A Framework for Safety Violation Identification and Assessment in Autonomous Driving

Lukas Heinzmann[1], **Sina Shafaei**[1], **Mohd Hafeez Osman**[1], **Christoph Segler**[2] and **Alois Knoll**[1]
[1] Department of Computer Science, Technical University of Munich, Germany
[2] BMW Group Research, New Technologies, Innovations, Germany
{lukas.heinzmann, sina.shafaei, hafeez.osman}@tum.de,
christoph.segler@bmwgroup.com, knoll@in.tum.de

Abstract

Safety in self-driving cars is essential and an interdisciplinary matter. Nevertheless, there exists a massive gap between system developers knowledge about safety concepts and the knowledge of safety engineers on autonomous driving. Thus, an approach to close this gap and integrating new ideas and concepts of the critical safety domain to self-driving cars is needed. This work presents a framework for mapping safety-critical situations based on safety measures in CARLA simulator. Through this framework, safety engineers can define basic safety measures such as respecting speed limits, keeping an appropriate distance to the vehicle ahead and keeping the suitable lane. Developers can quickly integrate their agent(s), and the framework generates a mapping of the safety-critical states by running an agent over several episodes in a simulated environment while maintaining the considerations of developers and safety engineers. In the simulation environment, our evaluations showed promising and intuitive results on identification of safety violations of two machine learning agents. Respectively, several safety-critical situations could be identified and analysed according to the outcome of the mappings.

1 Introduction

Context. The formal concept of safety is not easy to grasp from a development perspective. However, in a general overview, safety can be seen as a feeling based on the individual's own experience. Many metrics that are used for current self-driving car implementations are the accident-free driven kilometres, the count on necessary takeovers by the safety driver and the general well-being of the occupants [General Motors, 2018; Tesla, 2018]. However, from safety engineering perspective, there are fewer insights into the technical functionality of such a system. Besides the technical complexity and closed source problems, employing machine learning techniques in state-of-the-art approaches causes even bigger challenges. Machine learning-based approaches are seen as black boxes, with input and output streams, while the actual inner logic remains unknown even to most of the developers. This leads to new challenges regarding safety assessment of these systems.

Problem Statement. Establishing a safety framework for evaluating the developed applications of self-driving cars from safety perspective, is a challenging task due to various regulations of different countries, the complex and often unpredictable outcomes of the approaches and also lack of the proper standards. Machine learning-based approaches have several sources of uncertainty and Reinforcement Learning (RL) is the blackest black box in this area considering the fact that developer expert can only provide the "right" and "wrong" actions for the agent at initialisation phase. In this context, the argument that the agent always learns safe actions is challenging and can often not be generalised, because encoding the whole knowledge into a single numerical function is highly error-prone. A good example is a problem called reward hacking in which the RL algorithm collects much reward without reaching the actual goal by exploiting a bug in the reward function [Amodei *et al.*, 2016]. From the Automotive functional safety point of view [ISO 26262, 2011], the V-shaped development model is well accepted in product development. The V-shaped model carries a solid requirement that will be the main input of the product's safety validation. However, gathering a complete set of requirements for a machine learning-based application is a difficult task due to the uncertainty of these models. In autonomous driving the responsibility shifts from the human driver to the car itself in driving tasks, and behavioural safety is a fundamental part of a development. Here, an evaluation is more important to avoid incorrect behaviours that may lead to severe accidents.

Goal. This work aims to support the integration of safety concerns in development phase of the machine learning-based applications in autonomous cars. We provide a framework for an easy setup of safety measures and self-driving car agents with an exclusive focus on RL-based scenarios in CARLA simulator [Dosovitskiy *et al.*, 2017]. To validate our approach, we mainly focus on reinforcement learning and over several runs, and gather safety-related information about the agent. Those safety violations are mapped and visualised in the end and can be used by developers and safety engineers to analyse the performance of the agent regarding safety.

