Song, Per Runeson and Stefan Persson, Technical report, 2022.
DOI: Working manuscript

Abstract

While autonomous driving systems are expected to reduce road accidents and improve traffic safety, understanding intensive and complex traffic situations is essential to test such systems under realistic traffic conditions. In this work, aimed to generate critical test scenarios, we propose a new model that defines the distribution of TTC (Time-to-Collision) for the vehicle–pedestrian interactions at unsignalized crossings, based on the traffic density. The model is used as an input for the optimized identification of critical test scenarios. We validate the model using real traffic data collected in Sweden. The result indicates that the model is effective and consistently upholds the real distribution, especially for critical TTC below 3 seconds. We also demonstrate its use to test autonomous driving systems by connecting it to the critical test scenario identification for an auto-braking function from the industry. As a first step, our contribution is a worst-case model defining the TTC distribution that serves as input to testing autonomous driving systems and ensures the realism of the test scenarios.

1 Introduction

Autonomous driving systems, for example, auto-braking and auto-steering, are said to reduce road accidents and improve road safety, as they do not encounter challenges for human drivers, such as fatigue, distraction, and drunk driving [21]. Efficient sensor systems, like radars and cameras, have enabled capturing of the environment and traffic dynamics. Yet, the demand for testing autonomous driving systems based on understanding and analyzing complex traffic situations has never been greater [93]. The prevalence of autonomous driving systems calls for imminent efforts to model the traffic flow and road safety to constitute a basis for testing those systems under a realistic traffic setup.

One of the most critical places in the road traffic are intersections where a high concentration of vehicle–pedestrian interactions appear [21]. An existing study has revealed that 25% of the pedestrian fatalities in Europe occur while using a pedestrian crossing [39], which made pedestrian crossings a rather crucial scene to investigate especially at unsignalized crossings, where the right of way is not always clear [21]. According to Bella et al., most of the vehicle–pedestrian incidents are caused by the drivers fail to yield to a pedestrian [13]. In addition, pedestrians are more flexible and less regulated on their crossing behaviors, which makes their behaviors often unpredictable, or even illegal [29]. As a result, pedestrians have become one of the most vulnerable users on the road. Thus, autonomous driving systems, like autonomous braking function, to support vehicle–pedestrian interaction is of high interest, and consequently, methods and models for testing such systems.

We perform this study in an industrial, autonomous driving system context, where test scenarios are generated from a model of the system operation environment [100]. Based on the co-optimization of the test suites generated from the environment model and the autonomous driving system, critical test scenarios are identified with an objective function. For example, the objective may be to find test scenarios where the vehicle is close to collision, which is then turned into test cases for testing autonomous driving functions, which may occur both in simulated and real vehicle environments.

The research goal of this study is to propose and validate a worst-case model of vehicle–pedestrian interactions that facilitate testing of autonomous driving systems. The model takes the macroscopic characteristics of the traffic, namely the traffic density, and predicts the cumulative time-to-collision (TTC) distribution for vehicle–pedestrian interactions at unsignalized crossings.

The model should be as simple as possible while capturing the essential characteristics of the traffic situation. Thereby it contributes to defining the operational design domain (ODD) of the safety-critical system [40]. By worst-case, we mean careless drivers and pedestrians that do not pay sufficient attention to the ongoing traffic. We are particularly interested in the distribution of the high-risk scenarios with critical TTC values. The predicted TTC distribution serves as an input for test

scenario generation and optimization for autonomous driving systems. We validate the model using a real-world traffic data set collected in Sweden and demonstrate its use by deploying it to critical test scenario identification for an autonomous braking function. To the best of our knowledge, no such model has been established before, and no existing research was found on incorporating similar models for testing the autonomous driving systems, although they are prerequisites for safety assessment [40].

The rest of the paper is organized as follows. Chapter 2 describes the existing literature we found on traffic modelling for vehicle–pedestrian interactions and critical scenario identification. Chapter 3 describes the context and method we use for conducting this study. Chapter 4 describes the model construction and Chapter 5 presents the model validation. In Chapter 6, we particularly demonstrate the use of the model for testing autonomous driving systems by using an autonomous braking function. Lastly, we discuss the model and its validity in Chapter 7 and conclude the paper in Chapter 8.

2 Related Work

Existing studies reporting on testing of autonomous systems have proposed using model-based approaches and formal verification to facilitate automated test generation and execution [45]. One primary challenge with the approaches mentioned above for highly autonomous systems is the adaptive behavior of the system and the unpredicted environment that is unknown during design. It essentially leaves an infinite set of scenarios to be covered during test [99]. To address the said challenge, we and among other studies, proposed a critical scenario identification approach to reduce the test scenarios by optimizing the possible test scenarios and identifying the most critical ones for testing through simulation [58, 60, 100].

Our approach relates to a more general concept of operation design domains (ODD) [40]. An ODD is a definition of the operational environment of an autonomous driving system and is used to assess the safety of such systems. The ODD contains a world model that models use cases of the system. In our approach, we use the world model to generate test scenarios for both simulated and real vehicle testing.

Testing of autonomous driving systems has to be based on analyzing and modelling the traffic flow as well as the interactions between different road users [18, 20]. Jiang et al. collected urban midblock crosswalk data at multiple locations in Germany and China and observed a Weibull distribution of the TTC in their study [54]. Fu et al. focused on analysing the pedestrians' safety at unsignalized crossings during night time, using thermal video data and the surrogate measurement of Post-Encroachment-Time (PET), representing the time difference between a vehicle is leaving the area of encroachment and a conflicting vehicle entering the same area. Their result has shown a significant impact on the pedestrian's safety

during the night [34]. A study by Chen et al. used unmanned aerial video data and analysed the surrogate safety of pedestrian-vehicle conflicts at intersections, using both TTC and PET [22]. Other studies on pedestrian models for autonomous vehicles at crossings include Jayaraman et al. [51] modelling the pedestrians' decisions at intersections, Zhang et al. [113] predicting the path of pedestrians, and risk analysis at unsignalized crosswalks using data mining techniques [76]. While the studies introduced mainly focus on analysing or modelling the vehicle–pedestrian interactions based on collected traffic data, the data used is rather limited for deriving a generic model and location-specific. In addition, they do not include the potential on how such models can relate or contribute to the testing of autonomous driving systems. In contrast, our work proposes a model that is not relying on any existing traffic data and serves as an indispensable input for critical test scenario identification.

Recent studies also show the possibilities of using deep learning approaches for testing autonomous systems and have claimed the efficacy of such approaches. Specifically, Porres et al. describe an approach for identifying critical test scenarios for maritime collision avoidance systems, using a neural network that is trained in the runtime [82]; Parthasarathy et al. introduce a systematic framework for generating driving maneuvers for testing autonomous driving systems [77]. However, such approaches may lack robustness and generality due to the incompleteness and biases in the training data. Therefore, they may not identify scenarios that are not represented in the training data set, and for most of the time, the scenarios recorded are not critical [58]. As a comparison, our approach employs a traffic model that predicts the worst-case distribution of the driving scenarios and enables critical test scenario identification under a more realistic setup.

3 Research Context and Method

As already mentioned in Chapter 2, one of the principal challenges for testing autonomous driving systems is the infinite set of potential scenarios to be covered during test [99]. The limitations of exhaustive testing in the real world due to limited test budgets and rare occurrence of risky events have led to the exploration of more efficient approaches to reduce the number of tests through simulation, like our critical scenario identification approach [100], as summarized in Figure 1.

