Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
©2023 ESREL2023 Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P252-cd

Enhancing Safety Assurance for Automated Driving Systems by Supporting Operation Simulation and Data Analysis

Peng Su

Department of Engineering Design, KTH Royal Institute of Technology, Sweden E-mail: pensu@kth.se

ShuTing Kang

University of Chinese Academy of Sciences, China; Institute of Software Chinese Academy of Sciences, China
E-mail: kangshuting18@mails.ucas.ac.cn

Kaveh Nazem Tahmasebi

Department of Engineering Design, KTH Royal Institute of Technology, Sweden E-mail: kavent@kth.se

DeJiu Chen *

** Corresponding Author, Department of Engineering Design, KTH Royal Institute of Technology, Sweden*
E-mail: chendj@kth.se

Automated Driving Systems (ADS) employ various techniques for operation perception, task planning and vehicle control. For driving on public roads, it is critical to guarantee the operational safety of such systems by attaining Minimal Risk Condition (MRC) despite unexpected environmental disruptions, human errors, functional faults and security attacks. This paper proposes a methodology to automatically identify potentially highly critical operational conditions by leveraging the design-time information in terms of vehicle architecture models and environment models. To identify the critical operating conditions, these design-time models are combined systematically with a variety of faults models for revealing the system behaviours in the presence of anomalies. The contributions of this paper are summarized as follows: 1) The design of a method for extracting related internal and external operational conditions from different system models. 2) The design of software services for identifying critical parameters and synthesizing operational data with fault injection. 3) The design for supporting operation simulation and data analysis.

Keywords: Automated Driving Systems, Minimal Risk Condition, Condition Monitoring.

1. Introduction

Automated Driving Systems (ADS) employ embedded electrical and electronic (E/E) systems for advanced functionalities, including driving perception, localization, decision-making, and control strategies. Especially highly automated driving systems (L4 and L5), defined by the SAE automation levels SAE (2018), are expected to conduct Dynamic Driving Tasks (DDT) automatically according to perceived internal and external operational conditions without direct human interactions.

While the technologies for automated driving have made rapid progress over the last decade, challenges remain in supporting the trustworthiness of such systems. In general, a trustworthy system is believed to operate within defined levels of risk despite the presence of aleatory and epistemic uncertainties Chen et al. (2018). The support involves many aspects of system development, operation control, maintenance, and evolution. One key task is related to the assurance of operational safety by attaining Minimal Risk Conditions (MRC) despite unexpected environmental disruptions, human errors, functional faults, and security attacks. For ADS, this is however a more challenging task than for more conventional driving systems (below L3) as learning-enable components (LEC) are directly used in the operation perception, task planning, and control. Based on advanced ML/AI algorithms, such learning-enable components approximate some desired behaviours. They normally contain huge amounts of parameters (weights and biases), which are identified with machine learning techniques. This implies several challenges in safety engineering. Some well-known reasons include Salay et al. (2017): 1) Training data may not cover all possible operational conditions; 2) Safety requirements may be incomplete due to unknown probabilistic nature of LEC; 3) The explainability of LEC is often hampered by the non-transparency of such components. These call for novel methods and tools for ADS modelling, analysis, testing, and condition monitoring.

In this paper, we present a conceptual framework aimed at facilitating the safety assurance for ADS by providing: 1) The design of a method for integrating the internal and external operational conditions by modelling the vehicle architecture and environment models. 2) The design of software services for automatically identifying critical parameters and synthesizing operational data by fault injection. 3) The design for supporting operation simulation and data analysis.

2. Related Work

In the system development, the specifications of Operational Design Domain (ODD) describe the operational conditions where the ADS is intended to function concerning the roadway types, speed ranges, lighting conditions (day and/or night), weather conditions, and other operations constraints. The specifications play a key role in safety engineering by stipulating the intended functional system boundary Greenblatt and Shaheen (2015). For highly automated driving systems (L4 and L5), the ADS should be able to automatically achieve MRCs by detecting potential ODD excursion. To this end, the following measures in system development are expectedChen et al. (2018): 1) Identifying appropriate ODD as system requirements; 2) Developing suitable driving functions and components for meeting these requirements; 3) Preventing unsafe operating conditions when system failure or ODD exit occurs.

