How Can Simulation-based Safety Testing Help Understand the Real-World Safety of Autonomous Driving Systems?

Fauzia Khan
Dept. of Computer Science
University of Tartu,
Tartu, Estonia
fauzia.khan@ut.ee

Laima Dalbina
Dept. of Computer Science
University of Tartu
Tartu, Estonia
laima.anna.dalbina@ut.ee

Hina Anwar
Dept. of Computer Science
University of Tartu
Tartu, Estonia
hina.anwar@ut.ee

Dietmar Pfahl
Dept. of Computer Science
University of Tartu
Tartu, Estonia
dietmar.pfahl@ut.ee

Abstract—**An Automated Driving System (ADS) requires exhaustive safety testing before receiving a road permit. Moreover, it is not clear what exactly constitutes sufficient safety for an ADS. One would assume that an ADS is safe enough if it is at least as safe as a Human Driven Vehicle (HDV). However, evaluating the safety of an ADS by comparing its behavior with that of a typical HDV in the real world is costly and risky. In this paper, we give an overview of our approach to compare the performance of ADS with HDV. While the overall approach is still in progress and ongoing, we provide a detailed approach utilizing established guidelines to systematically generate test scenarios specifically aimed at safety testing. Using our approach, various scenarios could be generated and tested, contributing to autonomous vehicles' trustworthiness.**

Keywords - Autonomous Driving System (ADS); Human Driven Vehicle (HDV); Safety Testing; Scenario Generation; CARLA Simulator.

I. INTRODUCTION

The automotive industry and research community are working hard to deploy autonomous cars on roads in the future. However, it requires exhaustive safety testing before it is safe at an acceptable level. Simulation-based testing is a cost-effective way to evaluate ADS safety. We aim to compare the behavior of an ADS to that of a Human Driven Vehicle (HDV) via simulation. However, the conclusions drawn from simulation-based testing are not always clear. There are some open challenges and questions, which are as follows:

Challenge 1 - How can we transfer the real world into a simulation model, and what aspects should be considered? To transfer the real world into a simulator, many challenges are posed. For example, how to model the behavior of an ADS and a human driver, how to set up the environment, and on top of that, which scenarios must be tested? Vehicles encounter a wide range of scenarios based on the combination of scenery, traffic and road objects, environment, road geometry, and maneuvers [1]. Additionally, the complexity of driving tasks and the uncertainty of the driving environment grows exponentially, translating into infinite scenarios that ADS could encounter. It is practically impossible to test every scenario using a simulator. Thus, it leads to a few interesting sub-questions: **Q1)** How are scenarios derived systematically? **Q2)** How to select the critical scenarios for safety testing? **Q3)** How to generate exemplary test scenarios showing that ADS behaves better than HDV. There are also challenges to modeling an HDV in the simulator - as part of the context in which the ADS moves. The model of an HDV depends on the behavior of a human driver. A human driver is diverse in behavior as different drivers make different decisions under the same surroundings and driving situation, mainly due to aggression, attention, and experience. Furthermore, a human driver also makes mistakes, violates rules, and takes evasive maneuvers to prevent accidents that could cause by the mistakes of other drivers. All these aspects create behavior complexity in a simulation environment and make comparing an ADS with an average HDV difficult.

Challenge 2 - How can we build trust in simulations? That is, how to guarantee that simulation results correctly represent the real-world behavior of an ADS. This could be achieved by running scenarios in the simulator where the real-world reference behavior is known. We could consider scenarios where we know the human driver has more information and is better than ADS. If the ADS in the simulator reproduces the reference behavior, it supports the assumption that the ADS is modeled correctly in the simulator. Typical reference behavior could be the braking of the ADS on a straight road with dry streets and good visibility in daylight. Based on the physical characteristics of the car and its speed, one can calculate at what distance the car should be able to stop without hitting the obstacle.

Challenge 3 - How can we quantify the advantage and disadvantages of an ADS compared to an HDV with an average driver? In certain aspects, ADS performs better than an HDV. ADS can potentially reduce the mistakes that human drivers make while driving. For example, ADS are never distracted (e.g., using cell phones, drunk, or tired); it has better perception (e.g., no blind spot), faster response time, and more precise brakes, acceleration, and steering control. However, ADS might not perform well when it comes to certain situations. For example, corner cases (rare situations or not expected to happen in the real-world). Also, ADS lacks intuition and instinct compared to human drivers, which affects the response of ADS

in a particular situation. For example, a pedestrian is standing near the edge of a sidewalk, seemingly distracted and showing signs of potential jaywalking. A human driver might intuitively recognize the pedestrian's behavior and anticipate the possibility of them stepping into the road unexpectedly. However, an ADS may struggle to accurately interpret the pedestrian's behavior and intentions.

