2025 Design, Automation & Test in Europe Conference (DATE 2025)
Special Initiative "Autonomous Systems Design"

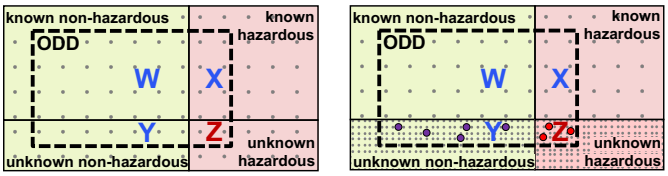
Identification of Hazardous Driving Scenarios using Cross-Channel Safety Performance Indicators

C.A.J. Hanselaar[1] M. M. Selva Kumar[2,3] Y. Fu[2] A. Terechko[2] R.R. Venkatesha Prasad[3] E. Silvas[1,4]

***Abstract*—Automated Driving (AD) vehicles are slowly being deployed on public roads. These AD vehicles will encounter hazardous (dangerous) scenarios due to unforeseen edge cases at design time and changing environments on the road after deployment. To allow developers of AD systems to mitigate such unforeseen risks, the safety of AD vehicles needs to be continuously monitored after deployment. To this end, the UL4600 standard and AVSC guidelines recommend the use of safety performance indicators (SPIs) by AD vehicle developers. Our paper presents a framework that uses SPIs to identify potentially hazardous scenarios specific to the evaluated AD system, covering both AD vehicles and cloud operations. The framework uses the perception systems and motion planners of heterogeneous redundant multi-channel architectures to detect hazards invisible in single-channel-based systems, provided one of the channels observes the environment correctly. We propose three cross-channel SPIs and use them to identify hazardous scenarios in the AD vehicle and validate this approach with a proof-of-concept implementation. In a test of 6 challenging routes in the CARLA simulator, our framework automatically identifies 86% of hazardous situations. Next, it identifies contributing issues in the AD vehicle, such as missed object detections or dangerous planned trajectories. With this proof of concept, we show that this framework provides evidence on the safety of deployed systems, identifies AD vehicle functions in need of improvement and provides lessons for the development of future AD systems.**

I. Introduction

Automated Driving (AD) vehicles are proposed as one of the ways to increase traffic safety. Currently, the AD vehicles are available only with limited Operational Design Domains (ODDs). As per Fig. 1a, restricting the ODD reduces the number of known scenarios where the AD vehicle needs to be safe. This reduces both development and testing complexity, as a restrictive ODD can limit the amount of hazardous scenarios that need to be handled without unreasonable risk, with risk defined as the combination of the probability of occurrence harm and the severity of that harm [1]. Despite limiting the ODD, AD vehicles deployed on public roads will be confronted with a changing environment over their lifetime. Changing environments are caused by, e.g., the emergence of new mobility means such as e-scooters, combinations of rare circumstances, or changes in traffic behavior [2]. As illustrated in Fig. 1b, this will lead to the AD vehicle encountering new and previously unknown scenarios inside their ODD. Some of these scenarios will create *edge cases* – novel or rare circumstances that need specific attention in AD systems for them to be dealt with appropriately [3].



(a) (b)

Fig. 1: At release (a), developers aim to limit risk by controlling exposure to known hazardous scenarios (gray dots in quadrants X), while testing is performed to reduce unknown scenarios (Y&Z). After deployment (b), new scenarios will emerge, increasing the number of unknowns (Y&Z) and exposing new edge cases (larger circled scenarios).

If an AD vehicle responds to an edge case with a dangerous action (red circles in Fig. 1b), the edge case is said to trigger a functional insufficiency (FI) [4], currently the main cause for AD test disengagement [5]. FIs are insufficiencies in (requirement) specification or performance that lead to hazardous behavior, defined in the Safety Of The Intended Functionality (SOTIF) standard [4]. Prior to release AD vehicles are extensively tested to uncover and mitigate as many FIs as possible inside the intended ODD [4]. However, this cannot prevent changing environments from exposing new edge cases after release. That means both online risk mitigation and subsequent monitoring for new edge cases and associated FIs is required to continuously maintain the safety of AD vehicles over their lifetime [4], [6]–[11].

Redundant heterogeneous multi-channel architectures (containing multiple perception systems and one or more motion planning systems) are proposed to mitigate the immediate risk caused by such edge cases, through cross-checking between included AD channels [5], [12]–[19]. These architectures continually evaluate the safety of proposed motion plans and switch to a safer alternative channel (if required). This is only possible if at least one of the channels handles the edge case correctly. Nevertheless, this approach is cited as an essential element in safe AD vehicles [20], [21].