In the automotive industry, it is evident that the established approach to functional safety as defined in ISO 26262 ISO26262 (2022) Functional Safety (FuSa) is no longer sufficient for ADS. One complementary standard is ISO 21448 Safety of the Intended Functionality (SOTIF). The SOTIF specification provides a general safety assurance framework and guidance on measures to ensure the safety of the intended functionality (SOTIF) ISO21448 (2022). One key effort in supporting the safety engineering is the usage of domain-specific models (DSM) to capture the system faults and operational knowledge of concern (e.g. Chen et al. (2013); Koopman and Wagner (2018); Gyllenhammar et al. (2020)). Such models stipulate the system parameters, design solutions and requirements and thereby constitute the basis for describing the internal operational conditions of ODD as well as for tracing their system-wide interdependence. Meanwhile, the environment of the ADS, are concerned to maintain the intended functional safety. Current work in Chen et al. (2022) proposes a generator based on OpenScenario, an environment model, to define the external operational conditions. This environment model specifies different elements such as movable objects and their actions, geo-spatial stationary features, traffic rules, laws, and policies.

In engineering practices, the provision of methods and tools connecting requirements, design decisions, formal analysis, testing outcomes and other operational feedback plays a key role in safety assurance. A methodology for combining system models and condition monitoring services to enable dynamic assessment of operational uncertainties and risks is given in Chen and Lu (2017). The analysis of field operational data, supported often by machine learning methods, allows the enrichment of knowledge about system operation Elgharbawy et al. (2019). Condition monitors help detect potentially unsafe and unexpected behaviors caused by unexpected environmental conditions and functional faults Koopman and Wagner (2016); Törngren et al. (2018); Koopman and Wagner (2018); Rahman et al. (2021). Formal methods have also been employed in current practices for verifying the safety of autonomous driving systems (ADS) Hekmatnejad et al. (2019); Zapridou et al. (2020); Rahman et al. (2021). Nevertheless, the capability in supporting the rea-

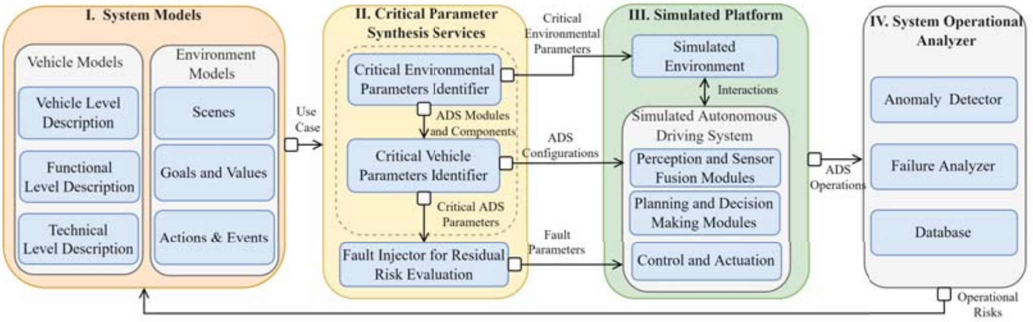

I. System Models
Vehicle Models
Vehicle Level Description
Functional Level Description
Technical Level Description
Environment Models
Scenes
Goals and Values
Actions & Events
Use Case
II. Critical Parameter Synthesis Services
Critical Environmental Parameters Identifier
ADS Modules and Components
Critical Vehicle Parameters Identifier
Critical ADS Parameters
Fault Injector for Residual Risk Evaluation
Critical Environmental Parameters
ADS Configurations
Fault Parameters
III. Simulated Platform
Simulated Environment
Interactions
Simulated Autonomous Driving System
Perception and Sensor Fusion Modules
Planning and Decision Making Modules
Control and Actuation
ADS Operations
IV. System Operational Analyzer
Anomaly Detector
Failure Analyzer
Database
Operational Risks

