

# Finding Critical Scenarios for Automated Driving Systems: The Data Extraction Form

Xinhai Zhang<sup>1,2,3</sup>, Jianbo Tao<sup>4,5</sup>, Kaige Tan<sup>1</sup>, Martin Törngren<sup>1</sup>, José Manuel Gaspar Sánchez<sup>1</sup>, Muhammad Rusyadi Ramli<sup>1</sup>, Xin Tao<sup>1</sup>,  
Magnus Gyllenhammar<sup>1,6</sup>, Franz Wotawa<sup>5</sup>, Naveen Mohan<sup>1</sup>, Mihai Nica<sup>4</sup>, and Hermann Felbinger<sup>4</sup>

<sup>1</sup>KTH – Royal Institute of Technology, Sweden. <sup>2</sup>Scania CV AB, Sweden. <sup>3</sup>Sigma Technology Consulting AB, Sweden.  
<sup>4</sup>AVL List GmbH, Austria. <sup>5</sup>TU Graz – Graz University of Technology, Austria. <sup>6</sup>Zenseact AB, Sweden  
xinhai.zhang@scania.com

This is the data extraction form for the systematic mapping study to find critical scenarios for automated driving systems. Table 1 lists all the acronyms used in the main paper. The extracted data from the primary studies is structured in the rest of the tables according to the taxonomy defined in Section 4 of the main paper. Table 2 shows how the content in these tables links to the taxonomy. Primary studies in Tables 2 to 6 correspond to the five clusters defined in Section 6 of the main paper.

Please note that some primary studies in these tables are classified as out of the scope of the literature study. These studies are marked in the Purpose column. Primary studies in Tables 7 and 8 are eventually considered as out of the scope. The tables are designed aligned with the taxonomy proposed in Section 4 of the main paper.

Table 1. List of acronyms used in the systematic mapping study

SOI	System Of Interest	KPI	Key Performance Indicator	DSE	Design Space Exploration
AEB	Automatic Emergency Braking	NCAP	European New Car Assessment Programme	SBT	Search-based Testing
LKA	Lane Keep Assist	FMEA	Failure Modes and Effects Analysis	NDS	Naturalistic Driving Study
ACC	Adaptive Cruise Control	V2X	Vehicle to Everything Communication	FOT	Field Operational Test
TTC	Time To Collision	RQ	Research Question	VAAFO	Virtual Assessment of Automation in Field Operation
HARA	Hazard Analysis and Risk Assessment	V&V	Verification and Validation	QOS	Quality Of Service
RRT	Rapidly Exploring Random Trees	FTA	Fault Tree Analysis	ODD	Operational Design Domain
MDP	Markov Decision Process	CV	Computer Vision	CSI	Critical scenario Identification
RSS	Responsibility Sensitive Safety	ADS	Automated Driving System	GTA	Grand Theft Auto
GPS	Global Positioning System	ADAS	Advanced Driving-Assistance System		
NLP	Natural Language Processing	FuSa	Functional Safety		
RNN	Recurrent Neural Networks	SOTIF	Safety of the Intended Functionality		

Table 2. Links between the taxonomy in the main paper and the content in the data extraction form

Elements in the taxonomy in the main paper		Elements in the data extraction form
Problem Definition	System of Interest	Purpose -> SOI
	Phase	Purpose -> Phase
	Purpose	Purpose
	Targeted ODD	Purpose -> ODD
	Definition of Criticality	Criticality definition & Surrogate Measure
	Level of Abstraction	Solution -> Input Sce. <b>AND</b> Solution -> Output Sce.
Solution	Scenario Space Construction	Scenario definition
	Scenario Space Construction -> Content	Scenario definition -> Covered layers
	Scenario Space Construction -> Representation	Solution -> Input Sce. <b>AND</b> Solution -> Output Sce. <b>AND</b> Scenario definition
	Scenario Space Exploration	Solution
	Required Information	Validation & other key observations -> Required knowledge
Evaluation		Validation & other key observations -> Validation

Table 3. Exploring logical scenarios without parameter trajectories

#	Purpose	Scenario definition	Criticality definition & Surrogate Measure	Solution	Validation & other key observations
[1][2][3]	<p><b>Activity:</b> formalization (Ontology meta-model) &amp; Instantiation &amp; criticality check</p> <p><b>Phase:</b> system verification / component verification, depending on the type of simulation (e.g., MIL, SIL or HIL)</p> <p><b>SOI:</b> decision-making and control system</p> <p>A general framework to automatically test the whole AD function(s) as a black box, aiming at finding critical scenarios, and thereby improve the AD function(s).</p>	<p>A scenario is defined as the development of the initial scene over time.</p> <p>The author utilizes a domain ontology to capture the environment of the system under test (SUT) in a degree of detail. The ontology consists of static (e.g. road type, infrastructure, lanes, road marking, traffic rules...) and dynamic information (e.g. objects type, speed, acceleration, direction...) of the environment. The ontology for the system under test is constructed using UML as modeling language.</p> <p>This paper proposes a UML-based Ontology meta-model, using which, one can propose</p>	<p>The criticality of scenarios is evaluated based on the different KPIs (e.g. TTC, yaw rate, longitudinal and lateral acceleration and deceleration) of the ADAS/AD functions.</p>	<p><b>Input Sce.:</b> A logical scenario described with the UML-based ontology model</p> <p><b>Output Sce.:</b> critical concrete scenarios</p> <p>Method: makes use of ontologies for describing the environment of autonomous vehicles and convert them to input models for combinatorial testing. The input model includes all the parameters and their representative values including their relations and constraints. The combinatorial test suite comprises abstract test cases that are mapped to concrete test cases that can be executed using CO-simulation platform. Criticality will be evaluated based on the defined KPIs.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1. A given functional scenario from EuroNcap</li><li>2. Ontology of the environment description for the formalization</li><li>3. Functional models of the AD system</li><li>4. Predefined KPIs</li><li>5. A Simulator (VTD)</li></ol> <p><b>Validation:</b></p> <p>A case study to show that this method can find critical and crashing scenarios.</p> <p><b>Similar paper:</b></p>

		<p>their own ontology according to their own case.</p> <p><b>Covered layers:</b> it depends on the final ontology. The given demo use case only considers layer 4 (including the initial states of the ego vehicle)</p>		<p>This paper considers <b>N-wise coverage</b>.</p>	<p>[4] Ponn2019a uses N-wise testing to generate test cases for the Lane Keeping Assist function (ADAS). It provides a detailed ontology of the logical scenario for this use case.</p>
[5]	<p><b>Activity:</b> Instantiation &amp; criticality check (requirement falsification) or refinement of a concrete scenario  <b>Phase:</b> system verification  <b>SOI:</b> the whole system</p> <p>This paper presents a simulation-based testing framework for test case generation and falsification for the whole AD system or a particular function.</p>	<p>A scenario in this paper refers to one simulation run with fixed time interval. It includes the intended behavior of the ego vehicle.</p> <p>A concrete scenario is a vector of relevant parameters such as the positions of other vehicles, the colors and models of other vehicles, etc.</p> <p><b>Covered layers:</b> 1 (fixed for each logical scenario), 4 (including the initial states of the ego vehicle), 5</p>	<p>In this paper, a critical scenario (in this paper it is call a glancing scenario) is defined as a boundary scenario between acceptable scenarios and unacceptable scenarios, e.g., a collision occurs with a small velocity.</p> <p>The acceptance criteria are described in signal temporal logic (STL).</p> <p>A concrete scenario will be run in a simulator to generate a simulation trace (a state trajectory). The simulation traces will be evaluated against a STL specification. The evaluation returns a value denoting the robustness of the satisfaction. If the value is positive and close to zero, it means that the scenario satisfies the specification, but it is very close to unsatisfaction. To this end, glancing scenarios are the ones whose robustness values are close to zero.</p> <p>Identified critical (glancing) scenarios can be used as test cases for falsification.</p>	<p><b>Falsification:</b>  <b>Input Sce.:</b> A logical scenario  <b>Output Sce.:</b> critical concrete scenarios</p> <p><b>Refinement:</b>  <b>Input Sce.:</b> A concrete scenario  <b>Output Sce.:</b> a more critical concrete scenario by tuning the continues parameters.</p> <p>The falsification process has two steps. The first step is to find all the glancing scenarios. The second step is to find collision scenarios around the glancing scenarios.</p> <p>Combinatorial test generation is used for the first step to generate a covering array guaranteeing N-wise coverage. To do this, continues variables are uniformly discretized. Among all the scenarios in the covering array, glancing scenarios are selected.</p> <p>The second step uses each identified glancing scenario as an initial point to optimize the continues variables to find more glancing scenarios.</p> <p>The second step can also be considered as a refinement of a given concrete scenario.</p> <p>This paper considers <b>N-wise coverage</b>.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. A given logical scenario</li> <li>2. Functional models of the ADS (as a black box)</li> <li>3. A simulator (Webots)</li> <li>4. Predefined specifications in STL</li> <li>5. S-TaLiRo, a falsification tool to calculate the robustness value.</li> </ol> <p><b>Validation:</b>  A case study to show that the approach can generate glancing scenarios.</p> <p>It assumes that glancing scenarios are more common than seriously hazardous scenarios.</p> <p>It assumes that the glancing scenarios are avoidable by improving the functionality, and are hence more important.</p>

[6]	<p><b>Activity:</b> Instantiation &amp; criticality check (requirement falsification) or refinement of a concrete scenario</p> <p><b>Phase:</b> system verification</p> <p><b>SOI:</b> the whole system, decision-making and control system, perception system</p> <p>This paper extends [5].</p> <p>It introduces methods to use their framework to falsify the perception systems (referring to their requirement 2 and 3) and the sensor fusion system (referring to their requirement 4).</p>	<p>Same as [5].</p> <p><b>Covered layers:</b>1,4,5</p> <p>Layer 1: Road</p> <p>Layer 4: Ego vehicle, agent vehicle, pedestrian</p> <p>Layer 5: e.g., building colors</p>	<p>Not like [5], this paper tries to find scenarios with the lowest robustness value.</p> <p>The requirements used for its case studies are:</p> <ol style="list-style-type: none"> <li>1. Not collide with an object (system-level)</li> <li>2. Detect visible obstacle within a certain time unit (sensor-level)</li> <li>3. Localization errors should provide sufficient accuracy (defined by an error threshold) (sensor-level)</li> <li>4. A sensor-related fault should not lead to a system-level fault (collision) (sensor-to-system level)</li> <li>5. The vehicle should not do excessive braking unnecessarily or too often (system-level performance)</li> </ol> <p>Different requirements can be used for different use cases.</p>	<p>The searching method is the same as the one proposed in [5].</p> <p>This paper considers <b>N-wise coverage</b>.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. A given logical scenario</li> <li>2. Functional models of the ADS (as black boxes)</li> <li>3. A simulator (Webots)</li> <li>4. Predefined requirements (defined by STL)</li> <li>5. Sensor models, including CCD camera, lidar, and radar</li> <li>6. Stochastic search optimization method for finding falsification</li> </ol> <p><b>Validation:</b></p> <p>Compared to [5], 3 case studies are introduced in the paper to evaluate the performance of the method and describe how to use the results to enhance the development process.</p>
[7]	<p><b>Activity:</b> Instantiation</p> <p><b>Phase:</b> system verification</p> <p><b>SOI:</b> decision-making and control system</p> <p>The proposed method is to generate test cases for the decision-making module of autonomous systems, i.e., the SUT.</p> <p>The generated test cases are diverse and close to decision boundaries, where minor changes to the environment may provide big impact to the behavior.</p>	<p>A test scenario in this paper is specified by one scenario configuration, which covers the environment, mission and vehicle parameters. It does not include the implementation of the SUT.</p> <p>A scenario configuration is formally modeled as a vector in a state space of continuous and discrete values. This state space consists of all the possible scenario configurations that could be tested.</p> <p>The distance between two scenario configurations in the state space is a measure of the similarity between the two scenarios.</p> <p><b>Covered layers:</b> Not explicitly given. We assess that it should cover many of the</p>	<p>In this paper, a scenario is critical if a small change of its configuration leads to significant changes of the SUT performance.</p> <p>Changes of scenario configurations are quantified by the distances in the configuration state space.</p> <p>SUT performance is evaluated in a score space, where each dimension is a performance metric, e.g., TTC, number of way points reached, fuel consumption, etc. – i.e. multiple metrics representing different types of properties are defined.</p>	<p><b>Input sce.:</b> A given logical scenario</p> <p><b>Output sce.:</b> critical concrete scenarios</p> <p>Adaptive sampling, a learning- based testing method, is used with simulation to generate boundary pairs, which contains two similar scenarios where the SUT will behave in two different performance modes.</p> <p>A surrogate model is used to improve the coverage of the sampling. It takes a set of samples as input and returns the estimated diversity (i.e., the mean distance) on the scoring space of the input samples.</p> <p>The sampled scenarios are given to a simulation platform. The simulation results are evaluated according to the score space.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. A given logical scenario</li> <li>2. Functional models of the ADS (as a black box)</li> <li>3. A simulator (APL Autonomy Toolkit)</li> </ol> <p><b>Validation:</b></p> <p>No validation. It only verifies the performance of the proposed method.</p> <p>This method assumes that the behavior of the SUT is deterministic under one scenario configuration.</p> <p>So far, the scenario model does not include reactive agents. The current</p>

		factors that may affect the decision-making module. The given case study uses an unmanned underwater vehicle as an example, which cannot represent automated vehicles.		<p>A clustering method is used to determine the performance modes among the simulation results.</p> <p>performance boundary pairs are identified beside the boundaries between different performance modes.</p> <p>Scenarios adjacent to the identified boundaries can be used as test cases.</p>	<p>version is only for single agent scenarios.</p> <p>It considers design of experiments (DOE) and test case generation as related domains.</p> <p>Similar paper: [8] Nabhan2019a also uses learning based testing methods to generate critical scenarios.</p>
[9]	<p><b>Activity:</b> criticality check  <b>Phase:</b> system verification and system design (sensor range)  <b>SOI:</b> the whole system</p> <p>This paper proposes an approach to identify the performance boundary in the parameter space. Knowledge of performance boundary helps to find corner cases, which are located on the performance boundary.</p>	<p>The case study is a traffic jam approach scenario on a left-turned curved road with a fixed radius of 50m. The ranges of three parameters are defined and values are provided:</p> <p>Ego vehicle speed (40 – 70 km/h)  Target vehicle speed (5 – 20 km/h)  aperture angle of the radar sensor of the ego vehicle (10 – 25°)</p> <p><b>Covered layers by the case study:</b>  Layer 4 + sensor configuration</p>	<p><b>Criticality definition:</b>  This paper interests in corner cases, where small changes in the parameter value can trigger the change from a safety scenario to a critical one. These corner cases are located around performance boundaries between critical scenarios and non-critical scenarios.</p> <p>In this paper, a scenario is considered as critical when the ego vehicle cannot prevent a collision in the simulation.</p>	<p><b>Input sce.:</b> A given logical scenario  <b>Output sce.:</b> criticality</p> <p><b>Process to train classifier:</b></p> <ol style="list-style-type: none"> <li>1. Create a set of concrete scenarios by sampling a given logical scenario with specified sampling methods i.e., Monte Carlo Simulation and Latin Hypercube (LHC).</li> <li>2. Simulate the sampled concrete scenarios to obtain simulation results (critical or non-critical) as labels of the concrete scenarios.</li> <li>3. Use these data to train a Gaussian Process Classifier (GPC) model, which yields the performance boundary.</li> </ol> <p>The current work focuses on the derivation of a classifier that distinguishes critical scenarios and non-critical scenarios. In the future work, an adaptive framework will be integrated to conduct sampling on the performance boundary to find corner cases.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Vehicle models in simulation platform (CarMaker), which provides training data and test data through simulation.</li> <li>2. A given logical scenario</li> <li>3. Pre-defined criteria of criticality</li> </ol> <p><b>Validation:</b>  The data acquired from simulation model is split into training set and test set. The evaluation of the test set can prove the effectiveness of trained model.  It used a case study to validate the estimation of the performance boundary. However, there are no validations on the criticality of the scenarios on the performance boundary.</p> <p><b>Similar paper:</b>  [10] 8431291 uses GPC to identify safety boundary in the parameter space (i.e. collision or not). It proposes a criterion to evaluate the approximation quality of classification results and employs a</p>

					method to accelerate the boundary searching process.
[11]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> the whole system</p> <p>An automated testing algorithm that builds on learnable evolutionary algorithms to achieve the following goals: First, classification models guide the search-based generation of tests faster towards critical test scenarios. Second, search algorithms refine classification models so that the models can accurately characterize critical regions.</p> <p>Aimed for improving effectiveness of the evolutionary search for large and multidimensional input spaces.</p> <p>It can be used in the system testing phase. The purpose is to find critical regions in the simulation environment.</p>	<p>The term ‘test scenario’ is used without clear definition.</p> <p>In a motivation case study, it is described that ‘scenarios capturing various road traffic situations and different pedestrian-to-vehicle and vehicle-to-vehicle interactions’ and ‘vary road-topologies, weather conditions and infrastructures.’</p> <p>Instead of explicitly defining ‘test scenario’, a related term ‘test input space’ (in our term, it refers to the logical scenario or the scenario configuration space) is used, which includes static input variables and dynamic objects. In our understanding, these variables constitute the driving scenario.</p> <p><b>Covered layers:</b> 1,4,5</p>	<p>Critical test scenarios are defined as test scenarios that contain critical behaviors (e.g., hitting a pedestrian with high speed) In this paper, criticality is evaluated on simulation results.</p> <p>Critical regions are defined as the regions of a test input space that are likely to contain most critical test scenarios.</p> <p>Critical behaviors: e.g., The main critical behavior of Automated Emergency Braking (AEB) is extracted from the AEB requirements: “AEB detects a pedestrian in front of the car with a high degree of certainty, but an accident happens where the car hits the pedestrian with a relatively high speed (i.e., more than 30km/h)". This critical behavior refers to any AEB simulation scenario exhibiting this behavior as a critical test scenario of AEB.</p>	<p><b>Input sce.:</b> A given logical scenario <b>Output sce.:</b> critical concrete scenarios</p> <p>Critical scenario identification is formulated as a multi-objective search optimization problem and solved by Non-dominated Sorting Genetic Algorithm version 2 (NS-GAII). A feasible solution is a vector of values to input variables of the ADAS.</p> <p>Decision tree (DT) learning is used to learn a classification model of critical behaviors.</p> <p>NSGAII-DT generates critical test scenarios and critical regions for ADAS.</p> <p>Subsequent search iterations are performed on the critical regions, generating and evolving more critical test scenarios within those regions using genetic operators.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1. A logical scenario</li><li>2. A simulator (PreScan)</li><li>3. Dynamic model of other road users</li><li>4. Definition of critical behaviors (extracted from specifications)</li><li>5. SUT (as black box)</li></ol> <p><b>Validation:</b> A case study to show the exploration and exploitation of their searching method.</p> <p>Assumption: It is argued that for testing at the system level, search-based techniques are best suited.</p> <p>Decision trees is not used to predict whether a given ADAS scenario is critical or not. Instead, the decision trees are used: (1) to better guide the search, and (2) to characterize the critical regions of the ADAS input space.</p> <p><b>Similar papers:</b> [12] use genetic algorithm to detect critical scenarios for emergency braking function. It applies TTC as critical criteria. It also compares the effectiveness of finding critical driving scenarios with random searching.</p>

					[13] use genetic algorithm and simulated annealing respectively to detect critical scenarios for emergency function. It applies TTC as critical criteria. It also compares their performance in test parameter optimization with random searching
[14]	<p>Activity: Instantiation Phase: system verification SOI: decision-making and control system</p> <p>In this article, a generic simulation-based toolchain for the model-in-the-loop identification of critical scenarios is introduced. The proposed methodology allows the identification of critical scenarios with respect to the vehicle development process.</p> <p>The toolchain is generic, but demonstrated using the example of automated highway chauffeur.</p> <p>The investigated automated driving function is an SAE Level 3 highway chauffeur.</p> <p>It can be used in system verification/ testing phase.</p>	<p>The full scope of a logical scenario is not explicitly given in the paper.</p> <p>In this paper, the scope of a scenario also includes disturbances (e.g., faults on the system, performance limitations and unexpected behaviors of other vehicles)</p> <p>How these disturbances are introduced into the simulation is not mentioned in this paper.</p> <p><b>Covered layers:</b> 1,4,6 + possible faults and performance limitations (including the ones caused by extreme environmental condition)</p> <p>The covered layers are judged by the proposed case studies.</p> <p>Usually, the investigated time interval for traffic quality varies from several minutes to hours. In this paper, this time interval is too large and is adjusted to 15 s.</p> <p>The spatial “domain of interest” (DOI) is chosen to be 450 m.</p>	<p>Critical scenarios are defined as scenarios that need to be tested. The criticality of the scenarios is determined by both standard safety metrics and newly developed traffic quality metrics. A scenario is critical if any of safety metric or the overall traffic quality metric exceeds its threshold.</p> <p>Standard safety metrics: 1. time to collision; 2. time to brake. 3. Introduced safety metrics: required deceleration (describes the deceleration of the ego-vehicle needed to generate a collision with 0 m/s)</p> <p>The overall traffic metrics is a weighted sum of:</p> <ol style="list-style-type: none"> <li>1. traffic density (macroscopic description)</li> <li>2. the velocity deviation divided by the velocity mean value of the ego-vehicle as an indication of the microscopic traffic quality</li> <li>3. Close-range interactions. (nanoscopic)</li> <li>4. Speed change of the ego vehicle (individual metric)</li> </ol> <p>The weights are optimized by training data (labeled by experts)</p> <p>Each metric has its own domain of interest (DOI).</p>	<p><b>Input sce.:</b> A given logical scenario <b>Output sce.:</b> critical concrete scenarios</p> <p>The identification process is realized by a coupled simulation framework combining a vehicle dynamics simulation, a traffic simulation and a cooperation simulation.</p> <p>The behavior of other traffic participants, like defensive or aggressive, is considered in the traffic simulation environment.</p> <p>According to our understanding, the parameters in the logical scenario are randomly sampled.</p> <p>For identification and evaluation of critical scenarios, the corresponding thresholds for the criticality classification are specified, including three safety metric values and an overall grade.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. A logical scenario</li> <li>2. Simulators (traffic simulation, vehicle dynamic simulation cooperation simulation)</li> <li>3. Dynamic model of other road users (in the traffic simulator)</li> <li>4. Metrics and thresholds for criticality</li> <li>5. SUT (as black box)</li> <li>6. Expert experience to label the overall traffic quality metric.</li> </ol> <p><b>Validation:</b> A case study is used to show the applicability of the approach.</p> <p>Limitation (assumption of a constant-velocity model): One of the major issues for the dynamic coupling between traffic and vehicle dynamics simulation is the discrepancy of sample time. Therefore, the absent motion of the traffic participants provided by the traffic simulation is predicted with a constant-velocity model.</p>