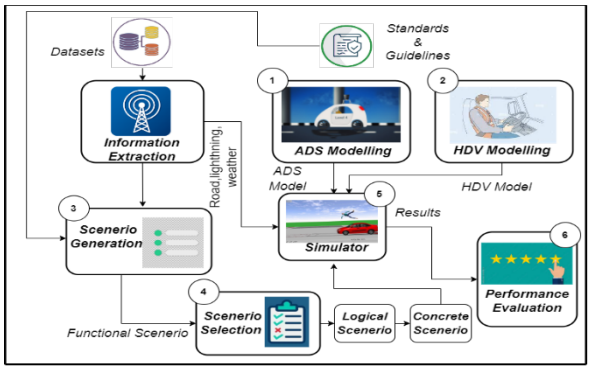

Figure 1. The proposed approach for comparing the behavior of ADS and HDV via Simulation-based Safety Testing.

In principle, we are still trying to resolve some of the above-mentioned challenges for translating the real world into the simulator. Figure 1 gives an overview of the proposed approach. As it is a work in progress, in this paper, we mainly explain steps in our approach for generating the test scenario systematically using public datasets by following the guidelines provided by the Center for Connect and Autonomous Vehicles [2] and existing literature [3], [4]. The work presented in this paper is related to the above-mentioned challenge # 1. More specifically, we tried to answer the sub-question Q1 by showing how a wide range of driving conditions can be effectively covered by systematically generating test scenarios, facilitating a better evaluation of ADS performance in diverse scenarios. It could also help identify potential risks and vulnerabilities of the ADS system, contributing to autonomous vehicles' overall development and trustworthiness.

II. Background

This section explains the key term "scenario" and abstraction levels of scenario representation.

Scenario: A specific situation or context that captures the essential elements of a particular driving experience or event. It is a quantitative description of the *ego vehicle*, its *activities*, *static environment*, and *dynamic environment* [5], [6].

Functional scenario: is the highest abstraction level that depicts possible situations on a real road as a brief text [7]. For example, in a highway merge scenario, the ego car merges from an on-ramp into highway traffic.

Logical Scenario: Functional scenarios are converted to logical scenarios by adding variables and parameter ranges [7]. To extend the previous example, the highway merge scenario incorporates different merge lane geometries, a range of surrounding vehicle speeds, varying traffic speed ranges, merging gaps, and timing. The combinations of all these attributes result in many logical scenarios.

Concrete Scenario: A concrete scenario refers to a specific instance of the logical scenario where precise values and conditions are defined for the various parameters of the scenario [7]. For example, specific road geometry, weather condition, speed, interaction time, etc.

III. Proposed Approach

Our proposed approach (see Figure 1) consists of six steps: (i) modeling an ADS in the simulator, (ii) modeling an HDV in the simulator, (iii) generating test scenarios, (iv) selecting test scenarios, (v) simulations, and (vi) performance evaluation. We give an overview of the first two steps in our approach. We explain step (iii), generating test scenarios, in more detail. Steps (iv) - (vi) are work in progress.

A. Modeling an ADS in Simulator

The choice of the simulator is important because it could significantly impact the reliability, accuracy, and validity of the results obtained from the performance evaluation of an ADS and HDV. The simulator must be capable of accurately replicating real-world conditions. Additionally, the ability to collect and analyze data on performance metrics such as travel time, fuel consumption, safety, and compliance with traffic rules is another important consideration when selecting a simulator. We opted for CARLA[1] because it possesses both capabilities [8]. To model an ADS in the simulator, the first step is to create a vehicle model by defining its physical properties, such as mass, dimensions, and color. The blueprint library *get blueprint_library* () function provided by the Carla Python API can be used to create the vehicle model. In the next step, sensors are integrated into the selected vehicle to enable perception of the surrounding environment. The desired parameters and sensors could be set in CARLA using *vehicle_blueprint.set _attribute* (). Using the above-mentioned steps, we model an ADS similar to a mini Cooper S with sensors such as a camera, Lidar, and radar. Additionally, the modeled ADS has all the built-in supporting systems that a regular HDV possesses.

B. Modeling an HDV in Simulator

To model an HDV in a simulator is challenging due to the diversity in human driver behavior, as individuals make different decisions when facing similar surroundings and driving scenarios. This is primarily attributed to variations in factors such as aggression, attention, and experience. We opt for CARLA's human-like driving behavior model. These models are based on real-world data distribution and include different driver behavior, e.g., aggressive, distracted, naturalistic, and low aggressive drivers. We select the human driver model by configuring the Non-Player Character (NPC) agent parameters in the CARLA configuration files. In the future, we plan to make different persona's according to each human-driver model, run a simulation, and compare the HDV behavior with an ADS.