Fig. 1. **The Architecture of the Proposed Framework**. We propose the conceptual framework to enhance safety assurance for ADS by supporting operation simulation and data analysis.

soning of the potential risks and their sources depends on the scope and richness of underlying system models. In Törngren et al. (2018); Hartsell et al. (2021); Chelouati et al. (2022), researchers also propose various learning-based techniques for condition monitoring and risk assessment. To optimize the performance, some of these faults could be labelled or isolated through historical experiments, system analysis and testing, and safety concepts (e.g., safety mechanism with redundancy design for fault tolerance).

3. Overview

In this paper, we present the design of a framework that aims at assuring the operational safety of ADS by connecting system models and operational data analysis (Fig. 1). The design models contain the specifications of system architecture and external environment. A software service (referred to as critical environmental parameters identifier) is introduced to derive potentially safety-critical environmental conditions and the corresponding operational scenarios to be simulated. To verify and validate the robustness of the intended functionalities of ADS, another software service (referred to as critical vehicle parameters identifier) is employed to elicit potentially critical parameters of a vehicle for a specification of fault injection. To verify and test the system models and their parameters, we support simulation platforms (e.g., CARLA) to collect operational data by using these generated specifications of operational scenarios and faults. A system operational analyzer is designed to collect and analyze the operational data generated by the simulation runs. This analyzer automatically classifies the failure cases and the related component anomalies. The results are then fed back to the system models for the enrichment of system knowledge.

4. System Modeling

The system models define and parameterize all design information of the ADS, including the internal and external operational conditions (Block I in Fig. 1). However, exploring the overall system design space and the implied operational conditions is a complex and intractable task. Therefore, we generate these operational conditions by providing: 1) Using Architecture Description Languages (ADL) to define and parameterize ADS and its functional components; 2) Using Scenario Description Languages (SDL) to model and quantify the external operational conditions.

4.1. *Specifying ADS Architecture*

Formally justifying AI component's behaviours, performance and other requirements are critical to ensure the safety of ADS. However, this could be a challenging task because of various purposes and techniques of AI components. To solve this issue, we elicit the internal operational behaviours by first modelling the ADS contexts with multiple abstractions ranging from vehicle-level features to platform-specific technical-level features (Fig. 1). The approach is based on EAST-ADL, an architecture analysis and design language for

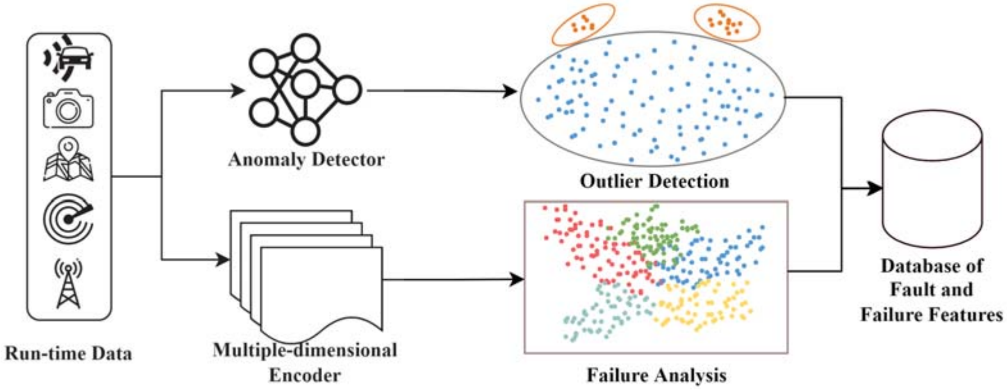

Run-time Data
Anomaly Detector
Multiple-dimensional Encoder
Outlier Detection
Failure Analysis
Database of Fault and Failure Features

Fig. 2. **The design of system operational analyzer based on condition monitoring.**

automotive development Chen et al. (2013). The ADS system is first described at the vehicle-level in a solution-independent way, where each module represents an intended driving task. The underlying functional level description specifies the underlying functional I/O and algorithmic features for each driving task. Towards the final system realization, the technical level description specifies corresponding software and hardware architectures, covering the needed run-time environment services, data models, I/O and communication networks, etc.