<p>[15] – [17] and [18]</p>	<p><b>Activity:</b> Formalization (including quantification) &amp; Instantiation  <b>Phase:</b> System verification  <b>SOI:</b> the whole ADAS system</p> <p>This paper presents an automatic test scenario generation method, which is based on combinatorial testing.</p> <p>When generating test scenarios, besides combinatorial coverage, they also maximize the complexity of the generated scenarios. The scenario complexity is defined based on the influence of each scenario factor on the performance of the SOI.</p>	<p>The input logical scenario is described by a predefined hierarchical ontology, which includes (1) environment, e.g., weather, time, road; (2) self-state, e.g., speed and (3) Other traffic participants' state, e.g., Road congestion. The leaf nodes of this ontology are the possible values of the scenario description parameters.</p> <p>A detailed table of these parameters is presented in the papers.</p> <p>The value of each parameter is discretized. A concrete scenario is a vector of the values of all the leaf nodes of the hierarchical ontology.</p> <p><b>Covered layers:</b> 1, 4, 5</p>	<p>The criticality of a scenario is evaluated by its complexity.</p> <p>Each leaf factor in the scenario description ontology has discrete values. Each value has an importance index (a real number) according to its potential influence on the performance of the SOI. The complexity of a concrete scenario is indicated by the sum of the importance indexes of all the values in the vector (i.e., the concrete scenario). The importance indexes are determined by expert knowledge.</p>	<p><b>Input Sce.:</b> a Logical scenario (a pre-defined ontology of scenario description)  <b>Output Sce.:</b> a set of concrete scenarios</p> <p>Process:</p> <ul style="list-style-type: none"> <li>Given the relative importance of the values under each leaf factor, the contribution of each factor or value to the complexity of the whole scenario can be determined by the Analytic Hierarchy Process (AHP).</li> <li>A covering array generation algorithm is proposed to guide the test case generation method to maximize the complexity of the scenarios in the generated covering array.</li> </ul> <p>The generated test scenario is constructed into simulation models.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>A predefined scenario description ontology</li> <li>The relative importance of the values under each leaf factor in the topology</li> <li>Simulator (PreScan)</li> </ul> <p><b>Validation:</b></p> <ul style="list-style-type: none"> <li>By comparing with other combinatorial test case generation methods, the proposed method generates more complex scenarios with acceptable number of total test cases to guarantee combinatorial coverage.</li> <li>Simulations in PreScan are used to show (as a proof of concept) that more complex scenario indicates higher failure rate.</li> </ul>
<p>[19]</p>	<p><b>Activity:</b> Formalization (derive the distributions of parameters) &amp; Instantiation  <b>Phase:</b> system verification  <b>SOI:</b> decision-making and control system</p> <p>The purpose of this method is to analyze naturalistic driving data (NDD) and generate critical test cases for safety evaluation for AD vehicles.</p> <p>The novel idea is to involve risk indices to constrain the sampling process, so as to find the critical</p>	<p>Parameters to sample in the example cut-in maneuver scenario:</p> <ul style="list-style-type: none"> <li>Speed of following/merging vehicle</li> <li>Max acceleration of merging vehicle</li> <li>Lateral/longitudinal distance</li> <li>Lateral speed of merging vehicle</li> </ul> <p>Distributions of these parameters are derived from NDD.</p> <p><b>Scenarios:</b> In the given example (the cut-in maneuver), scenario is the kinematic behaviors of following and merging vehicles incorporating with the risk indices, based on a cut-in maneuver on an expressway.</p> <p><b>Covered layers:</b> Layer 4</p>	<p>The proposed method tries to find common and risky scenarios.</p> <p><b>Criticality definition:</b> In this paper, risk index is used to represent criticality. Minimum TTC is set as risk index in the cut-in maneuver example. The definition of minimum TTC is not explained in the paper.</p> <p>The identified scenarios are <b>Safety critical for generic AD systems</b>.</p> <p>In result analysis and validation part, critical regions are identified. In the cut-in maneuver example, scenarios with short cut-in duration and high lateral (de)accelerations are identified as the critical region.</p>	<p><b>Input Sce.:</b> Logical Scenario  <b>Output Sce.:</b> a set of concrete scenarios</p> <p>The generation of concrete scenarios includes 3 steps:</p> <p><b>Step 1</b> is to build the kinematic model for automated driving function and get the parameter sets.</p> <p><b>Step 2</b> is to generate Gaussian mixture model (GMM) model to analyze the distributions for parameters using naturalistic driving data.</p> <p><b>Step 3</b> is to perform parameter sampling based on (Markov chain Monte Carlo) MCMC method, the sampling can be performed with and without the consideration of risky index.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>Dynamic model of certain use case(scenario)</li> <li>Naturalistic driving data</li> <li>Well-defined risk indices</li> </ul> <p><b>Validation:</b></p> <p>In a cut-in maneuver case study, the proposed method identifies that the criticality is statistically related to the lateral (de)acceleration and the cut-in duration. However, this finding is not further validated.</p>

	regions on the scenario configuration space.		Since the sampling also considers the distributions of the parameters, commonality of the scenario is also part of the criticality definition, since the sampling considers the distributions of the parameters.	When considering the risk index, it will be used as a constraint for the sampling.  Sampling considering the risky index will show the distribution of risky scenarios on the scenario configuration space.	
[20]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> ADAS system</p> <p>This paper proposes a method to identify faulty behavior regions in the scenario parameter space based on surrogate model and stochastic optimization.</p> <p>Surrogate model is used in optimization to reduce computational cost.</p>	<p><b>Covered layer:</b> In the case study scenario, acceleration, speed, moving trajectory and obstacle detection ability of the vehicle are listed but set with fixed number. 2-D position of static obstacle is the only variable in the search space (maybe layer 4).</p>	<p><b>Criticality definition:</b> This paper aims at finding fault behaviours by minimizing cost function through optimization.</p> <p>In the case study, critical scenario is found when a collision (a fault behavior) occurs. The presence of fault behaviour is evaluated through the calculation of cost function and it will stop the searching iteration. Cost function is defined by time to collision (TTC) value between vehicle and static obstacle (if there is no collision) or the collision speed. The scenario will be more severe if the ego vehicle crashes at a higher speed.</p>	<p><b>Input Sce.:</b> Logical Scenario <b>Output Sce.:</b> a set of concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Defining evaluation criterion and executing simulations on parameter set and calculating cost function.</li> <li>2. Building surrogate model. The surrogate model takes the same inputs with simulation model and estimate the result of the cost function as the outputs. Surrogate model is built with Radial Basis Function approximation and it is trained in each iteration.</li> <li>3. Based on the surrogate model, applying stochastic optimization to find most likely global minimum location.</li> <li>4. Generating new parameter set around the global minimum location (the result of the optimization) by zoom-in sampler for a new iteration. The iteration stops when faulty behaviour is found, or the maximum number of iterations is reached.</li> </ol> <p>The search is done in an iterative way. A concrete scenario is used as the input to the simulator. The output of the simulation is the traces of relevant states. The cost function is calculated on the traces.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. The model (can be a simulation model or a real system on HIL/VIL)</li> <li>2. A subset of the parameter space for accelerating searching and the cost function</li> <li>3. The regression surrogate model</li> <li>4. An optimization model to find the local minimum</li> <li>5. Simulator: MATLAB + Simulink</li> </ol> <p><b>Validation:</b> The case study applies 5 different optimization algorithms to evaluate the performance of proposed method. Optimization algorithms applied directly on the simulation models is compared with those on surrogate models. Performance on finding global minimum, computation time and accuracy are compared to prove the effectiveness of the method.</p>
[21], [22],	<p><b>Activity:</b> Instantiation &amp; Assessment <b>Phase:</b> system validation</p>	This paper talks about testing scenarios.	In this paper, the criticality of scenarios is defined based on safety and functionality	<p><b>Critical scenario library generation:</b> <b>Input Sce.:</b> A logical scenario</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. A given logical scenario</li> </ol>

<p>[23], [24]</p>	<p><b>SOI:</b> the whole system</p> <p>The purpose of this work is to generate a library of critical testing scenarios for connected automated vehicles. This method targets to find common and challenging scenarios. Scenarios in the library are critical for most of the AVs.</p> <p>Scenarios within this library will be randomly sampled as an importance sampling method to estimate the accident rate and failure rate. The sampled scenarios will be used as test cases in an augmented reality testing environment, where the ego-vehicle runs in the real world and receives other vehicles' information from a simulated environment. In other words, other vehicles are simulated, and their information is sent to the ego-vehicle through I2V.</p>	<p>The scenario model does not specify the duration of a scenario.</p> <p>When generating critical scenarios, vehicle behaviors are represented by surrogate models, e.g., polynomials.</p> <p><b>Covered layers:</b> The surrogate model only considers layer 4.</p>	<p>(if the ego-vehicle can complete the driving task).</p> <p>The scenario generation process is to find common and challenging scenarios. Challenging level is quantified as mnpETTC (a variance of TTC). The commonality is quantified as the distance (on the scenario configuration space) between the scenario and a high exposure frequency zone (i.e., <math>\Omega</math>) in Naturalistic Driving Data (NDD) (e.g., 95% percentile)</p> <p>For the evaluation with importance sampling phase, the metric (after testing) for safety is the accident rates; and the metric for functionality is the failure rate.</p>	<p><b>Output Sce.:</b> a library of critical concrete scenarios</p> <p>The scenario searching process is formulated as a two-step optimization problem. The decision variables are the parameterized scenario configurations of interest (normally referring to what can be changed in the given logical scenario e.g., the parameters of the dynamic objects). An example in this paper for a cut-in scenario is the cut-in distance and the speed difference with the vehicle in front. The objective function represents the criticality definition. Constraints are used to define the ranges of the parameters.</p> <p>The first step of the optimization tries to find multiple local optimal solutions with a multi-star optimization method (there is no simulation involved). The second step tries to search the neighborhood of the local optimal solutions to find all the scenarios whose criticalities are within a given threshold.</p> <p>If design variables contain profiles (e.g., acceleration profiles in one of the case studies), reinforcement learning is applied.</p> <p>[24] applies an adaptive method to change the importance distribution based the test results.</p>	<p>2. Predefined KPIs (e.g., mnpETTC) 3. A Surrogate model to fast compute the KPI 4. Naturalistic Driving Data (to determine the distributions of the parameters of the logical scenario)</p> <p><b>Validation:</b> They compared the failure rate estimated by their method with the failure rate measured from simulation/real testing.</p> <p>This paper has some arguments on which parameters should be fixed and which should be variable (in this paper, these variables are called design variables)</p> <p><b>Similar paper:</b> [25] uses real field test data and applies importance sampling method to accelerate the testing process and achieve the full coverage test.</p>
<p>[26]</p>	<p><b>Activity:</b> Instantiation &amp; assessment <b>Phase:</b> system validation <b>SOI:</b> decision-making and control system The proposed method in this paper is to validate the failure rate of an ADS via <b>importance sampling</b>. It</p>	<p>In this paper, a scenario is defined as a combination of the actions of the ego vehicle, the static environment (e.g. infrastructure and weather), and the ongoing activity of the dynamic environment (including the other traffic participants) for a certain period of time.</p>	<p>In this paper, criticality is defined based on KPIs and the corresponding thresholds. TTC is used in its example. In the case study, critical scenarios are those whose TTC belongs to the 5% lowest.</p>	<p><b>Input Sce.:</b> Logical Scenario <b>Output Sce.:</b> a set of concrete scenarios</p> <p>The main purpose of this method is to evaluate the failure rate of the ADS. It can also generate critical scenarios for testing.</p> <p>It has the following steps.</p>	<p><b>Required knowledge:</b> 1. A logical scenario 2. Surrogate model based on vehicle dynamics to calculate the indicators (TTC) 3. Metrics for criticality</p>

	can also fasten the speed to find critical scenarios.	<b>Covered layers:</b> 1, 2, 4, 5		<ol style="list-style-type: none"> <li>1. Monte Carlo simulation is used to find a set of critical scenarios (whose TTC belongs to the 5% lowest). The distributions of the scenario parameters are derived from real-life data by kernel density estimation (KDE).</li> <li>2. Approximate the importance sampling density function on the parameter space of the critical scenarios achieved from step 1.</li> <li>3. Use this density function for the importance sampling to get more critical scenarios and to get a more precise estimation of the failure rate.</li> </ol>	<ol style="list-style-type: none"> <li>4. Distribution of scenario parameters from Naturalistic driving database</li> <li>5. Simulator (Simulink)</li> </ol> <p><b>Validation:</b> They use a car-following scenario as an example. The generated test cases are given to a simulator (MATLAB + Simulink) and compare the result between the proposed method (important sampling) and normal Monte Carlo Simulation.</p>
[27]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> decision-making and control system</p> <p>The proposed method characterizes the deer model as the traffic participant, reacting with a startle response when a vehicle is approaching. Ego vehicle behavior is simplified into four rudimentary modes with different reactions after detecting deer. Genetic Algorithm (GA) is applied to optimize parameters in deer motion model, which helps to explore worst case interaction (collision) between deer model and vehicle.</p>	<p>In this paper, a scenario is constructed mainly with 3 phases in deer road crossing event, which are initial acceleration, deceleration and turn and final acceleration. Each phase is modelled with step response of transfer function. Parameters exist in the deer reaction model to describe its trajectory, including deer initial angel, acceleration phase time, time constant in transfer function and maximum speed. The only variable for ego vehicle is the number of 4 different pre-defined driver modes.</p> <p><b>Covered layers:</b> 4</p>	<p>In this paper, criticality is defined with minimum distance between a deer and the ego vehicle. The proposed method in the paper can be applied for generic AD function since predefined driver reaction modes are specified.</p>	<p><b>Input Sce.:</b> Logical Scenario <b>Output Sce.:</b> a set of concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Design deer motion model and define driver reaction modes (e.g. swerve, brake, hybrid and no reaction).</li> <li>2. Perform optimization with Genetic Algorithm to identify most challenging scenarios.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Relevant biological information for deer model setup.</li> <li>2. Defined driver reaction strategies.</li> </ol> <p><b>Validation:</b> Case study shows validity of proposed method and comparison is discussed about different performance with variant vehicle reaction.</p>
[28]	<p><b>Activity:</b> Criticality check <b>Phase:</b> system verification <b>SOI:</b> decision-making and control system</p>	<p>The scenario in this paper is described by the specified term situation and episode. A situation (scene in our terminology) is a single moment with information about all traffic participants, and episode (scenario in our terminology) represents the changes</p>	<p>In this paper, critical scenario is evaluated by criticality index, which calculates critical situations over the episode. Each situation can be evaluated with a criticality value from 0 – 1. In case study, a threshold is set</p>	<p><b>Input Sce.:</b> Logical Scenario <b>Output Sce.:</b> a classifier in situational space</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Design vehicle dynamic model with parameter ranges and distributions.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Vehicle dynamic model</li> <li>2. Real-world data for validation</li> <li>3. Matlab as simulator</li> </ol> <p><b>Validation:</b></p>

	<p>The proposed method designs a classifier to examine critical scenarios in situational space. The classifier is achieved with support vector machine. Real world data in defined situational space is recorded and used for validation.</p>	<p>over time of the underlying situation, which contains trajectory details of all related vehicles. Relevant parameters for vehicle dynamic model include velocities and positions for ego and other vehicles.</p> <p><b>Covered layers:</b> 4</p>	<p>to classify critical scenarios from non-critical ones.</p>	<ol style="list-style-type: none"> <li>2. Conduct sampling twice with different group sizes to generate training dataset and test dataset. Criticality of each scenario is labelled through simulation.</li> <li>3. Train SVM and evaluate the performance.</li> </ol>	<p>This paper analyses performance of the proposed method by processing 24 hours of driving sequences to generate a real-world dataset for validation.</p> <p>This paper applies simulation data for training and real test data for validation. The performance of predictability is validated.</p> <p><b>Similar papers:</b> [29] uses SVM to identify critical scenarios under cut-in situation. THW (time head way) and relative velocity are critical criteria.</p>
[30]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> the whole system (ADAS)</p> <p>This paper proposes a method to generate test scenarios for ADAS function. The proposed framework supports hardware-in-the-loop and test automation. Functional coverage is examined in the method to check how much testing has been completed against defined use case requirements.</p>	<p>In this paper, a scenario is the combination of test scenario and test case. A test scenario refers to the simulation environment, including other traffic participants and weather conditions. And in test case, actions by driver (human maneuver such as stepping on accelerator and braking) is specified. The target function in this paper is ADAS function (AEB in case study), which requires driver operations to trigger function. For this reason, the driver action is quantified with brake pedal value and accelerator pedal value.</p> <p><b>Covered layers:</b> 4, 5 + driver behavior</p>	<p>In this paper, criticality for scenario is defined as if longitudinal collision occurs.</p>	<p><b>Input Sce.:</b> Logical Scenario <b>Output Sce.:</b> a set of concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Perform randomization on scenario space and driver behaviour space. Set up HIL interface and start simulation in an automatic manner to search critical scenarios.</li> <li>2. Check functional coverage metrics. The coverage metrics are calculated and evaluated in Vitaq, the test automation tool. Coverage is achieved by partitioning parameters values in sub-ranges and conducting simulation in each range with required number of tests.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Hardware preparation including ECU with target ADAS function and hardware-in-the-loop test environment</li> <li>2. Tool for test automation and randomization</li> <li>3. Simulator</li> </ol> <p><b>Validation:</b> This paper is in the absence of validation part.</p>
[31]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> decision-making and control system</p> <p>This paper proposes a method to analyze the criticality of a certain</p>	<p>In this paper, a multi-level approach for safety consideration is proposed. The scenarios are divided into frequently occurring scenarios, less frequently occurring scenarios and scenarios beyond the crash boundary. The latent crashes happening in the first two types of scenarios can be</p>	<p>In this paper, criticality is evaluated based on Abbreviated Injury Scale (AIS) methodology. A lookup table is used to match contact velocity in both front-end crash and rear-end crash with injury probability for MAIS 2+. A threshold is set to evaluate whether the resulting risk is</p>	<p><b>Input Sce.:</b> Logical Scenario <b>Output Sce.:</b> a set of concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Build crash model and define logical scenario.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Vehicle dynamic model and simulation environment</li> <li>2. Correlation between simulation results and crash severity level.</li> <li>3. Real traffic measurement for driving behavior derivation</li> </ol>

	<p>virtual-simulated scenario and therefore define crash boundary in the parameter space. The focus of this paper is the criticality check for concrete scenario from a given logical scenario.</p>	<p>handled by ADAS/AD function and they are considered at design stage. The evaluation of scenarios beyond the boundary reveals edge cases and is referred to as ‘beyond-design-basis safety assessment’. In the scenario, vehicle braking behavior is specified and fitted from real traffic measurement. Parameter ranges are specified based on statistical information from real-world measurements.</p> <p><b>Covered layers: 4</b></p>	<p>acceptable or not. Through a full-scale grid search in parameter space, critical scenarios can be found, and crash boundary is identified. Crash boundary separates safe conditions from unsafe ones based on estimation of risk probabilities. Its location is decided by the defined threshold.</p>	<p>2. Setup corresponding relationship between simulation results and injury risk probability.</p> <p>3. Perform a grid search with simulation in parameter space and evaluate injury risk level.</p> <p>4. Obtain crash boundary using threshold of acceptable crash probability.</p>	<p>4. CarMaker</p> <p><b>Validation:</b></p> <p>This paper applies ACC with autonomous overtaking function as a case study. The crash boundaries of ego vehicle and front vehicle are analyzed.</p>
[32]	<p><b>Activity:</b> Instantiation</p> <p><b>Phase:</b> system verification</p> <p><b>SOI:</b> Control part of a Lane keeping system, excluding perception</p> <p>This paper presents an approach to generate test cases and expose safety-critical problems with AsFault. It focuses on the generation of road topology by procedural content generation and create critical scenarios by search-based testing.</p>	<p>In this paper, the scenarios are constructed with ego vehicle equipped with lane keeping functionality (SUT) and road network. Scenarios for testing is derived by combining different road segments and forming road networks with AsFault prototype, which uses genetic algorithm to iteratively refine road networks.</p> <p>Road network generation method in this paper ensures the validity of the generated road (gapless and non-self-intersecting) and specifies properties with regards to length, curvature and intersection locations.</p> <p>Segments of road: defined by back line (starting part of segment) and front line (polyline which defines angle and length of segment).</p> <p>Road network: generated by combination of roads, only feasible when intersection point of central polylines for different road occurs.</p> <p><b>Covered layers: 1</b></p>	<p>In this paper, a scenario is viewed as critical when the distance between the position of the ego-car and the center of the lane was bigger than the half of the lane width.</p>	<p><b>Input Sce.:</b> a logical scenario</p> <p><b>Output Sce.:</b> a set of critical concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Generate road network with procedural content generation method.</li> <li>2. Simulate and evolve road networks by AsFault with various search operators (e.g., road mutation, join crossover and merge crossover) through replacing or combining road segments to generate road network and find critical scenarios for Lane Keeping function. When an invalid road network is found, the combination will be aborted. Constraints are not used because ‘generate and validate’ scheme proves efficiency in this study.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. AsFault as test cases generation platform for road topology creation by procedural content generation.</li> <li>2. Simulator, BeamNG.research</li> <li>3. Search operators to mutate and recombine roads.</li> </ol> <p><b>Validation:</b></p> <p>This paper uses Lane Keeping function as a case study to evaluate the performance of proposed method. Two different control algorithms are implemented, and results are compared.</p>

[33]	<p><b>Activity:</b> Instantiation  <b>Phase:</b> system verification  <b>SOI:</b> not clear, use case is a highway traffic system</p> <p>This paper proposes a method to build templates for formulating fitness functions in order to test AD system and generate worst-case scenarios. The fitness function is used as constraints in searching and can be combined in the application of multi-objective searching for complex scenarios.</p> <p>The derivation of fitness function is the focus of this work, while how to perform searching in a certain scenario type with a given fitness function is not described in detail.</p>	<p>The derivation of a search space is not discussed in this work. In the paper, it is said that OpenScenario or CommonRoad format can be used. The paper takes use of scenario types defined in [34], where scenarios are grouped into 24 types. This paper analyzes 2 types of them as case studies. The use case in the paper, highway traffic system, only covers layer 4.</p> <p>Action of ego vehicle is considered in the scenarios, for example, ego vehicle must do lane change and then follow certain distance constraints with other vehicles. These conditions are formulated into fitness functions.</p> <p><b>Covered layers:</b> 4 (including the intention of the ego vehicle)</p>	<p>In this paper, scenarios are critical when they are near-collision and collision cases. Criticality is checked by fitness functions. This paper proposes several different templates to construct the fitness function for scenario search. Each template is a specification from one perspective. Fitness function is constructed by combining these templates. Fitness functions can be classified into two categories: 1) testing against safe operating envelopes (e.g., the distance between the ego vehicle and another vehicle) and 2) ensuring qualitative test goals (e.g., if an event happens at the right moment). The non-fulfilment of qualitative goal is assigned by the measurement of how far it is away from goal, which can be transformed and viewed as a quantitative way. The combination of fitness functions can be used in single-objective search and multi-objective search for complex scenarios.</p>	<p><b>Input Sce.:</b> a logical scenario of a particular scenario type  <b>Output Sce.:</b> a set of near-collision and collision concrete scenarios of one particular scenario type</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Create templates. They can be used for different primitive qualitative and quantitative test goals.</li> <li>2. Combine templates to generate fitness function so that it can be used for complex scenario searching.</li> <li>3. Search for the violation of test goals with fitness functions in the prescribed search space by simulation. The search engine and simulator are not described in the paper.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Fitness functions</li> <li>2. Optimization tool</li> <li>3. Simulator</li> </ol> <p><b>Validation:</b>  This paper is in the absence of validation part.</p>
[35]	<p><b>Activity:</b> scenario variation and criticality check  <b>Phase:</b> system verification  <b>SOI:</b> decision-making and control system</p> <p>This paper proposes a method to evaluate criticality of a scenario through risk based KPIs and generate new test scenarios by variation through PCA. The generated scenarios are similar to the original ones, with respect to trajectories, maneuver behaviors and risk evaluation results, which help to provide relevant scenarios from naturalistic data and add</p>	<p>In this paper, scenario is defined as a timely series of scenes. Scene is the snapshot in time, containing static and dynamic properties in the environment. JSCEN, proposed scenario description method, is used in the paper to characterize scenarios. It provides information of starting positions and driven traces for all traffic objects, including ego vehicle.</p> <p><b>Covered layers:</b> 4</p>	<p>In this paper, the criticality of a scenario is evaluated by risk based KPI. In the case study, time-to-react (TTR) [36] of every scene is calculated and mapped to risk probability by a weighting function. The risk of a scenario is then represented by the maximal risk value of the scene.</p>	<p><b>Input Sce.:</b> FOT/NDS data of a critical concrete scenario  <b>Output Sce.:</b> a set of critical concrete scenarios around the given scenario</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Extract a set of particular critical scenarios with the same maneuver (e.g., right lane change) and perform feature selection.</li> <li>2. Use principal component analysis (PCA) to segment scenarios into a lower-dimensional component space.</li> <li>3. On the component space (the output of PCA), conduct component variation. (i.e., adding noises to each dimension of the component space)</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. FOT/NDS data of the corresponding critical functional scenario</li> <li>2. JSCEN scenario format</li> <li>3. Simulator to evaluate scenario criticality</li> <li>4. Risk probability mapping to evaluate risk of scenarios based on simulation results. (e.g., weighting TTR values into risk values)</li> </ol> <p><b>Validation:</b>  This paper uses a cut-in and brake scenario as the case study. Risk of NDS data is evaluated and scenario variation is used to generate more</p>

	additional test cases in function assessment process.			<p>4. Perform reverse transformation and concatenate with other scenario segments which are not varied to form a completed scenario.</p> <p>5. Fit the scenario into JSCEN format to generate varied scenarios.</p> <p>6. After the generation of scenarios, perform criticality evaluation with risk based KPI through simulation.</p>	scenarios. The risks of generated scenarios are also evaluated to show their relevance to the original scenarios.
[36]	<p><b>Activity:</b> criticality check (risk assessment) for a given concrete scenario</p> <p><b>Phase:</b> system validation</p> <p><b>SOI:</b> decision-making and control system</p> <p>The paper proposed an approach to measure the criticality of a given driving scenario fitted on the requirements of safety testing: (1) no false negatives (i.e., critical scenarios that are predicted to be non-critical); and (2) less false positives.</p> <p>Multiple possible scene evolutions were simulated via Monte-Carlo simulation together with their probabilities of occurrence.</p> <p>Monte-Carlo simulation here means multiple simulations.</p>	<p>A scenario is a series of scenes connected by time.</p> <p>A scene is defined as an instance in a parameter space, which contains the position, speed, acceleration, heading and heading and turning angle of the ego vehicle and other traffic objects.</p> <p>A concrete scenario configuration is the configuration of the initial scene. (implicit)</p> <p><b>Covered layer: 4</b></p>	<p>This paper proposes a method to estimate risk for a particular scene.</p> <p>The risk of a scenario is determined by the highest scene risk.</p> <p>The risk of a scene is defined based on the severity and the probability of occurrence of a crash.</p> <p>The severity is estimated by time-to react (TTR) of the ego vehicle. The probability is estimated through a Monte-Carlo simulation.</p> <p>TTR describes the time which is left to avoid the collision within the physical constraints of the vehicle. It is the maximum value among time-to-steer (TTS), time to brake (TTB) and time to kickdown (TTK).</p> <p>The calculation of the TTR value itself is done by iterating stepwise through time, calculating paths for full left and right steering, full braking and accelerating and checking if the calculated trajectories avoid the collision.</p>	<p><b>Input Sce.:</b> a concrete scenario</p> <p><b>Output Sce.:</b> the risk of the concrete scenario</p> <p>How the Monte-Carlo simulation is conducted:</p> <ol style="list-style-type: none"> <li>1. The behavior of other traffic objects is modeled with a simple Constant Turn Rate and Acceleration (CTRA) model. The parameter distributions in the CTRA model are achieved from naturalistic driving data.</li> <li>2. The scene under evaluation is used as the initial scene. Its development over time is predicted within a given prediction horizon.</li> <li>3. Within the prediction horizon, the trajectory of the ego vehicle is planned (at the beginning) with its trajectory planning algorithm. For each traffic object, a given number of possible trajectories are predicted with their possibility calculated.</li> <li>4. TTRs are calculated for each trajectory pair (one from the ego vehicle and one from a traffic object).</li> <li>5. A risk value (between 0 and 1) is calculated according to all the TTRs and the probability of the corresponding trajectory of the traffic object.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. CTRA model</li> <li>2. Naturalistic driving data (euroFOT)</li> <li>3. Predefined parameter for the Monte-Carlo simulation: prediction horizon, prediction step, number of predicted trajectories per traffic object, point of no return (if TTR is below this point, the risk will be 1), maximum risk time (if TTR is larger than this, the risk will be 0), weighting slope (a coefficient to correlate RRT and the risk value) and consolidation threshold (this is used when multiple vehicles are involved in the scenario. Detail explanation can be found in the paper).</li> </ol> <p><b>Validation:</b></p> <p>This paper uses a case study to verify the applicability of the method.</p>
[37]	<p><b>Activity:</b> formalization and instantiation</p>	<p>In this paper, scenarios can be described in three levels, which are use case, test</p>	<p>In this paper, the critical criterion, as the optimization objective, is derived and</p>	<p><b>Input Sce.:</b> a use case as functional scenario</p>	<p><b>Required knowledge:</b></p>