[1] https://carla.org/

C. Generating Test Scenarios

To generate test scenarios, we perform the following steps: **1) Dataset Collection:** The first step is to collect a suitable dataset from public repositories based on the testing objectives. Choose a trustworthy source of high-quality datasets to ensure the dataset's reliability and quality. Formulate and run a search query using key terms to find relevant datasets. From the results, select a dataset that aligns with test objectives and contains the necessary information to generate functional scenarios. Keeping safety testing of ADS as objective, the necessary information a dataset should contain includes target object, provoking event, maneuver, etc.

The target object refers to the entity responsible for causing the accident, while the provoking event represents the specific action that triggered the accident. The accident directly affects the ego vehicle, and its corresponding driving situation is called the maneuver [9].

We formulated a search query based on our test objective to assess the advantages and disadvantages of ADS compared to HDV. We focused on key terms such as "road accident datasets" and "road accidents caused by human error" to find relevant datasets from the Kaggle repository. The chosen dataset[2] includes reports of traffic collisions in Addis Ababa, Ethiopia, spanning the period from 2017 to 2020.

2) Dataset Preprocessing: The next step is to preprocess and clean the dataset using the standard data preprocessing techniques to filter unnecessary, missing, and inconsistent data. In our case, the selected dataset is already preprocessed and clean. However, we refined the data by filtering the irrelevant attributes such as location, driver age, and experience. These attributes are unnecessary for generating functional scenarios. Additionally, we adjusted the column labels to align with our specific requirements using Python commands.

3) Feature Extraction and Categorization: Once data preprocessing is complete, elements for the functional scenario are extracted and categorized as the target object, provoking event, ego vehicle, and maneuver. Let us consider an example scenario where these elements are briefly explained to provide a clearer understanding.

Vehicle A abruptly changed the lane and hit Vehicle B, which was overtaking.

Here, Vehicle A is the target object, abruptly changing the lane is a provoking event, Vehicle B is the ego vehicle, and overtaking is a maneuver. We categorize the values of main elements (target object, provoking event, maneuver, road junction types) into subtypes based on their distinctiveness or uniqueness. We also merged similar categories to streamline the classification process. For instance, the categories of "Driving under the influence of drugs" and "Drunk driving" were considered similar, and thus, we merged them into a single category called "Drunk driver."

4) Generation of Functional Scenarios: In this step, the extracted main elements are combined to create a description known as a functional scenario. For example, we extract the Target object = animal, Maneuver = going straight, and Provoking Event = overspeeding. Combining these key elements, the functional scenario is "Vehicle A is going straight and overspeeding, hitting the animal." This step is repeated until all combinations of the defined category values are achieved.

The subsequent steps in the approach are in progress. We plan to develop a scenario selection method for safety testing to prioritize generated functional scenarios. The selected functional scenarios will be transformed into logical scenarios and refined into concrete scenarios by assigning specific values to the parameters. For example, information related to road junction type, weather conditions, lighting conditions, speed limits, and other relevant factors could be extracted from the dataset. Simulation results would be used i) to determine the reliability of the simulator and ii) to quantify ADS's advantages and disadvantages over the typical HDV.

IV. Preliminary Results

The results of steps (i) and (ii) of our approach will be published once the simulation is complete. We only show the preliminary result of step (iii) of our proposed approach for generating test scenarios from the chosen dataset (cf. Section III.C). Table I shows the extracted subtypes for each element of the functional scenario. We classify the target objects into seven categories, the provoking events into 18 subtypes, and the maneuvers into 11 subtypes. The total number of generated combinations without repetition is 1386. For each combination, it is possible to generate scenarios, However, not all scenarios are meaningful or relevant. We generated a list of functional scenarios using these combinations. The next step involves manually removing irrelevant or meaningless scenarios from the list. Due to space limitation, we only present a small initial set of generated functional scenarios in Table II. The last column in Table II shows the functional scenario after combining all the main elements captured in columns two to five.

TABLE I. Extraction of Functional Scenarios Elements and their Subtypes

Elements	Subtypes
Target object	Roadside-parked vehicles, Vehicles, Roadside objects, Animals, Rollovers, Pedestrians, and Train
Provoking event	Moving Backward, Overtaking, Changing lanes to the left, Changing lanes to the right, Overloading, No priority to a vehicle, No priority to pedestrian, No distancing, Getting off the vehicle improperly, Improper parking, Driving carelessly, Driving at high speed, Driving to the left, Over-speeding, Unknown, Overturning, Turnover, Drunk driving.
Maneuvers	Going straight, U-Turn, Moving Backward, Turnover, Waiting to go, Getting off, Reversing, Parked, Stopping, Overtaking, and Entering a junction.