To continuously monitor the safety of AD vehicles and allow for early edge case indication, Safety Performance Indicators (SPIs) are proposed [6], [9]. A SPI is ”a metric supported by evidence that uses a threshold comparison to condition a

[1]Department of Mechanical Engineering, CST Section, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands (c.a.j.hanselaar@tue.nl)
[2]NXP Semiconductors N.V, The Netherlands
[3]Technical University Delft, The Netherlands
[4]TNO, Dutch Organization for Applied Scientific Research, The Netherlands

978-3-9826741-0-0/DATE25/© 2025 EDAA

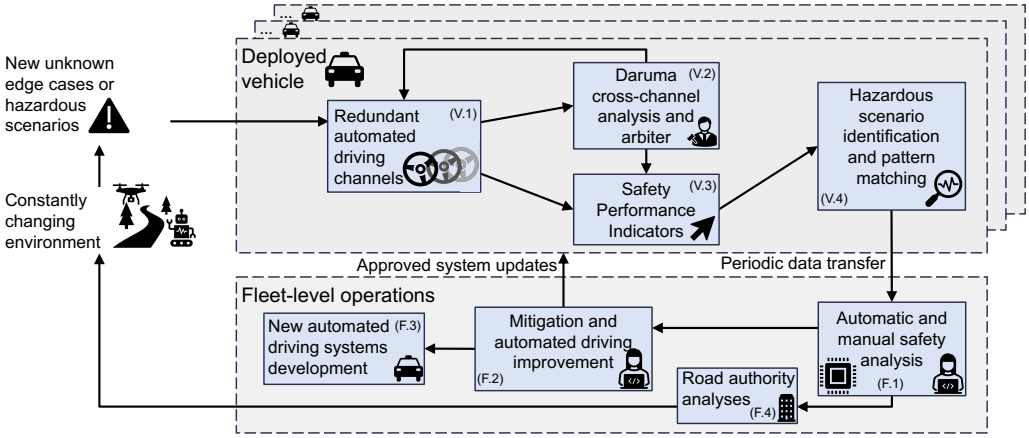


Fig. 2: Fleet-level framework to use vehicle-level SPIs and identified hazardous scenarios to provide continuous monitoring.

[safety] claim" [9]. The number of collisions, loss events, or fatalities can be considered lagging SPIs [9], as they indicate an issue in safety after harm has been done. Monitoring these is mandatory during testing [10], [11]. By contrast, leading SPIs are metrics that correlate to the safety of an AD, but do not cause harm every time they violate their threshold. For instance, detecting pedestrians at a suitable distance (e.g., $\geq 25\,\mathrm{m}$) is a leading metric, as not detecting them until they are closer (e.g., $15\,\mathrm{m}$) indicates a potential FI in the perception system, but does not cause a collision by itself. Monitoring leading SPIs by AD vehicle developers provides much-needed insight into the occurrence of previously unknown FIs and the general safety of their deployed vehicles. Some leading SPIs are challenging to measure, e.g., never detecting objects of a specific rare type, incorrect ego location estimation. Current published safety approaches in industry [22]–[24] do not describe how such SPIs will be detected during fully automated deployment.

To allow the detection of new edge cases and automate the identification and analysis of hazardous scenarios even after deployment, we propose to leverage the information in the aforementioned multi-channel architectures used to mitigate immediate risk, extending the initial work of [25]. As such, we tackle the question – *How can we use the capabilities of multi-channel architectures for automated driving to identify hazardous scenarios for enhanced continuous monitoring?* While answering this question, our contributions are:

(1) The development of SPIs that use cross-channel comparison to identify AD system-specific issues otherwise missed.

(2) Automated hazardous scenario identification using SPIs, providing initial probable cause analysis in case of FIs.

(3) Implementation of cross-channel comparison, arbitration, SPI tracking, and hazardous scenario identification on an automotive-grade embedded device coupled to the CARLA simulator. To the best of our knowledge, we are the first to use the information in these multiple heterogeneous channels to automate both FI detection and the initial post-processing efforts [2].

In Section II we briefly highlight related work, followed by Section III where we introduce the SPIs and Hazardous Scenario Identification algorithms used in our framework, shown in Fig. 2. Next, in Section IV we will show our proof of concept results, discuss them in Section V and finally summarize our findings in Section VI.

II. Related work

Prior to deployment, AD vehicle development heavily relies on Scenario Based Testing (SBT) to find and mitigate edge cases that trigger FIs [4]. Searches for high-risk scenarios are done via available general databases [26], [27] or via analysis of labeled data from vehicles fitted with AD-ready sensors [28], [29]. This allows for edge-case focused SBT [30]–[32].

Post-deployment edge case detection cannot rely on SBT or test-driver feedback to identify perception function FIs. Instead, significant work [2] focuses on online detection of elements in the environment that were not part of the training data of AD functions, so-called Out-Of-Distribution (OOD) detectors [33]–[36]. These OOD detectors are trained functions that necessarily share the training data of the AD functions they evaluate, resulting in specific and computationally heavy algorithms, making it difficult to apply them online.

As a more general and less computationally heavy approach, the quality of object tracking over time can be evaluated. For example, by using Time-Quality-Temporal-Logic feedback is provided on inconsistent object detections [37]. However, these systems cannot identify edge cases where objects are e.g., mislabeled or never identified.

Multi-channel architectures, e.g. monitor-actuator systems [14], [16], [17], [38] or cross-channel comparison systems [5], [12], [13], [15], [18], [19], have so far focused solely on immediate risk mitigation. Consequently, the proposed approaches do not determine contributing causes to detected hazards, limiting their use for continuous monitoring.