	<p><b>Phase:</b> system verification <b>SOI:</b> decision-making and control system</p> <p>This paper presents a method to find multiple critical scenarios in a set of test cases, which are derived from a use case. The method focuses on the improvement of Bayesian optimization algorithm to identify minimal points in multiple regions by iteratively searching.</p>	<p>scenario and test case, corresponding to functional scenario, logical scenario and concrete scenario in our terminology. The logical scenario is derived through the analysis of description of ego vehicle, together with its goals, activities and environment. The three levels of scenarios follow a pyramid structure. A use case can have multiple test scenarios and even more test cases. For example, a use case can be a pedestrian crossing road when ego vehicle reducing its speed. Test scenarios for this use case specify dynamic elements (e.g., pedestrian behavior) and test cases are generated with concrete values (pedestrian speed).</p> <p><b>Covered layers:</b> 4</p>	<p>formulated from Systems Theoretic Process Analysis (STPA). Scenarios are critical if the collision occurs between pedestrian and ego vehicle. The objective function in the case study is to minimize the longitudinal and lateral distance between the car and pedestrians.</p>	<p><b>Output Sce.:</b> a set of critical concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Use STPA to identify test scenarios. With the application of STPA, critical criteria and searching objective are also obtained.</li> <li>2. Perform Bayesian optimization to find minimal points. Search the minimum in the space and eliminate its local region afterwards with a predefined size. Each dimension in the space will then be divided into two new regions as upper and lower search region.</li> <li>3. Continue the process iteratively until all regions containing a minimum are identified.</li> </ol>	<ol style="list-style-type: none"> <li>1. Use cases for derivation of scenarios</li> <li>2. Carmaker</li> <li>3. Optimization tool</li> </ol> <p><b>Validation:</b> This paper applies the proposed method on a SAE Level-4 Low Speed Automated Driving system as a case study to evaluate the performance. Multiple test cases are identified, which validates the applicability of the method.</p>
[38]	<p><b>Activity:</b> Criticality check <b>Phase:</b> design and calibration <b>SOI:</b> decision-making and control system</p> <p>This paper proposes a method to estimate collision probability between a vehicle and an obstacle in a scenario. Collision probability is calculated based on existing trajectories of the ego vehicle and the obstacle. The results of collision probability will then be applied in the calibration of a controller.</p>	<p>Scenario contains trajectories and bounding boxes of the ego vehicle and the obstacle. Trajectory includes midpoint location of the rear end and orientation.</p> <p><b>Covered layers:</b> 4 (including the possible behaviors of the ego vehicle)</p>	<p>In this paper, criticality is defined as the estimated probability of collision for a trajectory. With uncertainties in environment, there will always be a possibility of a collision. For each scene, Monte Carlo simulation is used with probability density function of the possible trajectories of the ego vehicle and the obstacle to calculate if they overlap in every possible situation. In the end, collision probability is represented with its value at each time step.</p>	<p><b>Input Sce.:</b> a concrete scenario, size of vehicles, PDFs of waypoints of the trajectories <b>Output Sce.:</b> estimation of collision probability at each time step</p> <p><b>Implementation:</b> The method calculates collision probability by applying Monte Carlo simulation with position probability density functions of ego vehicle and obstacle in each simulation step, in order to check the proportion of overlapping. It will be evaluated iteratively with different parameter/threshold values and used in the application of calibration.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. probability density functions of position for ego vehicle and obstacle</li> </ol> <p><b>Validation:</b> This paper applies Lane Change Assistance System as a case study to evaluate the performance of the proposed method. Two different thresholds are set to demonstrate the calibration process. Their resulting collision probabilities are compared.</p>
[39]	<p><b>Activity:</b> Criticality check <b>Phase:</b> system verification <b>SOI:</b> decision-making and control system</p>	<p>The scenario is built up with system dynamic model of ego vehicle and other traffic participants (e.g., static obstacle and other vehicles). System dynamic model specifies</p>	<p>In this paper, criticality is defined by criticality metrics, which is formulated by objective function in MPC. By optimizing the trajectory in a scenario, an optimal trajectory is reached. The criticality is</p>	<p><b>Input Sce.:</b> a logical scenario (trajectory to be optimized) <b>Output Sce.:</b> criticality of the scenario</p> <p><b>Implementation:</b></p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. a logical scenario</li> <li>2. optimization toolbox in MATLAB</li> <li>3. a specific AD system</li> </ol>

	<p>This paper presents a method to assess a scenario with criticality metrics. The metric is calculated based on optimization of a defined criticality in MPC application for trajectory control. Four scenarios for highway traffic are evaluated and a parameter variation is conducted to analyze effects of parameters in the system dynamic model.</p>	<p>the position, velocity, acceleration, and yaw angle of the vehicle.</p> <p><b>Covered layers:</b> 4</p>	<p>denoted by the objective function value of the optimal trajectory.</p>	<ol style="list-style-type: none"> <li>1. Describe system dynamics with state space model. Define objective function and constraints. The objective function is regarded as a criticality metric.</li> <li>2. Define test scenarios.</li> <li>3. Execute optimization with MPC to optimize trajectories for scenarios and estimate criticalities.</li> </ol>	<p>4. criticality metrics</p> <p><b>Validation:</b> This paper applies four scenarios that are typical for highway traffic to evaluate their criticality with criticality metrics</p>
[40]	<p><b>Activity:</b> Formalization and instantiation <b>Phase:</b> system verification <b>SOI:</b> The act component (control algorithm)</p> <p>This paper presents an approach to generate test cases through search-based testing (SBT) in AD control application. Three industrial use cases, ACC, lane-keeping and steering control, are described in the paper, where the steps of specification analysis and critical scenario generation are studied. This paper also presents lessons learnt from the successful applications of SBT.</p>	<p>In the paper, scenario is initially described in a linguistic way with requirements. Scenario description is translated to test setup and specification is translated into system temporal logic in falsification tools (Breach and S-TaLiRo). For different scenarios, included parameters are in different aspects. In general, this paper considers vehicle parameters (e.g., velocity and distance), environment parameters (e.g., velocity of wind affecting the vehicle) and road parameters (e.g., slope and curvature).</p> <p><b>Covered layers:</b> 1, 4, 5</p>	<p>In this paper, a scenario is critical if it violates requirements from STL specification. Regarding case studies presented in the paper, criticality is identified with too close distance between vehicles.</p>	<p><b>Input Sce.:</b> a scenario description <b>Output Sce.:</b> a set of critical concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Analyse scenario requirements and build STL for specifications.</li> <li>2. Select parameters.</li> <li>3. Perform SBT with Bayesian optimization and detect critical scenarios.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. scenario description (functional)</li> <li>2. Specifications for criticality definition</li> <li>3. simulation tools (MATLAB)</li> </ol> <p><b>Validation:</b> This paper applies three use cases in an industrial AD setting to demonstrate the application of SBT.</p>
[41]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> Decision-making and control system (AD general)</p> <p>This paper proposes a rule-based algorithm to detect critical scenarios in the parameter space. The algorithm will select the driving behavior of other vehicles to</p>	<p>In the paper, scenario is defined with behaviors of the ego vehicle and other vehicles. The go point of the ego vehicle is specified to give the final target. The initial position, speed, acceleration and lane change behavior are specified and discretized. At most 5 vehicles can drive on the road.</p> <p><b>Covered layers:</b> 4</p>	<p>In this paper, a scenario is critical if a collision occurs between the ego vehicle and other vehicles.</p>	<p><b>Input Sce.:</b> a logical scenario <b>Output Sce.:</b> a set of critical concrete scenarios</p> <p><b>Implementation:</b> The method first gives parameter combination of ego vehicle and other vehicles on the road in a scenario, which defines potential paths of them. Exhaustive search algorithm tries to find cases when</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. simulation models</li> <li>2. AD simulator</li> </ol> <p><b>Validation:</b> This paper applies two case studies, high-way lane change and intersection. Collision scenarios are generated in the case studies.</p>

	prompt a collision between the ego vehicle and other vehicles. The method improves the efficiency in exhaustive search for test cases evaluation.			other vehicles drive in the path of the ego vehicle in a simulator. This is achieved with an iterative manner by firstly including one other vehicle in the scenario and iteratively adding more vehicles (up to 5). Other vehicles try to prevent ego vehicle to reach the goal and collide with the ego vehicle. Through short-distance simulations, uncritical scenarios are excluded and critical scenarios are detected.	
[42]	<p><b>Activity:</b> Instantiation and criticality check  <b>Phase:</b> system verification  <b>SOI:</b> control algorithms (ACC and AEB)</p> <p>This paper proposes a method to search a more critical scenario with a sequential learning approach based on kriging models. The method provides a way to approximate autonomous vehicle behaviors and thus can reduce on-track experimentation for evaluation.</p>	<p>In the paper, the scenario is limited into lane change situation. A scenario is constituted with three variables, namely 1) the frontal vehicle's velocity, 2) range between the frontal vehicle and the ego vehicle, and 3) time to collision.</p> <p><b>Covered layers:</b> 4</p>	<p>In this paper, the criticality of a scenario is estimated by the value of time to collision and distance between vehicles.</p>	<p><b>Input Sce.:</b> a logical scenario  <b>Output Sce.:</b> a set of concrete scenarios with higher levels of criticality</p> <p><b>Implementation:</b>  The method first utilizes some scenarios as samples and construct an initial kriging model. Then it searches a new scenario point by maximizing the gradient based on the distance of result estimations from the old and new scenario points. It works in an iterative way and the kriging model is updated with new points until the accuracy is satisfied.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Vehicle dynamic model</li> <li>2. Initial sampling points</li> <li>3. A small number of concrete scenario samples as initial observations</li> <li>4. Simulator</li> <li>5. A black-box controller</li> <li>6. SPMD database</li> </ol> <p><b>Validation:</b>  This paper applies the case study in a lane change scenario. The method shows the applicability of searching for the next best scenario with the accuracy of conflict probability evaluation.</p>
[43]	<p><b>Activity:</b> Instantiation  <b>Phase:</b> system validation  <b>SOI:</b> control algorithms (ACC and AEB)</p> <p>This paper proposes an acceleration evaluation method in test case generation. The aim of the study is to provide a more efficient sampling method to discover critical scenarios. The test cases are checked to ensure the</p>	<p>In the paper, the target scenario is the pedestrian-vehicle interaction at a crosswalk. A scenario is described by parameters of pedestrian walking speed, vehicle speed and longitudinal distance. The dataset is built with videos from cameras fixed on the roadside. The videos are processed and crossing events are collected. Parameters are fitted using a truncated Gaussian mixture model (TGMM).</p> <p><b>Covered layers:</b> 4</p>	<p>In this paper, a scenario is critical if a crash happens between the ego vehicle and the pedestrian. Risk is classified into different RLSs based on the intensity of actions required to avoid a collision (e.g., required braking acceleration). The safety performance is evaluated with the crash rate in the test.</p>	<p><b>Input Sce.:</b> raw videos from cameras  <b>Output Sce.:</b> a set of critical concrete scenarios under different RLSs</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Process video files and extract pedestrian-crossing events.</li> <li>2. Build a pedestrian-crossing model and fit with the TGMM models.</li> <li>3. Generate initial conditions for testing and classified with different RLSs.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Videos as raw data and processing tools (i.e., object detection, tracking, calibration, and motion estimation)</li> <li>2. Model Parametric Toolbox 3 (MPT3) to calculate backward reachable tube (BRT) for initial conditions generations.</li> <li>3. AD simulator</li> <li>4. Control algorithms</li> </ol>

	feasibility by reachability analysis and assigned with a defined degree of risk by Risk Level Sets (RLSs).			4. Perform importance sampling in each RLS to estimate the failure rate. Examine on the simulator to check the crash rate.	<b>Validation:</b> This paper applies the case study where the ego vehicle is equipped with AEB and ACC. Test samples are generated from both naturalistic distributions and importance sampling distributions. Results from both methods are compared to validate the fast crash rate convergence by the proposed method.
[44]	<p><b>Activity:</b> Instantiation <b>Phase:</b> system verification <b>SOI:</b> control algorithms (ACC and AEB)</p> <p>This paper proposes an acceleration evaluation method for autonomous vehicles with importance sampling approach. The results can be used to generate motions of vehicles and further applied in test case generation. The Cross Entropy method is used to guide the choice of importance sampling distribution.</p>	<p>In the paper, the target scenario is the lane change scenario. A lane change statistical model for importance sampling is developed with the data from Safety Pilot Model Deployment (SPMD) database. The scenario is described with initial states of vehicles, including distance between vehicles, ego vehicle's velocity and relative velocity.</p> <p><b>Covered layers:</b> 4</p>	<p>In this paper, three kinds of events, conflict, crash and injury, are analyzed respectively in the case study for the evaluation of criticality. A scenario is viewed as critical if the events occur in the simulation.</p>	<p><b>Input Sce.:</b> a logical scenario including the driver model and the vehicle model <b>Output Sce.:</b> a set of concrete scenarios</p> <p><b>Implementation:</b></p> <ol style="list-style-type: none"> <li>1. Identify lane change events from SPMD database and develop a lane change statistical model.</li> <li>2. Use importance sampling with Cross Entropy method to generate critical scenarios.</li> </ol> <p>Design an autonomous vehicle model and perform simulation to verify the generated scenarios.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. N-FOT data (SPMD database)</li> <li>2. AD simulator</li> <li>3. Control algorithms, AEB and ACC</li> </ol> <p><b>Validation:</b> This paper shows a lane change scenario case study where the ego vehicle is equipped with AEB and ACC. Results show the fast convergence of criticality detection by the proposed method.</p>
[45], [46]	<p><b>Activity:</b> Refinement of the logical scenario. <b>Phase:</b> Function verification <b>SOI:</b> Motion planning algorithm <b>ODD:</b> structured road</p> <p>The paper proposes a method to modify a logical scenario. The purpose is to reduce the range of the relevant parameters to the region that contains critical concrete scenarios.</p>	<p>The scenario is defined as the tuple: Initial state of the ego vehicle, trajectories of other traffic participants, and lanes (number and shapes).</p> <p>A scenario should have a fixed time interval.</p> <p>All the other vehicles are assumed to stay in the center of its lane with a fixed longitudinal acceleration. Therefore, the longitudinal positions of one trajectory are determined by the vehicle's initial position, initial speed,</p>	<p>A scenario is considered critical if the solution space is small enough.</p> <p>The solution space is not calculated by the ego vehicle's trajectory planning algorithm, which may consider real-time performance. It is calculated by a generic exhaustive search method. Therefore, the identified scenarios are critical for most of the motion planning algorithms.</p>	<p><b>Input Sce.:</b> a logical scenario with a specific parametrization. <b>Output Sce.:</b> a "more critical" logical scenario</p> <p><b>Implementation:</b> They define the criticality of the scenario based on the solution space that the ego vehicle has.</p> <p>To find a "more critical" logical scenario, they define an optimization algorithm where the objective is to minimize the solution space of</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Input logical scenarios</li> <li>2. A way to calculate the solution space (Drivable area).</li> <li>3. Dataset (NGSIM US 101 dataset)</li> </ol> <p><b>Validation:</b> They generate "critical" logical scenarios from initially uncritical logical scenarios. But they do not run any simulation to see if their planning algorithms are capable of planning properly.</p>

	<p>They do this by calculating the solution space of a scenario (They call it drivable area). This solution space depends on physical capabilities of the vehicle, and the idea is that the scenario will be more critical if the solution space is smaller.</p> <p>After this process, the motion planning algorithm would be tested by running on these scenarios and having to find a viable solution.</p>	<p>and its longitudinal acceleration. The lateral positions are the center points of the lane.</p> <p>The road network is modeled by “lanelets”, which are atomic, interconnected, and drivable road segments. They are defined by their left and right bound, where each bound is represented by an array of points (a polyline)</p> <p><b>Covered layer:</b> 1 and 4</p>		<p>the ego vehicle (They call it drivable area) and the design variables are the ranges (i.e., the upper and lower bounds) of the relevant parameters (in the case study, they are the initial speeds, initial positions and the accelerations of the other vehicles. These parameters determine the trajectories of other vehicles.)</p> <p>Then, they use evolutionary algorithms to solve this optimization problem. The output is a logical scenario with smaller parameter ranges.</p>	<p>Publication [45] is the newer version. [46] only allows the generation of simple scenarios on straight, non-intersecting roads and only realized translation of other traffic participants.</p>
--	---	--	--	---	--

Table 4. Exploring logical scenarios with parameter trajectories

#	Purpose	Scenario definition	Criticality definition& Surrogate Measure	Solution	Validation & other key observations
[47]	<p><b>Activity:</b> Instantiation  <b>Phase:</b> Function verification  <b>SOI:</b> Adaptive cruise control (ACC)  Level of Automation: L2 (It only considers the car following scenario with longitudinal control)  <b>ODD:</b> car-following within a lane</p> <p>The paper proposes a method to falsify ACC systems by generating motions of the leading vehicle such that the ACC under test causes a rear-end collision.</p> <p>They use Rapidly-exploring random trees (RRT) and they propose a forward search and a backward search. They implement the forward search as known in literature, with addition of the definition of unsafe states (When the distance is not enough to avoid an accident in case the leading vehicle performs an emergency brake).</p> <p>For the backward search they start in a randomly selected unsafe state and they generate random inputs for the ACC vehicle to try to get the vehicle in the previous state. Then, they simulate forward to ensure getting into an unsafe state again.</p>	<p><b>Covered layer:</b> 4 (only forward movement of one vehicle in front)</p> <p>They only consider the ego vehicle following another vehicle. The only layer that is covered is layer 4, but only because they define the motion of the vehicle in front. The motion of the ego vehicle and the vehicle in front is in one dimension.</p> <p>The car-following scenario is modeled as the motion of the leading vehicle (i.e., its <b>jerk profile</b>) and the initial state of the ego vehicle (ACC-equipped vehicle).</p> <p>Each scene contains the positions and speeds of the ego vehicle and the leading vehicle.</p> <p>The output of the ACC controller is the acceleration of the ego vehicle. The behavior (the state ([position, velocity]) at the next time step) of the ego vehicle can be calculated based on the motion of the leading vehicle, the ACC function and a simple one-dimensional vehicle dynamic model.</p> <p>The motion of the leading vehicle is determined by its initial states and its jerk profile.</p>	<p>They only consider the car-following scenario.</p> <p>A scenario is considered critical if it contains motions for the leading vehicle such that the ego vehicle causes a rear-end collision.</p> <p>Therefore, the proposed method tries to find system-specific critical scenarios.</p> <p>A critical scenario starts from a safe initial state (or scene) and ends up with a collision state.</p> <p>A state contains the positions and speeds of both vehicles at a particular time step.</p> <p>Criticality of a scene:  Each scene (i.e., a state) can be classified as:</p> <ol style="list-style-type: none"> <li>1) Safe: if the distance between the ego vehicle and the leading vehicle is at least the minimum safety distance to avoid a collision if the leading vehicle performs an emergency brake.</li> <li>2) Unsafe: If the distance is not enough to avoid an accident if case the leading vehicle performs an emergency brake.</li> <li>3) Colliding: If the distance between the two vehicles is smaller than 0.</li> </ol>	<p>They use rapidly-exploring random trees (RRT) to generate motions for a leading vehicle such that the ACC under test causes a rear-end collision.</p> <p><b>Input Sce.:</b> a logical scenario  <b>Output Sce.:</b> a set of critical concrete scenarios</p> <p>They use two methods to build the RRT:</p> <p><b>Forward search:</b> They start with a randomly generated safe state. They calculate the next state based on the ACC control law and the one-dimensional vehicle dynamic model. At the same time, they randomly generate the behavior of the leading vehicle (acceleration and jerk). Once they get to an Unsafe state, they make the leading vehicle perform an emergency brake until they get a colliding state.</p> <p><b>Backward search:</b> They start at a randomly generated unsafe state, and they try to search backwards in time until they find a safe state. They cannot simulate the ACC vehicle backwards in time since they cannot compute the inverse of the ACC control law. For this reason, they generate random inputs for the ACC vehicle to obtain the state of the ACC at the previous state. They need to ensure that the ACC system drives the ACC vehicle into unsafe states again.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- Knowledge about Rapidly-exploring random tress</li> <li>- Dynamic models of both the ego vehicle and the leading vehicle</li> <li>- The control algorithm of the ACC (as a black box)</li> </ul> <p><b>Validation:</b> They used their method to successfully falsify the safety of state-of-the-art ACC systems (They were able to generate a motion for the leading vehicle such that the ACC under test causes a rear-end collision). They also compared the results to that of existing approaches.</p>