Outline. The remainder of this paper is structured as follows: Section 2 summarises related work followed by the primary approach of this work in Section 3. Section 4 presents the evaluation with the conducted experiments and gained results followed by their discussion in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

Coming up with a formal specification of safe behaviour is not an applicable task for humans, because humans learn the most rules and behaviour through practical exercise a.k.a. "learning-by-doing", instead of remembering a specification of safe behaviour. NHTSA [Thorn *et al.*, 2018] has developed a set of "Behaviour Competencies" in which they listed 28 competencies regarding correct behaviour on the roads. Some instances are *Perform Low-Speed Merge*, *Perform Car Following (Including Stop and Go)* or *Navigate Roundabouts*. Waymo extended this set by 18 additional competencies [Waymo, 2018]. For example, *Detect and Respond to Animals*, *Detect and Respond to Unanticipated Weather or Lighting Conditions Outside of Vehicle's Capability (e.g. rainstorm)* or *Make Appropriate Reversing Manoeuvrers* are among the newly added competencies. Those sets give an excellent overview of the competencies of an autonomous car but still lacks from concretely defining an appropriate or critical behaviour. Further, these competencies result in a wide range of specific scenarios with variations of parameters like speed, road or weather conditions. Considering those, the number of testable situations will be enormous. An autonomous car normally is evaluated for those scenarios in either the simulation environment, or on closed-courses and real roads. Besides Waymo, PEGASUS [PEGASUS, 2019] and AdaptIVe [AdaptIVe, 2019] are also among the projects that address the problem of testing autonomous cars with regard to safety, but there was no evaluation measure or rating for the safety of agents that go further as "x kilometres without collision" or "x takeovers of the safety driver". During the recent years, several benchmarks or evaluation challenges are proposed for ensuring the safety of autonomous cars. An outstanding example in this area and related to core idea of our work, is the *CoRL Driving Benchmark* of CARLA [Codevilla, 2018] simulator that is followed by the *CARLA Autonomous Driving Challenge* [CARLA, 2019] or *The Grand Challenge for Autonomous Vehicles* (real world closed track) of the DARPA [DARPA, 2019]. The CARLA challenge integrated several scenarios based on the NHTSA *behavioural competencies* into a typical driving task. Nevertheless, the main goal of these challenges is mostly focused on comparing the overall performance of autonomous cars, rather than safety concerns.

As it was discussed before, identifying safety-critical situations is a crucial matter because avoiding such situations would lead to a considerable improvement from safety point of view, however, this remains still a challenging task. Here in this work, we differentiate between the ideas of *statistical* and *runtime* approaches. Statistical approaches use existing data such as reported accidents and accordingly visualizing them, and while they are currently only relevant for safety from the perspective of planning and defusing dangerous road segments, still could play a major role for automated vehicles. Traffic accident maps like *Unfallatlas* (Germany) [Statistische Ämter des Bundes und der Länder, 2019] or *CrashMap* (Great Britain) [Agilysis, 2019] can be seen as the most famous uses cases of such approaches. These maps display the accidents based on their location and further information such as severity, affected means of transport and the date of the incident. *Unfallatlas* also represents the accident frequency for a given stretch of road. Runtime approaches evaluate the safety during driving since some situations or locations are "labelled" as more safer in comparison to others. *Time to Collision* (TTC) or *Time to Brake* (TTB) are also among the metrics that are employed by researchers to define the safety level of situations [Eggert, 2014; González *et al.*, 2018; Hallerbach *et al.*, 2018; Mario Morando *et al.*, 2018]. One example for a runtime approach is the *Responsibility-Sensitive Safety* (RSS) proposed by Mobileye [Shalev-Shwartz *et al.*, 2017]. This approach is based on *safe distances* to define *dangerous situation*, for which *proper responses* are defined. A similar approach is proposed by NVIDIA with the *Saftefy Force Field* (SFF) [NVIDIA, 2019] which predicts the environment and mitigates harmful scenarios. Other approaches observe the autonomous driving safety by reading sensors or buses and evaluate it based on predefined rules [Kane *et al.*, 2015].

