Run-Time Monitoring Frameworks for Validation of Autonomous Vehicles: A Focused Review of Applications

1st Alexander Fertig
Technische Hochschule Ingolstadt
Ingolstadt, Germany
alexander.fertig@thi.de

2nd Michael Botsch
Technische Hochschule Ingolstadt
Ingolstadt, Germany
michael.botsch@thi.de

***Abstract*—Assuring safety in autonomous vehicles is a key challenge, as outlined in ISO 21448 (SOTIF). Run-time monitoring frameworks address this challenge by continuously supervising the system behavior of autonomous vehicles during live operation to detect deviations and unknown risks. While existing reviews and surveys cover anomaly detection or general monitoring, none focus specifically on run-time monitoring frameworks for validation along the signal processing chain of autonomous vehicles. This review fills this gap by providing a systematic taxonomy of monitoring frameworks organized by application level in the automotive signal processing chain. Anomaly, novelty, outlier detection, and clustering approaches are organized by application level, from sensor inputs through perception and prediction to end-to-end systems, enabling practitioners to identify suitable methods and integration points. Key development challenges and future research directions are discussed.**

***Index Terms*—Autonomous Driving, Monitoring Frameworks, ISMR, Safety Assurance Framework, Run-Time, Safety, SOTIF.**

I. Introduction

Ensuring safety in highly automated driving functions or Autonomous Vehicles (AVs) is challenging, as the complexity of AV systems and the diversity of real-world environments make it difficult to verify safe behavior through development and testing alone. Many critical system behaviors and potential failures only emerge during real-world deployment.

The standard ISO 21448, Safety Of The Intended Functionality (SOTIF), [1] addresses this deployment phase, referred to as operation phase. SOTIF prescribes measures to supervise system behavior during operation to identify unknown risks (SOTIF areas 3 and 4 [1]). Monitoring frameworks provide continuous supervision of systems during their lifecycle. These frameworks observe system behavior to detect deviations from normal operation, typically operating in the background to provide ongoing safety validation.

While anomaly detection for AVs [2]–[4] and general monitoring frameworks [5] have been reviewed, monitoring frameworks for AVs are underrepresented in literature. In [6] a review on monitoring frameworks on automotive visual data is presented, but no study reviews monitoring frameworks along the whole signal processing chain. However, monitoring frameworks for AVs are critical for safety assurance and are the subject of current research [7]. Therefore, this work presents a focused review on run-time monitoring frameworks for AV validation along the signal processing chain and contributes by

- a systematic taxonomy organizing methods by application level along the AV signal processing chain, enabling targeted method selection for specific monitoring requirements,
- definitions of monitoring frameworks and automotive-specific characteristics, and
- a discussion of current challenges and future directions.

Scope and Focus: This review focuses on external run-time monitoring frameworks for validation of AVs. Since few works explicitly present complete frameworks, this review also includes anomaly, novelty, outlier detection, and clustering methods suitable as detection components within a monitoring framework. Methods are organized by application level along the AV signal processing chain, from raw sensor inputs to high-level scenario representations. This taxonomy provides a structured view of where different methods can be integrated within a comprehensive monitoring architecture.

Review Methodology: This work follows a focused, narrative review approach with a structured taxonomy. Literature was identified through searches in IEEE Xplore, Google Scholar, Semantic Scholar using keyword combinations of *monitoring*, *run-time*, *online*, *anomaly*, *novelty*, *out-of-distribution* with *autonomous driving* and *automated vehicle*, up to February 2026. Excluded were works related to automotive bus systems, security, or predictive maintenance. Selection was guided by expert judgment rather than systematic exhaustive coverage. A quantitative cross-method comparison has been withheld, as currently no standardized benchmarks and consistent evaluation protocols are available [3], [4], [6].

II. Automotive Monitoring Frameworks

A. Definitions and Terminology

The objective of monitoring frameworks is to supervise a given system and provide ongoing safety validation. These

979-8-3315-6165-9/26/$31.00 ©2026 IEEE.