		Therefore, a concrete scenario consists of the initial states of both vehicles and the jerk profile of the leading vehicle.			
[48]	<p><b>Activity:</b> Refinement of a concrete scenario or (Instantiation + criticality check) <b>Phase:</b> system verification <b>SOI:</b> Motion controller</p> <p>This paper applies functional gradient descent on falsification. It can find a more critical concrete scenario around a given concrete scenario. It can also search within a given logical scenario to find critical concrete scenarios.</p> <p>The approach is based on multi-fidelity optimization technologies to save computation power. A low-fidelity dynamic model is used when computing the gradient.</p> <p>Compared to simulated annealing, which takes the system as a black box, the proposed method can provide a better performance in critical scenarios searching.</p>	<p><b>Covered layers:</b> Layer 4</p> <p>A concrete scenario may include Position, velocity (profile), acceleration/deceleration (profile) of the other vehicle and the ego vehicle.</p> <p><b>Scene &amp; Scenario:</b> Scene is defined by the state vector at each simulation time. After simulation, a scenario is represented as the trajectories of the states of both dummy vehicle and ego vehicle.</p> <p>Before simulation, a logical scenario includes the kinematic models and the initial states of both the ego vehicle and the other vehicle(s). A concrete scenario instantiates a logical scenario by fixing the variables e.g., the initial states and the parameters of the kinematic models.</p> <p>A Scenario should have a fixed duration, which is set 200s in the “stop and go” case study.</p> <p>The case study evaluates the performance of a Full Range Adaptive Cruise Control (FRACC) controller using maximum absolute jerk as a critical metrics. The variable of this logical scenario is the acceleration profile of the other vehicle. All the other parameters are fixed.</p>	<p><b>Criticality definition:</b> For the proposed method, the criticality of a scenario is determined by the worst scene criticality. The criticality of a scene is defined by relevant KPIs.</p> <p>The proposed method can find <b>Critical scenarios for a particular system.</b></p> <p>In the case study, Maximum absolute jerk (MAJ) is used to evaluate the comfort performance of the motion controller. Jerk is calculated for each scene, among which, the worst one is chosen as the MAJ.</p>	<p><b>Input:</b> Concrete scenario <b>Output:</b> Critical concrete scenario</p> <p><b>Implementation:</b> The method applies functional gradient descent to search critical scenarios. The objective function contains the relevant KPIs, which are calculated based on the initial states and the kinematic models.</p> <p>KPIs are calculated for each scene. When calculating the gradient at each iteration, the critical scene (i.e., the scene with the worst KPI) needs to be found first. The gradient is calculated on the critical scene with low-fidelity kinematic models.</p> <p>According to our understanding, the approach assumes that the critical time will converge during the gradient decent. In other words, when refining a concrete scenario, it normally finds a more critical scenario with similar critical time.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"><li>● A low-fidelity model for the implementation of functional gradient descent calculation.</li><li>● A high-fidelity model for the evaluation of system performance with certain input parameters.</li><li>● The algorithm to calculate functional gradient descent and update input signal.</li><li>● A given logical scenario for searching or a given concrete scenario for refinement.</li><li>● The control algorithm of the motion controller.</li></ul> <p><b>Validation:</b> The case study demonstrated in the paper proves the effectiveness of the method. In the case study, the results generated from functional gradient descent approach outperform that from simulated annealing in terms of worst-case performance value. Experiments generated from gradient descent has better average performance improvement than simulated annealing.</p>

		To save computation power, a low-fidelity dynamic model is also needed.			
[49]	<p><b>Activity:</b> Instantiation <b>Phase:</b> Function verification <b>SOI:</b> control policy</p> <p>They propose an adversarial testing methodology in which they train adversarial agents to demonstrate flaws in the behavior of the ego agent.</p> <p>The purpose of the adversarial agents is to force the ego vehicle to break its rules. The adversarial agents are constrained to follow the traffic rules. The rules for both the ego vehicle and the adversarial agents are described by Signal Temporal Logic.</p> <p>As pointed out in the paper, the proposed method can also be used to falsify the whole system, since the environmental conditions (e.g., weather) can also be modeled as an agent with changing states.</p>	<p><b>Covered layer:</b> 1, 2 and 4 (They also mention 5 but they do not show it in the examples)</p> <p>The variables can cover layer 2, 4 and 5</p> <p><b>1-Street:</b> The simulations show scenarios in CARLA, but the definition of the environment is not included in the scenario. <b>2-Traffic infrastructure:</b> One of the examples involves a traffic light changing the state. <b>4-Movable objects:</b> Some of the states of other vehicles around the ego vehicle is known (some states are hidden) <b>5-Environment conditions:</b> They mention that weather patterns can be modeled as being controlled by agents, but that it not shown in the examples.</p> <p>A scenario consists of a collection of agents together with rulebooks that specify constraints on their behaviors (together with the rulebook for the ego vehicle). Everything in the scenario with changing states can be modeled as an agent. Therefore, no environmental model is needed.</p> <p>The variables of a logical scenario are the potential states of the agent and actions that the agents can make at each time step, i.e., <b>the state and action profiles</b> of all the agents of interest (other than the ego vehicle).</p>	<p>A critical scenario refers to the one where the ego vehicle has to break its rule due to the legal behaviors of other agents.</p> <p>The rules for the adversarial agents are prioritized. Some of the rules of the adversarial agents are allowed to be broken when searching for action profiles. If necessary, rules with the lowest priorities will be broken first.</p> <p>During the searching, the objective function (to be maximized) is a weighted sum of a set of Boolean variables, each of which refers to the satisfaction of a rule. The ones for the rules of the ego vehicle should have negative weights. The others should have positive weights. The absolute values of the weights represent the priorities of the rules.</p>	<p>Input: A logical scenario Output: concrete critical scenarios</p> <p>Deep Q-learning (a reinforcement learning method) is used to learn the behaviors of the adversarial agents. At each time step, it updates a q-function (a look-up table). Its inputs are the current states of the agent, the observed states of the other agents and one action that the agent will take. Its output is the quality of that action. The q-function is trained based on all the previous data and the objective function mentioned before. If the action improves the objective function, the quality value of that action under those states will be increased.</p> <p>During simulation, the actions with the highest quality value will be chosen. Their method consists of training adversarial agents to challenge the ego vehicle (i.e., to force the ego vehicle to break its rules).</p> <p>Each simulation has a fixed duration. The q-function is trained by many simulations (episode). For each simulation, the initial states are randomly selected.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"><li>• A logical scenario (agents of interest)</li><li>• Rulebooks</li><li>• The dynamic model (the relating between the states and actions) of each agent</li><li>• A simulator (CARLA)</li></ul> <p><b>Validation:</b> The proposed method has been verified in three case studies: following an adversarial agent on a single-lane freeway, driving on the freeway when an adversarial agent performs a cut-in maneuver, coming to an intersection with a yellow light behind an adversary.</p> <p><b>Question:</b> The objective function is calculated based on all the previous states, which are determined by all the previous actions. Why does the update process only update the quality of the current action? I think it should update the quality values of all the previous actions.</p>

[50] [51] [52] [53]	<p><b>Activity:</b> Refinement of concrete scenarios and simulation</p> <p><b>Phase:</b> System Validation</p> <p><b>SOI:</b> Planning and control</p> <p>Adaptive Stress Testing (AST), which was first applied in the literature to test an aircraft collision avoidance system, formulates the scenario space as a Markov decision process (MDP). The original method uses Monte Carlo tree search (MCTS) to search for the most likely failure condition.</p> <p>The contribution in [50] is:</p> <ol style="list-style-type: none"> <li>To extend the AST methodology to use Deep Reinforcement Learning (DRL) as the solver technique.</li> <li>To apply AST to a set of autonomous vehicle scenarios.</li> </ol> <p>The contribution of [51] and [52] is to improve the reward function. They propose:</p> <ol style="list-style-type: none"> <li>A different reward function for the ATS using RSS to find scenarios where the ego vehicle performs unproper actions.</li> <li>A trajectory dissimilarity reward to promote the discovery of highly diverse failure scenarios.</li> </ol> <p>In [53], the authors extend the work developed in [50] by replacing the original Multi-layer perceptron (MLP) network with a Recurrent neural network (RNN).</p>	<p><b>Covered layer:</b> 1, 4 and performance limitation of the ego vehicle (i.e., the sensor noise).</p> <p>The variables of the logical scenario only cover layer 4 and the noises on the detected speeds and positions of other agents.</p> <p>A scenario is described by the (perceived) state profiles of all the relevant agents (ground truth states + noises).</p> <p>In their case study, there is a road with a crosswalk and a pedestrian walking on the crosswalk as the agent.</p> <p><b>Layer 1:</b> The road is contained in the model, and they also model a crosswalk</p> <p><b>Layer 4:</b> The pedestrian is a movable object</p> <p><b>Performance limitation:</b> the noises</p>	<p>In [50], a scenario is considered critical if a collision occurs, regardless of how the collision occurs. For this reason, in some of the results obtained, the pedestrian is considered responsible after analyzing the scenario. The algorithm is penalized for not finding a collision. This penalty also depends on the distance between the car and the pedestrian.</p> <p>In [51], a scenario is considered critical if it reaches a set of predefined critical states e.g., getting too close to another agent or colliding with it. In this work, the proposed method tries to find such critical scenarios with as many improper actions of the ego vehicle as possible. The improper actions are defined according to RSS.</p> <p>In [53], a scenario is considered critical if a collision occurs. In this study, all the transitions are deterministic with respect to the system under test (SUT). For this reason, the scenario is defined by the starting conditions and the sequence of actions taken by the SUT.</p>	<p><b>Input:</b> A non-critical scenario</p> <p><b>Output:</b> A critical scenario</p> <p>They use a variation of Adaptive Stress Testing (AST).</p> <p>AST formulates the scenario space as a Markov decision process (MDP) and is usually solved using a Monte Carlo Tree Search (MCTS) or a Trust Region Policy Optimization (TRPO).</p> <p>In [50], they propose to use a Multi-layer perceptron (MLP) as solver to find the failures. They compare the results obtained against a Monte Carlo Tree Search (MCTS) and Trust Region Policy Optimization (TRPO) because both have been shown to successfully find failures when combined with AST, showing that they can find failures more efficiently.</p> <p>The main contribution of [51] is on the improved reward function, which contains three parts:</p> <ol style="list-style-type: none"> <li>(1) The identified solutions should divers as much as possible.</li> <li>(2) The identified solutions should force the ego vehicle to make improper actions as many as possible.</li> <li>(3) The sampled actions of the agents at each time step should not diverge too much from their expectations.</li> </ol> <p>In [53], the authors propose two implementations:</p> <ul style="list-style-type: none"> <li>- Discrete recurrent deep reinforcement learning (DRDRL): The RNN implementation only requires the previous state of the network. This solver can only be run from a single initial condition.</li> </ul>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- RSS (Only in [51])</li> <li>- Simulator</li> </ul> <p><b>Validation:</b></p> <p>In [50], they apply AST with DRL on a scenario involving an autonomous vehicle and pedestrians and show that they are able to find scenarios that lead to a collision.</p> <p>In [51], they apply AST with an augmented reward on a scenario involving an autonomous vehicle and pedestrians and show that they are able to find more improper response when compared with an existing AST setup.</p> <p>In [53], they apply AST with DRL on a scenario involving an autonomous vehicle and pedestrians and show that they are able to find scenarios that lead to a collision. Also, they compare the results between the two architectures proposed, showing that the generalized architecture (GRDRL) outperforms the discrete architecture (DRDRL)</p>
------------------------------	---	---	--	--	---

	<p>The reason for this is that the previous implementation required the current state of the simulation as input for the network, which imposes many limitations for the implementation with real complex simulators.</p> <p>To solve this, the authors propose to define the scenario by the starting conditions and make all the transitions deterministic with respect to the system under test (SUT). This way, the solver does not need to get the state of the simulation at every step, allowing the simulation to be considered as a black box.</p>			<p>- Generalized recurrent deep reinforcement learning (GRDRL): The RNN implementation requires the initial state of the simulation as an input and the previous state of the network. The hypothesis is that by accepting the initial state as an input, the network can learn a policy that generalizes to the entire space of initial conditions. During the training, each rollout starts from a randomly sampled initial condition.</p>	
[54]	<p><b>Activity:</b> Instantiation  <b>Phase:</b> System verification  <b>SOI:</b> Planning and control</p> <p>The purpose of the study is to generate test cases in the boundary where the autonomous vehicle can no longer avoid a collision (almost-avoidable collisions or near-misses).</p> <p>They propose their custom version of RRT with a custom cost function. This cost function promotes collisions that are almost avoidable.</p>	<p><b>Covered layer: 4</b></p> <p>The street layer is represented when sampling target path segments. The example used is for a straight road, but the authors mention that a coordinate transformation can be applied for sampling from curved roads.</p> <p>Perception result is derived from a simulator and is assumed to be deterministic.</p> <p>The movement of the other movable objects is decided by the RRT algorithm to find the most critical scenarios.</p> <p>It searches for the initial condition and the control outputs of the other vehicles at each time step.</p>	<p>In this study, they are only interested in scenarios in the boundary where a collision is almost avoidable.</p> <p>They do not take into account what actor is responsible for the collision, only if a collision occurs.</p>	<p><b>Input:</b> A logical scenario  <b>Output:</b> A critical scenario</p> <p>The purpose of the study is to generate test cases in the boundary where the autonomous vehicle can no longer avoid a collision (almost-avoidable collisions or near-misses).</p> <p>They propose their version of RRT with a custom cost function. This cost function promotes collisions that are almost avoidable.</p> <ul style="list-style-type: none"> <li>- A collision between an Ego vehicle and an agent that could have been avoided with a minor change in the control applied or agent trajectories</li> <li>- An almost-collision (near-collision) which could have resulted in a collision with a minor change in the control applied or agent trajectories.</li> </ul> <p>They estimate how avoidable was the collision by using:</p> <ul style="list-style-type: none"> <li>- Ratio of the collision surface</li> <li>- Collision speed</li> <li>- Minimum time to collision</li> </ul>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- Knowledge about the simulator</li> <li>- Knowledge about the kinematic models</li> </ul> <p><b>Validation:</b> They have applied their method to a case study with 4 agents and one ego vehicle, being able to find scenarios that lead to a collision.</p>

				Each node in the tree includes the state of the system and the simulation time. When the tree grows, instead of running the simulation from the initial state, only a partial simulation is executed from an existing node of the tree to the new node.	
[55]	<p><b>Activity:</b> Refinement of concrete scenarios  <b>Phase:</b> System verification  <b>SOI: Decision and Control</b> parts of ACC (in the case study)</p> <p>The purpose of the study is to generate input trajectories (sequences of inputs) to falsify the requirements of a system.</p> <p>They use ACC as an example. The trajectory is a sequence (over time) of inputs. In the use case it is the acceleration profile of the leading vehicle. The same ACC algorithm is used to control two flowing vehicles.</p>	<p><b>Covered layer: 4</b></p> <p>In this paper, a scenario is defined by the trajectory of the control input of the SOI.</p> <p>In the case study, there are three vehicles platooning in the same lane. The SOI is a longitudinal ACC controller, which controls two vehicles following a leading vehicle. The scenario is then defined as the trajectory of the longitudinal acceleration values of the leading vehicle (i.e., the design variables) and the dynamics of the two following vehicles.</p> <p>Parameters for criticality evaluation include the distances between vehicles and the speeds of the two following vehicles.</p>	<p>In this study, a scenario is considered critical if it leads the system to violate the requirements. Each requirement is modeled on a critical scene. For example, the distance between the leading vehicle and the first following vehicle should not be less than 5m. The violation of this requirement is modeled as the advent of the first scene where the distance exceeds 5m.</p> <p>A strong assumption made by this paper is that the order of the advents of the critical scenes are known. It is possible that some critical scenes do not appear. But if they occur, they should follow the pre-defined order.</p> <p>The objective function of the searching process includes two parts. The first part maximizes the number of critical scenes. The second part minimizes the total length of the scenario.</p>	<p><b>Input:</b> A concrete scenario  <b>Output:</b> A critical concrete scenario</p> <p>The process to search for critical scenarios is formulated as an optimal control problem, which finds the optimal sequence of control input (in the case study, it is the acceleration profile of the leading vehicle and the time instances of critical scenes) to optimize the objective function.</p> <p>The system (including the states of the three vehicles and the ACC controller for the two following vehicles) is formulated as nonlinear differential equations. The proposed method assumes that the model of the system is not known except the order of the model.</p> <p>The optimization iteratively refines the given trajectory (as an initial nominal trajectory). In each iteration it:</p> <ol style="list-style-type: none"> <li>1. identifies the system (as a linear system) around the nominal trajectory with local time-varying model identification method by exciting the system n times (n = the order of the system) to get n independent state responses.</li> <li>2. gradient-based method is used to update the parameters (the trajectory and the time instance).</li> <li>3. Uses the new parameters to generate a new nominal trajectory.</li> </ol>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- The order of the system model</li> <li>- real test or simulation or some kinds of driving data to measure the state response (in the case study, they use a simplified dynamics model called “Optimal Velocity car-following Model”)</li> </ul> <p><b>Validation:</b> They have applied their method to generate a scenario that falsifies an ACC system. The identified critical trajectory is not validated.</p>

[56]	<p><b>Activity:</b> Instantiation  <b>Phase:</b> V&amp;V  <b>SOI:</b> Motion Control algorithms</p> <p>This paper proposes a method to find avoidable critical scenarios with controller synthesis method. Criticality is defined based on a set of predefined specifications. Avoidability is derived with controlled invariant set.</p>	<p><b>Covered layer:</b> 1, 4  From the use cases</p> <p>A falsifying scenario consists of two parts: an initial condition and a disturbance input profile.</p> <p>In the (ACC + Lane keeping) case study, the initial condition (also the system states) includes:</p> <ul style="list-style-type: none"> <li>• The longitudinal velocity of the ego vehicle,</li> <li>• The longitudinal velocity of the lead vehicle,</li> <li>• The distance between the two vehicles,</li> <li>• The lateral deviation from the lane center,</li> <li>• The lateral velocity,</li> <li>• The yaw-angle deviation, and</li> <li>• The yaw rate.</li> </ul> <p>The disturbance includes the longitudinal acceleration of the lead car, and the road curvature.</p>	<p>In this paper, a scenario is critical if it violates a given safety specification, criticality is defined based on specifications, together with a certificate that it is possible to satisfy the specification for this initial condition and disturbance, as long as the control inputs are within their range.</p> <p>Specifications are formally defined with first order logic. In the case study the specifications are:</p> <ul style="list-style-type: none"> <li>• the control input should stay within specified bounds.</li> <li>• whenever the lead car is closer than a certain value, the time headway needs to be greater than a threshold.</li> <li>• Lateral deviation, lateral velocity, yaw angle deviation and yaw rate should be within certain thresholds.</li> </ul> <p>The control inputs are:</p> <ul style="list-style-type: none"> <li>• the net force acting on the mass of the following car (the ego vehicle).</li> <li>• the steering angle of the front wheels.</li> </ul> <p>Note: In the case study, there is no specification for lateral control.</p>	<p><b>Input:</b> A logical scenario  <b>Output:</b> A critical concrete scenario</p> <p>The first step is to divide the system states into three sets according to system dynamics model:</p> <ol style="list-style-type: none"> <li>1. <b>Controlled invariant set:</b> It is a subset of safe states. For states in this set, as long as the disturbance is within its range, it is always possible to find a control input so that the next step state is also in this set.</li> <li>2. <b>Dual game winning set:</b> It is a subset of safe states. It is also the complementary set of the controlled invariant set.</li> <li>3. <b>Unsafe states:</b> States that violate the specifications.</li> </ol> <p>Step two is to sample the states (on the boundary or interior area) in the controlled invariant set to obtain potentially interesting initial conditions.</p> <p>Step three is to calculate the strategy of disturbance to lead the system moving from dual game winning set to unsafe states. In this step, the controller is not necessary.</p> <p>Step four is to calculate the strategy of disturbance to lead the system moving from a state in the controlled invariant set to dual game winning set. In this step, the controller is needed as a black box.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- System dynamics model (assumed to be linear)</li> <li>- Controller algorithm (black box)</li> <li>- Safety specifications</li> </ul> <p><b>Validation:</b> They have demonstrated their approach in two autonomous driving functions, adaptive cruise control and lane keeping. The applicability of this method is validated on two simple controllers and a realistic one provided by Comma AI. The efficiency of the method is validated by comparing the result with S-TaLiRo.</p>
[57]	<p><b>Activity:</b> Instantiation  <b>Phase:</b> System verification  <b>SOI:</b> Planning and control</p>	<p><b>Covered layer:</b> 4</p> <p>Layer 1 is represented by a map of the area discretized with a resolution of 1.5</p>	<p>A valid test scenario is the one that requires the AV to avoid a collision with a pedestrian, given that the pedestrian intercepts the AV path in</p>	<p><b>Input:</b> A logical scenario  <b>Output:</b> A critical concrete scenario</p> <p>At the beginning, the agents (pedestrians) are randomly placed using the spawn locations. At</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- Assumed behavior model of the pedestrians.</li> </ul>

	<p>This paper proposes the generation of test cases using agency-directed test generation.</p> <p>They propose to use a multi-agent system (a system where multiple intelligent agents interact in order to solve a problem) to generate test cases that would force the autonomous vehicle to react to avoid a collision.</p>	<p>meters. (not represented in the scenario space.)</p> <p>Layer 4 is represented by the pedestrians. They can have a random behavior or an agency-directed behavior where the agent’s perception is taken into account.</p>	<p>a way that the collision can be avoided.</p> <p>Scenarios are also scored to encourage more realistic scenarios. A living cost is assigned to promote shorter tests and a penalty is also given depending on how long the pedestrian spends on the road.</p>	<p>each simulation tick the agents behave depending on their class (random or directed). The possible actions are standing still, moving forward, backward, left or right. The test finishes when the assertion (forcing the AV to stop) is reached or when the AV gets to the end of the road.</p>	<p><b>Validation:</b> The have developed a case study where the emergency stopping of an AV is tested by pedestrians getting into the road.</p>
--	--	--	---	---	---

Table 5. Finding critical scenes for CV-based functions

#	Purpose	Scenario definition	Criticality definition& Surrogate Measure	Solution	Validation & other key observations
[58] Scenic	<p><b>Activity:</b> Logical scene modeling &amp; Instantiation &amp; improvisation &amp; assessment &amp; influential factor analysis</p> <p><b>Phase:</b> component design</p> <p><b>SOI:</b> CV function</p> <p>The proposed method can generate scenes (pictures) according to a given scenario description.</p> <p>It supports the training and testing of machine learning models. E.g.,</p> <ol style="list-style-type: none"><li>1. Generating Specialized Test Sets (e.g. partial occlusion, bad weather, or bad illumination)</li><li>2. Retraining to Improve Performance on Hard Cases (i.e., adding specific types of images (e.g., images with occluded vehicles) to the training set)</li><li>3. Generalizing from a Known Failure Case (e.g., to identify the most influential features to certain misclassifications and use that to guide the retraining)</li></ol> <p>It can be used in the function development phase</p>	<p>Scenario: Distributions over scenes.</p> <p>Scene: a set of configurations of objects in space. Generated scene: a picture</p> <p>According to our terminology, a scenario in this paper should be called a logical scene.</p> <p><b>A Domain-specific language is proposed to model a scenario</b> (i.e., a logical scene). It covers:</p> <p>Scenario Configuration:</p> <ul style="list-style-type: none"><li>• Configurations of Objects</li><li>• Relations between objects</li><li>• Time</li><li>• Weather</li></ul> <p>Object configuration</p> <ul style="list-style-type: none"><li>• Position</li><li>• Heading</li><li>• Road deviation: the heading of the car with respect to the local direction of the road</li><li>• Size</li><li>• Car model (including color)</li><li>• View range: to define the ‘can see’ predicate</li></ul> <p>Relations between objects</p> <ul style="list-style-type: none"><li>• Relative position</li><li>• Can see (if A can see B, B is in the view range of A.)</li></ul> <p>Not all these factors need to be specified in the scenario description. The user only needs to specify the ones of interest.</p>	<p>The purpose of the tool is to test the CV function under known critical scenes and to find the influential factors.</p> <p>Critical scenes are defined as the scenes that may cause the unintended behaviors of the camera perception functionality. E.g., miss detection, clutter detection, miss classification, etc.</p> <p>The proposed method can generate <b>both generic and system-specific</b> critical scenes.</p>	<p><b>Input:</b> logical scene described in the proposed language</p> <p><b>Output:</b> A set of scenes satisfying the given description</p> <p>Scene improvisation: Given a (partially) defined scenario, the improviser generates a set of scenes (pictures) that satisfy the scenario definition (i.e., the logical scene). The improviser uses a rejection sampling approach.</p> <p>Objects will be placed in a game engine (GTA V) according to the scenario description and physical constraints (e.g., cars are on the road, cars do not overlap, etc). Unspecified factors (e.g., the background, road structure, car model) will be randomly selected. For the background of the generated scene, the tool will put all the objects at a random place in the city in the game.</p> <p>A snapshot from the ego vehicle’s perspective is taken as one generated scene.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) Env. Model (appearance) (given by GTA V)</li><li>2) Implementation of the CV functions.</li><li>3) Interesting logical scene (either identified potentially risky scenarios or a complete set of scenarios)</li></ol> <p><b>Validation:</b></p> <p>Targeted data augmentation improves functional performance. Augmented data was designed according to an analysis of the influential factors.</p>