3 The Framework

In this section we propose a framework for evaluating the safety of an agent and detecting safety-critical situations in a defined environment. This framework can be used to visualize and expose the safety risk of the unknown situations that may be observed by a reinforcement learning agent in a suitable set of iterations defined by the application developer.

3.1 Concept and Architecture

The proposed framework employs the concept of *Safety Measures* which are activities, precautions or behavioural codes to avoid unnecessary risks and are taken to maintain safety. Moreover, it enables safety measures based on predefined rules, proven practices, and accepted guidelines in a real-world simulation environment. The original concept of safety measures is not new and already well established in the domain of behavioural safety, with prime examples like traffic rules or rules for defensive driving. Being quantifiable is the most important advantage of the safety measures. For instance, it is possible to determine whether drivers are violating the speed limit or are tailgating. In our proposed framework, safety measures are based on integrating the expert knowledge on top of simulated situations that statistically may have higher risks for injuries. A severity level is assigned to each safety measure to quantify the negative impact on safety. The respective measures are seen as *Safety Constraints* in our development and by violating a constraint, a *Safety Violation* is triggered. The framework is separated into three stages: **Initiation**, **Execution** and **Analysis**. The architecture is represented in Figure 1.

The *Initiation* phase consists of two different sections, one for application developers ( ) and the other for safety engi-

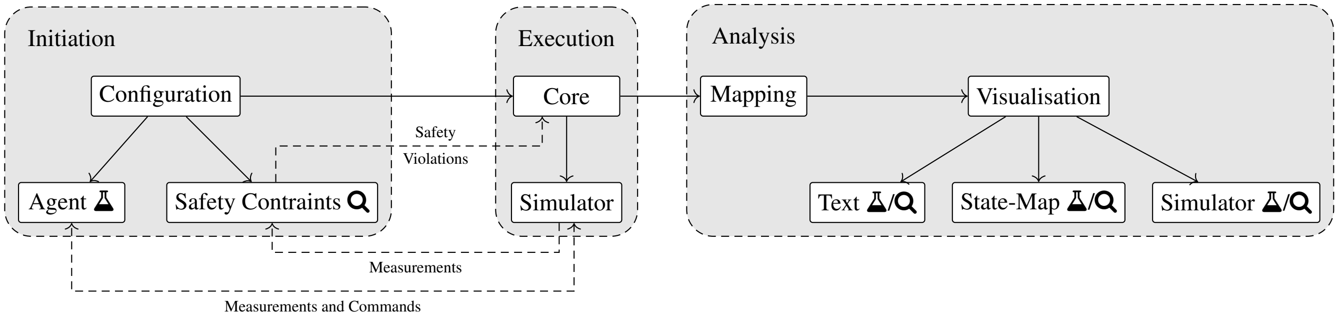


Figure 1: Architecture of the framework — Roles: Developer , Safety-Engineer

neers (). The *Agent* interface provides a platform for application developers to integrate the developed approach as an RL-based agent. *Safety Constraint* interface is also respectively a set of safety restrictions. In the *Execution* phase, the agent has to drive in the predefined environment and is evaluated by the given safety constraints. This phase is completed after a stop criterion is matched. The safety constraints are evaluated against the current situation and trigger a safety violation that contains relevant information about the current situation among other agents, type and location. In the end, the framework persists the given events. In the last stage *Analysis*, the safety violations are filtered, mapped and visualised. The location and the type are the primary parameter for the grouping but could vary in future implementations. The framework calculates different safety indicators for each group to make the groups comparable. The generated groups are visualised in a more intuitive way concerning the calculated safety indicators and gives the developers and safety engineers the possibility to better understand the system. To achieve this we use two types of safety measure that are implemented in the proposed framework.