In summary, current perception issue detection methods are either computationally heavy and AD version specific or are limited in the scope of their detections, while online hazard-mitigation methods using cross-channel comparisons do not identify contributing issues to the emergent hazard. Therefore, a framework to combine both strengths is desired.

III. Methodology

In this section, we will first introduce an overview of our framework. Next, we will introduce a segment of a safety case [9], define SPIs that condition some claims in that safety case, and introduce the way we identify hazardous scenarios with them. Finally, we will introduce our simulation environment to test our framework.

A. Framework

AD vehicles are at the core of our framework, each equipped with sensors, safety monitors, onboard computing systems, and, optionally, multiple heterogeneous AD channels, as shown in block (V.1) in Fig. 2. In our study, the redundant AD channels are employed via the Daruma architecture [5], [18] – selected for its modularity, real-time cross-channel analysis and information handling capabilities for SPI calculations, while remaining efficient enough to run on automotive-grade embedded processors. In Daruma, each AD channel contains a perception system and motion planner and sends information to the Daruma module (V.2) on, e.g., the world model (WM), ego location and motion plan (MP). We will use indices i and j to indicate these included enumerated AD channels. The Daruma block arbitrates between the AD channels (implementing immediate risk mitigation) and provides information to the SPI calculation (V.3). The framework detects hazardous scenarios by tracking the values of SPIs and assessing when one or multiple of them simultaneously violate their thresholds (V.4). Known combinations of SPIs are used to identify specific scenarios and contributing issues, while other SPI combinations indicate uncatalogued scenarios.

Periodic cloud updates provide the fleet section of the framework (bottom of Fig. 2) with information on the encountered hazardous scenarios. The analysis of fleet-level SPIs and identified hazardous scenarios (F.1) is used both by system developers to mitigate changed conditions (F.2) and enable future developments (F.3), as well as to satisfy regulation requirements (F.4). Approved system updates complete the feedback loop from fleet level to vehicle level to mitigate edge cases discovered with this framework.

B. Example safety case and SPIs

As suggested by UL4600 standard [9] and AVSC best practices [7], SPIs should be derived from the safety cases. A safety case is a structured argument that holds safety goals and assumptions and solutions as needed to support those goals. The authors of both AD channels included in our setup have not published a safety case [39], [40]. To allow us to provide a proof of concept of safety monitoring with SPIs derived from a safety case, we have created a segment of a plausible safety case using goal-structured notation, shown in Fig. 3. The structured development of effective SPIs is shown in [41]. For this proof of concept, we use SPIs easily derived from this safety case to cover hazards from perception to motion planning. As identified in Section II, perception FIs are difficult to track. Hence, we derive 2 leading SPIs on safety-critical perception system goals from goals G7 and G5 – Object Count Similarity (Ω_i) and Ego

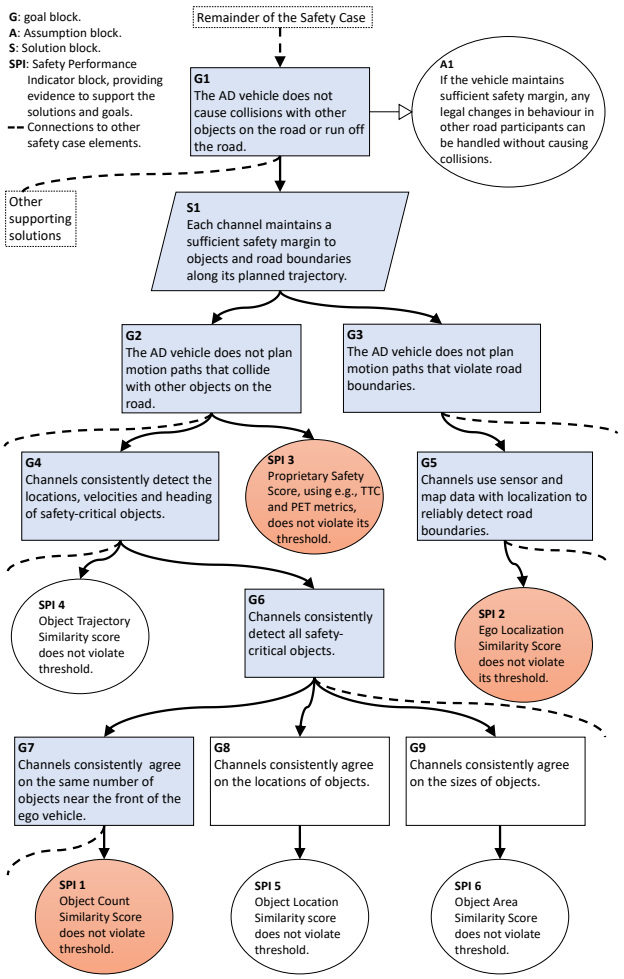


Fig. 3: Ensuring the AD vehicle does not cause collisions or run off the road. Goal and strategy blocks highlighted in blue and SPI blocks highlighted in orange are the focus of this paper. Dotted lines indicate connections to other necessary safety case elements not included in this example.

Location Similarity (Λ_i) (see Fig. 3). Next, we create a more general risk-indicating SPI connected to goal G2 – Safety Score (ζ_i).

