A Simulation-Aided Approach to Safety Analysis of Learning-Enabled Components in Automated Driving Systems

Peng Su, Fredrik Warg, DeJiu Chen

Published/presented in:

26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023) – Workshop on Beyond Traditional Sensing for Intelligent Transportation.

DOI: To be added when available

Postprint – To appear in 26^{th} IEEE International Conference on Intelligent Transportation Systems (ITSC 2023)
– Workshop on Beyond Traditional Sensing for Intelligent Transportation –

A Simulation-Aided Approach to Safety Analysis of Learning-Enabled Components in Automated Driving Systems

Peng Su
Mechatronics, Dept. of Engineering Design
KTH Royal Institute of Technology
Stockholm, Sweden
pensu@kth.se

Fredrik Warg
Dependable Transport Systems
RISE Research Institutes of Sweden
Borås, Sweden
fredrik.warg@ri.se

DeJiu Chen*
Mechatronics, Dept. of Engineering Design
KTH Royal Institute of Technology
Stockholm, Sweden
chendj@kth.se

***Abstract*—Artificial Intelligence (AI) techniques through Learning-Enabled Components (LEC) are widely employed in Automated Driving Systems (ADS) to support operation perception and other driving tasks relating to planning and control. Therefore, the risk management plays a critical role in assuring the operational safety of ADS. However, the probabilistic and nondeterministic nature of LEC challenges the safety analysis. Especially, the impacts of their functional faults and incompatible external conditions are often difficult to identify. To address this issue, this article presents a simulation-aided approach as follows: 1) A simulation-aided operational data generation service with the operational parameters extracted from the corresponding system models and specifications; 2) A Fault Injection (FI) service aimed at high-dimensional sensor data to evaluate the robustness and residual risks of LEC. 3) A Variational Bayesian (VB) method for encoding the collected operational data and supporting an effective estimation of the likelihood of operational conditions. As a case study, the paper presents the results of one experiment, where the behaviour of an Autonomous Emergency Braking (AEB) system is simulated under various weather conditions based on the CARLA driving simulator. A set of fault types of cameras, including solid occlusion, water drop, salt and pepper, are modelled and injected into the perception module of the AEB system in different weather conditions. The results indicate that our framework enables to identify the critical faults under various operational conditions. To approximate the critical faults in undefined weather, we also propose Variational Autoencoder (VAE) to encode the pixel-level data and estimate the likelihood.**

***Index Terms*—Automated Driving System, Learning-Enabled Components, Safety Engineering, Data Analysis, Fault Injection**

I. Introduction

With the advance of Artificial Intelligence (AI) technologies, Learning-Enabled Components (LEC) [1] are widely used in Automated Driving Systems (ADSs) and other intelligent systems for operation perception, task planning and control. The behaviours of these LEC rely heavily not only on the training data but also on the actual operational data and technical conditions. In general, there are two key issues of concern [2]: 1) *aleatory uncertainties*, which are due to the existence of *a priori* unknown environmental conditions, making it impossible to collect the data covering all driving scenarios; 2) *epistemological uncertainties*, which are due to the inherently probabilistic nature of LEC, making it impossible to identify all possible system fault behaviours and analyze their impacts preconceived during system development. In particular, the epistemological uncertainties are related to the specific design or model underlying LEC-based perception and control decision-making. Current industrial standards (e.g., ISO 26262 [3] and ISO 21448 [4]) demand systematic hazard identification/analysis and risk assessment [5] to mitigate the residual risks. However, for LEC, the effectiveness is restricted due to the insufficiency of conventional expert knowledge as well as the complexity of faults and operation assumptions [5], [6]. To cope with such challenges, in this work, we propose a simulation-aided approach to support these frameworks by exploring and evaluating the system's behaviours with simulation-aided data generation and fault injection. The contribution of this work is summarized as follows:

- A simulation-aided operational data generation service with the operational parameters extracted from the corresponding system models and specifications;
- A Fault Injection (FI) service aimed at high-dimensional sensor data to evaluate the robustness and residual risks of underlying LEC;
- A Variational Bayesian (VB) method for encoding the collected operational data and supporting an effective estimation of the likelihood of operational conditions.