Collisions Avoidance

A major safety measure is directly derived from the definition of safety. If the current situation causes injury at any object (e.g. humans, cars, other objects in the environment or even immaterial goods), safety is violated. In the context of cars, any injury is usually related to a collision. A collision occurs if a vehicle collides with another vehicle, pedestrians or other objects in the environment such as trees or animals. There are different types of collision such as a *single-vehicle collision*, where a vehicle collides with an object of the environment without the influence of another road user, or *longitudinal collision* if the vehicle collides with another vehicle that is driving in the same or the opposite direction. The severity of a collision depends on the collision type and parameters such as speed, crash worthiness or involved road users. Therefore, the highest safety goal is to prevent collisions of any kind and favour light damages on cars against heavy damages and casualties.

Since collisions are a violation to safety, therefore avoiding collisions is indispensable for detecting safety-critical situations. Intuitive examples for collision avoidance safety measures are appropriate distances to the vehicle ahead and the position in the assigned lane. It is essential from safety perspective to keep an appropriate distance to the leading vehicle since the time for reactions, and possible evasive manoeuvres are limited if the distance is too short. Computers are much faster in reaction, nevertheless these systems rely on measurements from the environment (e.g. radar sensors) which introduce latencies between measuring, detecting, and acting. For this case maintaining an appropriate distance, reduces the risk of a collision in most of the cases. Defining *appropriate* in this context is not as straightforward as it seems in the first place. Also, legislators do not specify this exactly for human drivers. Most countries specify formal or informal rules of thumb, popular is the *2-second-rule* or in countries with the metric system the *half speedometer*. The *2-second rule* enforces a cushion of at least the distance the car drives in two seconds (for 100km/h $\rightarrow$ 55,5m). In the framework as well, the distance constraint is variable based on an *x-second rule*; the safety engineer can specify the exact number of seconds.

The position orthogonal to the movement of the car is an important safety consideration. The primary focus is on staying in the correct lane. However, there are several cases, where it is necessary or is accepted to violate this rule. Examples are overtaking manoeuvre on a 2-lane road or the bypass over the side-walk if an accident or obstacle blocks the road. If the vehicle leaves the lane, either to the side-walk or to the other lane, this is declared as a *major* safety violation.

Safe Driving Behaviour

Traffic rules and guidelines for defensive driving are by far the biggest group of safety measures, and this is not a field that is only related to self-driving cars. The prevention of collisions is the primary goal for road users and countries for decades, and many rules are designed to reduce collisions and maintain the safety. Prominent examples are right of way regulations together with speed limits. Ignoring or misinterpreting right of way rules can cause *hazardous* or *catastrophic* accidents. Therefore, we enforce agents to remain in line and follow them.

3.2 Mapping

In the mapping phase, we group the violations by type, severity and location. Clustering by type and severity is trivial, but for the location, it is necessary to use a grid. The map is respectively divided into tiles of a predefined size. Each violation is added to a specific tile and grouped with the other violations of this tile. The degree of safety is measured in the quantity of safety violation in the situation. The quantity index ($score$) defines how relevant the tile is. Equation (3) represents the calculation of the score function. A low index ($score_s < 0$) indicates that the violation occurred only a few times compared to the average and is rather unspectacular. On the other hand, a high index ($score_s > 0$) indicates an interesting situation. A score of 0 indicates an average situation regarding safety violations, and yet does not imply any irrelevancy. The severities have weights to value more critical ones higher. Equation (1) depicts the weights for each severity.

$$m(violation) = \begin{cases} 1, \text{if severity is Negligible} & (S0) \\ 2, \text{if severity is Minor} & (S0) \\ 4, \text{if severity is Major} & (S1) \\ 8, \text{if severity is Hazardous} & (S2) \\ 16, \text{if severity is Catastrophic} & (S3) \end{cases} \tag{1}$$