		<b>Covered layers:</b> Scenario config. 4,5 Generated scenes: 1,4,5  Since it only generates scenes, therefore, content in layer 4 (e.g., vehicles and pedestrians) are static.			
[59] Dee pRo ad	<b>Activity:</b> Scene refinement <b>Phase:</b> component design / verification <b>SOI:</b> CV function  DeepRoad, an unsupervised framework <ul style="list-style-type: none"> <li>to automatically generate large amounts of accurate driving scenes with various weather conditions (including those with rather extreme conditions).</li> <li>to test the consistency of three well recognized DNN-based autonomous driving systems across different scenes.</li> </ul> It can be used in the system testing and verification phase.	Definition of scenario is not explicitly given in the paper.  This paper explored and synthesized various driving scenes including those with rather extreme conditions. In particular, images with two extreme weather conditions, including heavy snow and hard rain, were collected and synthesized.  As we understand, this paper was not aimed for identifying critical scenarios, but rather for testing autonomous driving functions with synthesized scene. From this perspective, driving scenes under extreme weather conditions, including heavy snow and hard rain, were initially regarded as critical scenes.  <b>Covered Layers:</b> It adds information on layer 5 to an existing scene, which contains layers 1, 2, 4, ,5.	In this paper, ‘critical scenario/scene identification’ is not mentioned.  In this paper, scenes with bad weather are considered as critical.  They assume that scenes with bad weather may cause unintended behavior of the CV functions.	<b>Input scene:</b> scene with good weather <b>Output scene:</b> scene with bad weather  Scene generation: Based on the Generative Adversarial Networks (GANs) technique, images with the two extreme weather conditions were collected from Youtube videos to transform real-world driving scenes and deliver them with the corresponding weather conditions.  Testing: Metamorphic testing. Metamorphic relations are defined such that regardless of how the driving scenes are synthesized to cope with weather conditions, the driving behaviors are expected to be consistent with those under the corresponding original driving scenes.  Metric: the number of inconsistent behaviors of autonomous driving Systems.	<b>Required knowledge:</b> 1) real-world driving scenes 2) Videos with extreme weather conditions  <b>Validation:</b> A case study to show how added bad weather conditions will affect the driving behaviors of E2E learning AD functions.  Here the driving behavior refers to steering angle, which is the output of different DNN-based autonomous driving models. It is claimed that ‘Since the road scenes should not largely impact the steering angles, any inconsistency may indicate correctness or robustness issues of the systems under test.’ Key assumption: the road scenes should not largely impact the steering angles.
[60] Dee pTes t	<b>Activity:</b> Assessment <b>Phase:</b> component design / verification <b>SOI:</b> end-to-end learning  DeepTest is proposed for automatically detecting erroneous behaviors of DNN-driven (end to end	Definition of scenario is not explicitly given in the paper.  This paper generates synthesized scenes by applying different image transformations.  This paper is aimed for testing the DNN capabilities for autonomous vehicle instead of identifying critical scenarios. From this perspective, driving	The critical scenario is not mentioned in this paper.  However, we can assume that the scene with bad weather conditions or by changing the lightning conditions can affect the capabilities of DNN, thus, may lead to critical scenario.	<b>Input Scene:</b> Scene before the implementation of image transformations.  <b>Output Scene:</b> A new scene resulted from implementing image transformations process.  <b>Scene generation:</b> By implementing nine different realistic image transformations	<b>Required Knowledge:</b> • Real-world driving condition scene.  <b>Validation Metrics:</b> • The correlation between input-output diversity and neuron coverage.  In this metric, they tried to make sure whether neuron coverage is a good

	<p>learning) vehicles that can potentially lead to fatal crashes. Also, it proposed to increase the neuron coverage of DNN models such that the accuracy of DNN models can be increased.</p> <ul style="list-style-type: none"><li>• Leverage the notion of neuron coverage to explore different parts of DNN logic.</li><li>• Applied different image transformations that mimic real-world conditions (e.g., changing contrast, adding rain) in driving condition to maximize the neuron coverage of DNN.</li><li>• They also generated a new image by using greedy algorithm in order to increase the neuron coverage.</li><li>• Automatically detect the erroneous behavior with neuron coverage guided the synthesized images.</li></ul>	<p>scenes under extreme weather conditions and lightning conditions can be regarded as critical scenes.</p> <p><b>Covered Layers:</b> It adds information on layer 5 (by doing image transformations) to an existing scene, which contains layer 1, 2, 4, 5.</p>		<p>In particular, for blurring and adding fog/rain effects, they used convolutional transformation.</p> <p>They used four different types of blurring filters (e.g., averaging, Gaussian, median, and bilateral) for the blurring. And Adobe photoshop was used for composing these multiple filters.</p> <p><b>Methodology:</b></p> <ul style="list-style-type: none"><li>• Systematic testing with neuron coverage. They mentioned the formula how to calculate neuron coverage. And based on the formula, will be used to guide their greedy optimization algorithm. They calculated neuron coverage by dividing the ratio of unique neurons get activated for given input and the total number of neurons in DNN.</li><li>• Increasing coverage with synthetic images. Implement image transformations to the scene to show how the results of image transformations can affect driving behaviors (e.g., steering angle and steering directions) of DNN-based AV. Moreover, it is also intended to show how the neuron coverage of DNN, the number of activated neurons, and the accuracy of DNN are affected because of the image transformations.</li><li>• Combining transformations to increase coverage.</li></ul>	<p>metric to capture the functional diversity of DNNs.</p> <p>For instance, the neuron coverage changes with the changes in output steering angles, i.e., different neurons get activated for different output.</p> <p>And based on the results showed above, the neuron coverage is correlated with input-output diversity of DNNs.</p> <ul style="list-style-type: none"><li>• The set number of activated different neurons affected by image transformations.</li><li>• The extent of neuron coverage affected by the combination of image transformations.</li><li>• The number of erroneous behaviors of DNN models that can be detected by DeepTest.</li></ul> <p>The accuracy of DNN models.</p>
--	---	--	--	---	--

				<p>Since each individual image transformation increased neuron coverage, they also tried to find whether combined image transformations can increase the coverage by using this method.</p> <p>In this method, they used greedy algorithm to find efficiently the combination of image transformations that result in higher coverage.</p> <p><b>Find critical scenarios:</b> They find the critical scenario based on the erroneous behaviors detected by DeepTest using synthetic images.</p>	
[61]	<p><b>Activity:</b> Criticality assessment</p> <p><b>Phase:</b> component design / verification</p> <p><b>SOI:</b> end-to-end learning</p> <p>This paper proposes a tool named DeepXplore to test and validate the performance of Deep Learning systems. It is the first testing tool to measure the performance of DNN according to the authors claim.</p> <p>In this paper, they are not particularly building the tool for DNN-based system for self-driving cars. However, they evaluate DeepXplore with one of datasets that contain driving video frame to evaluate DNN for self-driving cars.</p> <p>DeepXplore is the extended version of DeepTest.</p>	<p>Definition of scenario is not explicitly given in the paper.</p> <p><b>Covered layers:</b> 5 and noises on camera.</p> <p>This paper adds additional information on layer 5 such as intensifying the light condition and adding tiny black rectangles for simulating effects of dirt on camera lens by implementing image transformations.</p>	<p>This paper defines critical scenarios as corner cases.</p> <p><b>Corner case definition:</b> the unexpected behavior from the cars which may leads to critical scenario.</p> <p><b>Example:</b> a scene with different contrast or brightness could lead to corner case. This corner case should be detected by the DNN as erroneous behavior to achieve reliable systems.</p> <p>In this paper, they used transformed scene to generate a corner case.</p>	<p><b>Input Scene:</b> Original images from the datasets.</p> <p><b>Output Scene:</b> Transformed image by DeepXplore.</p> <p><b>Datasets:</b> They used five datasets in which three of datasets contain images (hand-written digits, general images, and driving video frames). And the other two datasets contain PDFs and android apps. Only the driving video frame database is related to our survey.</p> <p><b>Joint Optimization Problem:</b> DeepXplore formulated the joint optimization problem to generate new test input that can maximize the neuron coverage and exposing many differential behaviors (i.e., differences between multiple similar DL systems). And, after the optimal result is obtained, then it will implement specific-domain constraints to achieve a realistic and valid input tests.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"><li>• Image datasets</li><li>• DNN algorithms</li></ul> <p><b>Validation:</b> The evaluation is based on the numbers of erroneous behaviors found by DeepXplore and numbers of neuron coverage.</p> <p><b>Limitations:</b></p> <ul style="list-style-type: none"><li>• It only considers a small set of image transformations; thus, it does not fully capture all real-worlds conditions.</li><li>• The gradient information-only is not fully be able to compute many realistic transformations accurately.</li><li>• The proposed method cannot guarantee the absence of errors.</li></ul>

				<p>It has two objective functions:</p> <p><b>1. Maximizing differential behaviors.</b> The first objective of the optimization problem is to generate test inputs that can induce different behaviors in the tested DNNs, that is, different DNNs will classify the same input into different classes.</p> <p><b>2. Maximizing neuron coverage.</b> To achieve a maximized branch coverage of the testing Cases.</p> <p><b>Specific-domain constraints:</b> this paper have two major types of constraints which are:</p> <ol style="list-style-type: none"><li>1. Image constraints.</li><li>2. Other constraints.</li></ol> <p>The image constraints are used for image transformations to the image datasets.</p> <p><b>Image transformations:</b> They implemented image transformations based on the specific-domain constraints to generate new test input images. The image transformations that they used include adding lighting effects, occlusion by single rectangle, and occlusion by multiple rectangles).</p>	
[62]	<p><b>Activity:</b> Criticality (complexity) assessment <b>Phase:</b> offline testing and validation <b>SOI:</b> Perception tasks</p> <p>They assume that driving conditions with different complexity have</p>	<p>Definition of scenario is not explicitly given in the paper.</p> <p><b>Covered layers:</b> 1,(2), 4, 5</p> <p>Scenario Configuration:</p> <ul style="list-style-type: none"><li>• road type (layer 1)</li></ul>	<p>In this paper, criticality is considered as scenario complexity.</p> <p>Scenario complexity is not formally defined. It is classified into three levels as Simple, Medium and</p>	<p>The propose Graded Offline Evaluation (GOE) framework includes:</p> <ul style="list-style-type: none"><li>• multi-sensor data grading (in terms of complexity) with the semantic descriptor</li><li>• cognition task list</li></ul>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) Ungraded scenario data</li><li>2) Ontology of the scenario configuration (i.e. the sematic descriptor).</li></ol>

	<p>different requirements to the cognitive ability (i.e., complex driving condition requires high cognitive ability) to guarantee safety. Therefore, they propose a framework to determine the driving complexity (graded into three levels) of the scenes in naturalistic driving datasets so as to evaluate if the cognitive tasks (SOI) fulfills the required cognitive abilities for different driving complexities.</p>	<ul style="list-style-type: none"> <li>• scenario content (layer 1)</li> <li>• challenging conditions (layers 4 &amp; 5)</li> </ul> <p>scenes from classical naturalistic driving datasets are manually encoded into semantic descriptors (i.e., a vector).</p> <p>There is no clear definition of each scenario configuration, but a list of example factors is provided.</p>	<p>Complex, according to scenario configuration values.</p> <p>Complexity of each semantic descriptor is judged by expert experience.</p>	<ul style="list-style-type: none"> <li>• <i>Cascaded Tanks Model</i> based offline evaluation and results analysis</li> <li>• Visualization</li> </ul> <p><b>Criticality (complexity) checking:</b>  <b>Input sce.:</b> executable scene (image or sensor input at one time instance)  <b>Output:</b> The complexity of the scene</p> <p>Sensory data (image or sensor data) is manually encoded into a proposed Semantic Descriptor, which is further classified into 3 levels of scenario complexity by an SVM classifier. The SVM was trained according to expert knowledge.</p> <p><b>Function evaluation on scenarios with different complexity:</b></p> <ul style="list-style-type: none"> <li>• With the graded and annotated offline testing data and the predefined cognition tasks, the cognition abilities are evaluated with the Cascaded Tanks Model.</li> <li>• The Cascaded Tanks Model provides the cognition performance of the AV at different scenario complexity.</li> </ul>	<p>3) Expert knowledge to grade Scenario complexity for the training of the SVM.</p> <p><b>Inputs for the cognition validation:</b></p> <ol style="list-style-type: none"> <li>1) threshold indicators of fail/pass of cognitive algorithm /modules</li> <li>2) weights of different cognition task</li> </ol> <p><b>Validation:</b>  A case study to show the proposed framework can evaluate cognition tasks under different complexity. But they did not validate the classified complexity of each scenario.</p>
[63]	<p><b>Activity:</b> Criticality (Complexity) Assessment  <b>Phase:</b> Offline testing and validation  <b>SOI:</b> Perception Task</p> <p>Motivation: it often requires massive amount of traffic scenario datasets to perform an off-line testing and performance evaluation of UGV.</p> <p>These traffic scenario datasets provide data support which include</p>	<p>Definition of scenario is not explicitly given in the paper.</p> <p><b>Covered layers:</b> 1, 2, 4, 5</p> <p>This paper uses semantic descriptor to classify the scene. Specifically, the classification is considered from two levels:</p> <ul style="list-style-type: none"> <li>• Through vector to describe the semantic characteristics of road scene.</li> </ul>	<p>In this paper, criticality is considered as scenario complexity.</p> <p>Scenario complexity is not formally defined. It is classified into three levels as Simple, Medium and Complex, according to scenario configuration values.</p> <p>Furthermore, scenario complexity is measured on the basis of road types, scene types, challenging condition and traffic elements.</p>	<p>The scenario complexity is computed from two perceptual data levels:</p> <ul style="list-style-type: none"> <li>• Road semantic complexity (RSC) → It is based on support vector regression in which it predicts by learning the relationship between the road label and semantic descriptor.</li> <li>• Traffic element complexity (TEC) → to measure the complexity with the respect to the moving traffic entities such as nearby vehicles.</li> </ul>	<p><b>Required Knowledge:</b></p> <ul style="list-style-type: none"> <li>• Ungraded scene data</li> <li>• Ontology of the scenario configuration (i.e. the sematic descriptor).</li> </ul> <p><b>Validation:</b>  Implement the proposed method in their off-line test task to speed up the process.</p> <p>In specific, the off-line test data are having 2000km. The data are derived</p>

	<p>various annotations (e.g., types of scene and roadways).</p> <p>However, the existing database lacks the quantitative description of scene data complexity and scene characteristics.</p> <p>Thus, this paper proposes a method in quantifying scenario complexity to rank massive scene data.</p> <p>Moreover, they claimed that there is usually a negative correlation between the unmanned vehicle algorithm and scenario complexity.</p> <p>Hence, the complexity of the scene data needs to be incorporated for reliable evaluation of UGV systems.</p>	<p>Specifically, for describing the road scene, they defined the semantic descriptor from three levels: road types, scene types, and challenging conditions. And a vector is used to each level to represent the complexity value. The vector value is 0 or 1 for road types and scene types. And the vector value for challenging conditions is 0, 0.2, 0.4, 0.6, 0.8, 1.</p> <ul style="list-style-type: none"><li>• Through matrix to describe the topology information of traffic elements.</li></ul> <p>Particularly, they used Nx2 matrix where the first column is appointed for the value of distance from car to car. The second column is assigned for the value of car's viewpoint which obtained from 8-nearest neighbor of viewpoint from ego vehicle.</p>	<p>The final calculation of scenario complexity is obtained by the sum of quantified road semantic complexity and the traffic element complexity.</p>	<p><b>Input:</b> Scene data in datasets.</p> <p><b>Output:</b> The complexity of scene.</p> <p><b>Complexity checking for RSC:</b> Scene data are encoded to semantic descriptor. In particular, the semantic descriptor is defined from three levels:</p> <ul style="list-style-type: none"><li>• Road types</li><li>• Scene types</li><li>• Challenging conditions</li></ul> <p>In here, SVR is used to automate the scene complexity task by exploiting the relationship between the road marking complexity and the semantic descriptor.</p> <p><b>Complexity checking for Traffic Elements</b> The scene is encoded to the matrix of traffic elements. The traffic elements are reflected by semantic data in two aspects: distance and angle.</p> <p>Each row in the matrix of traffic element represents each car's information.</p> <ol style="list-style-type: none"><li>1. First column: for the distance from car to car.</li><li>2. Second column: for the car's viewpoint.</li></ol> <p>The matrix is obtained from the calculation of point cloud collected by LIDAR. And based on the matrix, the complexity of traffic elements can be calculated.</p>	<p>from the scene video data and the 3D laser point cloud collected by the real road.</p>
[64] Bolt e20 19	<p><b>Activity:</b> Criticality check of concrete scenarios (video streams)</p> <p><b>Phase:</b></p>	<p>In this work, no clear definitions are given to scenario and scene. Instead, they use the term "corner case".</p>	<p>According to our understanding, critical scenarios in this paper are interoperated as corner cases.</p>	<p><b>Input Sce.:</b> A video stream</p> <p><b>Output:</b> Criticality</p> <p><b>process:</b></p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"><li>• an image prediction algorithm</li></ul>

	<ul style="list-style-type: none"> <li>operation phase: Online corner case detection</li> <li>developing phase: offline corner case detection (for training algorithms)</li> </ul> <p><b>SOI:</b> perception system</p> <p>This paper presents a method to detect corner cases in video streams.</p> <p><b>Contribution:</b></p> <ul style="list-style-type: none"> <li>a formal definition of a corner case in a video stream;</li> <li>domain-specific challenge identification regarding corner case or anomaly detection in a video stream;</li> <li>Propose a system framework for corner case detection in a video stream.</li> </ul>	<p>The word ‘case’ is not defined. Corner case is clearly defined.</p> <p>Corner case definition: non-predictable relevant object/class in relevant location.</p> <p>So, according to our understanding, a scenario (or a case) mainly refers to a video stream.</p> <p><b>Covered layers:</b> The main concern is on layer 4</p>	<p>Corner case definition: non-predictable relevant object/class in relevant location.</p> <p>If a critical frame is detected, the video stream containing this frame and its previous frames is considered as a corner case.</p> <p>A frame in a video stream is critical if the frame contains a relevant moveable object (e.g., pedestrian, vehicle, bicycle, etc.) and its critical score is higher than a threshold.</p> <p>The critical score will increase if (1) this frame is largely different from a predicted frame (based on previous frames by a given image prediction algorithm); and (2) the relevant object is close enough (comparing to a given threshold) to the ego vehicle.</p>	<ul style="list-style-type: none"> <li>According to the previous frames in a video stream, the next frame will be predicted by a given image prediction algorithm.</li> <li>The predicted image is compared with the actual frame in the video stream.</li> <li>A semantic segmentation algorithm is used to detect if there is a relevant object in the frame. If there is no relevant object, the frame will be considered as non-critical.</li> <li>A corner case score is calculated according to (1) the difference between the predicted image and the actual frame; and (2) the distance between the relevant object and the ego vehicle.</li> <li>If the corner case score exceeds the threshold, a corner case is detected. The score and the location of the relevant object are outputted.</li> </ul> <p>To identify the location of the relevant object, they simply assume that objects being farther away from the bottom of the image are farther to the ego vehicle.</p>	<ul style="list-style-type: none"> <li>a semantic segmentation algorithm to identify relevant objects</li> <li>Cityscapes dataset</li> <li>Detection System to calculate corner case score</li> <li>threshold value of corner case score.</li> </ul> <p><b>Hypothesis:</b></p> <ul style="list-style-type: none"> <li>As a working hypothesis, they assume that it is possible to detect corner cases with a camera-based system.</li> </ul> <p><b>Limitation:</b> Within this work, they limit themselves to mobile objects, which determine the relevant classes for corner case detection.</p>
[65]	<p><b>Activity:</b> Criticality assessment</p> <p><b>Phase:</b> Component design / verification</p> <p><b>SOI:</b> CV function</p> <p><b>Motivation:</b> This paper tries to assess a CV algorithm (YOLO and Faster RCNN) by generating a photorealistic virtual world.</p>	<p>Definition of critical scenario is not explicitly given in this paper.</p> <p><b>Covered layers:</b> 5</p> <p>In this paper, they generate scene (i.e., picture) with various light conditions (environment conditions) and car colors to assess CV-based vehicle detection algorithms.</p>	<p>The purpose of this paper is to assess the ability of CV algorithm to detect an object with different color in the different light conditions.</p> <p>Based on our understanding, the critical scenario could be happened when the CV algorithm fails to detect an object (i.e., sedan car in this paper)</p>	<p><b>Input:</b></p> <ul style="list-style-type: none"> <li>Light modelling</li> <li>3D car modelling</li> <li>Environment modelling</li> </ul> <p><b>Output:</b></p> <ul style="list-style-type: none"> <li>Photo realistic virtual world with a 3d sedan car as a main object.</li> </ul> <p><b>Process of generating photorealistic virtual world:</b></p>	<p><b>Required knowledge(s):</b></p> <ul style="list-style-type: none"> <li>CV algorithms (YOLO and Faster RCNN)</li> <li>Generated photorealistic virtual world</li> <li>Five common colors for virtual car (i.e., black, white, red, blue, green)</li> <li>Intensity of illumination from low (0.001 refers to dark condition) to high (400 refers to very bright condition)</li> </ul> <p><b>Evaluation process:</b></p>

	<p>In particular, they tried to reproduce various illumination intensities in their photo-realistic virtual world scene. Then, based on the reproduced scene, they conduct assessment for two CV algorithms to detect a sedan car with different colors.</p>			<p>First, they conducted an accurate light modelling. Light is essential part to generate a photorealistic virtual world since it can affect or determine appearance of the object in the environment. And in particular, they tried to capture the different behaviors of light that arise from nature.</p> <p>Second, they created a 3d car by using 3ds Max as the modelling software. The car is made with 203,522 polygons and 610,556 vertices.</p> <p>Third, they used Unreal Engine 4 (UE4) to create an environment/scene modelling. They used diffuse bidirectional reflectance distribution function (BRDF), Microfacet Specular BRDF, and environment BRDF as for lighting system.</p>	<p>In this paper, they evaluated the CV algorithms based on various intensities of light. Moreover, they tested the capability of algorithms to detect car colored with common color that are mentioned in the required knowledge.</p> <p>The results show car under very high condition can hardly be detected. Moreover, cars with red and blue color are more robust to the variation of illumination.</p>
[66]	<p><b>Activity:</b> Assessment / validation</p> <p><b>Phase:</b> Component design</p> <p><b>SOI:</b> Sign detection function</p> <p><b>Motivation:</b> This paper tries to automatically generate corner cases by using their proposed method named cycle-consistent generative adversarial networks (CycleGANs). And the generated corner case images are used to validate the robustness and reliability of ADAS to ensure the safety.</p>	<p><b>Covered layer(s):</b> 2 and 5</p> <p>This paper generated corner cases based on “stop” sign images. And the images are transformed based on the various temperatures (e.g., room temperature, high temperature). Thus, the covered layers are the road infrastructure and environment conditions.</p>	<p>They defined critical scenarios as corner cases.</p> <p>And CV algorithm is validated by its capability to detect stop sign. If the CV algorithm cannot be able to detect the sign, then it could lead to critical scenario.</p>	<p><b>Input:</b></p> <ul style="list-style-type: none"><li>• Image belonging to class A (nominal condition)</li><li>• Image belonging to class B (corner case)</li></ul> <p><b>Output:</b> Image belonging to class B (corner case). Specifically, the expected output is a corner case image that appropriately carry the important feature of “corner case” while retain other fundamental characteristics of image (e.g., sizes and shapes of background objects).</p> <p><b>Process of generating corner case images</b></p> <p><b>Generator structure:</b></p>	<p><b>Required knowledge(s):</b></p> <ul style="list-style-type: none"><li>• Three image datasets</li><li>• Information regarding important features of corner case image (e.g., the mosaic noise textures used by high temperature or distortion patterns caused by rain drops).</li><li>• Three high temperature datasets.</li><li>• One normal temperature dataset.</li></ul> <p><b>Evaluation process:</b> They evaluated the proposed method by using three datasets as follows:</p> <ul style="list-style-type: none"><li>• Full dataset including 164k high-temperature images.</li><li>• Small dataset including 2k high-temperature images.</li></ul>