SPI 1 – Object Count Similarity (Ω_i). This is a measure of the number of objects in a safety-critical region in front of the ego vehicle that have not been detected by the evaluated channel (i) but have been detected by other channels. To compute this at time t, objects seen by channel i are paired to objects seen by channel j, if possible. We start this by placing all objects detected by a channel into a set of unpaired objects of that channel. So for channels i and j all objects k and l are defined to be part of sets $k \in O_i^{\mathrm{u}}$ and $l \in O_j^{\mathrm{u}}$, respectively. Next, we loop over each object $k \in O_i^{\mathrm{u}}$ and assess if a suitable pairing to $l \in O_j^{\mathrm{u}}$ exists via

$$\arg\min_{l}\{\Delta d^{\mathrm{p}}_{k,l}(t) \text{ s.t. } \Delta d^{\mathrm{p}}_{k,l}(t) < \Delta d^{\mathrm{p\ max}}\}, \tag{1}$$

where $\Delta d^{\mathrm{p}}_{k,l}(t)$ is the euclidean distance between objects k, l and $\Delta d^{\mathrm{p\ max}}$ the maximum permissible distance to pair objects (determined through evaluation of reasonable position variation). If there exists an l that satisfies (1), both k and l are

removed from the sets O_i^{u} and O_j^{u}, respectively. The number of unpaired objects in O_j^{u} that remain after looping over all $k \in O_i^{\mathrm{u}}$ is denoted (1) $\Delta N_{i,j}$. The Object Count Score for channel i (Ω_i) is then computed as

$$\Omega_i = \frac{1}{n_{\mathrm{c}} - 1} \sum_{j=1}^{n_{\mathrm{c}}} \frac{\mathbf{1}\,(i \neq j)}{1.0 + \Delta N_{i,j}}, \tag{2}$$

with $\mathbf{1}\,(i \neq j)$ the indicator function that is 1 when $i \neq j$ and 0 otherwise.

SPI 2 – Ego Location Similarity (Λ_i). This is a measure of the difference in estimated ego positions as reported by the included AD channels, calculated via

$$\Lambda_i = 1 - \frac{1}{1 + e^{-\rho\left(\sum_{j=1}^{n_{\mathrm{c}}} \left(\Delta d_{i,j}^{\mathrm{e}}\right) - d_0^{\mathrm{e}}\right)}}, \tag{3}$$

with $\Delta d_{i,j}^{\mathrm{e}}$ the Euclidean distance between ego vehicle locations in AD channels i and j, $\rho = 10$ the gradient parameter and $d_0^{\mathrm{e}} = 0.6$ the inflection point of the function. These parameters have been set to ensure that deviations over 0.6 m, approximately the distance from the center of a lane to the lane boundary, will result in $\Lambda_i < 0.5$.

SPI 3 – Safety Score (ζ_i). This is a proprietary combination of, among others, conventional safety metrics. The safety score for channel i (with $i \in \{1, 2, ..., n_{\mathrm{c}}\}$ and n_{c} channels) is calculated at each time t by comparing the trajectory $\mathrm{T}_i(t)$ to every channel's World Model $\mathrm{WM}_j(t)$, which contains all relevant perception information gathered by channel $j \in \{1, 2, ..., n_{\mathrm{c}}\}$. The safety score is calculated in a two-step process via,

$$\zeta_{i,j}^*(t) \leftarrow (\mathrm{T}_i(t)\ \&\ \mathrm{WM}_j(t))\ \ \forall j = (1, 2, ..., n_{\mathrm{c}}) \tag{4}$$

and

$$\zeta_i(t) = \sum_{j=1}^{n_{\mathrm{c}}} w_j(t) \zeta_{i,j}(t), \tag{5}$$

with $\zeta_{i,j}^*(t) \in (0, 1)$ the pair-wise-safety-score, the result of the mapping ($\leftarrow$) of the comparison of Trajectory ($\mathrm{T}_i(t)$) and World Model j ($\mathrm{WM}_j(t)$). If $i = j$, then $\zeta_{i,j}^*(t)$ in (4) represents the *self-check* of channel i, and if $i \neq j$, then (4) represents a *cross-check* of channel i vs the observations and predictions of channel j. The final Safety Score in (5) is the weighted sum of individual safety scores, with the time-varying weights $w_j(t)$ such that $\sum_j^{n_{\mathrm{c}}} w_j(t) = 1$, a reflection of the trust in the different channels.

Benefits of redundant channels. Note that the Ω_i and Λ_i SPIs require comparing at least two AD channels i and j, while the ζ_i is augmented by the multi-channel comparison. By having access to heterogeneous redundant AD channels with sufficient capabilities, these SPIs are available to give meaningful insights into leading SPIs of individual AD channels.

C. Hazardous Scenario Identification

To identify edge cases, we implement our Hazardous Scenario Identification (HSI) function. The primary requirement to detect a hazardous scenario is the presence of a Safety Score threshold violation ($\zeta_i < \zeta^T = 0.8$ for any channel i). Following the structure of the segment of the safety case (Fig. 3), we then assess the SPIs as in hierarchical order in Algorithm 1.