The rest of this paper is organized as follows: Section II presents the related work on simulation-aided fault injection. Section III describes the design of the framework to support hazard identification and risk assessment. Section IV introduces a case study by injecting faults in Autonomous Emergency Braking (AEB) system with various operational conditions in terms of weather parameters. We also present the hazardous faults of the AEB system. A discussion of future work is described in Section V.

[*] Corresponding author.

II. Related Work

Safety engineering for ADS involves a wide range of concerns across ADS life-cycle, including system use cases, requirements, estimated faults and failures and their impacts on the intended system functionalities under different operational conditions [6]. The safety analysis of a proposed design [3] is focused on the identification of potential hazards and the assessment of corresponding risks. To justify the sufficiency of a system design or a safety measure, counterexamples in terms of specific operational scenarios and conditions where some intended system requirements are violated with potential hazardous situations, become useful. To support this, FI is an experiment-based technique widely adopted. The purpose is to reveal the actual system behaviours in the presence of faults and thereby support the safety analysis with counterexamples. The overall process is centred on the definition or selection of faults that likely occur in the system, the injection of such faults into the target system and observing how the system responds under different system work conditions. FI approaches can differ according to the availability of targets. A simulation-based fault injection is useful for the safety analysis and the design of safety mechanisms due to its low cost and high efficiency by employing virtual operation centring on system models (e.g., the configuration of the simulated scenarios and vehicle functionalities). The risks in regard to intended system requirements could be revealed directly according to the simulated behaviours [6]. Nevertheless, in engineering practices, the configuration and execution of FI are still challenging tasks, relating to the configurations of system faults and workloads, as well as the optimization of test execution, data collection and result analysis. Many approaches [7]–[9] consider multiple fault types (e.g. bit flip, stuck-at, and Gaussian noise) for FI, but fail to consider the likelihood of occurrence of these faults in different operational situations. For example, specific bit-flips could be very likely to occur in a specific system [8]. For the design of LEC, it is also common to inject pixel-level faults in the sensory data to evaluate the algorithmic performance [10]–[13]. On the other hand, the support for the integration of such analysis with fault modeling and hazard analysis is often not provided. For the optimization of FI test cases, optimization-based methods have been proposed to guide the configurations [14], [15] but experience scalability challenges due to the existence of a wide range of operational scenarios. In this work, we present an approach to safety analysis of LEC using simulated operational data by FI cases that are configured systematically according to the corresponding system models. It facilitates the work of revealing the safety critical operational conditions and faults under various operational conditions.

III. Methodology

The overall structure of our approach to a simulation-aided safety analysis for LEC based on FI is illustrated in Figure 1. It consists of three major services: I. Simulated Data Generation; II. Fault Injection; and III. Hazard Identification and Risk Evaluation. We present the simulated data generation service in Section III-A. The design of service for fault injection is described in Section III-B. In Section III-C, we introduce the support for an automated data-driven safety assessment using an auto-encoder (AE).

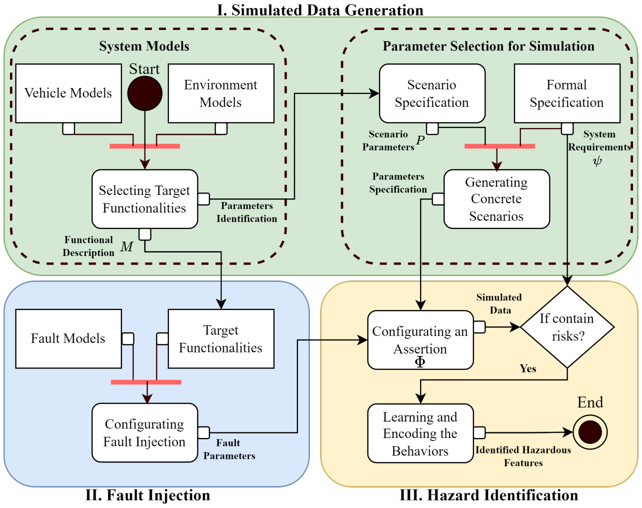
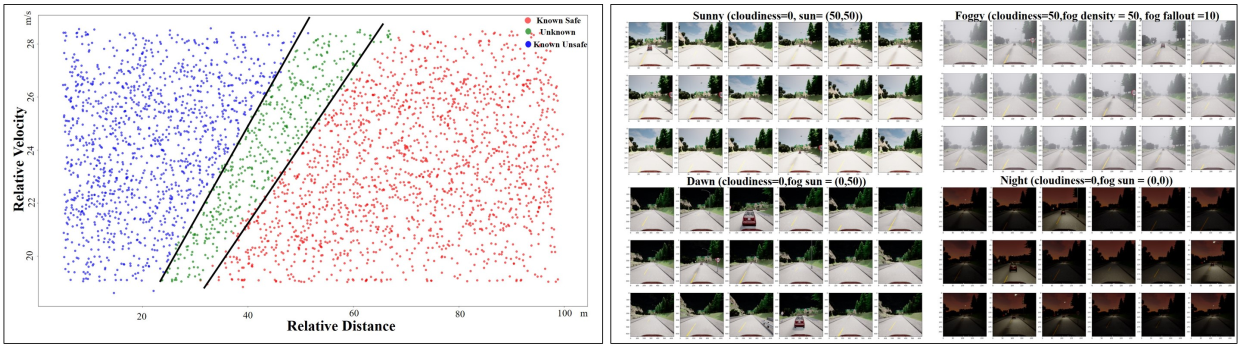