The critical scenario identification approach is used for generating critical scenarios for testing autonomous driving systems. The general idea of this approach is to first model the operation environment and system under test, based on selected parameters and objective functions, and generate an initial test suite. Next, the selection of critical test scenarios is optimized through simulation to identify the scenarios that are beyond the criticality thresholds, similar to search-based testing [44]. The identified critical scenarios are then executed during testing, either

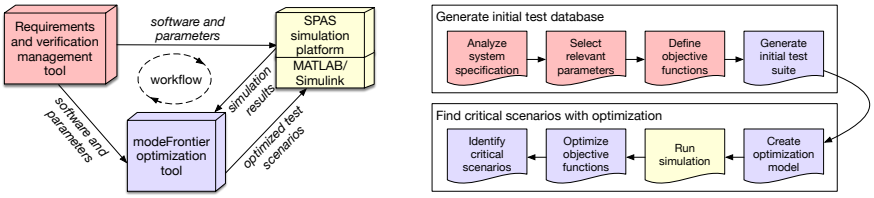


Figure 1: Overview of the critical scenario identification approach (adapted from our previous work [100]).

in simulation platforms or real-world traffic. We refer to our previous work [100] for more information on this approach and related tools that are involved.

One major step in this approach is selecting relevant parameters that define a driving scenario, including both the value range and distribution of each parameter and ensuring that the scenarios are within the desired boundaries and frequency. Even though an enriched set of common distribution functions are provided in applications like modeFrontier, customized and realistic distribution is more favourable to ensure the feasibility of the test scenarios. Yet the distribution of a specific parameter, for example, TTC, under a certain test scene, for example, at unsignalized crossings with different traffic densities, is still not well understood and properly modelled.

We conducted our study in the context of the said approach. We aim to construct a worst-case model that predicts the distribution of TTC for vehicles–pedestrian interactions based on the traffic density. We use the model as an input for critical scenario identification for testing autonomous driving systems. We have conducted the study in four steps.

1. The model is constructed based on a Poisson distribution of the vehicles' and pedestrians' arrival. The model takes two parameters, namely the vehicle mean arrival rate and pedestrian mean arrival rate and outputs the predicted cumulative distribution of TTC under the given density.

2. The model is implemented in Matlab, and examples of the prediction by the model under different traffic densities are generated as a preview.

3. The model is validated through a real traffic data set collected in Sweden by Viscando. The TTC distribution predicted by the model is compared against the distribution from real traffic data.

4. We study an industrial autonomous braking function and demonstrate using our model to facilitate the critical scenario identification for testing the function.

The work is conducted under the design science paradigm [89]. The problems of test scenario generation challenges are conceptualized in the industrial con-

text [100]; we report our design of the traffic model and optimization procedures (1 and 2 above) and validate the solution against real data (3) and a potential usage context (4).

4 Model Construction and Simulation

In this section, we describe the construction of the vehicle–pedestrian model based on the Poisson distribution of vehicles' and pedestrians' arrival at an intersection. In addition, we also implement the model in Matlab and generate examples of prediction of the cumulative TTC distribution under different traffic densities as a quick check of the model.

4.1 Model Construction

The Poisson distribution [2] is selected as the primary component for modelling the arrival of vehicles and pedestrians at an intersection since their occurrences in the traffic are random and independent. Therefore, given the mean rate of the vehicles and pedestrians, hereafter referred to as λ_v and λ_p respectively, the Poisson distribution expresses the probability of having a certain number of vehicles or pedestrians occurring within a given time frame. Chen uses the same distribution at al. for modelling the arrival time of pedestrians at an unsignalized crossing and is used for evaluation of the safety of the driving strategy of the autonomous vehicles [21]. Besides, Hu et al. discuss using the Poisson distribution for modelling the number of crashes for road traffic safety [46].

Further derivation of the standard Poisson distribution leads to the probability of a certain time interval between the occurrence of two events, that is, the likelihood of having a specific time interval between two vehicles, given the mean rate of λ_v. More specifically, Equation (1) gives the mathematical representation of the probability of the time interval between vehicles, herein $Intv$ is defined as the time interval between vehicles, k number of vehicles, t the specified time frame in seconds, and P is short for probability.

$$\begin{aligned} P\left(Intv \leq t\right) &= 1 - P(Intv > t) \\ &= 1 - Poisson(\textit{0 vehicle in time } t) \\ &= 1 - \frac{\left(\lambda_v t\right)^k e^{-\lambda_v t}}{k!}, \text{ given k = 0} \\ &= 1 - e^{-\lambda_v t} \end{aligned} \tag{1}$$

Based on the previous equation, Equation (2) is the probability of a certain time interval between two vehicles. $\boldsymbol{t}$, in bold, is an interval instead of an exact value. A constant time step Δt is introduced in this equation, which is a small time unit

subject to the desired level of precision. We have selected 0.1 seconds as Δt in the current study, as it is a typical precision for TTC in previous studies [24, 91] and an adequate precision to model the distribution in our work. The derived equation is used for model construction, and integration of the equation in the model is detailed in a later paragraph in the same section.

$$\begin{aligned} P\left(Intv = \boldsymbol{t}\right),\ &\textit{herein } \boldsymbol{t} \textit{ is interval } (t - \Delta t,\ t] \\ &= P\left(Intv \leq t\right) - P\left(Intv \leq t - \Delta t\right) \\ &= (1 - e^{-\lambda_v t}) - (1 - e^{-\lambda_v (t - \Delta t)}) \\ &= e^{-\lambda_v (t - \Delta t)} - e^{-\lambda_v t} \end{aligned} \tag{2}$$

TTC is another primary component used in our model, and according to Schwarz, TTC is originally defined as the time until two road users collide if they continue with their current speed and direction [93]. TTC is a surrogate measurement of road safety and describes how imminent a collision will happen. A lower TTC value means a higher risk for a collision to take place, and vice versa [54]. A low TTC value, for example, under 3 seconds, is considered critical for the vehicle–pedestrian interaction at crosswalks and autonomous driving in the urban environments by Schneemann et al. [92]. Another study by Kluck et al. has considered TTC below 1 second as critical for optimizing parameters for testing the advanced driver-assistance systems [60]. The TTC family of measures is reported to be the most commonly used safety indicator in a review study by Laureshyn et al. [63].

In the current work, we define TTC as *the time remaining before an oncoming vehicle hits a pedestrian at the moment that the pedestrian is entering the intersection, given that the vehicle and pedestrian continue in their current speed and direction*.

$$\begin{aligned} P(\boldsymbol{TTC} \leq t) &= P\left(D \leq t \cdot V\right) \\ &= P(Intv \leq t) + \sum_{Intv=t}^{M_{intv}} P(Intv) \cdot \frac{t \cdot V}{Intv \cdot V} \\ &= P(Intv \leq t) + \sum_{Intv=t}^{M_{intv}} P(Intv) \cdot \frac{t}{Intv} \\ &= 1 - e^{-\lambda_v t} + \sum_{Intv=t}^{M_{intv}} \frac{(e^{-\lambda_v (t - \Delta t)}) - e^{-\lambda_v t}) \cdot t}{Intv} \end{aligned} \tag{3}$$

Equation (3) is the derived model for cumulative TTC distribution by integrating the distribution of vehicles' interval, as listed in Equation (1) and (2), and the aforementioned definition for TTC. Herein, D is the distance between vehicle and

$$P(TTC \leq t) = \begin{cases} N/A & \text{, if no pedestrian occurs} \\ Equation(3) & \text{, if pedestrian(s) occur} \end{cases} \tag{4}$$

$$P(TTC \leq t) = \underbrace{\left(1 - e^{-\lambda_p t}\right)}_{①} \underbrace{\left(1 - e^{-\lambda_v t} + \sum_{Intv=t}^{M_{intv}} \frac{t \cdot (e^{-\lambda_v (t-\Delta t)} - e^{-\lambda_v t})}{Intv}\right)}_{②}$$

① Probability of pedestrian(s) occurrence, as pre-condition for computing TTC

② Probability of vehicle–pedestrian TTC $\leq$ t

(5)

pedestrian, and V is the vehicle's speed. Since we consider the worst-case situations where drivers and pedestrians cross the intersection without paying enough attention to the traffic, the model is, in essence, the summation of the probability of each possible vehicle interval multiply by the probability of a pedestrian occurs in a specific time in front of the vehicle, given that the vehicle and pedestrian move as intended and no interference while crossing the intersection. Besides, for vehicle intervals that are smaller than a given time *t*, TTC is naturally smaller than *t* regardless of where the pedestrian is located between vehicles.