Algorithm 1 The hierarchical order of contributing issues

```
if ζ_i ≥ ζ^T, ∀i ∈ [1, 2, ..., n_c] then
    No hazard
else
    if ∃i s.t. Ω_i < Ω^T & ζ_i < ζ^T, then
        Object detection issue
    else if ∃i s.t. ζ_{i,i} < ζ^T, then
        Trajectory planning issue
    else if ∃i s.t. Λ_i < Λ^T & ζ_i < ζ^T, then
        Ego localization issue
    else
        Unknown hazardous issue
    end if
end if
```

The pattern filter shown in Fig. 4 reports the start, initial identified issue, changes of contributing issues, and duration of an identified hazardous scenario. We group iteratively occurring contributing issues into a single hazardous scenario, by enforcing a minimum duration of $n_{\mathrm{ts}} = 20$ timesteps or 2 s.

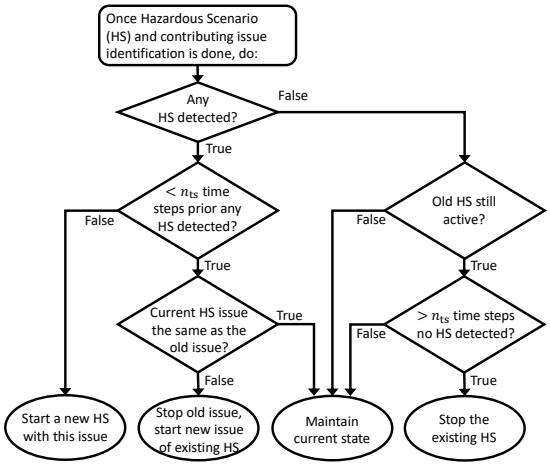


Fig. 4: Pattern filter for identified Hazardous Scenarios (HS), to report onset, change of contributing issue and end of HS.

D. Test environment

To evaluate the proposed framework, we employ it in the simulation environment developed in [19], summarized in Fig. 5. We use Learn from All Vehicles [39] (LAV, shown as red in figures) and Transfuser [40] (shown as green) as heterogeneous AD channels. Routes 1 through 6 from the longest six benchmark [40] are used to evaluate the performance of the framework. In contrast to [19], we substitute the MATLAB-based Safety Shell arbitration software with an embedded implementation of the Daruma framework using C++ running on an S32G274A automotive-grade processor [42], to evaluate the behavior in an embedded automotive system.

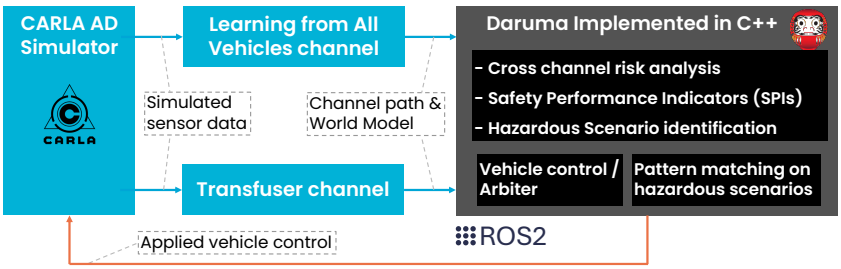


Fig. 5: The testbed used to evaluate the SPI and hazardous scenario identification framework, adapted from [19].

IV. Results

The framework identified 63 separate hazardous scenarios in the 6 completed routes (Table I). Of those, 33 are labeled *true positive* hazardous scenarios by an operator, as a collision seemed likely or imminent (e.g., Fig. 6). In addition to the automatically identified scenarios, five other scenarios were subjectively assessed as dangerous by the operator, where cyclists or pedestrians crossed in front of the ego vehicle. In these five cases, one of the AD channels mitigated the hazard by slowing down sufficiently to avoid violating the safety threshold ζ^T for either channel's trajectory plan. This means that of 38 true hazardous scenarios, the framework identified 33 or a recall rate of $33/38 = 86.8\%$.

Logically, the other 30 of 63 identified scenarios were *false positive* hazards, as, according to the evaluating operator, no conflicting object paths or dangerous ego vehicle behavior was planned or displayed. Fig. 7 shows an example of a false positive hazardous scenario, where the vehicle following the ego vehicle is predicted to drive into the rear of the ego vehicle for a single timestep, as per the wireframe shown in Fig. 7b. False-positive causes were varied, e.g., an incorrect object motion prediction ($15\times$, e.g., Fig. 7), an incorrect object state (size or position, $5\times$), an overly conservative stopping distance near red-lights ($5\times$) or ghost-object detection issues ($3\times$). Consequently, the precision of the framework is $33/63 = 52.4\%$, using the combination of Transfuser and LAV as the two AD channels.