Note: S0, S1, S2 and S3 indicate the severity class defined in ISO 26262 [ISO 26262, 2011]

$$x_s = \sum_{v \in V_s} m(v) \tag{2}$$

where V_s are all violations at location s

$$score_s = \begin{cases} \dfrac{\mu - x_s}{\sigma}, & \text{if } \sigma \neq 0 \\ 0, & \text{else} \end{cases} \tag{3}$$

where μ is the mean and σ the standard deviation of all x_s

3.3 Visualisation

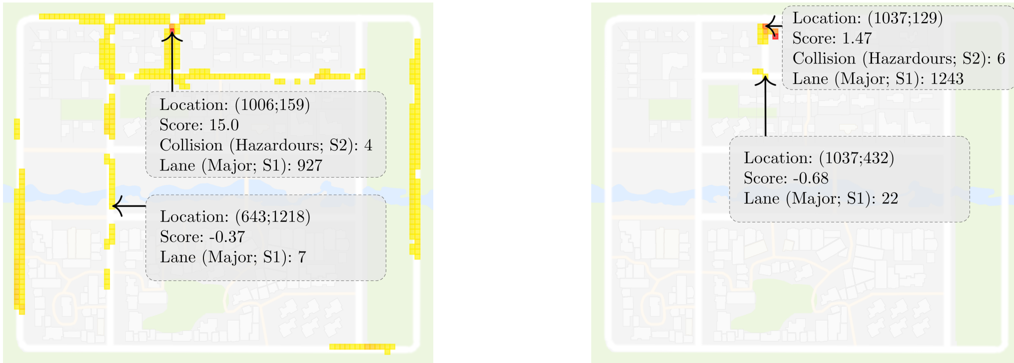
An indispensable part of this framework is the visualisation of the given mapping. Scores and counts are calculated in tiles with the given grid size. The visualisation helps the developer and safety engineer to identify and understand the problems of the agent in an intuitive way. We propose to use three different types of visualisation methods: a simple text output, a 2D map with highlights of the safety violations and an overlay in which violations can be displayed directly in the simulation environment (cf. Figure 2)

4 Evaluation and Results

Evaluation Setup. To evaluate or approach we compare two agents within the simulation environment CARLA. As agents we use a reinforcement Learning (RL) agent [Dosovitskiy *et al.*, 2017] and an Imitation Learning (IL) agent [Codevilla *et al.*, 2018]. The RL agent is trained as a proof of concept in the context of the first CARLA draft. It is based on the asynchronous advantage actor-critic (A3C) algorithm and is trained for goal-directed navigation in CARLA. The reward is based on speed, distance to the goal, collision and position in the assigned lane. From our point of view, the agent is driving acceptable for an evaluation of the safety. Nevertheless, the agent faces a considerable amount of issues, especially in the task of navigating and it has only limited awareness regarding other road users. The second agent is trained using Conditional Imitation Learning (CIL) and is an improved version of an imitation agent presented in the first CARLA draft. Imitation Learning uses knowledge of an expert and imitates the behaviour of the expert; a human driver in this case. This agent is much better at navigating, driving and awareness regarding other road users. Nevertheless, this agent has several limitations, e.g. preserving right of way rules.

Test Environment. The agents initially are set to drive a distance of 100km in the simulated environment with a number of iterations according to the preference of the safety engineer. This test environment uses a distance stop criterion over time or episodic criterion because the navigating capabilities of the agent strongly influence the episodes. We do not specify a time criterion to avoid punishing agents driving with higher speed. The episodes have a fixed number of critical situation (like intersections) and driving slower through them will decrease the number of critical situations in total. On the selected map, the route is set to be straight from the origin to the destination, therefore no advanced navigation capabilities are required. Nevertheless, the routes still contain critical situation such as intersections, pedestrians or slower driving vehicles. Situations with traffic, are considered as well as traffic-free scenarios for the testing. The test environment with traffic includes 100 other cars and 40 pedestrians. With this configuration, the scenarios are crowded by cars and pedestrians but without stop-and-go or traffic jams. We apply the safety constraints *Distance*, *Lane*, and *Collision* for evaluating the agents regarding safety and testing the framework. In the traffic-free scenarios, the distance constraint is not relevant since no other cars are involved. The value for an appropriate distance is set to two seconds, as a common practice.