	<p>The main reasons are to minimize the effort and budget for physically collecting data.</p>			<ul style="list-style-type: none"><li>• <b>Encoder:</b> It consists of convolutional layers. It aims to extract from a given image using CNN.</li><li>• <b>Transformation:</b> Based on ResNets to achieve accurate refinement for feature extraction. It aims to modify the extracted features.</li><li>• <b>Decoder:</b> It takes the modified feature after transformation and generates the corresponding image from these features. It consists of deconvolutional layers.</li></ul> <p><b>Training the generator:</b> It is challenging to train the generator to generate corner case automatically. In this paper, they used CycleGANs, and it consists of four major components as follows:</p> <ul style="list-style-type: none"><li>• <b>Forward generator:</b> It aims to generate synthetic images that are identical to real one.</li><li>• <b>Forward discriminator:</b> It is created to guide the training of forward generator. It is designed to judge whether a given corner case is real or synthetic.</li><li>• <b>Backward generator:</b> The function of backward generator and backward discriminator are to prevent the multiple input images to be mapped to the same output image. In particular, backward generator the inverse transformation of the forward generator.</li><li>• <b>Backward discriminator:</b> It aims to distinguish between real and synthetic nominal recordings.</li></ul>	<ul style="list-style-type: none"><li>• Hybrid dataset including 2k high-temperature images and 162k synthetic images.</li></ul> <p>The results showed that the generated synthetic images have similar features to the actual images. Moreover, the synthetic images retained the important characteristics of nominal recordings.</p>
--	---	--	--	--	---

[67]	<p><b>Activity:</b> Assessment / validation</p> <p><b>Phase:</b> Component design</p> <p><b>SOI:</b> CV / ML function</p> <p><b>Motivation:</b> This paper tries to find falsifying executions, such that, they will know whether the output from learning components can lead to a failure of CPS.</p> <p>They conducted two case studies in which the first case study is conducted based on Simulink for advanced emergency braking systems (AEBS) that connected to DNN-based image classifier.</p> <p>In their second case study, they conducted a validation for a CNN algorithm to detect a cow in a self-driving simulator. If the algorithm fails to detect a cow, the car will collide with the cow.</p>	<p><b>Covered layer(s):</b> 4 and 5</p> <p>This paper generated scene with different intensity of object color and rotation of the object. They also displaced the object to different location. Hence, the covered layer is movable objects and environment condition.</p>	<p>The definition of critical scenario is not explicitly given in this paper. However, the critical scenario could be happened when the CV algorithms fails to detect the object.</p>	<p><b>Case study 1:</b> The AEBS determines a braking mode depending on the speed of vehicle (vs), its velocity (vp), and the distance between vehicle and the obstacle (dist). Then, the AEBS will compute the times to collision and longitudinal safety indices (e.g., safe, warning, braking, collision mitigation mode).</p> <p>First, they tried to find the region of uncertainty (ROU) where they identify it as counterexample candidate. The ROU found when vs= 25 and dist= 40.</p> <p>Then, they generated an image that can lead to misclassification by changing the brightness and location of the object.</p> <p><b>Case study 2:</b> In this case study, they implemented CNN that classifies the pictures captured by the onboard camera in two categories, e.g., cow and not cow.</p> <p>Then, they try to misclassify the scene by changing the rotation of cow and the color of cow.</p> <p>Based on this case study, they tried to find a case where the AEBS fails to operate.</p>	<p><b>Required knowledge(s):</b></p> <ul style="list-style-type: none"> <li>• Images.</li> <li>• Values to define counterexample candidates.</li> </ul> <p><b>Evaluation process:</b> They validated the DNN algorithms by using the corner case misclassifications identified from the steps that are mentioned in the previous column.</p> <p><b>Note: Use case 1 is out of the scope of literature study.</b></p>
[68]	<p><b>Activity:</b> Assessment</p> <p><b>Phase:</b> Component design / verification</p> <p><b>SOI:</b> CV function</p>	<p>Definition of critical scenario is not given in the paper.</p> <p><b>Covered Layers:</b> 1, 2, and 5.</p>	<p>No critical scenario is addressed in this paper.</p> <p>However, we can assume the critical scenario could be happened if the LD systems cannot</p>	<p><b>Input:</b> extracted images from videos.</p> <p><b>Output:</b> lane detection with marking annotations (e.g., white dashed line is annotated light yellow, yellow single line</p>	<p><b>Required Knowledge:</b></p> <ol style="list-style-type: none"> <li>1. Image dataset</li> <li>2. DNN algorithms</li> </ol> <p><b>Evaluation process:</b></p>

	<p><b>Motivation:</b> This paper establishes the reference systems to evaluate the quality of line marking built by transportation agencies to support ADAS functions that rely on pavement markings.</p> <p>Also, the test system can be used by developers as benchmark for their proprietary systems.</p> <p>They gathered an extensive video dataset of various roads in Central Texas at different times of day and weather conditions to support the proposed test system.</p> <p><b>Out of the scope</b></p>	In this paper, they evaluate the line detection algorithms performance based on the environmental factors, lane marking types, color, material, and the retro reflectivity of pavement markings. Also, the dataset they created includes different types of roads (e.g., asphalt and concrete road surfaces)	detect the line marking when it is activated for ADAS.	<p>with orange color, and road curb with yellow color)</p> <p><b>Process of gathering the Dataset:</b> The video data were gathered in various roads at different times of day (morning 11 am CST, evening 6 pm CST, and night 10 pm CST). To maintain the homogeneity of the video, the camera was set at 25 fps during the recording of roads.</p> <p>Moreover, they used retro reflectometer to measure how visible the marking will be at night.</p> <p><b>DNN-based LD algorithms they tested:</b></p> <ol style="list-style-type: none"> <li>1. SCNN</li> <li>2. LaneNet</li> <li>3. ENet</li> </ol>	<p>They used Scalable tool to annotate each image in dataset. However, they still annotated the image manually. The performance metric they used is based on conventional pixel-accuracy such as True Positives (TP), False Positive (FP), F-Measure, etc.</p> <p>In particular, they set a threshold for line markings with the width equal to 30 pixels to evaluate whether the line marking is successfully detected.</p> <p><b>Limitation:</b> The annotation of each image is given manually.</p>
[69] CV- HAZ OP	<p><b>Activity:</b> influential factor analysis <b>Phase:</b> Requirement analysis, component design <b>SOI:</b> CV function</p> <p>In this paper, a complete HAZOP is conducted on a generic CV system for automated driving.</p> <p>The result is a checklist with more than 900 identified hazards (it is called hazard in the paper, but it actually means failure mode). The checklist can be used to evaluate and improve test datasets.</p> <p>This paper is <b>out of the scope</b> of the survey.</p>	<p>Scenario and scene are not explicitly mentioned in this paper.</p> <p>According to our interpretation, a scene in this paper includes the content of the image captured by the camera and the whole perception system including sensors and algorithms.</p> <p><b>Covered layers:</b> 1,2,4,5 and the potential problems of the CV functions of the ego vehicle. It covers all the items that need to be identified/classified by the CV functions.</p>	<p>The purpose of the HAZOP is to analyze how different deviations from the design intent of the CV system will lead to perception information loss (i.e., the definition of hazard in this paper).</p> <p>Example of loss of information: A glare may cause a loss of texture information.</p>	<p>HAZOP performed by CV experts and safety experts on the whole CV process.</p> <p>The whole CV process is divided into 5 nodes (or locations in the paper’s term), namely light sources, medium, object, observer and algorithm. Different parameters are defined for different nodes.</p> <p>Generic key words and CV-specific additional key words are used.</p> <p>A deviation is identified by a combination of a node’s parameter and a keyword.</p> <p>The consequences of the identified deviations are judged and reviewed by</p>	<p><b>Required knowledge:</b> 1) Knowledge of CV systems.</p> <p><b>Validation:</b> 1) Expert validation to check the relevance of the identified hazards (i.e., information loss) 2) They use stereo vision as a case study. For the identified hazard “no texture”, failure rate at the identified hazardous area (where “no texture applies”) on a picture is higher than the average failure rate for various algorithms on multiple databases.</p> <p>The hardware running the algorithms is not included in this paper. Therefore, it only considers SOTIF problem but not ISO 26262.</p>

				multiple experts with different backgrounds.	
--	--	--	--	--	--

Table 6. Deductive reasoning methods

#	Purpose	Scenario definition	Criticality definition& Surrogate Measure	Solution	Other key observations
[34]	<p><b>Activity:</b> Reasoning <b>Phase:</b> system validation <b>SOI:</b> the whole system <b>ODD:</b> Highway</p> <p>It provides a systematically defined test case catalogue with good coverage of critical highway traffic situations. Each entry of the catalogue is one functional scenario.</p> <p>The goal of this catalogue is to cover all the pre-crash scenarios on a highway.</p>	<p>This paper only considers highway scenarios. The scenario definition includes (1) the relative positions between the ego vehicle and the threatening vehicles; and (2) the intended maneuvers of the ego vehicle and the threatening vehicles.</p> <p>All the vehicles have only two types of maneuvers, namely lane keep (accelerate, decelerate, keep speed) and lane change (steer left and steer right). A complex maneuver will be a combination of these maneuvers.</p> <p><b>Covered Layer: 4</b></p>	<p>A critical scenario should contain at least one accident threat (i.e., one accident type may potentially happen)</p> <p>This paper assumes three types of accidents from the perspective of the ego vehicle, namely rear-end collision, front-end collision and sideswipe.</p> <p>In this work, a scenario starts from a threat-free condition and ends with either a crash or another threat-free condition.</p> <p>In other words, this work tries to catalogue all the pre-crash scenarios.</p>	<p>This paper proposes a qualitative and systematic approach to derive the scenario catalogue.</p> <p>The approach starts with basic critical scenarios, where only one threatening vehicle is considered, and the ego vehicle has only one intended maneuver (lane keep or lane change). Nine basic critical scenarios are identified.</p> <p>When encountering an accident threat, the ego-vehicle may start an evasive maneuver (brake, accelerate or lane change), which may lead to a secondary accident threat. These scenarios with evasions are called complex scenarios. One complex scenario can be seen as a connection of multiple basic scenarios with evasive maneuvers. In total, 13 complex critical scenarios are identified.</p> <p>When identifying complex critical scenarios, they limit the number of threatening vehicles within 3 and the number of evasive maneuvers within 2.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) Potential behaviors of vehicles on the highway</li><li>2) Possible collision types</li><li>3) Traffic rules on the highway.</li></ol> <p><b>Validation:</b></p> <ol style="list-style-type: none"><li>1) The coverage of the identified test case catalogue is evaluated by a naturalistic driving behavior database.</li><li>2) One entry is parameterized and sampled to get concrete scenarios for simulation, so as to identify the performance boundary of this entry.</li></ol>
[70]	<p><b>Activity:</b> Reasoning <b>Phase:</b> system validation <b>SOI:</b> the whole system <b>ODD:</b> highway</p> <p>This paper introduces a framework to define safety relevant functional scenarios, i.e., scenarios ending up with a crash.</p>	<p>The paper mentioned the six-layer scenario model. It says that in this work, only the movements of objects (i.e., the layer 4) are considered.</p> <p>In this paper, a base scenario shows how other vehicle (challenger) may collide with the ego vehicle assuming the ego vehicle keeps its lane and speed.</p>	<p>A scenario is critical if it may lead to a collision.</p> <p>Each critical scenario is described by the collision type and the initial position of the challenging vehicle.</p>	<p>Critical functional scenarios are identified through a qualitative and systematic approach.</p> <p>First, it defines all the possible ways to collide between two vehicles on a highway.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) Potential behaviors of vehicles on the highway</li><li>2) Possible collision types</li><li>3) Traffic rules on the highway.</li></ol> <p><b>Validation:</b></p> <p>No validation</p>

	It emphasizes that by only considering relevant objects (e.g., the vehicles close to the ego vehicle) the number of possible scenarios can be significantly reduced.	Covered layer: 4		Second, it considers other situations that may increase the challenge to avoid the crash, e.g., the evasive trajectory is blocked by a third vehicle.	So far, they only considered the collisions that involve only two vehicles.
[71]	<p><b>Activity:</b> Reasoning  <b>Phase:</b> system V&amp;V  <b>SOI:</b> The whole system</p> <p>The purpose of this method is to identify and quantify hazardous scenarios for highly automated driving vehicles with an emphasis on AD function limitations and failures.</p>	<p>In this paper, a scenario is defined as the temporal development between several scenes in a sequence of scenes. However, the necessary content in a scenario description is not given. According to our interpretation, the hazardous scenarios include a hazard and the environmental condition, which may trigger the hazard.</p> <p><b>Covered layers:</b> (1, 2), 4, 5 + hazard of the ego vehicle</p>	This paper uses the term hazardous scenario, which refers to a scenario that may lead to a harm, caused by the functional limitation or failures of the system.	<p>Hazardous scenarios are identified with the following steps:</p> <p>Step 1 is to model the automated driving functions as components and interfaces</p> <p>Step 2 is to identify hazards. They firstly identify generic vehicle-level hazards that are independent of the underlying implementation with a HAZOP-like method. Secondly, they identify the Functional Insufficiencies with a HAZOP-like method. Here, each AD function is treated as a black box.</p> <p>Step 3 is to identify the causes of the functional insufficiencies and the corresponding environmental conditions with “environmental fault tree” (a new method proposed in this paper to combine triggering condition with fault tree).</p> <p>Step 4 is to quantify the risk of the identified hazardous scenarios (functional insufficiencies + environmental condition)</p>	<p><b>Required Knowledge for the identification of the generic hazards (step 2.1)</b></p> <ol style="list-style-type: none"> <li>1) A set of base (pre-crash) scenarios (e.g., the ones identified in [70])</li> <li>2) A set of predefined basic maneuvers (e.g., start, follow, turn left/right)</li> <li>3) A set of keywords to determine possible Incorrect Vehicle Behavior</li> <li>4) A set of top-level events (e.g., collision types)</li> <li>5) Possible behaviors of vehicles on the road</li> </ol> <p><b>Required Knowledge for the identification of the generic hazardous scenarios (steps 2.2 &amp; 3)</b></p> <ol style="list-style-type: none"> <li>1) Interfaces of each AD function</li> <li>2) A set of predefined key words to determine Functional Insufficiency</li> <li>3) A set of basic scenarios</li> <li>4) Understanding of the system and functions to analyze the causes of each functional insufficiency</li> <li>5) Influential environmental factors</li> </ol> <p><b>Validation:</b> no validation</p>
[72], [73]	<p><b>Activity:</b> Deductive reasoning to generate test case  <b>Phase:</b> Function validation/evaluation  <b>SOI:</b> adaptive cruise control, active lane changing control and active lane keeping control for L2 function.</p>	<p>The scenario includes:</p> <ul style="list-style-type: none"> <li>• structured road (straight, curved and ramp sections of urban road and highway)</li> <li>• ego-vehicle and its intension</li> <li>• intension (9 possibilities such as going straight, left/right lane change, left/right</li> </ul>	The scenario importance is used as main criteria to judge the test value of the designed test scenarios. The author defined four level form to classify the scenario importance based on their	<p><b>Input:</b> function features analysis and specified traffic condition.  <b>Output:</b> valuable test scenarios (logical scenario)  Method:  Step1: define road structure and condition of the ego vehicle</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- Function feature</li> </ul> <p><b>Validation:</b> the coverage of generated valuable scenarios was validated with test regulations of level 1 automated vehicles, the relevant traffic accidents database, the classification of test</p>

	<p><b>ODD:</b> straight, curved and ramp sections of urban road and highway This paper presented a deductive method of testcase generation for the function validation.</p>	<p>lateral moving, and left/right U turn) and relative position (in the paper the author uses 8 possible positions around the ego-vehicle) of surrounding vehicles.</p> <p><b>Covered layers:</b> 1, 4</p>	<p>impact on the ego-vehicle's movement (longitudinal and lateral movement.)</p> <ul style="list-style-type: none"> <li>• Level A: possible collision</li> <li>• Level B: longitudinal and lateral movement impact</li> <li>• Level C: longitudinal or lateral movement impact</li> <li>• Level D: no impact</li> </ul> <p>Levels A, B and C are considered as valuable scenarios.</p>	<p>Step 2: analysis of the scenario importance with one surrounding vehicle (define corresponding levels from A to D), scenarios with an importance degree of D have no test significance and were excluded.</p> <p>Step 3: adding another vehicle to the identified valuable scenarios by step 2 to form new scenarios and analysis of the scenario importance</p> <p>Step 4: adding more vehicles analog to step 3.</p> <p>If the impact of detection and response of ego-vehicle interfered by the movement of multiple obstacle vehicles in the higher-level scenarios is same as the that of lower one level scenarios with one less obstacle vehicle, those higher scenarios are no longer considered as valuable scenarios.</p>	<p>conditions defined by some projects (EuroNCAP (AEB), ISO (ACC), NHTSA, Waymo, ADaptIVe and GES 2015)</p>
[74]	<p><b>Activity:</b> Deductive reasoning method to refine the given logical scenario <b>Phase:</b> Function validation</p> <p>This paper proposes a methodology for an intelligent selection of relevant scenarios for the certification of automated vehicles.</p> <p>The aim of this paper enables the technical service to carry out scenarios that are relevant from its perspective, thus preventing so-called gaming of tests, and at the same time perform an efficient evaluation of the vehicle to be tested.</p>	<p>The authors only proposed a framework. No concrete scenario model was proposed.</p>	<p>In this paper, they proposed two concepts, complex scenarios and critical scenario. This paper uses TTC (Time-To-Collision) as a metric for the criticality definition using simulation.</p> <p>The following factors are considered to increase the complexity of the logical scenario:</p> <ul style="list-style-type: none"> <li>• Number of elements</li> <li>• Number of states per element</li> <li>• Interdependency</li> <li>• Self-dynamics</li> <li>• Intransparency</li> <li>• Multiple conflicting goals</li> <li>• Openness of the target situation</li> <li>• Novelty</li> </ul>	<p><b>Input:</b> the logical scenarios will be taken from: laws, regulations and system specification <b>Output:</b> relevant concrete scenarios</p> <p>According to the methodology first the logical scenarios are taken from three sources.</p> <p>They proposed a two-stage optimization framework to generate concrete scenario. A detailed optimization method was not proposed. In the first optimization stage the parameters of layer one, two and five (refer to 5-layer model) are first optimized by sensor analysis and consideration of driving behavior. In addition, the trajectory of the potential conflict partner (L4) is determined. In the second stage, further</p>	<p><b>Required knowledge:</b> System specification, specific driver safety training information and driving license directive, understanding of complex scenarios</p> <p><b>Limitation:</b> With the method presented, it is not possible to determine all parameters of the five layers model precisely, because it is not entirely possible to determine the relevance of a parameter for a scenario.</p> <p>The method developed does not cover the entire parameter space there is no guarantee that all errors of the SUT will be detected.</p>

				objects are defined (to refine the logical scenario) by considering complexity and their trajectories are optimized.	
--	--	--	--	--	--

Table 7. Inductive reasoning methods

#	Purpose	Scenario definition	Criticality definition& Surrogate Measure	Solution	Other key observations
[75]	<p><b>Activity:</b> Reasoning <b>Phase:</b> design + V&amp;V <b>SOI:</b> the whole system</p> <p>This paper proposes a method to identify critical scenarios from textual accident database through text-weight analysis.</p>	<p>The purpose of this paper is to find critical functional scenarios. In their example method, a functional scenario is a template sentence to combine the identified high-frequent wordings, or a variance of this template sentence.</p> <p>The template sentence in this paper is “A crash is likely to occur when ‘maneuvers’ during ‘traffic operations’ at ‘crash location’ due to ‘triggering event’.”” Where the words in “ needs to be replaced by one wording in the corresponding category.</p>	<p>In this paper, a critical scenario means a crash scenario.</p> <p>The proposed method also defines a scope for the CSI. E.g., in this paper, the scope is urban scenarios.</p>	<p><b>Input:</b> textual accident database (detailed functional scenarios) <b>Output:</b> critical functional scenarios</p> <p>In the database, information for each scenario includes time, location, vehicle and driver details, before-crash maneuver, triggering events, crash descriptions of police officers, etc.</p> <p>Process:</p> <ol style="list-style-type: none"><li>1) Scenarios in the database are first filtered according to the scope of interest (e.g., urban scenarios).</li><li>2) Text weight analysis (searching for frequent wordings)</li><li>3) Select the high-frequent wording</li><li>4) Categorizing wordings by types</li><li>5) Scenario developments as descriptive sentences based on the sentence template.</li></ol>	<p>Required knowledge:</p> <ol style="list-style-type: none"><li>1) Crash scenario database</li></ol> <p>Validation:</p> <p>This proposed Big Data technique-based method is validated in comparison with the resulting test scenarios from a manual investigation of crash data, and it was found that 14 of a total of 18 scenarios correspond to the scenarios from manual investigation and the other four scenarios are additionally derived by the proposed approach.</p>
[76] – [78] AC3 R	<p><b>Activity:</b> generating critical concrete scenarios from accident databases (functional scenarios) <b>Phase:</b> V&amp;V <b>SOI:</b> the whole AD system</p> <p>This paper proposes a method to parse each entry of an accident database to transform it into a simulation scenario for the purpose of test case generation.</p>	<p>Covered layers: 1, 4, 5</p> <p>In this paper, a scenario is defined based on a given topology, which contains the environment (weather, lighting, and roads), the traffic participants and their actions (e.g., movements), as well as the accidents.</p>	<p>A scenario is critical if in the simulation, the ego vehicle (i.e., the system under test) cannot avoid the accident extracted from the accident database.</p>	<p><b>Input:</b> accident database (set of functional accident scenarios) <b>Output:</b> a set of concrete critical scenarios</p> <p>Simply speaking, this approach has three steps:</p> <ol style="list-style-type: none"><li>1) Parse each accident description according to a pre-defined scenario ontology with Natural Language Processing (NLP) techniques.</li><li>2) According to the parsed information, generate simulations which can represent the accident scenarios depicted in the database. Trajectories of</li></ol>	<p><b>Required Knowledge:</b></p> <ol style="list-style-type: none"><li>1) Accident database (NHTSA)</li><li>2) A simulator (BeamNG.research)</li><li>3) Model of the SOI as a black box.</li></ol> <p>Validation:</p> <p>The approach is validated from 4 perspectives:</p> <ol style="list-style-type: none"><li>1) How many simulations can be successfully generated from accident descriptions? This is answered by checking if a simulation can be generated from a description and comparing the</li></ol>

				<p>dynamic objects are modeled as a vector of way points.</p> <p>3) Replacing one vehicle in the simulation with the ego-vehicle and check if its behavior passes the oracle, which is to safely reach a goal point within a given time. The goal point is behind the accident point to guarantee that the ego vehicle avoids the accident.</p>	<p>simulated damage with the recorded damage.</p> <p>2) How well the generated scenarios represent the description? This is answered by an empirical study, people are asked to read the description and the generated simulation videos to provide comments.</p> <p>3) The computation time for simulation generation is acceptable.</p> <p>4) They compare the critical test cases generated by AC3R with non-critical test cases in terms of code coverage and neuron coverage and critical scenarios outperform non-critical ones.</p> <p>5) The effectiveness of AC3R is validated by comparing the test results on critical scenarios and non-critical scenarios in terms of fault (unavoidable crashes) detection.</p>
[79]	<p><b>Activity:</b> Inductive reasoning method to generate critical scenario</p> <p><b>Phase:</b> System verification</p> <p>This paper proposes a scenario analysis and simplification method for reducing the number of test scenarios for HAV test and evaluation.</p>	<p>Covered Layers: all layers (qualitative analysis)</p> <p>The scenario is defined by SCPs (scenario characteristic parameters). The SCPs include qualitative parameters and quantitative parameters.</p> <p><b>Quantitative parameters:</b> the number of interfering vehicles, the initial position and initial speed of each interfering vehicle, the initial position and initial speed of the VUT, the number of lanes, and the curvature of the lane.</p> <p><b>Qualitative parameters:</b> vehicle type (passenger or commercial vehicle), road facilities conditions (such as if road markings and traffic signs are clear and</p>	<p>A trajectory is critical if it leads to a collision or an unfinished task.</p> <p>A scenario is critical if there exist a possible critical trajectory. In other words, a scenario is critical if it is possible to have a collision in this scenario due to FuSa or SOTIF problem of the ego vehicle.</p>	<p><b>Input:</b> a set of concrete scenarios</p> <p><b>Output:</b> the minimum group of scenarios covering the test content of the initial scenarios</p> <p><b>Method:</b></p> <p>First, input the concrete scenarios set that needs to be analyzed. Then, by analyzing concrete scenarios through the traversal of trajectories, trajectories which lead to collisions or test tasks uncompleted are obtained. By analyzing these trajectories, the SCPs of the corresponding scenario are obtained using functional decomposition [80], combined with the fault tree analysis (FTA). By analyzing the overlap or contain relationship among the SCPs, the inclusion</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"><li>- Logical scenario</li><li>- FTA analysis</li></ul> <p>No validation available in paper.</p>

		unobstructed), lighting conditions (normal lighting, backlighting, streetlights, darkness, etc.), weather conditions (sunny, fog, rain, snow, etc.), road conditions (dry, stagnant water, snow, ice, etc.).		relationship among scenarios are obtained according to the SCPs included in different scenarios. By searching for the combination that contains the fewest scenarios but still cover all the SCPs and use this set of scenarios to replace the original combination of test scenarios, the redundant evaluation scenarios were deleted. Therefore, this method further reduces the number of test scenarios.	
[81]	<p><b>Activity:</b> Inductive reasoning to generate pre-crash scenario from accident data base</p> <p><b>Phase:</b> Function validation</p> <p><b>SOI:</b> Autonomous Emergency Braking (AEB) system.</p> <p>This paper presented an inductive method (based on scenario clustering) of testcase (logical scenario) generation for the AEB system based on the accident data.</p>	The scenario description in the paper includes environmental parameters, static parameters, and the dynamic parameters of traffic participants. The paper presented 11 fields related to scenario features: subject participant, object participant, accident type, collision type, road type, road condition, road separation, light condition, weather condition, participant type, speed.	In this paper, critical scenarios refer to potential accident scenarios.	<p><b>Input:</b> Accident data</p> <p><b>Output:</b> Set of typical pre-crash (logical) scenarios</p> <p>First, data from the IGLAD database were analyzed and divided into four categories based on differences in traffic environments among countries and regions. In each group scenarios were clustered.</p> <p>Clustering method:</p> <ul style="list-style-type: none"><li>• Feature selection: after a comparative analysis 11 fields related to scenario features were presented</li><li>• Feature representation using concept of “flip distance”</li><li>• Clustering analysis using SPSS</li></ul> <p>One typical scenario was chosen in each cluster. Chi-square was used to select typical scenario. If an actual chi-square value was found to be greater than the standard value, the value of the variable was considered significant, and a variable with a high proportion was selected as a scenario element. Otherwise, a variable with a large number was selected as a scenario element.</p> <p>In total 21 typical pre-crash scenarios were found.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"><li>- Accident data (IGLAD)</li></ul> <p><b>Validation:</b></p> <p>The found typical pre-crash scenarios were compared with the EuroNCAP AEB scenarios. The typical scenarios mined in the paper were highly consistent with scenarios mapped by Euro-NCAP.</p>