Fig. 1: Overall structure of the proposed approach.

A. Data Generation Service based on System Parameters

The system models specify the target ADS and its operational environments, supported by the domain-specific language EAST-ADL [16]. An additional scenario description method is used to capture the related operational conditions for the configuration of simulations [17]. The configuration of each FI experiment is given by a set of parameters for the internal and external conditions. However, the configuration of such FI parameters typically relies on sampling-based methods that assess the distribution of generated simulation data, challenging the validity and effectiveness of FI experiments. For example, FI experiments could cover specific scenarios where ADS and traffic participants keep large distances, exceeding the operational range of the ADS camera. Under this circumstance, the generated operational data may fail to reveal the actual behaviours of object detection. To cope with this issue, we propose the workflow shown in Block I of Fig. 1 where a more comprehensive set of parameters for stipulating various simulation cases (referred to as P), relating to both ADS functional manoeuvres and ADS external conditions (e.g. traffic participants, weather parameters, and road conditions). These parameters collectively define the driving functionalities according to the system models. Next, we transform the critical parameters which determine such behaviors of the ADS to the parameter sets for the simulations of ADS operation. More precisely, the assessment of ADS behaviors with the scenario of P and ADS system $\mathcal{M}$ is given by the assertion Φ in regard to some system requirements ψ:

$$\Phi(P, \mathcal{M}) \models \psi \tag{1}$$



(a) Characterization of ADS operation by simulation data

(b) Simulated Data with Different Parameters

Fig. 2: Validation regions shown by the collected data and the simulated data according to different parameter specifications.

TABLE I: Configurations (P, M and ψ) for generating concrete scenarios.

Parameter Name		Scenario Parameter
ADS	Interested Functionaility	AEBS
		Relative Distance Estimation
	Initial State	$v_r, a_r = \beta$, location = (x,y)
Leading Vehicle	Initial State	$v_l < v_r, a_l = 0$ location = (x,y+d_r^0), $d_r^0 = [d_{safe}^{max}, d_{safe}^{min}]$
Road Ontology	High way	\
Fault Type	Salt and Pepper	Intensity = 80% Radnom Location
	Solid Occulsion	
	Water Drop	
Weather	Fog	Fog = True or False
		Fog Fallout = (0-100)
		Fog Density = (0-100)
	Sun	Sun Angle = (0-100, 0-100)
		Cloudiness = (0-100)

B. *Fault Injection Service based on Simulated Data*

A *hazard* is a system failure that represents a potential source of harm. Such a failure, in combination with specific environmental conditions, could cause violations of system requirements in terms of unacceptable safety risks [18]. To identify and analyze potential hazards as well as related faulty behaviours of LEC, it is critical to investigate the possible faulty and operational conditions systematically.