Given such a modelling framework, we extract a set of key vehicle parameters. These parameters define and quantify the ADS internal operational conditions across different abstraction levels.

4.2. *Specifying ADS Environment*

To ensure the operational safety of ADS, we also need to specify the environmental conditions. Our approach follows the environment modelling method introduced in Chen et al. (2022); Fremont et al. (2020). The scope of modelling includes 1) Scene, which describes a snapshot of the environment, including traffic participants, weather conditions, and map information; 2) Permanent goals & values, which encompass traffic rules, laws, and policies. 3) Actions & events, which support actions with traffic participants. We also develop an environment modelling language Kang et al. (2022) to integrate and manage the external operational conditions of concern. The approach also allows detailed description of scenes. Furthermore, the language employs functional definitions to encapsulate related actions & events.

5. Synthesizing Critical Parameters from the System Models

Services shown in Block II of Fig. 1 provide support for identifying and configuring the cases of simulation for synthesising operational data. The causes of ODD excursion under consideration include unexpected environmental conditions, faults from the ADS, and their combinations.

5.1. *Identifying Critical Parameters*

Two functions are developed to identify the safety-critical parameters in the vehicle and environment models (the dash-lined box in Block II of Fig1). To identify the critical environmental conditions, we adopt a Reinforcement Learning (RL) based approach as introduced in Kang et al. (2022) to explore state space. Based on the environment models, we also describe action sequences of traffic participants. To understand the corresponding effects and robustness of ADS components, another identifier is used to search the critical vehicle parameters. The backbone of this critical vehicle parameters identifier is to search causal relationships based on the predefined attributes (e.g., failure logic) in the ADS models. Compared with time-consuming simulations for all the vehicle parameters, the identified results explicitly reveal correlations between

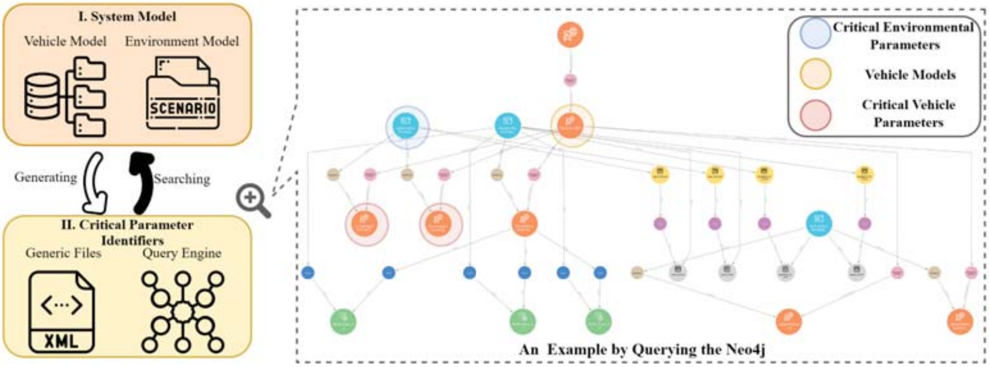

I. System Model
Vehicle Model
Environment Model
SCENARIO
Generating
Searching
II. Critical Parameter Identifiers
Generic Files
Query Engine
<...>
XML
Critical Environmental Parameters
Vehicle Models
Critical Vehicle Parameters
An Example by Querying the Neo4j

Fig. 3. **Parameter treatment based on Neo4j.** In this case, the internal and external operational conditions are described as vehicle and environment models, exported as generic files into the Neo4j. According to the predefined attributes, their dependencies are extracted automatically by Neo4j tool.

vehicle and environment models under specific operational conditions, accelerating to verify and validate the operational safety of the ADS.