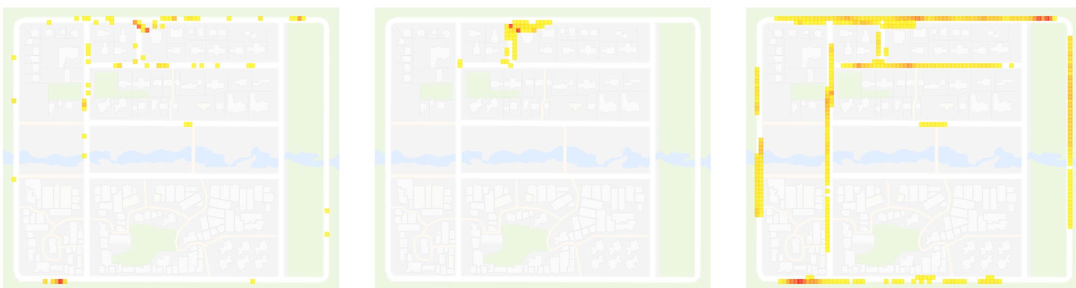
Results. Figure 2a represents the violations of the RL and Figure 2b depicts the results of the IL agent in the traffic-free scenario. For the RL agent only 19 out 71 violations ($\sim 26\%$) did not occur in this area and the IL agent did not collide outside of this region. This is an indication of a problem for the agents here. The amount and distribution of lane violations of the RL agent imply a broader issue regarding lane keeping and collision avoidance. We assume there exist a relationship between the collisions and the lane violations, but there are plenty of lane violations observed without any related collision. We assume that there are no collisions detected since there is no other traffic specified in the scenarios in which the car may face a collision possibility. Driving on the wrong lane or on the side-walk causes no collisions if there are no objects to collide with. According to the results, it is obvious that the IL agent performs a safer drive in comparison to the RL agent with better performance in lane keeping. There are no lane violations or collisions recorded outside the mentioned hot spot. Contrary to the mapped states of Figure 2, Figure 3 clarifies the safety violations of the IL agent separated by the violation type in the scenario with traffic. Again, the IL agent performs a much safer driving compared to the RL agent. The lane violations are similar to the traffic-free scenario. Most violations occurred in the same area, but have a higher variance. There is a massive increase in collisions and in this scenario many violations got recorded all over the



(a) State-Map of the RL agent without traffic

(b) State-Map of the IL agent without traffic

Figure 2: State-Map of the evaluation scenario without traffic



(a) Collision

(b) Lane violations

(c) Distance violations

Figure 3: State-Map of the IL agent with traffic

map. The safety-critical areas is as before, but several new hot spots are also added to the consideration afterwards.

5 Discussion

Interpretation of Results. The results reflect our previous intention regarding the safety of the agents. The IL agent drives much safer than the RL agent, but both still have many limitations. The IL agent caused in every category fewer violations (cf. Table 1). This table only represents a high level overview of the safety violations but can be extended with the type of a collision, traffic situation, time and road conditions as well. Further, there is a relation identified between a collision and distance/lane violations, which shows the direct connection to safety. It is worth to mention that this framework is used in order to exploit the safety violations rather than enhancing the uncertainty, however the results can be used to increase the confidence of the developed intelligent features w.r.t. safety factors.

The framework was able to identify several safety-critical situations. Interesting to mention are the ones that got identified as safety-critical for both agents. There are no obvious causes of the turbulence in this area, but it seems to be a general problem. Furthermore, the framework demonstrates that the IL agent is driving much safer comparably, hence this reflects the findings of Dosovitskiy et al. [Dosovitskiy *et al.*, 2017]. We were able to enlighten the relation between collisions (main symptom of insufficiency in safety) and the safety measures of lane and distance. In the vicinity of collisions, we also observed either a hot spot of lane/distance violations. This demonstrates the impact of those two measures on safety. Additionally, the framework detected several hot spots of lane/distance violations without collisions in the predefined environment scenario. This indicates either the safety measures are too strict or there are not sufficient episodes to provoke any collision.