A workflow for FI is presented in Block II of Fig. 1. To inject faults into different modules of ADS, we specify the faults f according to the models of the target ADS system $\mathcal{M}$. We parameterize the faults by their location l, type t, and intensity i of occurrence. The FI process is referred to as $f_{\mathcal{M}}(l, t, i)$. The results of $f_{\mathcal{M}}$ are dependent on the operational conditions parameterized by P. For example, $\mathcal{M}$ could be represented by LEC for object detection. Faults are injected into the pixel-level input data $\mathbf{d}$ of this LEC with multiple fault types t, through the l-th pixels with an intensity i. The behaviours of $\mathcal{M}$ given by the corrupted data $\mathbf{d}_c$ are evaluated in regard to the requirements ψ for assessing the risks. A hazardous fault $f_{\mathcal{M}}(l, t, i)$ is revealed when $\Phi(P, \mathcal{M}) \nvDash \psi$.

C. *Encoding Operational Conditions*

The basic FI service mentioned above constitutes the basis for safety analysis. However, for high-dimensional operational data in sensory systems, the interpretation of FI results can be challenging. To this end, we use Variational Auto-Encoder (VAE) [19]–[22], which is a generative model combining auto-encoder (AE) and probabilistic models, to process the FI data, while leveraging the VAE latent space presentation for treating the high-dimensional inputs. This encoding process can be represented as follows:

$$p_{\varphi}(\mathbf{z}^i) = \int_{\mathbf{d}} p_{\varphi}(\mathbf{z}^i|\mathbf{d}_k^i) p_{\varphi}(\mathbf{d}_k^i) d(\mathbf{d}) \tag{2}$$

where $d(\mathbf{d}) = \prod_i d(\mathbf{d}_k^i)$, $\mathbf{d}_k^i$ refers to the simulated data at timestamp k in the operational conditions $i \in \{1, 2, 3, \ldots, n\}$, which is generated by parameters P^i in the FI services. φ is the parameters of the encoder networks, which extract the features of high dimensional $\mathbf{d}^i$ to a low dimensional distribution of $\mathbf{z}^i$.

To optimize the performance of the encoder network under various simulated data, $\hat{\mathbf{d}}_k^i$ is the recovered data from $\mathbf{z}^i$ by using decoder networks with parameters θ as Eq. 2. This network tries to reconstruct the input data by sampling $\mathbf{z}^i$ from latent states, which can regularize to Gaussian Distribution with *reparameterization tricks* as follows [21], [23]:

$$\mathbf{z}^i \sim p_{\varphi}(\mathbf{z}^i|\mathbf{d}_k^i) = \mathcal{N}(\boldsymbol{\mu}^i, \boldsymbol{\sigma}_i^2). \tag{3}$$

Eq. 4 refers to the decoding process of the VAE. In Eq. 4, $d(\mathbf{z}) = \prod_i d(\mathbf{z}^i)$. Such an encoding-decoding structure supports a straightforward way for inferring latent states, avoiding the over-fitting and model collapse of neural networks. We use Kullback–Leibler (KL) divergence to guide the training of the VAE by measuring the loss between $\hat{\mathbf{d}}_k^i$ and $\mathbf{d}_k^i$.

$$q_{\theta}(\hat{\mathbf{d}}_k^i) = \int_{\mathbf{z}} q_{\theta}(\hat{\mathbf{d}}_k^i|\mathbf{z}^i) q_{\theta}(\mathbf{z}^i) d(\mathbf{z}) \tag{4}$$

Since the latent space $\{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n\}$ represents the distribution clusters of n operational conditions, they reveal different operational conditions by specific parameters of

TABLE II: Fault injection results under different scenarios.

	Water Drop		Salt and Pepper		Solid Occlusion	
	False-trigger	Omit-trigger	False-trigger	Omit-trigger	False-trigger	Omit-trigger
Sunny	5.6%	3.2%	16.2%	12.1%	60.1%	**48.4%**
Night	**79.2%**	18.4%	65.5%	23.4%	74.2%	**56.3%**
Foggy	21.4%	14.7%	61.2%	**45.0%**	77.9%	38.1%
Dawn	55.5%	10.3%	35.2%	24.6%	80.5%	**61.4%**

means $\boldsymbol{\mu}$ and standard deviations $\boldsymbol{\sigma}$. Therefore, we evaluate the likelihood of input data $\mathbf{d}_k^u$ with undefined operational conditions by comparing their $\mathbf{z}^u$ with the already collected $\mathbf{z}^{\{1,2,\ldots,n\}}$. Based on the generative results from the VAE, we can further assess the likelihood which indicates the similarity between the undefined condition and the already collected conditions.