[82]	<p><b>Activity:</b> Inductive reasoning to generate test case from accident data base</p> <p><b>Phase:</b> System design</p> <p><b>SOI:</b> Collision Warning System at intersection.</p> <p>This paper presented an inductive method of testcase generation for the Collision Warning system based on the accident data. The corresponding scenarios were the collision between two vehicles at the intersection.</p>	<p>Main data source for this study is from Naturalistic Driving Study (NDS), a database developed by a research project Second Strategic Highway Research Program (SHRP2) which is supported and operated by the Virginia Tech Transportation Institute (VTTI). 16 scenario classes were defined according to the intentions of two involved vehicles.</p>	<p>This paper uses Safety- remaining distance (SRD) for the scenario evaluation. It is based on the concept of safety stopping distance, as a quantitative indicator to measure their performance.</p>	<p><b>Input:</b> Accident data (two sensor system: camera-based and radar-based data)</p> <p><b>Output:</b> evaluation of the two sensor systems under 16 predefined pre-crash scenario classes</p> <p>The author first selected 363 accident scenarios from two sets of intersection accident data. All these selected scenarios were classified into sixteen vehicle-to-vehicle pre-crash classes. Corresponding SRD were calculated to each scenario for two sensor system. If the SRD is greater than zero, it means that it is possible to prevent an accident. Prevention rate of each system in different scenario class was calculated. The prevention rate was then compared with the compliance rate of each scenario to evaluate which sensor system is necessary for each scenario class.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>• Accident data from Naturalistic Driving Study (NDS)</li> <li>• 16 pre-crash scenario classes at</li> </ul> <p>No validation.</p>
[83]	<p><b>Activity:</b> Inductive reasoning to generate test case from accident data base</p> <p><b>Phase:</b> Function validation</p> <p><b>SOI:</b> Autonomous Emergency Braking (AEB) system.</p> <p>This paper presented an inductive method of testcase generation for the AEB system based on the accident data.</p>	<p>The scenario features in the paper include: Time, Time of crash, obstruct, visual obstruction, intention of car, pre-crash driving behavior of the car driver, intention of the TW and pre-crash driving behavior of the TW driver.</p>	<p>Pre-crash scenarios</p>	<p><b>Input:</b> Accident data</p> <p><b>Output:</b> Six typical car-to-TW (two-wheels) scenarios.</p> <p>Clustering method:</p> <ul style="list-style-type: none"> <li>• Feature selection: five features were selected according to experience</li> <li>• Correlation analysis between the features was calculated to show that the selected features were independent enough for clustering.</li> <li>• K-modes method was applied to cluster the scenarios in the database.</li> </ul> <p>Medoid scenarios (i.e., the typical scenario in each cluster), created from clustering algorithm, were combinations of these five crash characteristics.</p>	<p><b>Required knowledge:</b></p> <p>Accident data</p> <p>Validation:</p> <p>The found six typical pre-crash scenarios were compared with the EuroNCAP AEB 2018 scenarios. The scenarios identified by the author covered all the scenarios in EuroNCAP. Furthermore, the identified scenarios included scenarios in which the car or the TW was turning, while Euro NCAP scenarios only include cars or the bicyclist moving straight ahead.</p>

[84]	<p><b>Activity:</b>  <b>Phase:</b> Function verification  <b>ODD:</b> urban or motorway  <b>SOI:</b> ACC, AEB and LKA</p> <p>This paper presented a systematic approach for assessing the effectiveness of ADAS using accident data.</p>	<p>In this paper a concrete scenario is an entry of GIDAS PCM (precrash matrix), which contains information on environment and vehicle dynamics for the precrash phase of 5 s before the first impact for selected GIDAS cases.</p>	<p>Criticality KPIs:</p> <ul style="list-style-type: none"> <li>Time gap (longitudinal front vehicle distance divides ego vehicle velocity) for ACC</li> <li>TTC (time-to-collision) for AEB</li> <li>TTLC (time-to-line-crossing) and DLC (distance-to-line crossing) for LKA</li> </ul> <p><b>Green spots:</b> the situation can be solved by the system.  <b>Grey spots:</b> the situation cannot be solved by the system, unless system boundaries can be extended.  <b>White spots:</b> The situation cannot be addressed by the system in either basis or extended configuration.</p>	<p><b>Input:</b> Accident database  <b>Output:</b> set of concrete critical scenarios</p> <p>The author first selected scenarios from the accident database based on injury severity score (&gt; 2.5). The selected scenarios were then simulated to calculate criticality KPIs. Three classes (green, grey and white spots) were categorized based on the simulation results.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>Accident data (GIDAS PCM)</li> <li>A simulator (rateEFFECT)</li> <li>System as black-box</li> </ul> <p>No validation.</p>
[85]	<p><b>Activity:</b> Inductive reasoning method to generate concrete crash scenario  <b>Phase:</b> Function V&amp;V  <b>ODD:</b> urban traffic</p> <p>This paper proposes a method based on Microscopic Traffic Simulation to obtain a large number of urban traffic scenarios.</p>	<p>Scenario on microscopic traffic level (position, x,y speed. posLat, speedLat)</p>	<p>The Scenario Risk Index calculation: the magnitude of risk is equal to the probability of a risk event times the degree of loss. The loss calculation was based on the kinetic energy of collision.</p>	<p><b>Input:</b> public traffic data for MTS (microscopic traffic simulation) parameterization  <b>Output:</b> concrete crash scenarios</p> <p>Method described in the paper:</p> <ol style="list-style-type: none"> <li>Using public data to calibrate the traffic simulator.</li> <li>Perform simulation.</li> <li>Data postprocessing of the simulation results to extract the concrete crash scenarios. Each vehicle in the simulation is regarded as Ego.</li> </ol> <p>The calculation of SRI (scenario risk index).</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>Traffic data</li> <li>SUMO</li> </ul> <p>No validation.</p>
[86] VAA FO	<p><b>Activity:</b> creating the database of critical concrete scenarios  <b>Phase:</b> the database can be used for statistical validation / system verification  <b>SOI:</b> the whole AD system</p> <p>Virtual Assessment of Automation in Field Operation (VAAFO) tool runs in parallel with human driving. It</p>	<p>In this paper, a scenario is defined by all the sensor data collected by a human-driving vehicle. The vehicle is equipped with all the sensors that the AD function needs. The logged sensor data will be processed off-line to represent the real world more correctly. The corrected world model includes constant objects and the behaviors of the dynamic objects.</p>	<p>The logged decision made by the AD function during runtime will be mapped to the corrected perceived world model.</p> <p>The paper says that for the assessment module, the simplest measure is whether a real collision has occurred or not. Further, multiple metrics for preventive or active safety can be implemented. Those can range from the safety in the current</p>	<p><b>Input:</b> human driving  <b>Output:</b> critical concrete scenarios</p> <p>The AD functions make decisions based on the on-line world model and this decision is verified on the corrected world model.</p> <p>The AD function may make a decision that is different from the human driver's. This difference may consequently affect the</p>	<p><b>Required Knowledge:</b></p> <ol style="list-style-type: none"> <li>Human driving sensor data</li> <li>Simulator running online to calculate the planned behavior of the AD functions.</li> </ol> <p><b>Validation:</b> not mentioned</p>

	<p>receives information from all the sensors, but it is not connected to the actuator. AD functions are running (simulating) within the perceived world. The trajectory planned by the AD functions is compared with human driving trajectory.</p> <p>This paper argues that, accident scenarios made by human drivers will be recorded by the police. This method will record the scenarios that are risky for ADs but not for human drivers.</p>	<p>The perceived world model (after off-line correction) is more accurate than the online version because:</p> <ol style="list-style-type: none"> <li>1) Further measurement can be used.</li> <li>2) More complex sensor fusion algorithm can be used since there is no time limit for computation.</li> <li>3) The driver is in the loop to correct manually.</li> </ol> <p>The AD function is making decision during runtime based on the sensor inputs. Its decision is also logged.</p>	<p>motion state (TTC, constant reserve time, etc.) to motion prediction models (physics-based, maneuver-based, and interaction-aware). However, no particular metrics are given.</p>	<p>behavior of other vehicles. To solve this, VAAFO shorten the duration of each simulation to 2 seconds, and a new simulation is initiated every 0.5 second. VAFFO assumes that the behaviors of other vehicles will not be significantly affected by a different decision made by the AD within 2 seconds.</p>	
[87]	<p><b>Activity:</b> Refinement of concrete scenarios  <b>Phase:</b> System verification  <b>SOI:</b> The whole system</p> <p>The purpose of the study is to generate new collision scenarios from already recorded ones.</p> <p>For that, the authors propose to train Recurrent Neural Networks (RNNs) to generate new collision scenarios.</p>	<p><b>Covered layer:</b> 2, 4</p> <p>They use “speed and direction of vehicles for in-vehicle data and traffic light data to represent V2X data”.</p> <p>Layer 4 is represented by the speed and direction of other movable objects.</p> <p>Layer 2 is represented by the traffic light data</p> <p>They mention V2X data, which would be layer 6, but they only use traffic light information.</p>	<p>In this study they only consider scenarios where there is a collision.</p> <p>They do not consider a specific system under test for this process. The aim of the study is to generate new accident scenarios from prerecorded data using a RNN.</p>	<p><b>Input:</b> A set of concrete scenarios  <b>Output:</b> A set of generated concrete scenarios</p> <p>To generate new collision data from prerecorded data using ANN, they use Long Short Term Memory Networks (LSTM). The data used to train the network comes from a simulation environment, but it could be real accident data. In the example developed in the study, the data includes speed, direction of vehicles and traffic light data.</p> <p>Once trained, the network can be used to generate new collision data starting from an initial seed that contains the initial speed, direction and traffic light state.</p>	<p><b>Required knowledge:</b></p> <ul style="list-style-type: none"> <li>- Recurrent neuronal networks (RNNs)</li> <li>- Accident data or a simulator to generate accident data</li> </ul> <p><b>Validation:</b></p> <p>The generated scenarios are compared with the original scenarios in terms of:</p> <ul style="list-style-type: none"> <li>- The distance travelled from both vehicles.</li> <li>- The total speed at collision from both vehicles.</li> </ul> <p>The cumulative angular change from both vehicles.</p>

Table 8. Formal methods

#	Purpose	Scenario definition	Criticality definition& criteria	Solution	Other key observations
[88]	<p><b>Activity:</b> Formalization &amp; criticality check</p> <p><b>Phase:</b> requirement analysis (verification) or component design (early-phase specification verification)</p> <p><b>SOI:</b> the decision-making function (implicit)</p> <p><b>ODD:</b> structured road</p> <p>At requirement analysis phase, this method can <b>formally verify</b> if the specifications of the AD functions are safe within the predefined scenario catalogue. If it is not safe, a counter example (i.e., a critical scenario) will be provided.</p>	<p>Definition of scenario is not explicitly given in the paper.</p> <p>Our Interpretation: A scenario is defined by the lane structure and the behaviors of all the involved objects.</p> <p>A scenario catalog is defined by the lane structure and all the possible behaviors (including interactions with other objects) of objects other than the ego vehicle. This is the closest definition to logical scenario.</p> <p>Scenario specification: A formal way to model acceptable behaviors of the ego vehicle within a scenario catalog.</p> <p><b>Covered layers:</b> 1 (fixed for each logical scenario),4 (including the ego vehicle)</p>	<p>Criticality is defined based on the formally defined scenario specifications.</p> <p>Critical scenarios are the ones where the AD function cannot satisfy the corresponding scenario specification. (According to our understanding, scenario specifications can be either functional requirement or safety constraints)</p> <p>The AD function needs to be modeled (maybe simplified) as a hybrid automaton.</p>	<p><b>Inputs:</b> Executable (logical scenario) model for model checking <b>Output:</b> pass or a counter example (a simulated scenario)</p> <p>Scenario specifications are modeled as traffic sequence charts. The behaviors of all the objects and their interactions with the ego vehicle are modeled as hybrid automata. These two are input to a model checker, which consequently evaluates if the AD function of the ego vehicle satisfy all the scenario specifications. If not, a critical scenario will be returned.</p> <p>Model checking can guarantee <b>full coverage on the parameter space</b>.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>Behaviors and reactions of other vehicles</li> <li>Specifications of the decision making and control functions.</li> <li>Dynamic model of the ego-vehicle</li> <li>Lane structure of the road</li> <li>Scenario specifications</li> </ol> <p><b>Validation:</b> No Validation</p>
[89]	<p><b>Activity:</b> Instantiation &amp; criticality check (falsification)</p> <p><b>Phase:</b> early-phase component verification</p> <p><b>SOI:</b> Decision making + motion planning + control</p> <p>This paper provides a <b>formal</b> way to describe the scenarios, the behaviors of the vehicles and the safety requirements. The formalization is used for falsification (i.e., to find scenarios that violates the safety requirement).</p>	<p><b>Covered layers:</b> 1,2 (traffic laws),4</p> <p>In this paper, a logical scenario contains the road structure (including the possible variation), the behavior models of the traffic objects (including the ego vehicle),The traffic laws, the goal of the ego vehicle (i.e., its destination), the initial states and the exit condition (it can be either time triggered or event triggered).</p> <p>All the Behaviors of the traffic objects are modeled in one hybrid automaton.</p> <p>A logical scenario can be instantiated into a concrete scenario by giving fixed values to the initial states.</p>	<p>A critical scenario is a concrete scenario where the ego vehicle violates the traffic laws.</p>	<p><b>Inputs:</b> Executable (logical scenario) model for model checking <b>Output:</b> pass or a counter example (a simulated scenario)</p> <p>The falsification process: The formal scenario models will be translated to suitable formats for two falsification tools, S-TALIRO and dReach.</p> <p>S-TALIRO is a statistical model checking tool which gives robustness values of the satisfaction, denoting how good it satisfies (positive value) the requirement or how bad it violates the requirement (negative value). Scenarios with low</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>Behaviors of other traffic objects modeled by a hybrid automaton.</li> <li>The decision making and control algorithms of the ego vehicle modeled in a hybrid automaton.</li> <li>The falsification tools, S-TALIRO and dReach.</li> </ol> <p>Validation: Three case studies are conducted to show that this method can detect controller faults.</p>

		<p>The traffic laws (e.g., speed limitations) are modeled by Metric Temporal Logic (MTL).</p>		<p>positive robustness values can also be considered as risky scenarios.</p> <p>dReach is based on model checking. It will provide either an assertion “safe” or a counterexample that violates the requirement.</p> <p>S-RALIRO runs faster than dReach, however, S-RALIRO cannot guarantee completeness.</p>	
--	--	---	--	--	--

Table 9. Other methods

#	Purpose	Scenario definition	Criticality definition& Surrogate Measure	Solution	Other key observations
[90] DriveFI	<p><b>Activity:</b> Simulation + fault injection <b>Phase:</b> HARA for the preliminary architecture or the verification of the resilience of the AD system. <b>SOI:</b> the whole system</p> <p>DriveFI is a fault injection engine to fast identify safety critical faults of the AD systems and the corresponding critical scenes. It can be used for the safety analysis (e.g., HARA) phase.</p> <p>Fault injection is <b>out of the scope</b> of this literature review.</p>	<p>A scenario in this paper refers to one simulation run with fixed time interval. Examples of simulator include Carla and DriveSim.</p> <p>A scene is a snapshot of one scenario at a specific time instance.</p> <p>Detailed model of a scenario is not given in this paper.</p> <p><b>Covered layers:</b> mainly focus on the implementation. Other layers are provided by the simulator (e.g., 1, 2, 4, 5).</p>	<p>If a fault was injected to the AD system at a particular time instance in one driving scenario, and this fault will lead to an accident, this fault is defined as a safety critical fault and the scene at this time instance is defined as a safety critical scene.</p> <p>To judge the criticality of a scene, it calculates the difference between stop distance and safe distance (the distance that the ego vehicle can travel without a collision) on both lateral and longitudinal directions. If the stop distance is larger than the safety distance, the scene will be considered as critical.</p>	<p>DriveFI will decide which faults to be injected to one scenario at which time instance. If the simulation shows that the injected fault leads to an accident, the fault and the scene are identified as safety critical.</p> <p>The proposed method to select the injected faults:</p> <ul style="list-style-type: none"><li>• A Dynamic Bayesian Network (DBN) model is designed and trained to represent the relation between relevant states (internal and external) and the controlled values (e.g., steering angle, acceleration and brake)</li><li>• A simulation trace will be derived by running the ego vehicle in the given concrete scenario without any injected faults.</li><li>• At each time step of this simulation trace, the states will be given to the trained DBN model together with an injected fault. The DBN model will accordingly estimate the control output of the ego vehicle for the next time step.</li><li>• The behavior of the ego vehicle at the next time step is estimated via a vehicle dynamic model. Based on this behavior, criticality will be checked. If it is critical, this fault and this time step will be logged.</li></ul>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) A given concrete scenario</li><li>2) Functional models of the ADS</li><li>3) Fault model of the ADS</li><li>4) A simulator (Carla &amp; DriveSim)</li><li>5) Predefined KPIs</li></ol> <p><b>Validation:</b></p> <p>In a case study, comparing to random FI, DriveFI finds much more safety-critical faults with much less time.</p> <p>Scenarios are pre-defined. DriveFI can only find safety critical scenes from the given scenario.</p>
[91]	<p><b>Activity:</b> Reasoning &amp; ontology design (influential factors analysis) <b>Phase:</b> system validation</p> <p>This article lists the factors that need to be considered during the</p>	<p>Scenario is not explicitly defined in this paper. The validation space can be represented with four dimensions, namely ODD, OEDR (Object and Event Detection and Response), Maneuvers, and Fault Management. ODD relates to the</p>	<p>The main purpose of the proposed approach is to find emergent effects that cause some combinations of those factors to cause unexpected and dangerous results.</p>	<p>During validation process, experts should check all the combinations of factors within the four-dimensional validation space.</p>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) Experience with a variety of autonomous vehicle projects</li></ol> <p><b>Validation:</b></p> <p>No validation</p>

	<p>validation (i.e., the validation space) of highly autonomous vehicles in a systematic way, according to the authors' multiple relevant project experience.</p> <p>This paper is considered as <b>out of the scope</b> since it only says what aspects need to be considered to validate automated vehicles, but not a concrete method to identify critical scenarios.</p>	<p>environment, OEDR and Maneuvers relate to the intended functionality of the ego vehicle, where OEDR focuses on the perception and prediction; and the Maneuvers dimension focuses on planning and control.</p> <p>ODD factors include operational terrain, Environmental and weather conditions, operational infrastructure, rules and expectations, and distributions of operational state space elements.</p> <p><b>Covered layers:</b> 1,2,3,4,5</p>			
[80] Fun c. Dec omp	<p><b>Activity:</b> test suite reduction during Instantiation  <b>Phase:</b> system V&amp;V  <b>SOI:</b> the whole system</p> <p>Given a logical scenario, the proposed method tries to reduce the number of generated test cases, while guaranteeing N-wise coverage.</p> <p>The method decomposes the whole AD/ADAS function into several sub-functions. Each sub-function will be affected by only a subset of the total factors of a logical scenario. Testing the sub-functions one by one may reduce the total number of required test cases w.r.t. coverage.</p> <p>It is <b>out of the scope</b> of the survey, since it does not talk about the identification of critical scenarios. However, this topic is highly relevant.</p>	<p><b>Covered layers:</b> The examples cover layers 1,4,5.  The proposed test suite reduction method can be applied to any test case generation method as long as a concrete scenario is modeled as a vector of relevant parameters.</p>	<p>Criticality is not clearly defined in this paper.</p> <p>It is not the main focus of this paper.</p>	<p>This paper mentions three principles to reduce the size of the test suite.</p> <ol style="list-style-type: none"> <li>(1) Testing for only one function (e.g., perception, decision making or control) may shrink the parameter space since some parameters are only influential on particular functions. E.g., illumination will not affect the performance of the trajectory following function.</li> <li>(2) Less complex subsystems may require a smaller test coverage.</li> <li>(3) The test of the perception layers can be aggregated for a set of similar scenarios. E.g., vehicle detection function can be tested only once (with one aggregated test suite) for all the logical scenarios on a two-lane highway, since they assume that the perception functions will be affected by the same set of factors within those scenarios.</li> </ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"> <li>(1) Which parameters are (or are not) influential for which functions.</li> <li>(2) Which function may request a lower test coverage</li> </ol> <p><b>Validation:</b>  This method is verified on 9 logical scenarios. The results show that the functional decomposition approach potentially reduces the size of the required test suite by a factor of 20, ..., 130, depending on the required test coverage.</p> <p>It is not verified if the reduced test suite can still find the same number of hazards.</p> <p>This paper points out that the majority of parameters with a high number of possible values only have an influence on the perception functions.</p>

[92]	<p><b>Activity:</b> Assessment <b>Phase:</b> system validation <b>SOI:</b> the whole system</p> <p>The paper is <b>out of the scope</b>. It talks about the assessment of automated driving function without the identification of critical scenarios. However, it provides an interesting perspective to evaluate automate driving function.</p> <p>a scenario-based assessment approach based on real world driving field data of both human driving and automated driving.</p> <p>The method uses human driving behavior as a reference for the assessment of automated driving.</p>	<p><b>Covered layer:</b> 4 + the behavior of the ego vehicle</p> <p>In the scenario classification method, a scenario has three representations: (1) time series sensor data; (2) extracted feature; and (3) classified functional scenarios (i.e., the labels, e.g., cut-in, lane change).</p> <p>Suitable features are: 1. Extended features (e.g. criticality-indicators such as Time-To-Collision (TTC) or the estimated time to the next Cut-In maneuver of a traffic participant) from real world driving data; 2. Derivatives of extended features; 3. Segmentation of the extended features.</p> <p>The finally used features are selected by the filter and wrapper method.</p> <p>In this paper, they use field operation test data.</p>	<p>This paper does not explicitly talk about criticality.</p> <p>The AD function under a functional scenario is assessed according to its behavior deviation from the human driving behavior.</p> <p>The deviation is assessed on some predefined performance KPIs. In the vehicle following scenario <b>longitudinal acceleration</b> and <b>time headway</b> are used.</p> <p>To quantify the deviation from normal driving behavior, the quantitative measure ‘<b>effect size</b>’ (calculated based on the means and variances of the performance KPIs) is proposed in the paper. An effect size less than <math>d_{small} = 0.2</math> can be considered as a ‘small’ influence.</p>	<p><b>Assessment process:</b></p> <ol style="list-style-type: none"><li>(1) a scenario classifier is trained by expert knowledge. The input feature is extracted from time series sensor data. The scenario classes are labelled by experts.</li><li>(2) Both human driving data and automated driving data are classified into a set of predefined classes (i.e., free driving, vehicle following, lane change and cut in)</li><li>(3) Under each class, the differences (in terms of the predefined KPIs) between human driving behavior and automated driving behavior are used as the metrics to assess the automated driving function.</li></ol>	<p><b>Required knowledge:</b></p> <ol style="list-style-type: none"><li>1) Scenarios and scenario classes</li><li>2) Real driving data/measurement for both human driving and automated driving</li><li>3) Expert knowledge to label scenarios.</li><li>4) Expert knowledge to design the feature.</li><li>5) KPIs for the statistical comparison between human driving behavior and automated driving behavior under one scenario class (i.e., functional scenario)</li></ol> <p><b>Validation:</b> A case study is used to evaluate the performance of the scenario classifier and the show the assessment result.</p>
------	---	--	---	---	--

## Reference:

- [1] Y. Li, J. Tao, and F. Wotawa, "Ontology-based test generation for automated and autonomous driving functions," vol. 117, no. October 2019, 2020, doi: 10.1016/j.infsof.2019.106200.
- [2] J. Tao, Y. Li, F. Wotawa, H. Felbinger, and ..., "On the industrial application of combinatorial testing for autonomous driving functions," *2019 IEEE Int. ...*, 2019, [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8728928/>.
- [3] F. Klueck, Y. Li, M. Nica, J. Tao, and F. Wotawa, "Using Ontologies for Test Suites Generation for Automated and Autonomous Driving Functions," in *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Oct. 2018, pp. 118–123, doi: 10.1109/ISSREW.2018.00-20.
- [4] T. Ponn, D. Fratzke, C. Gndt, and M. Lienkamp, "Towards Certification of Autonomous Driving: Systematic Test Case Generation for a Comprehensive but Economically-Feasible Assessment of Lane Keeping Assist Algorithms," in *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2019, no. Vehits, pp. 333–342, doi: 10.5220/0007678603330342.
- [5] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, vol. 2018-June, no. Iv, pp. 1555–1562, doi: 10.1109/IVS.2018.8500421.
- [6] C. E. Tuncali, G. Fainekos, D. Prokhorov, H. Ito, and J. Kapinski, "Requirements-Driven Test Generation for Autonomous Vehicles with Machine Learning Components," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 265–280, Jun. 2020, doi: 10.1109/TIV.2019.2955903.
- [7] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, and S. K. Gupta, "Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles," *J. Syst. Softw.*, vol. 137, pp. 197–215, Mar. 2018, doi: 10.1016/j.jss.2017.10.031.
- [8] M. Nabhan, M. Schoenauer, Y. Tourbier, and H. Hage, "Optimizing coverage of simulated driving scenarios for the autonomous vehicle," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Nov. 2019, pp. 1–5, doi: 10.1109/ICCVE45908.2019.8965211.
- [9] F. Batsch, A. Daneshkhah, M. Cheah, S. Kanarachos, and A. Baxendale, "Performance Boundary Identification for the Evaluation of Automated Vehicles using Gaussian Process Classification," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 419–424, doi: 10.1109/ITSC.2019.8917119.
- [10] J. Zhou and L. del Re, "Safety Verification Of ADAS By Collision-free Boundary Searching Of A Parameterized Catalog," in *2018 Annual American Control Conference (ACC)*, Jun. 2018, pp. 4790–4795, doi: 10.23919/ACC.2018.8431291.
- [11] R. Ben Abdesslem, S. Nejati, L. C. Briand, and T. Stifter, "Testing vision-based control systems using learnable evolutionary algorithms," *Proc. 40th Int. Conf. Softw. Eng. - ICSE '18*, pp. 1016–1026, 2018, doi: 10.1145/3180155.3180160.
- [12] F. Klück, M. Zimmermann, F. Wotawa, and ..., "Genetic algorithm-based test parameter optimization for ADAS system testing," *2019 IEEE 19th ...*, 2019.