Table 1: Total violations occurred by the RL and IL agent

	Traffic	Time	Episodes	Collision	Distance	Lane
RL	w/o traffic	243min	586	71	–	10109
	w/ traffic	286min	1201	626	17418	17332
IL	w/o traffic	327min	641	29	–	5758
	w/ traffic	437min	836	154	14116	4951

As mentioned in Section 4, we evaluated our approach using three main safety constraints that are Collision, Lane and Distance Violation. These safety constraints can be seen as high-level safety requirements. To this end, the approach does not provide an automated way to transform the high-level requirements to more detailed ones (e.g. safe distance violation (high level) to 2-seconds (rule)). However, since the simulator is able to provide more information on the test environment as well as the features of the car (e.g. sensor data), we believe that detailed requirements could be achieved accordingly (top-down approach). As stated in ISO 26262:2011-3, the safety requirements should be evaluated to determine their effectiveness, therefore we suggest to use our proposed framework as a prototype for this purpose.

This work is in coordination with the methodology of Salay and Czarnecki [Salay and Czarnecki, 2018] on considerations for developing a safety-critical software and is also a suitable application for supporting the iterative *Hazard Analysis and Requirement Refinement* [Warg *et al.*, 2016] in order to determine the hazardous condition of autonomous driving applications. With the help of this framework, a system prototype can be used in the simulated environment and respectively a safety engineer can analyse the safety level of the driving application in conjunction with road and environment conditions.

Threats to Validity. In terms of *internal threats to validity*, the evaluation of the approach may not be generalizable to the divers driving environments due to the limitation of the map and algorithm that are provided by CARLA simulator. Nevertheless, we have minimised the risk of this threat by evaluating different driving environment within the provided map. In terms of *external threats to validity*, we attempt to introduce a generalised safety violation identification and assessment framework that can be used for multiple types of autonomous driving scenarios in simulated environment. However, we represent our development only in CARLA simulator. Implementing this framework to other autonomous driving simulators remains as our future work.

6 Conclusion

In this work, we presented a framework for evaluating the safety of agents. Safety engineers and Automated Driving Systems (ADS) developers can use this framework to develop, improve and evaluate the ADS. Initially, we highlighted the problem of quantifying safety and showed the concept of *Safety Measures* as a solution to this problem. We apply different types of safety measures and showed their relevance to safety. Most mentionable ones are *Collision Avoidance* and *Safe Driving Behaviour*. We evaluate the proposed framework by checking two learning agents over several episodes implemented in the CARLA simulator based on the defined safety constraints. We could demonstrate promising results regarding the detection and identifying the relationships of safety violations and respectively, recognizing safety-critical situations. Furthermore, this framework allows to easily setup self-driving car approaches by employing safety measures. Developers can simply set up a self-driving car agent, and safety engineers can build a framework of safety measures on top of it. Both groups can evaluate and improve their ideas and will be able to build better and safer approaches for the applications of autonomous driving.

For future work, we would like to (i) extend the framework to include other types of safety violation to be identified, (ii) improve the visualisation of safety violation by presenting more relevant information, and (iii) explore the possibility of this approach to support the determination of Automotive Safety Integrity Level (ASIL) for autonomous driving functions. Furthermore, we also would like to evaluate the effectiveness and suitability of this approach in identifying and assessing safety violations of driving functions from a system developer and safety engineer perspective.

References

[AdaptIVe, 2019] AdaptIVe. Automated driving applications and technologies for intelligent vehicles - adaptive fp7 project- automated driving applications and technologies for intelligent vehicles. http://www.adaptive-ip.eu/, 2019. (Accessed on 03/26/2019).

[Agilysis, 2019] Agilysis. Crashmap - uk road safety map. https://www.crashmap.co.uk/, 2019. (Accessed on 03/26/2019).

[Amodei *et al.*, 2016] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Francis Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *CoRR*, abs/1606.06565, 2016.