[2] https://www.kaggle.com/datasets/saurabhshahane/road-traffic-accidents

V. FUTURE WORK

In the future, the generated test scenario will be prioritized and converted into logical and concrete scenarios. The concrete scenarios would be loaded in the simulation environment and executed. The results from the simulation would be used to quantify the performance of both ADS and HDV based on performance criteria. The possible metrics to evaluate the performance criteria could be the number of accidents, the severity of the accident, or parts of a vehicle damaged in simulated scenarios. We also plan to answer the many open questions and challenges (cf. section I) and run simulations that help identify the advantages or disadvantages of ADS behavior over that of HDVs in simulated safety-critical traffic situations and to translate these findings into corresponding real-world behavior.

TABLE II. AN INITIAL SET OF GENERATED TEST SCENARIOS AFTER APPLYING STEP (III) OF OUR PROPOSED APPROACH

Maneuvers	Target Object	Provoking Event	Functional Scenario
Driving Straight	Vehicle	Change lane to the left	The ego vehicle is driving straight. The target object (vehicle) is lane changing into ego's vehicle driving ahead.
Driving Straight	Vehicle	Overtaking	The ego vehicle is driving straight, while the target object (vehicle) is overtaking.
Reversing	Roadside objects	-	The ego vehicle is reversing and colliding with a roadside object.
U-Turn	Animal	-	The ego vehicle takes a U-turn and hits the animal.
Driving Straight	Pedestrian	Drive carelessly	The ego vehicle is driving straight carelessly while the target object (pedestrian) is crossing.

VI. RELATED WORK

Due to space limitations, we present only a few recent relevant studies related to simulation-based safety testing of ADS. Matthew et al. [10] presented a simulation framework based on an adaptive sampling method to test an entire ADS. Jha et al. [11] proposed a fault injection tool that systematically injects faults into the hardware and software of an ADS to evaluate safety and reliability. Ben et al. [12] presented an approach to test ADS in a simulation environment (Simulink). They used multi-objective search and surrogate models based on a neural network to identify critical test cases regarding an ADS behavior. Wicker et al. [13] performed black box testing to evaluate the robustness of neural networks against adversarial attacks in traffic sign recognition in self-driving cars. Our approach is different from existing work in simulation-based safety testing. We have a black-box point of view on the ADS when observing its behavior and focus on identifying ADS behavior that differs from that of an HDV, both positive and negative.

ACKNOWLEDGMENT

This work is supported by Estonian Research Council grant PRG1226, the Bolt Technology OU grant, and the Estonian state¨ stipend for doctoral studies.

REFERENCES

[1] F. Duarte and C. Ratti, "The impact of autonomous vehicles on cities: A review," *Journal of Urban Technology*, vol. 25, no. 4, 2018.

[2] "Uk standards: Natural language description for abstract scenarios for automated driving systems," *Specification-BSI Flex 1889v1.0:202207*,

[3] Y. Zhu, J. Wang, F. Meng, and T. Liu, "Review on functional testing scenario library generation for connected and automated vehicles," *Sensors*, vol. 22, no. 20, p. 7735, 2022.

[4] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation-a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[5] H. Elrofai, D. Worm, and O. Op den Camp, "Scenario identification for validation of automated driving functions," in *Advanced Microsystems for Automotive Applications 2016: Smart Systems for the Automobile of the Future*, Springer, 2016, pp. 153–163.

[6] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder, *et al.*, "Concept and development of a unified ontology for generating test and usecase catalogues for assisted and automated vehicle guidance," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 183–189, 2014.

[7] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1821–1827.

[8] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, IEEE, 2021.

[9] S. Park, S. Park, H. Jeong, I. Yun, and J. So, "Scenario-mining for level 4 automated vehicle safety assessment from real accident situations in urban areas using a natural language process," *Sensors*, vol. 21, no. 20, p. 6929, 2021.

[10] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," *Advances in neural information processing systems*, 2018.

[11] S. Jha, T. Tsai, S. Hari, M. Sullivan, Z. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "Kayotee: A fault injection-based system to assess the safety and reliability of autonomous vehicles to faults and errors," *arXiv preprint arXiv:1907.01024*, 2019.

[12] R. Ben Abdessalem, S. Nejati, L. C. Briand, and T. Stifter, "Testing advanced driver assistance systems using multi-objective search and neural networks," in *IEEE/ACM international conference on automated software engineering*, 2016.

[13] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black box safety testing of deep neural networks," in *24th International Conference, Held as Part of the European Joint Conferences on Theory and Practice of Software, Greece*, Springer, 2018.