- [13] F. Klück, M. Zimmermann, F. Wotawa, and M. Nica, "Performance Comparison of Two Search-Based Testing Strategies for ADAS System Validation," *IFIP Int. Conf. ...*, 2019, [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-030-31280-0\\_9](https://link.springer.com/chapter/10.1007/978-3-030-31280-0_9).
- [14] S. Hallerbach, Y. Xia, U. Eberle, and F. Koester, "Simulation-Based Identification of Critical Scenarios for Cooperative and Automated Vehicles," *SAE Int. J. Connect. Autom. Veh.*, vol. 1, no. 2, pp. 2018-01-1066, Apr. 2018, doi: 10.4271/2018-01-1066.
- [15] F. Gao, J. Duan, Y. He, and Z. Wang, "A Test Scenario Automatic Generation Strategy for Intelligent Driving Systems," *Math. Probl. Eng.*, vol. 2019, pp. 1-10, Jan. 2019, doi: 10.1155/2019/3737486.
- [16] Q. Xia, J. Duan, F. Gao, T. Chen, and C. Yang, "Automatic Generation Method of Test Scenario for ADAS Based on Complexity," in *SAE Technical Papers*, Sep. 2017, vol. Part F1298, no. September, doi: 10.4271/2017-01-1992.
- [17] Q. Xia, J. Duan, F. Gao, Q. Hu, and Y. He, "Test Scenario Design for Intelligent Driving System Ensuring Coverage and Effectiveness," *Int. J. Automat. Technol.*, vol. 19, no. 4, pp. 751-758, Aug. 2018, doi: 10.1007/s12239-018-0072-6.
- [18] J. Duan, F. Gao, and Y. He, "Test Scenario Generation and Optimization Technology for Intelligent Driving Systems," *IEEE Intell. Transp. Syst. Mag.*, p. 1, 2020, doi: 10.1109/MITS.2019.2926269.
- [19] Y. Akagi, R. Kato, S. Kitajima, J. Antona-Makoshi, and N. Uchida, "A Risk-index based Sampling Method to Generate Scenarios for the Evaluation of Automated Driving Vehicle Safety \*," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 667-672, doi: 10.1109/ITSC.2019.8917311.
- [20] H. Beglerovic, M. Stolz, and M. Horn, "Testing of autonomous vehicles using surrogate models and stochastic optimization," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2018-March, pp. 1-6, 2018, doi: 10.1109/ITSC.2017.8317768.
- [21] S. Feng *et al.*, "Testing Scenario Library Generation for Connected and Automated Vehicles, Part I: Methodology," *IEEE Trans. Intell. Transp. Syst.*, pp. 1-10, May 2020, doi: 10.1109/TITS.2020.2972211.
- [22] S. Feng *et al.*, "Testing Scenario Library Generation for Connected and Automated Vehicles, Part II: Case Studies," pp. 1-13, 2019, [Online]. Available: <http://arxiv.org/abs/1905.03428>.
- [23] S. Feng, Y. Feng, X. Yan, S. Shen, S. Xu, and H. X. Liu, "Safety assessment of highly automated driving systems in test tracks: A new framework," *Accid. Anal. Prev.*, vol. 144, p. 105664, Sep. 2020, doi: 10.1016/j.aap.2020.105664.
- [24] S. Feng *et al.*, "Testing Scenario Library Generation for Connected and Automated Vehicles: An Adaptive Framework," *IEEE Trans. Intell. Transp. Syst.*, pp. 1-12, May 2020, doi: 10.1109/TITS.2020.2972211.
- [25] W. L. Huang, "Accelerate the autonomous vehicles reliability testing in parallel paradigm," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2018, pp. 922-927, 2018, doi: 10.1109/ITSC.2017.8317959.

- [26] E. De Gelder and J. P. Paardekooper, "Assessment of Automated Driving Systems using real-life scenarios," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 589–594, 2017, doi: 10.1109/IVS.2017.7995782.
- [27] S. Cutrone, C. W. Liew, B. Utter, and A. Brown, "A Framework for Identifying and Simulating Worst-Case Animal-Vehicle Interactions," in *Proceedings - 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018*, 2019, pp. 1995–2000, doi: 10.1109/SMC.2018.00344.
- [28] D. Stumper and K. Dietmayer, "Towards Criticality Characterization of Situational Space," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018, vol. 2018-Novem, pp. 3378–3382, doi: 10.1109/ITSC.2018.8569505.
- [29] X. Ma, Z. Ma, X. Zhu, J. Cao, and F. Yu, "Driver Behavior Classification under Cut-In Scenarios Using Support Vector Machine Based on Naturalistic Driving Data," in *WCX SAE World Congress Experience*, Apr. 2019, doi: <https://doi.org/10.4271/2019-01-0136>.
- [30] S. Khastgir, G. Dhadyalla, S. Birrell, S. Redmond, R. Addinall, and P. Jennings, "Test Scenario Generation for Driving Simulators Using Constrained Randomization Technique," in *SAE Technical Paper*, Mar. 2017, doi: 10.4271/2017-01-1672.
- [31] F. Reiterer, J. Zhou, J. Kovanda, V. Rulc, V. Kemka, and L. del Re, "Beyond-Design-Basis Evaluation of Advanced Driver Assistance Systems," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 2119–2124, doi: 10.1109/IVS.2019.8813893.
- [32] A. Gambi, M. Mueller, and G. Fraser, "Automatically testing self-driving cars with search-based procedural content generation," in *ISSTA 2019 - Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 273–283, doi: 10.1145/3293882.3330566.
- [33] F. Hauer, A. Pretschner, and B. Holzmüller, "Fitness functions for testing automated and autonomous driving systems," *Int. Conf. ...*, 2019, [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-030-26601-1\\_5](https://link.springer.com/chapter/10.1007/978-3-030-26601-1_5).
- [34] J. Zhou and L. del Re, "Reduced Complexity Safety Testing for ADAS & ADF," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5985–5990, 2017, doi: 10.1016/j.ifacol.2017.08.1261.
- [35] S. Wagner, A. Knoll, K. Groh, T. Kühbeck, D. Watzenig, and L. Eckstein, "Virtual Assessment of Automated Driving: Methodology, Challenges, and Lessons Learned." Dec. 2019.
- [36] S. Wagner, K. Groh, T. Kuhbeck, M. Dorfel, and A. Knoll, "Using Time-to-React based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, vol. 2018-June, no. Iv, pp. 1521–1528, doi: 10.1109/IVS.2018.8500624.
- [37] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, "Identification of Test Cases for Automated Driving Systems Using Bayesian Optimization," *2019 IEEE Intell. Transp. Syst. Conf. ITSC 2019*, pp. 1961–1967, 2019, doi: 10.1109/ITSC.2019.8917103.
- [38] V. Bithar and A. Karumanchi, "Application of collision probability estimation to calibration of advanced driver assistance systems," *SAE Tech. Pap.*, vol. 2019-April, no. April, 2019, doi: 10.4271/2019-01-1133.

- [39] P. Junietz, F. Bonakdar, B. Klamann, and H. Winner, "Criticality Metric for the Safety Validation of Automated Driving using Model Predictive Trajectory Optimization," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018, vol. 2018-Novem, pp. 60–65, doi: 10.1109/ITSC.2018.8569326.
- [40] C. Gladisch, T. Heinz, C. Heinzemann, J. Oehlerking, A. von Vietinghoff, and T. Pfitzer, "Experience Paper: Search-Based Testing in Automated Driving Control Applications," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Nov. 2019, pp. 26–37, doi: 10.1109/ASE.2019.00013.
- [41] S. Masuda, H. Nakamura, and K. Kajitani, "Rule-based searching for collision test cases of autonomous vehicles simulation," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 1088–1095, 2018, doi: 10.1049/iet-its.2018.5335.
- [42] Z. Huang, H. Lam, and D. Zhao, "Sequential experimentation to efficiently test automated vehicles," in *2017 Winter Simulation Conference (WSC)*, Dec. 2017, pp. 3078–3089, doi: 10.1109/WSC.2017.8248028.
- [43] X. Wang, H. Peng, and D. Zhao, "Combining Reachability Analysis and Importance Sampling for Accelerated Evaluation of Highway Automated Vehicles at Pedestrian Crossing," *ASME Lett. Dyn. Syst. Control*, vol. 1, no. 1, Jan. 2021, doi: 10.1115/1.4046610.
- [44] D. Zhao *et al.*, "Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 595–607, 2017, doi: 10.1109/TITS.2016.2582208.
- [45] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," *IEEE Intell. Veh. Symp. Proc.*, vol. 2019-June, no. Iv, pp. 2352–2358, 2019, doi: 10.1109/IVS.2019.8814230.
- [46] M. Althoff and S. Lutz, "Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles," *IEEE Intell. Veh. Symp. Proc.*, vol. 2018-June, no. Iv, pp. 1326–1333, 2018, doi: 10.1109/IVS.2018.8500374.
- [47] M. Koschi, C. Pek, S. Maierhofer, and M. Althoff, "Computationally Efficient Safety Falsification of Adaptive Cruise Control Systems," *2019 IEEE Intell. Transp. Syst. Conf. ITSC 2019*, pp. 2879–2886, 2019, doi: 10.1109/ITSC.2019.8917287.
- [48] C. E. Tuncali, S. Yaghoubi, T. P. Pavlic, and G. Fainekos, "Functional gradient descent optimization for automatic test case generation for vehicle controllers," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Aug. 2017, vol. 2017-Augus, pp. 1059–1064, doi: 10.1109/COASE.2017.8256245.
- [49] X. Qin, N. Aréchiga, A. Best, and J. Deshmukh, "Automatic Testing and Falsification with Dynamically Constrained Reinforcement Learning," pp. 1–16, 2019, [Online]. Available: <http://arxiv.org/abs/1910.13645>.
- [50] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive Stress Testing for Autonomous Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, vol. 2018-June, pp. 1–7, doi: 10.1109/IVS.2018.8500400.
- [51] A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, "Adaptive Stress Testing with Reward Augmentation for Autonomous Vehicle Validation," in *2019 IEEE Intelligent*

*Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 163–168, doi: 10.1109/ITSC.2019.8917242.

- [52] P. Du and K. Driggs-Campbell, “Finding Diverse Failure Scenarios in Autonomous Systems Using Adaptive Stress Testing,” *SAE Int. J. Connect. Autom. Veh.*, vol. 2, no. 4, pp. 12-02-04–0018, Dec. 2019, doi: 10.4271/12-02-04-0018.
- [53] M. Koren and M. J. Kochenderfer, “Efficient Autonomy Validation in Simulation with Adaptive Stress Testing,” in *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, Oct. 2019, pp. 4178–4183, doi: 10.1109/ITSC.2019.8917403.
- [54] C. E. Tuncali and G. Fainekos, “Rapidly-exploring Random Trees for Testing Automated Vehicles,” *2019 IEEE Intell. Transp. Syst. Conf. ITSC 2019*, pp. 661–666, 2019, doi: 10.1109/ITSC.2019.8917375.
- [55] N. Li, I. Kolmanovsky, and A. Girard, “Model-free optimal control based automotive control system falsification,” in *2017 American Control Conference (ACC)*, May 2017, pp. 636–641, doi: 10.23919/ACC.2017.7963024.
- [56] G. Chou, Y. E. Sahin, L. Yang, K. J. Rutledge, P. Nilsson, and N. Ozay, “Using Control Synthesis to Generate Corner Cases: A Case Study on Autonomous Driving,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2906–2917, Nov. 2018, doi: 10.1109/TCAD.2018.2858464.
- [57] G. Chance, A. Ghobrial, S. Lemaignan, T. Pipe, and K. Eder, “An Agency-Directed Approach to Test Generation for Simulation-based Autonomous Vehicle Verification,” in *2020 IEEE International Conference On Artificial Intelligence Testing (AITest)*, Aug. 2020, pp. 31–38, doi: 10.1109/AITEST49225.2020.00012.
- [58] D. J. Fremont, X. Yue, T. Dreossi, S. Ghosh, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, “Scenic: Language-Based Scene Generation,” Sep. 2018.
- [59] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, “DeepRoad: GAN-based Metamorphic Autonomous Driving System Testing,” Feb. 2018, [Online]. Available: <http://arxiv.org/abs/1802.02295>.
- [60] Y. Tian, K. Pei, S. Jana, and B. Ray, “DeepTest: automated testing of deep-neural-network-driven autonomous cars,” in *Proceedings of the 40th International Conference on Software Engineering*, May 2018, vol. 2018-May, pp. 303–314, doi: 10.1145/3180155.3180220.
- [61] K. Pei, Y. Cao, J. Yang, and S. Jana, “DeepXplore,” *Commun. ACM*, vol. 62, no. 11, pp. 137–145, Oct. 2019, doi: 10.1145/3361566.
- [62] C. Zhang, Y. Liu, Q. Zhang, and L. Wang, “A Graded Offline Evaluation Framework for Intelligent Vehicle’s Cognitive Ability,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, vol. 2018-June, no. Iv, pp. 320–325, doi: 10.1109/IVS.2018.8500622.
- [63] J. Wang, C. Zhang, Y. Liu, and Q. Zhang, “Traffic Sensory Data Classification by Quantifying Scenario Complexity,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, vol. 2018-June, no. Iv, pp. 1543–1548, doi: 10.1109/IVS.2018.8500669.
- [64] J.-A. Bolte, A. Bar, D. Lipinski, and T. Fingscheidt, “Towards Corner Case Detection for Autonomous Driving,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, vol. 2019-June,

no. Iv, pp. 438–445, doi: 10.1109/IVS.2019.8813817.

- [65] S. Yang, W. Deng, Z. Liu, and Y. Wang, “Analysis of Illumination Condition Effect on Vehicle Detection in Photo-Realistic Virtual World,” in *Intelligent and Connected Vehicles Symposium*, Sep. 2017, vol. Part F1298, no. September, doi: 10.4271/2017-01-1998.
- [66] H. Yu and X. Li, “Intelligent corner synthesis via cycle-consistent generative adversarial networks for efficient validation of autonomous driving systems,” in *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, 2018, vol. 2018-Janua, pp. 9–15, doi: 10.1109/ASPDAC.2018.8297275.
- [67] T. Dreossi, A. Donzé, and S. A. Seshia, “Compositional Falsification of Cyber-Physical Systems with Machine Learning Components,” *J. Autom. Reason.*, vol. 63, no. 4, pp. 1031–1053, Dec. 2019, doi: 10.1007/s10817-018-09509-5.
- [68] A. Nayak, S. Rathinam, A. Pike, and S. Gopalswamy, “Reference Test System for Machine Vision Used for ADAS Functions,” in *WCX SAE World Congress Experience*, Apr. 2020, doi: <https://doi.org/10.4271/2020-01-0096>.
- [69] O. Zendel, M. Murschitz, M. Humenberger, and W. Herzner, “CV-HAZOP: Introducing Test Data Validation for Computer Vision,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2066–2074, doi: 10.1109/ICCV.2015.239.
- [70] H. Weber *et al.*, “A framework for definition of logical scenarios for safety assurance of automated driving,” *Traffic Inj. Prev.*, vol. 20, no. sup1, pp. S65–S70, Jun. 2019, doi: 10.1080/15389588.2019.1630827.
- [71] B. Kramer, C. Neurohr, M. Büker, E. Böde, M. Fränzle, and W. Damm, “Identification and Quantification of Hazardous Scenarios for Automated Driving,” vol. 1, 2020, pp. 163–178.
- [72] L. Huang, Q. Xia, F. Xie, H.-L. Xiu, and H. Shu, “Study on the Test Scenarios of Level 2 Automated Vehicles,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 49–54, doi: 10.1109/IVS.2018.8500600.
- [73] F. Xie, T. Chen, Q. Xia, L. Huang, and H. Shu, “Study on the Controlled Field Test Scenarios of Automated Vehicles,” in *SAE Technical Papers*, Aug. 2018, vol. 2018-Augus, no. August, doi: 10.4271/2018-01-1633.
- [74] T. Ponn and C. Ghandt, “AN OPTIMIZATION-BASED METHOD TO IDENTIFY RELEVANT SCENARIOS FOR TYPE APPROVAL OF AUTOMATED VEHICLES.”
- [75] J. J. So, I. Park, J. Wee, S. Park, and I. Yun, “Generating Traffic Safety Test Scenarios for Automated Vehicles using a Big Data Technique,” *KSCE J. Civ. Eng.*, vol. 23, no. 6, pp. 2702–2712, Jun. 2019, doi: 10.1007/s12205-019-1287-4.
- [76] A. Gambi, T. Huynh, and G. Fraser, “Generating effective test cases for self-driving cars from police reports,” in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering - ESEC/FSE 2019*, 2019, pp. 257–267, doi: 10.1145/3338906.3338942.
- [77] T. Huynh, A. Gambi, and G. Fraser, “AC3R: Automatically Reconstructing Car Crashes from Police Reports,” in *2019 IEEE/ACM 41st International Conference on Software Engineering:*

*Companion Proceedings (ICSE-Companion)*, May 2019, pp. 31–34, doi: 10.1109/ICSE-Companion.2019.00031.

- [78] A. Gambi, T. Huynh, and G. Fraser, “Automatically Reconstructing Car Crashes from Police Reports for Testing Self-Driving Cars,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, May 2019, pp. 290–291, doi: 10.1109/ICSE-Companion.2019.00119.
- [79] Y. Qi, K. Li, W. Kong, Y. Wang, and Y. Luo, “A trajectory-based method for scenario analysis and test effort reduction for highly automated vehicle,” *SAE Tech. Pap.*, vol. 2019-April, no. April, pp. 1–8, 2019, doi: 10.4271/2019-01-0139.
- [80] C. Amersbach and H. Winner, “Functional decomposition—A contribution to overcome the parameter space explosion during validation of highly automated driving,” *Traffic Inj. Prev.*, vol. 20, no. sup1, pp. S52–S57, Jun. 2019, doi: 10.1080/15389588.2019.1624732.
- [81] W. Hu *et al.*, “Mining and comparative analysis of typical pre-crash scenarios from IGLAD,” *Accid. Anal. Prev.*, vol. 145, no. March, p. 105699, Sep. 2020, doi: 10.1016/j.aap.2020.105699.
- [82] Y. Kim, S. Tak, J. Kim, and H. Yeo, “Identifying major accident scenarios in intersection and evaluation of collision warning system,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2017, vol. 2018-March, pp. 1–6, doi: 10.1109/ITSC.2017.8317660.
- [83] B. Sui, N. Lubbe, and J. Bärghman, “A clustering approach to developing car-to-two-wheeler test scenarios for the assessment of Automated Emergency Braking in China using in-depth Chinese crash data,” *Accid. Anal. Prev.*, vol. 132, p. 105242, Nov. 2019, doi: 10.1016/j.aap.2019.07.018.
- [84] L. Stark, M. Düring, S. Schoenawa, J. E. Maschke, and C. M. Do, “Quantifying Vision Zero: Crash avoidance in rural and motorway accident scenarios by combination of ACC, AEB, and LKS projected to German accident occurrence,” *Traffic Inj. Prev.*, vol. 20, no. sup1, pp. S126–S132, 2019, doi: 10.1080/15389588.2019.1605167.
- [85] B. Yue, S. Shi, S. Wang, and N. Lin, “Low-Cost Urban Test Scenario Generation Using Microscopic Traffic Simulation,” *IEEE Access*, vol. 8, pp. 123398–123407, 2020, doi: 10.1109/ACCESS.2020.3006073.
- [86] P. Junietz, W. Wachenfeld, V. Schönemann, K. Domhardt, W. Tribelhorn, and H. Winner, “Gaining Knowledge on Automated Driving’s Safety—The Risk-Free VAAFO Tool,” in *Lecture Notes in Control and Information Sciences*, vol. 476, Springer International Publishing, 2019, pp. 47–65.
- [87] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder, “Accident Scenario Generation with Recurrent Neural Networks,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018, vol. 2018-Novem, pp. 3340–3345, doi: 10.1109/ITSC.2018.8569661.
- [88] W. Damm, E. Möhlmann, T. Peikenkamp, and A. Rakow, “A Formal Semantics for Traffic Sequence Charts,” in *Simulation and Modeling of Systems of Systems*, vol. 2, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2018, pp. 182–205.
- [89] M. O’Kelly, H. Abbas, and R. Mangharam, “Computer-aided design for safe autonomous vehicles,” *Proc. - 2017 Resil. Week, RWS 2017*, pp. 90–96, 2017, doi: 10.1109/RWEEK.2017.8088654.

- [90] S. Jha *et al.*, “ML-Based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection,” in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jun. 2019, pp. 112–124, doi: 10.1109/DSN.2019.00025.
- [91] P. Koopman and F. Fratrick, “How Many Operational Design Domains, Objects, and Events?,” in *Proceedings of AAAI Workshop on Artificial Intelligence Safety*, Jan. 2019.
- [92] C. Roesener, F. Fahrenkrog, A. Uhlig, and L. Eckstein, “A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. November, pp. 1360–1365, 2016, doi: 10.1109/ITSC.2016.7795734.