IV. Case Study

A. Simulated Data Generation

We use Autonomous Emergency Braking (AEB) System as a case study to verify our framework. The sensory data from the camera are processed by LEC to detect objects and estimate relative distances. Next, the results from LEC support preventing a collision by initializing the brake. The simulation verifies the performance by initializing with a range of relative distance and velocity. To efficiently simulate this scenario, we generate appropriate parameters by modifying Responsibility-Sensitive Safety (RSS) [24]–[26] for the formal specification ψ. In particular, we use the same assumption as in [27], [28] where the AEB system relies on partial brake $\beta_{partial}$ and full brake β_{full} to maintain a safe distance. The minimum safety distances under these brake modes are presented as follows:

$$d_{safe} = v_r\rho + \frac{\beta\rho^2}{2}; \quad \psi ::= \Box\ d_{safe}^{max} \leq d_r \tag{5}$$

where $\beta \in \{\beta_{partial}, \beta_{full}, 0\}$; v_r refers to the relative velocity captured by the radar; d_r refers to the relative distance which a LEC estimates from image; ρ refers to the response time of the AEB system. In Fig. 2 (a), all the nodes refer to the possible relative distance and velocity by using the Uniform Sampling. Such a state space needs to generate a large amount parameter sets, leading to inefficiency of the simulation. Relying on the formal specification defined in Eq. 5, we conclude that a collision is inevitable when d_r is less than the safe distance with full brake d_{safe}^{max}. This unsafe area is represented by the blue nodes in Fig. 2 (a). On the other hand, the red nodes refer to safe states that the relative distance is larger than the distance with minimum brake d_{safe}^{min}. In this case, the minimum brake refers to the partial brake $\beta_{partial}$. However, the region in Fig. 2 (a) between the d_{safe}^{min} and d_{safe}^{max} should be verified in the simulation by evaluating the performance of LEC. Especially, when these critical parameters are in this region, the AEB system is initialized with *false-trigger* if the detected distance is smaller than the safety distance in Eq. 5, leading to the ADS braking in advance. Otherwise, the AEB system is initialized with *omit-trigger*, causing a violation of the specification. Compared with *false-trigger*, it is obvious that the *omit-trigger* is a more hazardous event. In this case, it is critical to assure the safety of the ADS by addressing the faults and operational conditions leading to *omit-trigger*.

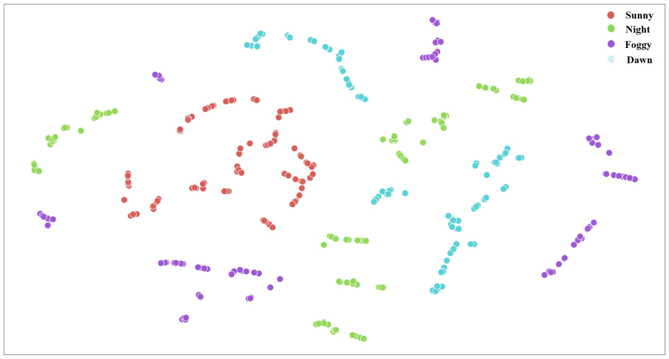


Fig. 3: Characterization of ADS operation by latent space clustering using t-SNE (t-distributed Stochastic Neighbor Embedding)