[CARLA, 2019] CARLA. Carla ad challenge. https://carlachallenge.org/, 2019. (Accessed on 03/26/2019).

[Codevilla *et al.*, 2018] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *International Conference on Robotics and Automation (ICRA)*, 2018.

[Codevilla, 2018] Felipe Codevilla. Carla 0.8.2: Driving benchmark. http://carla.org/2018/04/23/release-0.8.2/, 4 2018. (Accessed on 05/08/2019).

[DARPA, 2019] DARPA. The grand challenge for autonomous vehicles. https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles, 2019. (Accessed on 03/01/2019).

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[Eggert, 2014] J. Eggert. Predictive risk estimation for intelligent adas functions. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 711–718, 10 2014.

[General Motors, 2018] General Motors. 2018 self-driving car report, 2018.

[González *et al.*, 2018] Leonardo González, Enrique Martí, Isidro Calvo, Alejandra Ruiz, and Joshue Pérez. Towards risk estimation in automated vehicles using fuzzy logic. In Barbara Gallina, Amund Skavhaug, Erwin Schoitsch, and Friedemann Bitsch, editors, *Computer Safety, Reliability, and Security*, pages 278–289. Springer International Publishing, 2018.

[Hallerbach *et al.*, 2018] Sven Hallerbach, Yiqun Xia, Ulrich Eberle, and Frank Köster. Simulation-based identification of critical scenarios for cooperative and automated vehicles. 04 2018.

[ISO 26262, 2011] ISO 26262. ISO 26262:2011 Road vehicles – Functional safety, 2011.

[Kane *et al.*, 2015] Aaron Kane, Omar Chowdhury, Anupam Datta, and Philip Koopman. A case study on runtime monitoring of an autonomous research vehicle (arv) system. In Ezio Bartocci and Rupak Majumdar, editors, *Runtime Verification*, pages 102–117, Cham, 2015. Springer International Publishing.

[Mario Morando *et al.*, 2018] Mark Mario Morando, Qingyun Tian, Long Truong, and Hai L. Vu. Studying the safety impact of autonomous vehicles using simulation-based surrogate safety measures. *Journal of advanced transportation*, 2018, 02 2018.

[NVIDIA, 2019] NVIDIA. Safety force field. https://www.nvidia.com/en-us/self-driving-cars/safety-force-field/, 2019. (Accessed on 06/19/2019).

[PEGASUS, 2019] PEGASUS. Home - pegasus-en. https://www.pegasusprojekt.de/en/, 2019. (Accessed on 03/26/2019).

[Salay and Czarnecki, 2018] Rick Salay and Krzysztof Czarnecki. Using machine learning safely in automotive software: An assessment and adaption of software process requirements in iso 26262. *arXiv preprint arXiv:1808.01614*, 2018.

[Shalev-Shwartz *et al.*, 2017] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *CoRR*, abs/1708.06374, 2017.

[Statistische Ämter des Bundes und der Länder, 2019] Statistische Ämter des Bundes und der Länder. Unfallatlas — kartenanwendung. https://unfallatlas.statistikportal.de/, 2019. (Accessed on 03/26/2019).

[Tesla, 2018] Tesla. Tesla vehicle safety report — tesla. https://www.tesla.com/VehicleSafetyReport?redirect=no, 2018. (Accessed on 03/11/2019).

[Thorn *et al.*, 2018] Eric Thorn, Shawn Kimmel, and Michelle Chaka. A framework for automated driving system testable cases and scenarios, 09 2018.

[Warg *et al.*, 2016] Fredrik Warg, Martin Gassilewski, Jörgen Tryggvesson, Viacheslav Izosimov, Anders Werneman, and Rolf Johansson. Defining autonomous functions using iterative hazard analysis and requirements refinement. In *International Conference on Computer Safety, Reliability, and Security*, pages 286–297. Springer, 2016.

[Waymo, 2018] Waymo. Waymo safety report, 2018.