The system identified 125 contributing issues to the detected hazardous scenarios, shown in Table II. Most scenarios have up to 3 (possibly repeating) issues, while $< 15\%$ of scenarios are registered with 4 or more issues. The occurrence of multiple issues in a single hazardous scenario is a consequence of them resolving and recurring, such as iteratively occurring and resolving duplicate object detection and dangerous trajectory generation[1]. Of all identified issues, most are attributed to MP insufficiencies, instances where a trajectory created by one of the channels is deemed dangerous in the safety self-check of (4). This occurs when no trajectory is safe (e.g., a collocated object with ego) or when the trajectory planner accepts more risk than the safety score computation allows (e.g., a dangerous trajectory is planned despite correctly observing surrounding objects). WM issues occur when one channel observes duplicate detections (e.g., Fig. 6) or when one channel fails to detect an object. A case in point is the lack of pedestrian detection by Transfuser, a large portion of the number of WM issues.

[1]See the short video of this occurring at https://youtu.be/kl0AehmLMME

TABLE I: Overview of the scenarios identified in the first 6 routes, with overall precision and recall scores.

Route	1	2	3	4	5	6	Total
Detected scenarios	14	10	15	8	7	9	63
True positive scenarios	9	6	5	5	5	3	33
False positive scenarios	5	4	10	3	2	6	30
Missed scenarios	1	1	1	2	0	0	5
Precision							52.4%
Recall							86.8%

TABLE II: Overview of the identified contributing issues detected via the hazardous scenario identification logic.

Route	1	2	3	4	5	6	Total
Identified issues	32	26	28	15	10	14	125
WM issues	13	12	10	5	5	4	49
MP issues	15	12	17	9	6	9	67
Location issues	2	1	1	1	0	1	6
Unknown issues	2	1	0	0	0	0	3

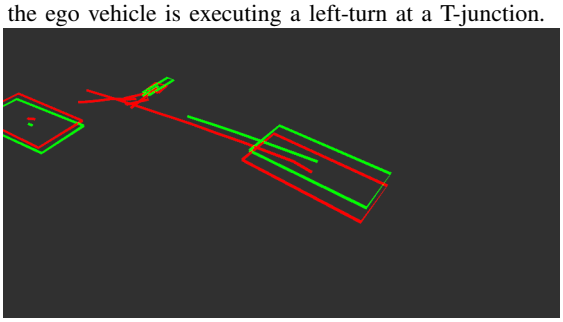
Running the SPI calculations on the NXP S32G274 embedded processor [42] showed that the majority of SPIs were run well within time budgets of 25 ms, indicating that these and similar SPIs are suitable for online application.

V. Discussion

A review of the identified hazardous scenarios and contributing issues of Section IV shows that the LAV channel causes few hazardous scenarios but triggers most false positive hazard identifications. These involve, for example, predicting that the following cars will collide with the ego vehicle and false positive object detections that coincide with the ego vehicle. By contrast, the Transfuser channel causes the majority of hazards, either by not slowing down when observing crossing cyclists or completely failing to either detect or respond to crossing pedestrians. The cross-channel SPIs are able to detect this glaring issue in the Transfuser channel. At the same time, the online risk mitigation arbiter switches to LAV to prevent simulated collisions. Without our framework, the Transfuser channel would not be aware of this issue until it hit its first pedestrian.

The heterogeneity of the redundant channels in our framework is used both for immediate risk mitigation and for con-

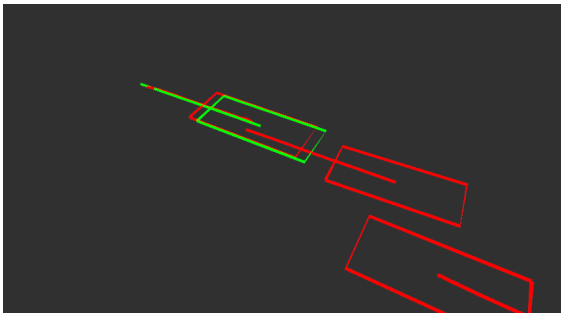
(a) 3D rendering of the identified hazardous scenario of a cyclist crossing in front of the moving ego vehicle while the ego vehicle is executing a left-turn at a T-junction.



(b) Relevant perception shown via wireframe, showing the red LAV channel, which sees the cyclist as 2 separate objects and plans a collision trajectory.

Fig. 6: An example of an identified hazardous situation.

(a) The traffic light turns red, resulting in a sudden deceleration of the red LAV channel.



(b) The motion of the rear vehicle is predicted to continue forward and to collide with the ego vehicle.

Fig. 7: An example of a false-positive hazardous situation.

tinuous monitoring through various cross-channel comparisons. This approach can only detect issues or unmitigated hazards if the included channels do not suffer the same FIs at the same time. Therefore, adopting multi-channel architectures should be seen as an additional safety layer, and not a guarantee of complete safety. To avoid eroding the required heterogeneity through shared interface requirements (as seen in aviation [43]), we have adopted relatively simple cross-comparing SPIs. Intrinsic differences in channels (e.g., deviating type definitions) are subsequently handled in this simplified interface. Even when passed through this simplified interface, the cross-comparison still requires some form of interoperability of the MP of one channel with the WM of another. When channels are too different in concept or implementation, they may not work in the Daruma or Safety Shell architectures. More research into optimal levels of heterogeneity is needed.