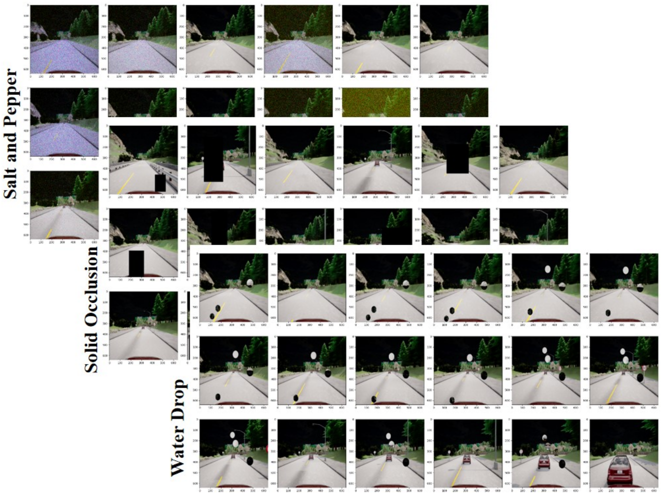
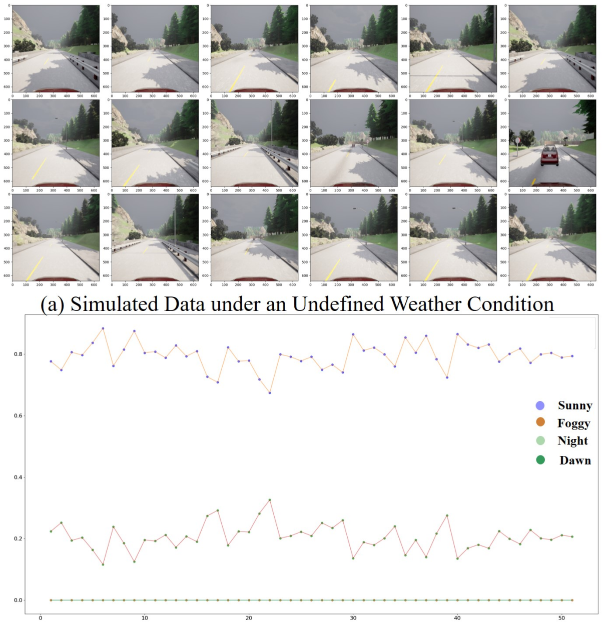


Fig. 4: An example of Corrupted Data after FI services

We use CARLA [29] as a simulator to generate data according to the parameters listed in Table I. We implement the functionality $\mathcal{M}$ as target functionalities for detecting the relative distance d_r. A YOLO network is used in the perception module by analyzing images $\mathbf{d}$ captured by the camera [30], [31]. This neural network shows robustness under different driving section (e.g., urban road and high way) by the KITTI training dataset [32]. However, the weather is a critical element regarding the image resolution. Therefore,



(a) Simulated Data under an Undefined Weather Condition

(b) Likelihood between the Undefined and Predefined Weather

Fig. 5: Likelihood Estimation of the Unknown Weather

we investigate different operational conditions according to a variety of weather parameters (shown in Table I), including: *sunny, foggy, dawn, and night* (Fig. 2 (b)). For the relative distance estimation, the perception module shows acceptable performance in these weather conditions. We select then *water drop, solid occlusion, salt and pepper* (Fig. 4), which also are used in [11], [12], as fault types t to corrupt the images $\mathbf{d}$. These faults are injected into around 500 images under each weather condition with $i = 80\%$ and random pixels. In Table II, we summarize the consequences of such faults with the following observations: 1) *Solid occlusion* is highly hazardous in all weathers, with the *dawn* scenario as the worst case where this fault leads to 61.4% of the *omit-trigger*. 2) *Salt and pepper* is highly hazardous in *foggy*, where 45.0% of this fault could lead to *omit-trigger*. 3) *Water drop* is sensitive during the *night* scenario, where 79.2% of this fault leads to the *false-trigger*. In additional, the faults are most significant to *dawn* and *night*, indicating that the YOLO network is sensitive to the brightness of the environment. This conclusion is also found in [10], [13].

The results indicate that even for the same faults, there are different effects in these weather conditions. Therefore, we use VAE to encode the operational conditions under these weathers and identify the latent states, supporting classifying the hazards and evaluating the risks under undefined weathers. In Fig. 3, the latent states of $\mathbf{d}$ are clustered by the t-SNE algorithm [33]. The latent states of *dawn* and *night* show similar features. Simulated data in *Sunny* cluster to a centring area, indicating unique features compared with other weather conditions. These facts indicates that the VAE could support revealing the likelihoods of different operational conditions. To verify this, we generate the simulated data $\mathbf{d}^u$ by a random weather parameter set shown as Fig. 5(a). We use the same method in [34] to evaluate this likelihood by computing the similarity of $\mathbf{z}^u$ and $\mathbf{z}^{\{1,\dots,4\}}$, referring to the latent states with the predefined weathers in Fig. 2 (b). The results in Fig. 5(b) indicate that the simulated environment is more similar to the *sunny*, implying the *solid occlusion* is the most hazardous faults in this undefined weather. Meanwhile, the undefined conditions still has a certain similarity of *foggy* (around 20%). This means that a prevention of *false-trigger* from *salt and pepper* is of particular concern during the operation.