5.2. *Specifying Fault Injection*

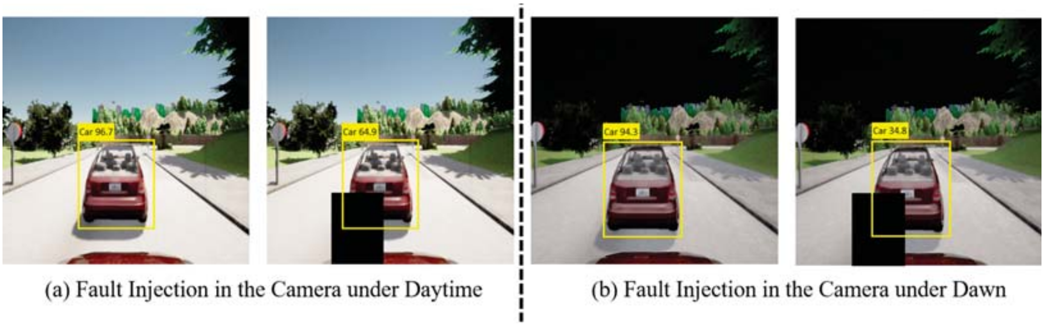
Fault injection provides support for evaluating system robustness and residual risks. We consider a wide variety of fault types across the compositional hierarchy of ADS. For example, a perception system in ADS contains: 1) Sensors (e.g., camera and radar), 2) Learning-enable functions (e.g. neural networks), 3) Sensor fusion function for world-modeling. These components can be associated with different fault parameters: 1) Failure-in-Time (FIT) and ageing factor Fabarisov et al. (2022). 2) Gaussian noise and solid occlusions Jha et al. (2018). 3) Hardware specific faults like bit-flip and stuck-at Su and Chen (2022). We use a fault injector to assign these fault parameters into ADS components under different environmental conditions.

6. Analyzing Simulated Operational Data by Condition Monitoring

The simulation platform is shown as Block III of Fig. 1. Given the specifications of critical environmental and vehicle parameters, the simulation cases are configured and implemented. The simulation operational data are collected and analyzed by an analyzer, shown as the system operational analyzer in Block IV of Fig. 1 . The generated data characterize the interactions of ADS and its environment by sampling the simulated operational data and signals. A system operational analyzer is then introduced to detect the anomalies and failure cases for the enrichment of system knowledge. This helps assure ADS safety by: 1) Improving the tolerance of the ADS for functional faults by synthesizing the results of the anomaly detector; 2) Enhancing the robustness of the ADS under different external operational conditions by analyzing features from the failure analyzer; 3) Completing the parameters of the ADS and environment models.

6.1. *Anomaly Detection*

Anomalies can exhibit significantly different features (e.g., data value) from the rest of the data. Anomalies usually indicate the observations of errors occurred in the system. Some of these errors cause system failures according to the system design and failure logic. Therefore, to ensure system safety, associating possible component anomalies with critical system failures by monitoring and analyzing the operational behaviours becomes important. In Fig. 2, we illustrate an example using deep learning-based methods to detect outliers, a common type of anomalous data. The detector infers the reasons and consequences of the outliers by comparing them with fault features acquired from the fault injection, providing a reference for ensuring operational safety.



(a) Fault Injection in the Camera under Daytime

(b) Fault Injection in the Camera under Dawn

Fig. 4. **Examples of fault injection for camera sensor** The weather conditions are visualized by the corresponding parameters from the environmental model. The solid occlusion is defined in the fault injector.