A practical part of this model is the selection of M_{intv}, which is the theoretical maximum time interval between two vehicles. M_{intv} can be selected as either being the maximum vehicle interval based on the empirical data at an intersection if available or the maximum interval that remains a certain probability under a Poisson distribution. We have set M_{intv} to 460 seconds, since the probability of a vehicle appearing beyond this interval is lower than 0.01 even under an extremely low vehicle density, like λ_v equals 0.01/s (36 vehicles per hour). Furthermore, the probability of having a pedestrian occurs at a specific time in front of the vehicle, e.g., 3 seconds, is low, given this vehicle interval. Thus, the probability of having a critical TTC, e.g., below 3 seconds for vehicle interval above 460 seconds, is negligible.

Equation (3) indicates the distribution of TTC given a pedestrian enters the intersection. However, TTC is valid only when a vehicle–pedestrian interaction occurs; otherwise, it will be illogical to compute a TTC without a pedestrian occurring. Thus, the model should be further divided into two branches under two different circumstances, as shown in Equation (4). In the first branch, TTC is not applicable without a pedestrian occurring, and the second branch adopts Equation (3) with pedestrian(s) occurring between two vehicles. Integration of these two branches will be the final model, as in Equation (5), which handles the overall TTC distribution of vehicle–pedestrian interactions at the intersection.

4.2 Model Implementation

We implemented the model in Matlab to preview how the model predicts the cumulative distribution of TTC under different traffic densities. We have selected three arbitrary traffic densities from low to high, with λ_v = 0.02/s and λ_p =0.01/s; λ_v = 0.04/s and λ_p =0.02/s; λ_v = 0.1/s and λ_p =0.04/s, respectively. The plotting results are given in Figure 2, where we show the low TTC part (below 5 seconds) since the low TTC section is the most critical part in road traffic. The lines in the figure show the cumulative distribution of TTC under the given traffic densities. The differences are obvious: higher traffic density has a higher probability of getting a low TTC situation in the traffic. This result is fully consistent with the findings in another empirical study reported by Loukaitou-Sideris et al. in their traffic safety study [66], where higher traffic density shows a significant relationship to the number of pedestrian–vehicle collisions.

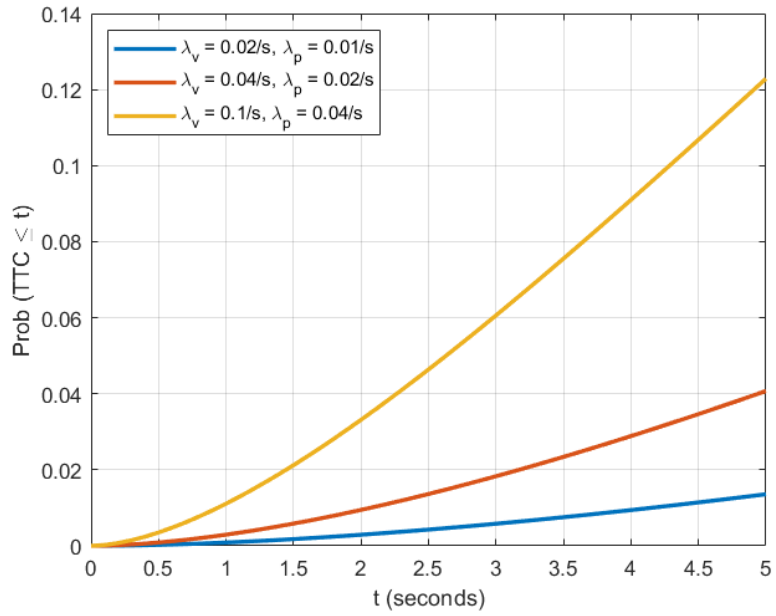


Figure 2: Model prediction with three different example traffic densities, λ_v is λ_p are mean rate of vehicles and pedestrians per second. x axis represents specific time t in seconds, and y axis represents the probability of TTC $\leq t$ in given traffic density.

5 Model Validation

In this section, we describe the procedures that we carried out for model validation and analysis of the results. Specifically, we first introduce the data set collected from real traffic, then the pre-processing of the data and the actual validation. The validation is conducted by comparing the prediction by the model against the real distribution of TTC based on the collected data under the same traffic density.

5.1 Validation Data Set

The real-world data set for model validation is provided by the Swedish company Viscando and contains the traffic data from recording at an unsignalized intersection in Linköping, Sweden, from 2017/09/04 15:12 to 2017/09/07 11:17. Generally, the data set represents time-resolved trajectories of all types of road users during these dates. More specifically, it captures different road users, such as pedestrians, bicyclists, and vehicles, through edge processing of stereo-vision data collected using Viscando's sensor system. The trajectory info includes the timestamp and location coordinates speed and type of per road user.

The data set is provided as a CSV(comma-separated values) spreadsheet. Each row is a trajectory snapshot that associates with a user id, timestamp, x and y coordinates, speed and type of the road user. The position and speed of each road user are measured up to 20 times per second, and all users within an approximately 30x30 meter square around the intersection are tracked. Figure 3 is a bird-view picture of the location that explains the possible traffic flow at the location under-recording. There are three entry–exit locations for vehicles clock-wise: one at the lower-left (refer as 'A'), one at upper-middle (refer as 'B'), and one at upper-right (refer as 'C'); thus, six directions of vehicle movement are expected. Besides, one unsignalized pedestrian crossing is constructed close to 'A'.

The pre-processing of the initial data set is mainly conducted in three steps, each implemented and automated in a corresponding Python script. First, we divide and sort the data by hour, as the traffic density is dynamically changing and is different in each hour. We filter only the pedestrians with a trajectory that overlaps the unsignalized pedestrian crossing and vehicles approaching the crossing from the lower-left bound (the 'A' location). The vehicles that are occurring from the upper side (location 'B' and 'C') are either in a right angle turn or highly interrupted by vehicles in other directions; thus, they are not considered for proper TTC computation. Second, we identify the partial pedestrian trajectories within the unsignalized pedestrian crossing and identify the closet vehicle that is approaching. Thirdly, we compute the TTC based on the location of the vehicle and pedestrian as well as the vehicle speed and multiply the TTC distribution by the probability of pedestrians' occurrence based on Poisson distribution, according to the first component in Equation (5).

Figure 3: Bird view of the location for traffic recording at Linköping, Sweden (picture provided by Viscando)

After initial pre-processing of the original data set, we have obtained trajectories for vehicles and pedestrians in 61 hours and 50 hours, respectively. We filter the data with less than 20 vehicles and pedestrians in an hour since too few TTC samples can be generated based on that low traffic volume. The distribution of TTC would be rather discrepant and random. As a result, 28 hours remain with trajectories for both vehicles and pedestrians. We selected the 6 hours with most pedestrians or vehicles and got a total of 10 hours of pre-processed data for validation and analysis purposes, as listed in Table 1. The selected hours are distributed in all four days during recording and are typically distributed from $7-9$ and $15-19$. Pedestrian counts average 63 per hour and range from 30 to 104. The vehicles have a mean rate of 159 per hour, with the least 100 and the most 332 vehicles.