V. Discussion and Future Work

For safety analysis of ADS, our approach can effectively select the critical parameters relating to the vehicle maneuvers and external traffic conditions, generate the operational data by simulation, and estimate the likelihood of violating system requirements by VB method. In future work, a Scenario Description Language (SDL) (e.g., [35], [36]) can be used to generate the external parameters and describe more complex operational scenarios. We would combine these parameters with formal methods or declarative programming. The logical solvers support generating the related parameters of the functionalities automatically. Furthermore, a systematical architecture could be the future work to specify the definition of the parameters. For real-life applications, the discrepancy caused by the gap between simulated and real-world data needs to be considered.

We inject random pixel-level faults to evaluate the behaviors of LEC. To further investigate the hazards in the ADS, especially the LEC, fault injection based on hardware level (e.g., autonomous embedded system) could be the next topic. The optimization-based method can be used in future work to improve the efficiency of fault injection. Next step, we prepare to combine the fault injection with adversarial learning to find the critical pixels in the input image.

VAE is used to encode the pixel-level data in the predefined weathers, supporting to compute the likelihood of an undefined operational condition. However, this method still shows some limitations: For example, some clusters of *foggy* are outliers in Fig. 3, indicating that the VAE misinterprets the latent states. This phenomenon can be improved by generating more training data or improving the resolution of the related parameters. Future work could investigate other generative models (e.g., Generative Adversarial Networks) to improve the performance for encoding the operational parameters. Moreover, one critical work in the future is the interpretation of latent states by mapping the latent space to the human-understandable parameters. The latent space of the multi-sensor such as sensor-fusion by using deep neural networks could be a future topic for assuring the safety of ADS.

Our framework can also be extend to support Predictive Health Monitors (PHM) in future by including the diagnosis and prognosis services. We need to investigate the correlation and causation between the target and the related functionalities by considering the Fault Tree Analysis (FTA). Combining the PHM with safe and hazardous states in the latent space, a self-evolving monitors could be used for mitigating hazards.

Acknowledgment

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

References

[1] F. Cai and X. Koutsoukos, "Real-time out-of-distribution detection in learning-enabled cyber-physical systems," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2020, pp. 174–183.

[2] D. Chen, K. Östberg, M. Becker, H. Sivencrona, and F. Warg, "Design of a knowledge-base strategy for capability-aware treatment of uncertainties of automated driving systems," in *Computer Safety, Reliability, and Security: SAFECOMP 2018, Sweden, September 18, 2018, Proceedings 37*. Springer, 2018, pp. 446–457.

[3] *ISO 26262:2018 - Road vehicles – Functional safety*. International Organization for Standardization. [Online]. Available: https://www.iso.org/standard/43464.html

[4] ISO, *21448:2022 Road Vehicles – Safety of the Intended Functionality*. International Organization for Standardization, 2022. [Online]. Available: https://www.iso.org/standard/77490.html

[5] K. Madala, C. Avalos-Gonzalez, and G. Krithivasan, "Workflow between iso 26262 and iso 21448 standards for autonomous vehicles," *Journal of System Safety*, vol. 57, no. 1, pp. 34–42, 2021.

[6] P. Koopman and M. Wagner, "Toward a framework for highly automated vehicle safety validation," *SAE Technical Paper, Tech. Rep*, 2018.

[7] P. Su and D. Chen, "Using fault injection for the training of functions to detect soft errors of dnns in automotive vehicles," in *Proceedings of the 17th International Conference on Dependability of Computer Systems, June 27–July 1, 2022, Wrocław, Poland*. Springer, 2022, pp. 308–318.

[8] M. Moradi, B. J. Oakes, M. Saraoglu, A. Morozov, K. Janschek, and J. Denil, "Exploring fault parameter space using reinforcement learning-based fault injection," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2020, pp. 102–109.