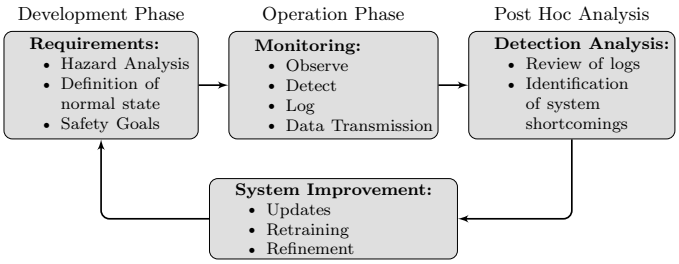

Fig. 1: Lifecycle view of passive monitoring frameworks.

frameworks observe system behavior to detect deviations from normal operation, typically operating in the background to provide continuous supervision. This includes detecting unusual or undesirable behavior in system inputs, outputs, or states, such as anomalies, Out-Of-Distribution (OOD) samples, novel samples, or safety violations. Monitoring frameworks typically consist of observation mechanisms, a detection component, and response mechanisms. This aligns with ISO 21448 (SOTIF) that prescribes *operation phase activities* to monitor AVs during operation to detect unknown states [1].

In the automotive context, monitoring frameworks are also referred to as In-Service Monitoring and Reporting (ISMR) [8]–[10]. ISMR typically represents one pillar within a high-level safety architecture, such as a Safety Assurance Framework (SAF) [7], [10], [11].

Monitoring frameworks are typically applied after development and testing when the system operates in real-world conditions. In literature, live operation is also referred to as *online* [12]–[14], *run-time* [15]–[19], *post-deployment* [7], or *operation phase* [1]. Figure 1 illustrates the lifecycle view of monitoring frameworks.

B. Core Components

Monitoring frameworks may be divided in three core components: (i) observation mechanisms, (ii) detection algorithms, and (iii) response mechanisms. These components work together to continuously assess the system state and identify any deviations that require attention.

1) Observation Mechanisms: Data is collected from the system during operation. In automotive applications, this includes sensor interfaces (camera, LiDAR, radar), internal system states (object lists, trajectories, control signals), and contextual information (vehicle dynamics, environmental conditions). Observation mechanisms must provide sufficient information for downstream detection while minimizing computational overhead.

2) Detection Component: This represents the core methodology within a monitoring framework and can be realized by e. g., statistical, model-based, or Machine Learning (ML)-based methods. In particular, clustering methods or techniques from anomaly, novelty, or outlier detection are often utilized. For instance, statistical methods use probability distributions or hypothesis tests to identify outliers. Rule-based approaches incorporate failure modes identified through hazard analysis. Detection methods may also be based on physical or kinematic models that evaluate system behavior. Finally, data-driven methods, particularly those based on ML, are increasingly utilized to identify normal behavior patterns in data and detect deviations [6].

In literature many methods are not explicitly presented as monitoring frameworks but can be integrated as detection components in such. Section III organizes such methods by their applicable level in the AV processing chain.

3) Response Mechanisms: Upon detection, monitoring frameworks usually execute a response pipeline. This involves logging relevant data, such as sensor readings, system states, and timestamps. Active systems can trigger immediate interventions in accordance with safety protocols. For passive systems, detections are queued for backend transmission, often accompanied by contextual data such as infrastructure information and environmental conditions.

C. Design Trade-Offs

1) Onboard and Offboard Frameworks: Data processing can be executed within the vehicle (onboard) or outside the vehicle (offboard). Given AV data volumes, offboard processing is impractical for large AV fleets. Monitoring frameworks are designed to operate throughout the system lifecycle and onboard systems can be updated over-the-air as the system and operational environment evolve.

2) Internal and External Frameworks: Internal frameworks monitor signals from within the AV system itself, such as internal system states or occupant monitoring [6]. External frameworks, by contrast, supervise the AV's driving performance and environmental interactions independently. The focus of this review is on external frameworks.

3) Active and Passive Application: Monitoring frameworks can also be classified as either active or passive. Active systems are designed to intervene in the current situation during run-time [17], [18]. In contrast, passive monitoring systems aim to detect events of interest during run-time without interfering with ongoing processes. Detected events may be sent to an evaluation entity (e. g., Original Equipment Manufacturer (OEM)) for post hoc analysis [15], [21]. The majority of monitoring frameworks in the literature are passive systems that detect and log events without intervention. An evaluation entity can utilize detections to improve the system development (see Figure 1), for example through:

- expert-based analysis to identify design flaws,
- detection of data drift or missing training and testing data,
- expansion of the training data set by detected samples,
- implementation of open-world learning frameworks [22].

4) Development Constraints: Two critical factors must be considered during the development of monitoring frameworks.

First, onboard systems may use only in-vehicle information, while offboard systems can exploit supplementary post-processed data, though post-processing is costly.