5.2 Validation Result and Analysis

The validation results are separated in two figures, as Figure 4 and Figure 5, due to the limitation of the page height. More specifically, Figure 4 contains six sub-figures, corresponding to the first six selected hours in Table 1, and Figure 5 contains four sub-figures for the last four selected hours. In each sub-figure, the TTC distribution predicted by the model is plotted against the TTC distribution of the real traffic data. The tail part with TTC less than 3 seconds is highlighted in yellow, which is the most critical part that is of interest for safety purposes in existing

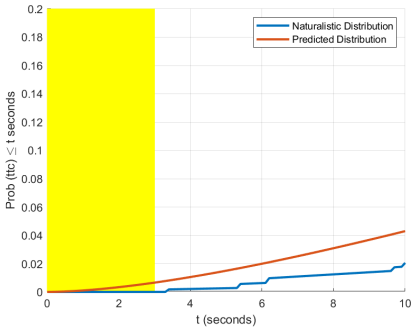
Table 1: Summary of the pre-processed data set for model validation

Date with Hour	**Pedestrian Count (λ_p)**	**Vehicle Count (λ_v)**
2017-09-04 17	30 (0.008/s)	118 (0.033/s)
2017-09-05 07	67 (0.019/s)	332 (0.092/s)
2017-09-05 15	77 (0.021/s)	115 (0.032/s)
2017-09-05 16	69 (0.019/s)	106 (0.029/s)
2017-09-05 18	42 (0.012/s)	146 (0.041/s)
2017-09-06 07	51 (0.014/s)	260 (0.072/s)
2017-09-06 08	45 (0.013/s)	161 (0.045/s)
2017-09-06 15	73 (0.020/s)	100 (0.028/s)
2017-09-07 07	104 (0.029/s)	152 (0.042/s)
2017-09-07 08	71 (0.020/s)	100 (0.028/s)

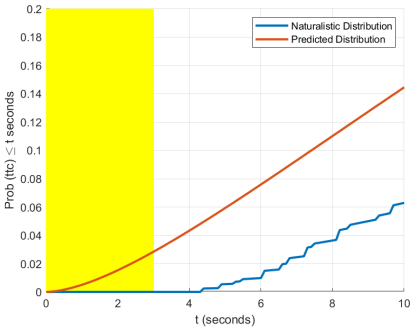
studies [92] and in our work. We have also plotted the TTC distribution and comparison up to 10 seconds in the figures to show the general trend of the cumulative TTC distribution. Yet, larger TTC values are considered safer in general and not interested in safety analysis [60]. Our model can also give predictions above 10 seconds, yet they are beyond the scope and discussion in this work.

As indicated by the two figures and their sub-figures, the real distribution of the low TTC (less than or equal to 3 seconds) is consistently under the prediction by the model. In particular, 9 out of 10 selected hours have no distribution on this part and only one (Figure 5a) hour has approximately 0.0009 distribution on getting a low TTC. The mean difference between the real and predicted distribution for the selected 10 hours is 0.0164, with the maximum 0.0284 and minimum 0.0066. Since we aim to construct a worst-case model for the TTC distribution, the validation result positively supports the expectation. It shows no exceptions under different traffic densities for the low TTC part.

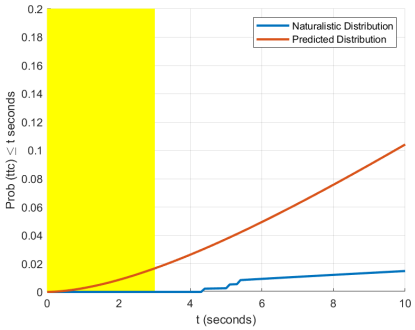
Even though scenarios with high TTC values are not considered critical in the typical traffic scenarios, we still analyze the second part – a section with greater than 3 seconds TTC in the plots. Generally, the model is still defensible and valid in 9 of the 10 selected hours. One exception can be found and is shown in Figure 5d, where the real distribution of TTC surpasses the predicted distribution in roughly 8.2 seconds. It is due to the randomness of vehicles or pedestrians' arrival in real traffic. In some cases, we may get very different vehicle–pedestrian interactions with the same traffic density in an hour. To further clarify, our model is not a worst-case model in the statistical sense, but it is instead a feasible approximation based on Poisson distribution, especially for low TTC. In addition, the result in Figure 5d could also be due to that we have a relatively low pedestrian count, 71 in this hour; thus, the actual distribution of TTC could be highly stochastic that depends on the crossing behaviors and vehicle–pedestrian interactions that appeared in this hour. Given a larger volume of pedestrians and vehicles, which implicitly



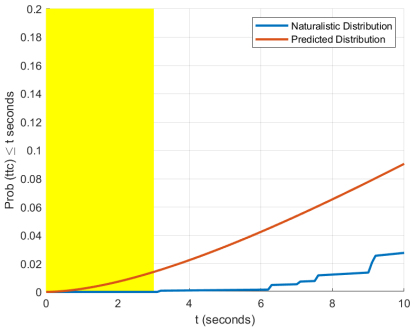
(a) 2017-09-04 17: 30/118



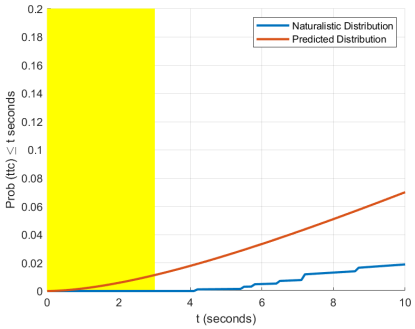
(b) 2017-09-05 07: 67/332



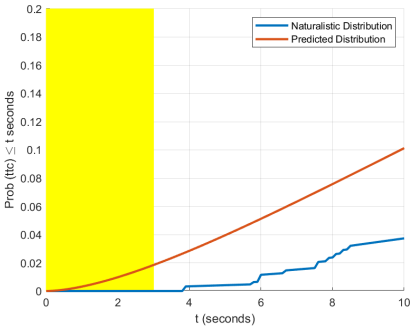
(c) 2017-09-05 15: 77/115



(d) 2017-09-05 16: 69/106

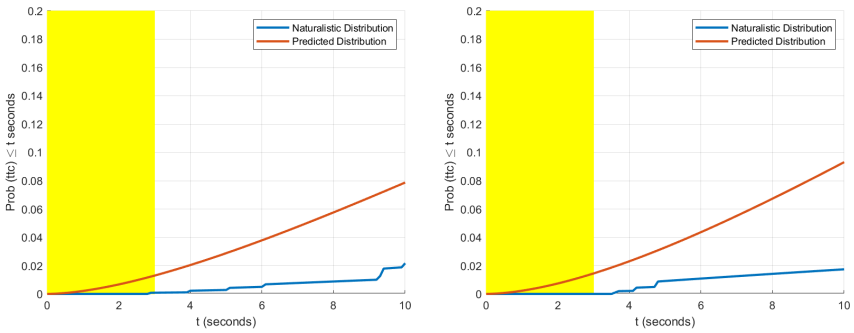


(e) 2017-09-05 18: 42/146



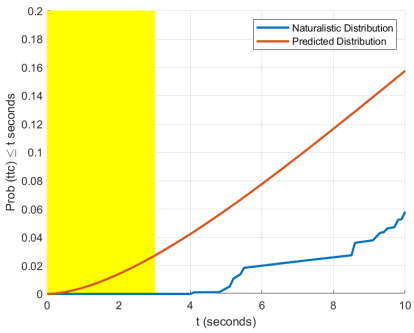
(f) 2017-09-06 07: 51/260

Figure 4: Comparison of cumulative TTC distribution, below 10 seconds, from the model prediction and real traffic recording, for the first six selected hours. The critical section with TTC less than 3 seconds is highlighted in yellow. The caption in the sub-figures starts with the date, the hour, a colon sign, and then the count for pedestrians and vehicles in the mentioned order.

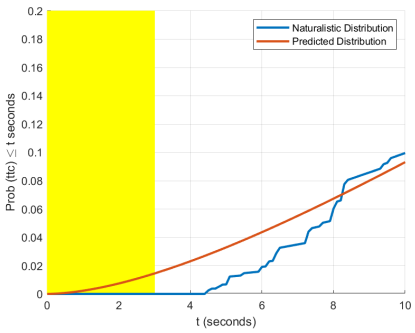


(a) 2017-09-06 08: 45/161

(b) 2017-09-06 15: 73/100



(c) 2017-09-07 07: 104/152



(d) 2017-09-07 08: 71/100

Figure 5: Comparison of cumulative TTC distribution from the model prediction and real traffic recording, for the last four selected hours for validation, as given in Table 1

means more TTC values can be obtained, the distribution of TTC from different hours is expected to converge and be more stable.