6.2. *Failure Analysis*

Anomalies or errors in system components can propagate to system failures which in turn lead to undesired ODD excursions. To capture the behaviour of error propagation from a component, we use multiple-dimensional encoders to model and classify the patterns implied by run-time data. To cope with the complexity, a generative model (e.g., Variational Autoencoder) is preferable due to the advantages of handling multiple-dimensional data with latent space representation (e.g,. the motion estimation in Karl et al. (2016)). When combined with pattern recognition, the approach can effectively classify even unexpected operational situations. As illustrated in Fig. 2, the sampled operational data are encoded and grouped into different patterns (e.g., manifolds), where each color indicates an observed specific operational condition. When an encoded model mismatches with its neighbours, unexpected operational conditions would have occurred. By classifying these observations, a database synthesizes potential operational risks, which support to enrich the system models.

7. Case Study: Safety Analysis of the Object Detection Module

To evaluate our framework, a case study with the modeling and simulation of an ADS system with object detection module has been carried out. The system is expected to work in all weather conditions. We define and qualify the weather parameters in the environment models. Following these environment parameters, we use Neo4j (Fig. 3), a graphical database, to identify critical vehicle parameters. In the functional level description, the object detection module is decomposed into a camera component and an AI component. To achieve the intended functionalities of these components, the technical level description refines these components with specific behaviors (e.g., the camera should capture the RGB-D image, the AI component should detect objects with the regions of interest).

Next, the vehicle and environment parameters generate different ADS operational scenarios to be simulated with the injections of functional fault (e.g., solid occlusions) in the object detection module. The results show that although this module works well both daytime and dawn, it is vulnerable to solid occlusions. Furthermore, such results reflect the functional behaviors of the AI component should be enhanced during the dawn. Therefore, the system operational analyzer indicates that the anomaly detector should monitor images with solid occlusions. Meanwhile, the failure analyzer encoding data from different sensors should classify different weather conditions (e.g., if the current time is dawn) to prevent accidents from the low precision of the AI component.

8. Discussion and Future Work

In this paper, we have presented a conceptual framework for enhancing safety assurance by supporting fault-injection simulation and data-driven analysis of failure behaviours. In future work, critical parameters can be identified by Reinforcement Learning with formal methods. The formal methods support to improve the performance of the RL agent and provide an interpretation for selecting critical parameters. Moreover, the risk assessment approaches proposed by ISO 26262 (e.g., HARA) can be combined for the refined safety goals and requirements specification. Fault Tree Analysis (FTA) based description of faults and error propagation can also be combined as labels for machine learning or integrated as holistic models for simulation design. Another desired feature is that the operation analyzer should provide a support for quantifying the probability of ODD excursions.

Acknowledgement

This work is supported by the SALIENCE4CAV project (ref. 2020-02946) funded by the Strategic vehicle research and innovation programme (FFI), Sweden.

References

Chelouati, M., A. Boussif, J. Beugin, and E.-M. El Koursi (2022). A framework for risk-awareness and dynamic risk assessment for autonomous trains. In *32nd European Safety And Reliability Conference*.

Chen, D. and Z. Lu (2017). A model-based approach to dynamic self-assessment for automated performance and safety awareness of cyber-physical systems. In *Model-Based Safety and Assessment*. Springer.

Chen, D., N. Mahmud, M. Walker, L. Feng, H. Lönn, and Y. Papadopoulos (2013). Systems modeling with east-adl for fault tree analysis through hip-hops. *IFAC Proceedings Volumes 46*(22), 91–96.

Chen, D., K. Östberg, M. Becker, H. Sivencrona, and F. Warg (2018). Design of a knowledge-base strategy for capability-aware treatment of uncertainties of automated driving systems. In *Computer Safety, Reliability, and Security*, pp. 446–457. Springer.

Chen, H., H. Ren, R. Li, G. Yang, and S. Ma (2022). Generating autonomous driving test scenarios based on openscenario. In *2022 9th International Conference on Dependable Systems and Their Applications*.