The framework identifies a significant number of false-positive hazardous scenarios (see Table I). This limits the precision of this implementation compared to other custom-made OOD detection algorithms. However, because the identified false positives stem from cross-comparisons of observations of the used AD channels, they expose limitations in the AD channel's implementation. This is by design, following the view of UL4600 regarding the tracking of unnecessary risk mitigation activation occurrences [9]. As such, our framework is able to identify issues that limit the permissiveness of journey continuation of the included AD channels.

VI. Conclusions

We have presented and implemented[2] a framework to continuously monitor leading safety performance indicators and identify hazardous scenarios by using high-level information from redundant heterogeneous AD channels. With our proof-of-concept, we have shown that this framework is able to identify 86.8% of hazardous scenarios in complex situations simulated in the CARLA simulator, thereby exposing limitations of the tested AD channels with respect to object detection, safe motion planning, and localization. Furthermore, each identified hazardous scenario is associated with contributing issues, facilitating partially automated analysis of identified scenarios. In contrast to the previous works based on computationally intensive neural networks, our framework easily runs on embedded automotive systems. Overall, the presented method identifies hazardous scenarios, enabling continuous monitoring and improvement of automated driving systems after deployment.

Acknowledgments

This work was funded by the project NEON, through the Dutch Research Council (NWO) Crossover Programme (project number 17628).

References

[1] "Road vehicles – Functional safety (ISO Standard No. 26262:2018)."

[2] S. Rahmani, S. Rieder, E. de Gelder, M. Sonntag, J. L. Mallada, S. Kalisvaart, V. Hashemi, and S. C. Calvert, "A systematic review of edge case detection in automated driving: Methods, challenges and future directions," *arXiv preprint arXiv:2410.08491*, 2024.

[2] See a short demonstration video at https://youtu.be/kl0AehmLMME

[3] L. Vater, M. Sonntag, J. Hiller, P. Schaudt, and L. Eckstein, "A systematic approach towards the definition of the terms edge case and corner case for automated driving," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 2023, pp. 1–6.

[4] "Road vehicles – Safety of the intended functionality (ISO Standard No. 21448:2022)."

[5] Y. Fu, J. Seemann, C. Hanselaar, T. Beurskens, A. Terechko, E. Silvas, and W. P. M. H. Heemels, "Characterization and mitigation of insufficiencies in automated driving systems," in *The 27th International Technical Conference on the Enhanced Safety of Vehicles*, 2023.

[6] "AVSC best practice for metrics and methods for assessing safetyperformance of automated driving systems," Automated Vehicle Safety Consortium, Tech. Rep., 2021.

[7] "AVSC best practice for continuous monitoring and improvement after deployment," Automated Vehicle Safety Consortium, Tech. Rep., 2023.

[8] "AVSC best practice for developing ADS safety performance thresholds based on human driving behavior," Automated Vehicle Safety Consortium, Tech. Rep., 2023.

[9] "UL4600, Standard for Evaluation of Autonomous Products," 2020. [Online]. Available: https://www.shopulstandards.com/ProductDetail.aspx?productid=UL4600

[10] European Commision, "Regulation (EU) 2022/1426," online, 2022. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022R1426

[11] State of California, "Article 3.7 - Testing of Atonomous Vehicles," online, 2022. [Online]. Available: https://www.dmv.ca.gov/portal/file/adopted-regulatory-text-pdf/

[12] S. Fürst, "Scalable architecture for autonomous driving," *9th Vector Congress*, 2018.

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

[14] A. Mehmed, W. Steiner, M. Antlanger, and S. Punnekkat, "System architecture and application-specific verification method for fault-tolerant automated driving systems," in *Intelligent Vehicles Symposium*. IEEE, 2019, pp. 39–44.

[15] T. Fruehling, A. Hailemichael, C. Graves, J. Riehl, E. Nutt, R. Fischer, and A. K. Saberi, "Architectural safety perspectives & considerations regarding the ai-based av domain controller," in *International Conference on Connected Vehicles and Expo*. IEEE, 2019, pp. 1–10.

[16] A. Mehmed, W. Steiner, and A. Čaušević, "Systematic false positive mitigation in safe automated driving systems," in *International Symposium on Industrial Electronics and Applications*. IEEE, 2020, pp. 1–8.

[17] J. Weast, "Sensors, safety models and a system-level approach to safe and scalable automated vehicles," *arXiv preprint arXiv:2009.03301*, 2020.

[18] C. Hanselaar, E. Silvas, A. Terechko, and W. P. M. H. Heemels, "The safety shell: An architecture to handle functional insufficiencies in automated driving," *IEEE Transactions on Intelligent Transportation Systems*, 2024. [Online]. Available: https://arxiv.org/abs/2311.08413

[19] C. Hanselaar, Y. Fu, A. Terechko, J. Seemann, T. Beurskens, E. Silvas, and W. P. M. H. Heemels, "Evaluation of the safety shell architecture for automated driving in a realistic simulator," 2024.

[20] A. A. A. B. C. D. F. H. I. I. VW, "Safety first for automated driving," 2019.

[21] G. Escuela, N. Stroud, K. Takenaka, J.-K. Tiele, U. Dannebaum, J. Rosenbusch, M. Törngren, G. Niedrist, M. Antlanger, C. Mangold, F. Reisenberger, N. B. Mosbeh, C. Shinde, A. Mehmed, C. Schulze, S. More, L. Fryzek, and M. Storr, "Safe Automated Driving: Requirements and Architectures," The Autonomous, resreport, Dec. 2023.