[9] S. Jha, S. Banerjee, T. Tsai, S. K. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "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)*. IEEE, 2019, pp. 112–124.

[10] T. Dreossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components," *Journal of Automated Reasoning*, vol. 63, pp. 1031–1053, 2019.

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

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

[13] T. Dreossi, S. Jha, and S. A. Seshia, "Semantic adversarial deep learning," in *Computer Aided Verification: 30th International Conference, CAV 2018*. Springer, 2018, pp. 3–26.

[14] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems," in *Tools and Algorithms for the Construction and Analysis of Systems: 17th International Conference, Saarbrücken, Germany, March 26–April 3, 2011. Proceedings 17*. Springer, 2011, pp. 254–257.

[15] S. Kang, H. Guo, L. Zhang, G. Liu, Y. Xue, and Y. Wu, "Ecsas: Exploring critical scenarios from action sequence in autonomous driving," 2023.

[16] D. Chen, R. Johansson, H. Lönn, H. Blom, M. Walker, Y. Papadopoulos, S. Torchiaro, F. Tagliabo, and A. Sandberg, "Integrated safety and architecture modeling for automotive embedded systems," *Elektrotechnik und Informationstechnik*, vol. 128, no. 6, p. 196, 2011.

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

[18] S. Khastgir, S. Birrell, G. Dhadyalla, H. Sivencrona, and P. Jennings, "Towards increased reliability by objectification of hazard analysis and risk assessment (hara) of automated automotive systems," *Safety Science*, vol. 99, pp. 166–177, 2017.

[19] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[20] J. Wörmann, D. Bogdoll, E. Bührle, H. Chen, E. F. Chuo, K. Cvejoski, L. van Elst, T. Gleißner, P. Gottschall, S. Griesche *et al.*, "Knowledge augmented machine learning with applications in autonomous driving: A survey," *arXiv preprint arXiv:2205.04712*, 2022.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[22] Z. Zhu, P. Su, S. Zhong, J. Huang, S. Ottikkutti, K. N. Tahmasebi, Z. Zou, L. Zheng, and D. Chen, "Using a vae-som architecture for anomaly detection of flexible sensors in limb prosthesis," *Journal of Industrial Information Integration*, p. 100490, 2023.

[23] Y. Liu, Z. Liu, S. Li, Z. Yu, Y. Guo, Q. Liu, and G. Wang, "Cloud-vae: Variational autoencoder with concepts embedded," *Pattern Recognition*, vol. 140, p. 109530, 2023.

[24] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[25] P. Su, T. Fan, and D. Chen, "Scheduling resource to deploy monitors in automated driving systems," in *Proceedings of the 18th International Conference on Dependability of Computer Systems (accepted)*, 2023.

[26] P. Koopman, B. Osyk, and J. Weast, "Autonomous vehicles meet the physical world: Rss, variability, uncertainty, and proving safety," in *Computer Safety, Reliability, and Security: 38th International Conference, SAFECOMP 2019, Turku, Finland, September 11–13, 2019, Proceedings 38*. Springer, 2019, pp. 245–253.

[27] M. Bouton, J. Karlsson, A. Nakhaei, K. Fujimura, M. J. Kochenderfer, and J. Tumova, "Reinforcement learning with probabilistic guarantees for autonomous driving," *arXiv preprint arXiv:1904.07189*, 2019.

[28] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Safe reinforcement learning with scene decomposition for navigating complex urban environments," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1469–1476.

[29] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[30] B. Strbac, M. Gostovic, Z. Lukac, and D. Samardzija, "Yolo multi-camera object detection and distance estimation," in *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE, 2020, pp. 26–30.

[31] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-yolo: Fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, p. 1354, 2022.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[33] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[34] D. Chen, P. Su, S. Ottikkutti, P. Vartholomeos, K. N. Tahmasebi, and M. Karamousadakis, "Analyzing dynamic operational conditions of limb prosthetic sockets with a mechatronics-twin framework," *Applied Sciences*, vol. 12, no. 3, p. 986, 2022.

[35] R. Queiroz, T. Berger, and K. Czarnecki, "Geoscenario: An open dsl for autonomous driving scenario representation," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 287–294.

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