Elgharbawy, M., A. Schwarzhaupt, M. Frey, and F. Gauterin (2019). Ontology-based adaptive testing for automated driving functions using data mining techniques. *Transportation research part F: traffic psychology and behaviour 66*, 234–251.

Fabarisov, T., A. Morozov, I. Mamaev, and P. Grimmeisen (2022). Fidget: Deep learning-based fault injection framework for safety analysis and intelligent generation of labeled training data. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*.

Fremont, D. J., E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia (2020). Scenic: A language for scenario specification and data generation. *arXiv preprint arXiv:2010.06580*.

Greenblatt, J. B. and S. Shaheen (2015). Automated vehicles, on-demand mobility, and environmental impacts. *Current sustainable/renewable energy reports 2*, 74–81.

Gyllenhammar, M., R. Johansson, F. Warg, D. Chen, H. M. Heyn, M. Sanfridson, J. Söderberg, A. Thorsén, and S. Ursing (2020). 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)*.

Hartsell, C., S. Ramakrishna, A. Dubey, D. Stojcsics, N. Mahadevan, and G. Karsai (2021). Resonate: A runtime risk assessment framework for autonomous systems. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pp. 118–129. IEEE.

Hekmatnejad, M., S. Yaghoubi, A. Dokhanchi, H. B. Amor, A. Shrivastava, L. Karam, and G. Fainekos (2019). Encoding and monitoring responsibility sensitive safety rules for automated vehicles in signal temporal logic. In *Proceedings of the 17th ACM-IEEE International Conference on Formal Methods and Models for System Design*, pp. 1–11.

ISO21448 (2022, Jun). Road vehicles—safety of the intended functionality.

ISO26262 (2022, Jun). Road vehicles—functional safety.

Jha, S., S. S. Banerjee, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer (2018). Avfi: Fault injection for autonomous vehicles. In *2018 48th annual ieee/ifip international conference on dependable systems and networks workshops (dsn-w)*, pp. 55–56. IEEE.

Kang, S., H. Guo, L. Zhang, P. Su, G. Liu, Y. Xue, and Y. Wu (2022). Ecsas: Exploring critical scenarios from action sequence in autonomous driving.

Kang, S., H. Hao, Q. Dong, L. Meng, Y. Xue, and Y. Wu (2022). Behavior-tree based scenario specification and test case generation for autonomous driving simulation. In *2022 2nd International Conference on Intelligent Technology and Embedded Systems (ICITES)*, pp. 125–131. IEEE.

Karl, M., M. Soelch, J. Bayer, and P. Van der Smagt (2016). Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*.

Koopman, P. and M. Wagner (2016). Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety 4*(1).

Koopman, P. and M. Wagner (2018). Toward a framework for highly automated vehicle safety validation. *SAE Technical Paper, Tech. Rep.*

Rahman, Q. M., P. Corke, and F. Dayoub (2021). Runtime monitoring of machine learning for robotic perception: A survey of emerging trends. *IEEE Access 9*, 20067–20075.

SAE (2018). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE international 4970*(724).

Salay, R., R. Queiroz, and K. Czarnecki (2017). An analysis of iso 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*.

Su, P. and D. Chen (2022). Using fault injection for the training of functions to detect soft errors of dnns in automotive vehicles. In *Proceedings of the 17th International Conference on Dependability of Computer Systems, 2022, Poland*, pp. 308–318. Springer.

Törngren, M., X. Zhang, N. Mohan, M. Becker, L. Svensson, X. Tao, D.-J. Chen, and J. Westman (2018). Architecting safety supervisors for high levels of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1721–1728. IEEE.

Zapridou, E., E. Bartocci, and P. Katsaros (2020). Runtime verification of autonomous driving systems in carla. In *Runtime Verification: 20th International Conference, RV 2020, Los Angeles, CA, USA, October 6–9, 2020, Proceedings*, pp. 172–183. Springer.