[22] Aurora, "Aurora's Safety Case Framework," accessed: Dec 20, 2023. [Online]. Available: https://safetycaseframework.aurora.tech/gsn

[23] T. Victor, K. Kusano, T. Gode, R. Chen, and M. Schwall, "Safety performance of the Waymo rider-only automated driving system at one million miles," 2023. [Online]. Available: https://storage.googleapis.com/waymo-uploads/files/documents/safety/SafetyPerformanceofWaymoROat1Mmiles.pdf

[24] EasyMile, "Safety report," online, 2023. [Online]. Available: https://easymile.com/sites/default/files/easymile_safety_report_2023_4.pdf

[25] M. M. Selva Kumar, "Continuous improvement of driving automation," TU Delft, Tech. Rep., 2024. [Online]. Available: https://resolver.tudelft.nl/uuid:bb70525e-7197-460b-b81b-74d94e7b057a

[26] A. Bálint, V. Labenski, M. Köbe, C. Vogl, J. Stoll, L. Schories, L. Amann, G. B. Sudhakaran, P. H. Leyva, T. Pallacci, M. Östling, D. Schmidt, and R. Schindler, "D2.6 use case definitions andinitial safety-critical scenarios," SAFE-UP, Tech. Rep., 2021.

[27] P. Nitsche, P. Thomas, R. Stuetz, and R. Welsh, "Pre-crash scenarios at road junctions: A clustering method for car crash data," *Accident Analysis & Prevention*, vol. 107, pp. 137–151, 2017.

[28] J.-P. Paardekooper, S. van Montfort, J. Manders, J. Goos, E. de Gelder, O. O. den Camp, A. Bracquemond, and G. Thiolon, "Automatic identification of critical scenarios in a public dataset of 6000 km of public-road driving," in *26th International Technical Conference on the Enhanced Safety of Vehicles*, no. 19-0255, 2019.

[29] E. De Gelder, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. O. Den Camp, and B. De Schutter, "Real-world scenario mining for the assessment of automated vehicles," in *23rd IEEE International Conference on ITSC*. IEEE, 2020, pp. 1–8.

[30] E. De Gelder, H. Elrofai, A. K. Saberi, J.-P. Paardekooper, O. O. Den Camp, and B. De Schutter, "Risk quantification for automated driving systems in real-world driving scenarios," *Ieee Access*, vol. 9, pp. 168 953–168 970, 2021.

[31] I. Souflas, L. Lazzeretti, A. Ahrabian, L. Niccolini, and S. Curtis-Walcott, "Virtual verification of decision making and motion planning functionalities for automated vehicles in urban edge case scenarios," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 4, no. 2022-01-0841, pp. 2135–2146, 2022.

[32] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner case generation and analysis for safety assessment of autonomous vehicles," *Transportation research record*, vol. 2675, no. 11, pp. 587–600, 2021.

[33] F. Cai and X. Koutsoukos, "Real-time out-of-distribution detection in learning-enabled cyber-physical systems," in *2020 ACM/IEEE 11th IC-CPS*. IEEE, 2020, pp. 174–183.

[34] T. Vojir, T. Šipka, R. Aljundi, N. Chumerin, D. O. Reino, and J. Matas, "Road anomaly detection by partial image reconstruction with segmentation coupling," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 651–15 660.

[35] S. Ramakrishna, Z. Rahiminasab, G. Karsai, A. Easwaran, and A. Dubey, "Efficient out-of-distribution detection using latent space of β-vae for cyber-physical systems," *ACM Transactions on Cyber-Physical Systems (TCPS)*, vol. 6, no. 2, pp. 1–34, 2022.

[36] A. Stocco, M. Weiss, M. Calzana, and P. Tonella, "Misbehaviour prediction for autonomous driving systems," in *ACM/IEEE 42nd international conference on software engineering*, 2020, pp. 359–371.

[37] A. Balakrishnan, J. Deshmukh, B. Hoxha, T. Yamaguchi, and G. Fainekos, "Percemon: online monitoring for perception systems," in *RV 2021, October 11–14, 2021, Proceedings 21*. Springer, 2021, pp. 297–308.

[38] E. Schwalb, "Analysis of hazards for autonomous driving," *Journal of Autonomous Vehicles and Systems*, vol. 1, no. 2, p. 021003, 2021.

[39] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 222–17 231.

[40] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *arXiv preprint arXiv:2205.15997*, 2022.

[41] D. Ratiu, T. Rohlinger, T. Stolte, and S. Wagner, "Towards an argument pattern for the use of safety performance indicators," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2024, pp. 160–172.

[42] NXP, "S32G274A data sheet," Tech. Rep., 2022. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/S32G2.pdf

[43] Y. C. Yeh, "Design considerations in boeing 777 fly-by-wire computers," in *Proceedings Third IEEE International High-Assurance Systems Engineering Symposium (Cat. No. 98EX231)*. IEEE, 1998, pp. 64–72.