Overall, the validation results give a positive indication of the model as a prediction of the worst-case distribution of the cumulative TTC for vehicle–pedestrian interactions at unsignalized crossings, both in the critical TTC part and the other part. Further, the model effectively predicts TTC distribution under different densities in realistic traffic.

6 Model Utilization for Testing

In this section, we briefly introduce the auto-braking function from Volvo Cars based on existing publications. We also explain how the distribution of TTC predicted by the model can be used as an input for critical test scenario generation for this autonomous braking function.

6.1 Autonomous Braking Functions

Several automated emergency braking functions have been developed, such as City Safety by Volvo Cars, Front Assist by Volkswagen, and Pre-SAFE Brake by Mercedes-Benz [42]. While the techniques involved are diverse, the general purpose of such systems remains the same: to support the human drivers in avoiding collisions by automatically performing an immediate braking maneuver.

The autonomous braking function from Volvo Cars gives braking support to the drivers in emergent situations where a collision is likely to happen, under a certain speed limit [25]. The function supports the drivers in two ways: one is to raise warnings if a frontal object is detected and identified as risky; the other is to apply certain braking torque to decelerate the vehicle if it is too close to the frontal object and the driver performs no intervention. The systems can be effective under many different traffic conditions. For example, when the host vehicle is approaching another vehicle standing still in front, a pedestrian illegally coming across the road without noticing the approaching vehicle, or an animal passing the road during nighttime etc.

Information of the braking function can be found through Volvo Cars and open access pages [1]. Particularly, specifications of the early version of the braking function have been published [24, 25]. Some details of the development, components, and mechanisms of the function are reported by Coelingh et al., where TTC to the frontal object and speed of the vehicle are two primary factors that decide whether the braking function should intervene or not, and when. According to the publication, the system should only intervene when the estimated TTC to the frontal object and the speed of the vehicle are below certain thresholds [24].

6.2 Critical Test Scenario Identification

Our research context, as described in Chapter 3, is to deploy the critical scenario identification approach for testing the autonomous driving systems. One principal challenge is to model the distribution of the parameters, for example, the distribution of TTC for vehicle–pedestrian interactions at unsignalized crossings for testing the aforementioned auto-braking function.

Our model is a prerequisite for addressing this gap and serves as an input to the critical scenario identification approach in testing the auto-braking function. Specifically, the model provides a worst-case TTC distribution of the vehicle–pedestrian interactions under different traffic densities to enforce the feasibility and probability of a given scenario in the real world, particularly for those with a very low TTC value and are of interest for the test. By using this model, the critical test scenario generation is based on real distribution, rather than assumptions based on no ground, and still includes the potentially rare events that are not recorded from the traffic. In principle, as long as the system under test resists the frequencies of the scenarios with critical TTC that our model generates, it is safe in practice under the same traffic density, as it provides a worst-case prediction.

7 Discussion of Results and Limitations

We have established a model that predicts a worst-case distribution of TTC for vehicle–pedestrian interactions at unsignalized crossings. The model is based on assumptions that careless drivers and pedestrians are involved, for example distracted drivers or pedestrians that are not properly assessing the ongoing traffic. A typical example of this worst-case scenario for autonomous driving is the fatal accident by Uber where the autonomous vehicle missed the pedestrian in detection and eventually hit the innocent person at an intersection [101]. In realistic traffic, drivers are obligated to obey the regulations to remain safe. A recent study has indicated that drivers' behavior follows the Thread Avoidance Model – drivers focus on avoiding the adverse events which also affects their decision to give the road to pedestrians or not [13]. Besides, pedestrians' crossing behavior is not always careless and is said to respect the Gap Acceptance Theory – pedestrians will check and determine if the gap between two vehicles is large enough to cross [51,115]. Thus, the real distribution of critical TTC is expected to be lower. Our model serves as a worst-case probability that can take place, based on the Poisson distribution of the vehicles and pedestrians' arrival.

The model is simple because it includes only the macroscopic property of the traffic – the density – and predicts merely the cumulative distribution of TTC at unsignalized crossings. Risk factors for road safety can originate from many other perspectives, such as the behavioral characteristics of the drivers and pedestrians, vehicle conditions, and geometric uniqueness, such as the number of lanes or road shoulder width, etc. [70]. Thus, the model can be extended in the future to be

more sophisticated by including additional features of the traffic dynamics. Nevertheless, the model also benefits from its simplicity. It can be utilized without too much overhead and does not require extensive traffic information. Given that road safety and testing the autonomous driving systems under realistic traffic setup is a major concern, the investigation and exploration on modelling the traffic further to provide realistic distribution of testing scenarios is significant and has great potential. Our model can be used for further modelling and testing purposes.

A thorough validation of the model requires more real traffic data sets with different traffic densities to evaluate the model's accuracy and to study the differences of the prediction since the traffic is different by time and location. Even though the validation shows a positive result by comparing the prediction by our model against the real traffic data, the actual traffic flow and vehicle–pedestrian interactions in a particular hour could lead to the actual TTC distribution exceeding the prediction from the model. Our model provides a feasible approximation of the worst-case distribution of TTC, especially for the critical TTC. Consequently, the model also provides the worst-case distribution of critical scenarios, with low TTC for vehicle–pedestrian interactions, for testing the autonomous driving systems under specific test scenes.

Our model, constructed in the context of testing autonomous driving systems, aims to predict the distribution of TTC for critical scenario identification. Existing models reported in previous studies require the collection of real traffic data, where the collection is usually performed only in specific periods and locations [34, 54]. In addition, curation and annotation of the data can be quite cost- and time-consuming. In contrast, our model does not rely on traffic recordings and is easy to use. Similarly, other approaches for critical scenario identification for autonomous driving, for example, using deep learning approaches [82], are also depending on the collection of data and may still not represent all possible scenarios in a real distribution. Thus, rare-occurring scenarios, like crashing or near-crashing cases, may not be recognized correctly by the mentioned approach. More specifically, studies that generate test scenarios by mining through the collected sensor data [53, 73] and reported accident records [36] will only be able to produce scenarios within the existing data sets and represents only a subset of the possible scenarios in the real world. In addition, scenarios recorded in the regular road traffic, most of the time, are considered not critical [58]. Our approach is not subject to the limitations of the data collection by exploring the parameters of the operational environment and optimizing the selection of parameters for generating critical scenarios for testing the autonomous driving systems based on realistic distribution.

The critical scenario identification approach sheds light on the generation and selection of risky scenarios for testing and will substantially reduce the number of test scenarios and total test efforts. The approach must be grounded on realistic distributions of the parameters involved in constituting a scenario. The absence of the realistic distribution in the critical scenario identification approach would

result in a biased frequency of scenarios or scenarios that are not even feasible in the actual traffic. The significance of our model is to produce and provide such distribution under different traffic densities and facilitate the testing of autonomous driving systems. As a specific case, the model provides the necessary input to the testing of an auto-braking function as discussed in Section 6. In principle, it could also be used to test other autonomous driving systems that involve TTC distribution at the unsignalized crossings. As a future plan, we aim to deploy the model for generating critical scenarios for testing such autonomous driving systems in real vehicles.

8 Conclusion

In this study, we have established a model for predicting the cumulative TTC distribution of vehicle–pedestrian interactions at unsignalized crossings. The model is used as input for the critical test scenario identification for autonomous driving functions [100]. The model takes the traffic density as input and indicates a worst-case TTC distribution. We have validated the model using a real traffic data set collected in Sweden and have demonstrated using the model for critical test scenario identification for an auto-braking function.

The model is the first step in improving the efficiency of testing autonomous driving functions and can be extended in future work. Nonetheless, the value and novelty of such a model and deploying it to test autonomous driving systems is clear and significant.

9 Acknowledgement

This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP). Thanks to Viscando AB for providing real traffic data set for validation, and Volvo Cars for review of earlier versions of the manuscript.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] City safety. https://en.wikipedia.org/wiki/City_safety. Accessed: 2021-06-10.

[2] Poisson distribution. https://en.wikipedia.org/wiki/Poisson_distribution. Accessed: 2021-06-10.

[3] Complementary materials to concepts in testing of autonomous systems: Academic literature and industry practice. https://serg.cs.lth.se/experiment-packages/testing-of-autonomous-systems/, 2021.

[4] Sherif M Abuelenin and Adel Y Abul-Magd. Empirical study of traffic velocity distribution and its effect on VANETs connectivity. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 391–395. IEEE, 2014.

[5] Venkatesh Agaram, Frank Barickman, Felix Fahrenkrog, Edward Griffor, Ibro Muharemovic, Huei Peng, Jeremy Salinger, Steven Shladover, and William Shogren. Validation and verification of automated road vehicles. In *Road Vehicle Automation 3*, pages 201–210. Springer, 2016.

[6] Matthias Althoff and Sebastian Lutz. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1326–1333. IEEE, 2018.

[7] Adina Aniculaesei, Daniel Arnsberger, Falk Howar, and Andreas Rausch. Towards the verification of safety-critical autonomous systems in dynamic environments. *Electronic Proceedings in Theoretical Computer Science*, 232:79–90, 12 2016.

[8] Adina Aniculaesei, Jörg Grieser, Andreas Rausch, Karina Rehfeldt, and Tim Warnecke. Toward a holistic software systems engineering approach for dependable autonomous systems. In *1st Int. Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, pages 23–30. IEEE, 2018.

[9] Johannes Bach, Jacob Langner, Stefan Otten, Marc Holzäpfel, and Eric Sax. Data-driven development, a complementing approach for automotive systems engineering. In *IEEE International Systems Engineering Symposium (ISSE)*, pages 1–6. IEEE, 2017.

[10] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based scene creation for the development of automated vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1813–1820. IEEE, 2018.

[11] Felix Batsch, Alireza Daneshkhah, Vasile Palade, and Madeline Cheah. Scenario optimisation and sensitivity analysis for safe automated driving using gaussian processes. *Applied Sciences*, 11(2):775, 2021.

[12] Halil Beglerovic, Michael Stolz, and Martin Horn. Testing of autonomous vehicles using surrogate models and stochastic optimization. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[13] Francesco Bella and Manuel Silvestri. Effects of safety measures on driver's speed behavior at pedestrian crossings. *Accident Analysis & Prevention*, 83:111–124, 2015.

[14] Hanna Bellem, Thorben Schönenberg, Josef F Krems, and Michael Schrauf. Objective metrics of comfort: developing a driving style for highly automated vehicles. *Transportation research part F: traffic psychology and behaviour*, 41:45–54, 2016.

[15] Anand Bhat, Shunsuke Aoki, and Ragunathan Rajkumar. Tools and methodologies for autonomous driving systems. *Proceedings of the IEEE*, 106(9):1700–1716, 2018.

[16] Prashant Shridhar Bokare and Akhilesh Kumar Maurya. Acceleration-deceleration behaviour of various vehicle types. *Transportation research procedia*, 25:4733–4749, 2017.

[17] Markus Borg, Cristofer Englund, Krzysztof Wnuk, Boris Duran, Christoffer Levandowski, Shenjian Gao, Yanwen Tan, Henrik Kaijser, Henrik Lönn, and Jonas Törnqvist. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. *Journal of Automotive Software Engineering*, 1:1–19, 2019.

[18] Najia Bouha, Gildas Morvan, Hassane Abouaissa, and Yoann Kubera. A first step towards dynamic hybrid traffic modeling. *arXiv preprint arXiv:1505.07257*, 2015.

[19] Oliver Buehler and Joachim Wegener. Evolutionary functional testing of an automated parking system. In *Proceedings of the International Conference on Computer, Communication and Control Technologies (CCCT'03) and the 9th. International Conference on Information Systems Analysis and Synthesis (ISAS'03), Florida, USA*, 2003.

[20] Fanta Camara, Nicola Bellotto, Serhan Cosar, Florian Weber, Dimitris Nathanael, Matthias Althoff, Jingyuan Wu, Johannes Ruenz, André Dietrich, Gustav Markkula, et al. Pedestrian models for autonomous driving part ii: high-level models of human behavior. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[21] Baiming Chen, Ding Zhao, and Huei Peng. Evaluation of automated vehicles encountering pedestrians at unsignalized crossings. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1679–1685. IEEE, 2017.

[22] Peng Chen, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang. Surrogate safety analysis of pedestrian–vehicle conflict at intersections using unmanned aerial vehicle videos. *Journal of advanced transportation*, 2017, 2017.

[23] Yu Chen, Shitao Chen, Tangyike Zhang, Songyi Zhang, and Nanning Zheng. Autonomous vehicle testing and validation platform: Integrated simulation system with hardware in the loop. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 949–956, 2018.

[24] Erik Coelingh, Andreas Eidehall, and Mattias Bengtsson. Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 155–160. IEEE, 2010.

[25] Erik Coelingh, Lotta Jakobsson, Henrik Lind, and Magdalena Lindman. Collision warning with auto brake: a real-life safety perspective. *Innovations for Safety: Opportunities and Challenges*, 2007.

[26] On-Road Automated Driving (ORAD) committee. J3016 taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *Surface Vehicle Recommended Practice*, 2021.

[27] Daniela S Cruzes and Tore Dybå. Recommended steps for thematic synthesis in software engineering. In *International symposium on empirical software engineering and measurement*, pages 275–284. IEEE, 2011.

[28] Maya Daneva. Focus group: Cost-effective and methodologically sound ways to get practitioners involved in your empirical RE research. In *REFSQ Workshops*, pages 211–216, 2015.

[29] Brigitte Cambon de Lavalette, Charles Tijus, Sébastien Poitrenaud, Christine Leproux, Jacques Bergeron, and Jean-Paul Thouez. Pedestrian crossing decision-making: A situational and behavioral approach. *Safety science*, 47(9):1248–1253, 2009.

[30] Wenhao Ding, Minjun Xu, and Ding Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. *arXiv preprint arXiv:2003.01197*, 2020.

[31] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[32] Ahmetcan Erdogan, Emre Kaplan, Andrea Leitner, and Markus Nager. Parametrized end-to-end scenario generation architecture for autonomous vehicles. In *6th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–6. IEEE, 2018.

[33] Shuo Feng, Yiheng Feng, Chunhui Yu, Yi Zhang, and Henry X Liu. Testing scenario library generation for connected and automated vehicles, part i: Methodology. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1573–1582, 2020.

[34] Ting Fu, Luis Miranda-Moreno, and Nicolas Saunier. Pedestrian crosswalk safety at nonsignalized crossings during nighttime: use of thermal video data and surrogate safety measures. *Transportation research record*, 2586(1):90–99, 2016.

[35] Alessio Gambi, Tri Huynh, and Gordon Fraser. Generating effective test cases for self-driving cars from police reports. In *Proceedings of the 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 257–267, 2019.

[36] Alessio Gambi, Marc Mueller, and Gordon Fraser. Automatically testing self-driving cars with search-based procedural content generation. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 318–328, 2019.

[37] Alessio Gambi, Marc Müller, and Gordon Fraser. Asfault: Testing self-driving car software using search-based procedural content generation. In *41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 27–30. IEEE, 2019.

[38] Ahmad Nauman Ghazi, Kai Petersen, Sri Sai Vijay Raj Reddy, and Harini Nekkanti. Survey research in software engineering: Problems and mitigation strategies. *IEEE Access*, 7:24703–24718, 2018.

[39] Nicolas Guéguen, Sébastien Meineri, and Chloé Eyssartier. A pedestrian's stare and drivers' stopping behavior: A field experiment at the pedestrian crossing. *Safety science*, 75:87–89, 2015.

[40] Magnus Gyllenhammar, Rolf Johansson, Fredrik Warg, DeJiu Chen, Hans-Martin Heyn, Martin Sanfridson, Jan Söderberg, Anders Thorsen, and Stig Ursing. Towards an operational design domain that supports the safety argumentation of an automated driving system. In *10th European Congress on Embedded Real Time Systems (ERTS)*, 2020.

[41] Sven Hallerbach, Yiqun Xia, Ulrich Eberle, and Frank Koester. Simulation-based identification of critical scenarios for cooperative and automated vehicles. *SAE International Journal of Connected and Automated Vehicles*, 1(2018-01-1066):93–106, 2018.

[42] Umar Zakir Abdul Hamid, Fakhrul Razi Ahmad Zakuan, Khairul Akmal Zulkepli, Muhammad Zulfaqar Azmi, Hairi Zamzuri, Mohd Azizi Abdul Rahman, and Muhammad Aizzat Zakaria. Autonomous emergency braking system with potential field risk assessment for frontal collision mitigation. In *2017 IEEE Conference on Systems, Process and Control (ICSPC)*, pages 71–76. IEEE, 2017.

[43] David Harel, Assaf Marron, and Joseph Sifakis. Autonomics: In search of a foundation for next-generation autonomous systems. *Proc. of the National Academy of Sciences*, 117(30):17491–17498, 2020.

[44] Mark Harman, S. Afshin Mansouri, and Yuanyuan Zhang. Search-based software engineering: Trends, techniques and applications. *ACM Comput. Surv.*, 45(1), December 2012.

[45] Philipp Helle, Wladimir Schamai, and Carsten Strobel. Testing of autonomous systems–challenges and current state-of-the-art. In *INCOSE International Symposium*, volume 26, pages 571–584. Wiley Online Library, 2016.

[46] Qiong Hu, Miao Cai, Nasrin Mohabbati-Kalejahi, Amir Mehdizadeh, Mohammad Ali Alamdar Yazdi, Alexander Vinel, Steven E Rigdon, Karen C Davis, and Fadel M Megahed. A review of data analytic applications in road traffic safety. part 2: prescriptive modeling. *Sensors*, 20(4):1096, 2020.

[47] WuLing Huang, Kunfeng Wang, Yisheng Lv, and FengHua Zhu. Autonomous vehicles testing methods review. In *19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 163–168. IEEE, 2016.

[48] Casidhe Hutchison, Milda Zizyte, Patrick E Lanigan, David Guttendorf, Michael Wagner, Claire Le Goues, and Philip Koopman. Robustness testing of autonomy software. In *40th Int. Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, pages 276–285. IEEE, 2018.

[49] International Organization for Standardization, Geneva, Switzerland. *ISO 16787. Intelligent transport systems – Assisted parking system (APS) – Performance requirements and test procedures*, December 2017.

[50] Muhammad Iqbal, Jia Cheng Han, Zhi Quan Zhou, and Dave Towey. Enhancing euro NCAP standards with metamorphic testing for verification of advanced driver-assistance systems. In *IEEE/ACM 6th International Workshop on Metamorphic Testing (MET)*, pages 37–41. IEEE, 2021.

[51] Suresh Kumaar Jayaraman, Lionel P Robert, Xi Jessie Yang, Anuj K Pradhan, and Dawn M Tilbury. Efficient behavior-aware control of automated vehicles at crosswalks using minimal information pedestrian prediction model. In *2020 American Control Conference (ACC)*, pages 4362–4368. IEEE, 2020.

[52] Ian Rhys Jenkins, Ludvig Oliver Gee, Alessia Knauss, Hang Yin, and Jan Schroeder. Accident scenario generation with recurrent neural networks. In *21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3340–3345. IEEE, 2018.

[53] Ian Rhys Jenkins, Ludvig Oliver Gee, Alessia Knauss, Hang Yin, and Jan Schroeder. Accident scenario generation with recurrent neural networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3340–3345, 2018.

[54] Xiaobei Jiang, Wuhong Wang, and Klaus Bengler. Intercultural analyses of time-to-collision in vehicle–pedestrian conflict on an urban midblock crosswalk. *Ieee transactions on intelligent transportation systems*, 16(2):1048–1053, 2014.

[55] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[56] Yue Kang, Hang Yin, and Christian Berger. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. *IEEE Transactions on Intelligent Vehicles*, 4(2):171–185, 2019.

[57] Dhanoop Karunakaran, Stewart Worrall, and Eduardo Nebot. Efficient statistical validation with edge cases to evaluate highly automated vehicles. In *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[58] Moritz Klischat and Matthias Althoff. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 2352–2358. IEEE, 2019.

[59] Lars Klitzke, Carsten Koch, Andreas Haja, and Frank Köster. Real-world test drive vehicle data management system for validation of automated driving systems. In *VEHITS*, pages 171–180, 2019.

[60] Florian Klück, Martin Zimmermann, Franz Wotawa, and Mihai Nica. Genetic algorithm-based test parameter optimization for ADAS system testing. In *IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*, pages 418–425. IEEE, 2019.

[61] Alessia Knauss, Jan Schröder, Christian Berger, and Henrik Eriksson. Paving the roadway for safety of automated vehicles: An empirical study on testing challenges. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1873–1880. IEEE, 2017.

[62] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[63] Aliaksei Laureshyn, Carl Johnsson, Tim De Ceunynck, Åse Svensson, Maartje de Goede, Nicolas Saunier, Paweł Włodarek, Richard van der Horst, and Stijn Daniels. Review of current study methods for VRU safety. appendix 6–scoping review: surrogate measures of safety in site-based road traffic observations: Deliverable 2.1–part 4. 2016.

[64] Sam Levin. Tesla fatal crash: 'autopilot' mode sped up car before driver killed, report finds. https://www.theguardian.com/technology/2018/jun/07/tesla-fatal-crash-silicon-valley-autopilot-mode-report, 2018.

[65] Mikael Lindvall, Adam Porter, Gudjon Magnusson, and Christoph Schulze. Metamorphic model-based testing of autonomous systems. In *2nd International Workshop on Metamorphic Testing (MET)*, pages 35–41. IEEE, 2017.

[66] Anastasia Loukaitou-Sideris, Robin Liggett, and Hyun-Gun Sung. Death on the crosswalk: A study of pedestrian–automobile collisions in Los Angeles. *Journal of Planning Education and Research*, 26(3):338–351, 2007.

[67] S.M. Sohel Mahmud, Luis Ferreira, Md. Shamsul Hoque, and Ahmad Tavassoli. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS research*, 41(4):153–163, 2017.

[68] Malte Mauritz, Falk Howar, and Andreas Rausch. Assuring the safety of advanced driver assistance systems through a combination of simulation and runtime monitoring. In *International Symposium on Leveraging Applications of Formal Methods*, pages 672–687. Springer, 2016.

[69] Jens Mazzega, Frank Köster, Karsten Lemmer, and Thomas Form. Testing of highly automated driving functions. *ATZ worldwide*, 118(10):44–48, 2016.

[70] Amir Mehdizadeh, Miao Cai, Qiong Hu, Mohammad Ali Alamdar Yazdi, Nasrin Mohabbati-Kalejahi, Alexander Vinel, Steven E Rigdon, Karen C Davis, and Fadel M Megahed. A review of data analytic applications in road traffic safety. part 1: Descriptive and predictive modeling. *Sensors*, 20(4):1107, 2020.

[71] Haoran Meng, Junyi Chen, Xingyu Xing, Tianyang Liu, Zhuoping Yu, and Renjing Zuo. Quantitative evaluation system of automated valet parking. *Tongji Daxue Xuebao/Journal of Tongji University*, 46:116–121 and 195, 2018.

[72] Till Menzel, Gerrit Bagschik, and Markus Maurer. Scenarios for development, test and validation of automated vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1821–1827. IEEE, 2018.

[73] Noah Metzger, Lars Hoffmann, Christian Bartelt, Heiner Stuckenschmidt, Michael Wommer, and Maria Belen Bescos del Castillo. Towards tracegraphs for data-driven test case mining in the domain of automated driving. In *The Third IEEE International Conference On Artificial Intelligence Testing – short papers*. IEEE Computer Society, 2021.

[74] Hafida Mouhagir, Reine Talj, Véronique Cherfaoui, François Aioun, and Franck Guillemard. Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation. In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1114–1119. IEEE, 2016.

[75] Galen E Mullins, Paul G Stankiewicz, and Satyandra K Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In *IEEE international conference on robotics and automation (ICRA)*, pages 1443–1450. IEEE, 2017.

[76] Byeongjoon Noh, Wonjun No, Jaehong Lee, and David Lee. Vision-based potential pedestrian risk analysis on unsignalized crosswalk using data mining techniques. *Applied Sciences*, 10(3):1057, 2020.

[77] Dhasarathy Parthasarathy and Anton Johansson. Silgan: Generating driving maneuvers for scenario-based software-in-the-loop testing. *arXiv preprint arXiv:2107.07364*, 2021.

[78] Michael Paulweber. Validation of highly automated safe and secure systems. In *Automated Driving*, pages 437–450. Springer, 2017.

[79] Raphael Pfeffer and Tobias Leichsenring. Continuous development of highly automated driving functions with vehicle-in-the-loop using the example of Euro NCAP scenarios. In *Simulation and Testing for Vehicle Technology*, pages 33–42. Springer, 2016.

[80] Thomas Ponn, Matthias Breitfuß, Xiao Yu, and Frank Diermeyer. Identification of challenging highway-scenarios for the safety validation of automated vehicles based on real driving data. In *15th International Conference on Ecological Vehicles and Renewable Energies (EVER)*, pages 1–10. IEEE, 2020.

[81] Thomas Ponn, Christian Gnandt, and Frank Diermeyer. An optimization-based method to identify relevant scenarios for type approval of automated vehicles. In *Proceedings of the ESV—International Technical Conference on the Enhanced Safety of Vehicles, Eindhoven, The Netherlands*, pages 10–13, 2019.

[82] Ivan Porres, Sepinoud Azimi, and Johan Lilius. Scenario-based testing of a ship collision avoidance system. In *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 545–552. IEEE, 2020.

[83] Maria Priisalu, Aleksis Pirinen, Ciprian Paduraru, and Cristian Sminchisescu. Generating scenarios with diverse pedestrian behaviors for autonomous vehicle testing. In *5th Annual Conference on Robot Learning*, 2021.

[84] Nijat Rajabli, Francesco Flammini, Roberto Nardone, and Valeria Vittorini. Software verification and validation of safe autonomous cars: A systematic literature review. *IEEE Access*, pages 4797–4819, 2020.

[85] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477, 2020.

[86] Francisca Rosique, Pedro J Navarro, Carlos Fernández, and Antonio Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3):648, 2019.

[87] Jennifer Rowley. Conducting research interviews. *Management research review*, 35(3/4):260–271, 2012.

[88] Per Runeson, Emelie Engström, and Margaret-Anne Storey. The design science paradigm as a frame for empirical software engineering. In *Contemporary empirical methods in software engineering*, pages 127–147. Springer, 2020.

[89] Per Runeson, Emelie Engström, and Margaret-Anne Storey. *The Design Science Paradigm as a Frame for Empirical Software Engineering*, pages 127–147. Springer, Cham, 2020.

[90] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131, 2009.

[91] Mahmoud Saffarzadeh, Navid Nadimi, Saber Naseralavi, and Amir Reza Mamdoohi. A general formulation for time-to-collision safety indicator. In *Proceedings of the Institution of Civil Engineers-Transport*, volume 166, pages 294–304. Thomas Telford Ltd, 2013.

[92] Friederike Schneemann and Irene Gohl. Analyzing driver-pedestrian interaction at crosswalks: A contribution to autonomous driving in urban environments. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 38–43, 2016.

[93] Chris Schwarz. On computing time-to-collision for automation scenarios. *Transportation research part F: traffic psychology and behaviour*, 27:283–294, 2014.

[94] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.

[95] Joseph Sifakis. Autonomous systems–an architectural characterization. In *Models, Languages, and Tools for Concurrent and Distributed Programming*, pages 388–410. Springer, 2019.

[96] Joseph Sifakis. Can we trust autonomous systems? Boundaries and risks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 65–78. Springer, 2019.

[97] Hannah Snyder. Literature review as a research methodology: An overview and guidelines. *J. of Business Research*, 104:333–339, 2019.

[98] Selim Solmaz and Franz Holzinger. A novel testbench for development, calibration and functional testing of ADAS/AD functions. In *IEEE Int. Conf. on Connected Vehicles and Expo (ICCVE)*, pages 1–8. IEEE, 2019.

[99] Qunying Song, Emelie Engström, and Per Runeson. Concepts in testing of autonomous systems: Academic literature and industry practice. In *IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, pages 74–81, 2021.

[100] Qunying Song, Kaige Tan, Per Runeson, and Stefan Persson. An industrial workbench for test scenario identification in autonomous driving software. In *IEEE International Conference on Artificial Intelligence Testing (AITest)*, pages 81–82. IEEE Computer Society, 2021.

[101] Jack Stilgoe. Who killed Elaine Herzberg? In *Who's Driving Innovation?*, pages 1–6. Springer, 2020.

[102] Kaige Tan. Building verification database and extracting critical scenarios for self-driving car testing on virtual platform. Master's thesis, KTH, School of Industrial Engineering and Management (ITM), 2019.

[103] Jianbo Tao, Yihao Li, Franz Wotawa, Hermann Felbinger, and Mihai Nica. On the industrial application of combinatorial testing for autonomous driving functions. In *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 234–240. IEEE, 2019.

[104] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proc. of the 40th international conference on software engineering*, pages 303–314, 2018.

[105] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. Defining and substantiating the terms scene, situation, and scenario for automated driving. In *IEEE 18th International Conference on Intelligent Transportation Systems*, pages 982–988. IEEE, 2015.

[106] Simon Weckert. Google maps hacks. Performance & installation. http://www.simonweckert.com/googlemapshacks.html, 2020.

[107] Lukas Westhofen, Christian Neurohr, Tjark Koopmann, Martin Butz, Barbara Schütt, Fabian Utesch, Birte Kramer, Christian Gutenkunst, and Eckard Böde. Criticality metrics for automated driving: A review and suitability analysis of the state of the art. *arXiv preprint arXiv:2108.02403*, 2021.

[108] Franz Wotawa. Testing autonomous and highly configurable systems: Challenges and feasible solutions. In *Automated Driving*, pages 519–532. Springer, 2017.

[109] Zhang Xinxin, Li Fei, and Wu Xiangbin. CSG: Critical scenario generation from real traffic accidents. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1330–1336. IEEE.

[110] Bo Yang, Xuelin Cao, Xiangfang Li, Chau Yuen, and Lijun Qian. Lessons learned from accident of autonomous vehicle testing: An edge learning-aided offloading framework. *IEEE Wireless Communications Letters*, 9(8):1182–1186, 2020.

[111] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 2020.

[112] Jin Zhang and Jingyue Li. Testing and verification of neural-network-based safety-critical control software: A systematic literature review. *Information and Software Technology*, page 106296, 2020.

[113] Xi Zhang, Hao Chen, Wenyan Yang, Wenqiang Jin, and Wangwang Zhu. Pedestrian path prediction for autonomous driving at un-signalized crosswalk using w/cdm and msfm. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[114] Xinhai Zhang, Jianbo Tao, Kaige Tan, Martin Törngren, José Manuel Gaspar Sánchez, Muhammad Rusyadi Ramli, Xin Tao, Magnus Gyllenhammar, Franz Wotawa, Naveen Mohan, et al. Finding critical scenarios for automated driving systems: A systematic literature review. *arXiv preprint arXiv:2110.08664*, 2021.

[115] Xiangling Zhuang and Changxu Wu. Pedestrians' crossing behaviors and safety at unmarked roadway in China. *Accident analysis & prevention*, 43(6):1927–1936, 2011.