Second, particularly for data-driven approaches, it is essential to distinguish between data from the development/testing phase and the operation phase. Data from development/testing is considered *known* (normal), as problematic cases are either resolved or lie outside the Operational Design Domain (ODD),

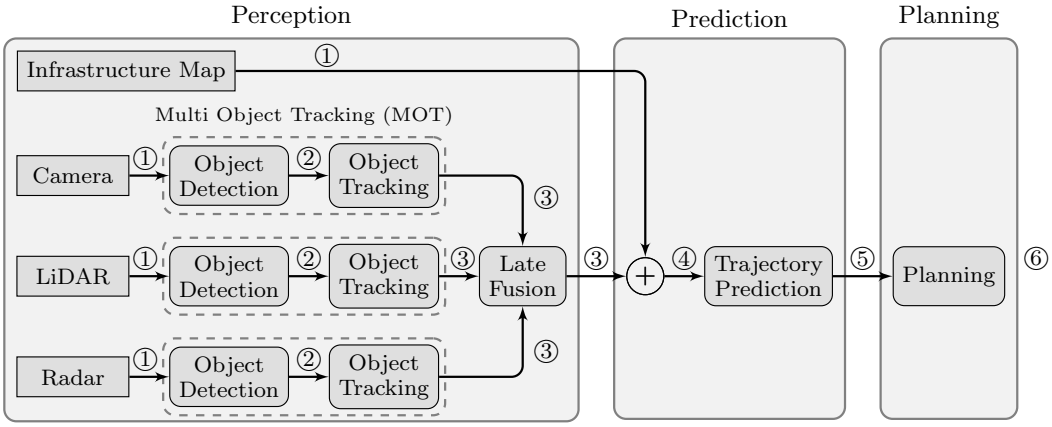

Fig. 2: Overview of an AV signal processing chain using late fusion, embedded within the Perception-Prediction-Planning scheme [20]. The numbers refer to different application levels: ① is input data level (Sec. III-A1), ② is object detection level (Sec. III-A2), ③ object state level (Sec. III-A3), ④ scenario level (Sec. III-B1), ⑤ is the trajectory prediction level (Sec. III-B2), and ⑥ represents the planning and end-to-end level (Sec. III-C and III-D1).

and is typically used to represent normal system behavior. During operation, the AV encounters new scenes and environmental conditions that may contain *unknown* risks and safety violations. Thus, *known* data corresponds to SOTIF areas 1 and 2, while *unknown* risks align with areas 3 and 4 [1], [23]. Furthermore, labeled anomalies and rule-based logic (e. g., from hazard analysis) confine detection to pre-anticipated failures and therefore cannot address the *unknown* risks required by ISO 21448 [1]. Thus, monitoring frameworks should avoid reliance on labeled anomaly data during training.

III. Application of Detection Methods along the Signal Processing Chain

Monitoring frameworks require detection components capable of identifying anomalies, novelties, and safety violations at various stages of the AV processing pipeline. While few works explicitly present complete monitoring frameworks, numerous methods have been identified that can serve as detection components within such frameworks. This section presents a taxonomy that organizes such methods and monitoring frameworks by their application level along the AV signal processing chain, illustrated by the Perception-Prediction-Planning scheme [20] in Figure 2, which represents a late fusion approach. The taxonomy generally applies also to alternative fusion architectures, such as early and mid fusion. Table I provides a tabular overview.

A. Perception-Level Monitoring

1) Input Data Level: Input data primarily comes from automotive sensors (camera, LiDAR, and radar), but may also include infrastructure maps. In [24], a method is proposed to segment sparse radar point clouds using a neural network to detect anomalous ghost targets and clutter. In [25], a variational autoencoder is utilized to encode camera images for anomaly detection. Further, deep learning techniques for camera-based anomaly detection are reviewed in [6]. Monitoring frameworks can also be applied at the infrastructure level. In [26], an entropy-based outlier score is used for novelty detection in infrastructure images. These approaches enable, for example, the detection of ODD violations. Additional input-level methods are surveyed in [2].

Common detection methods are designed to identify data points (map, image, radar and LiDAR data) that do not align with the training distribution and could lead to incorrect system behavior. Monitoring at this level enables early detection of ODD violations and sensor failures. However, distinguishing between novel but safe input data and actual hazards remains challenging, due to high diversity.

2) Object Detection Level: This level represents the output of object detection, usually as frame-based detections without temporal information.

Detection methods include autoencoder-based anomaly detection [27] and consistency checking using mirrored point clouds [28]. Inconsistencies between object detection results [29] or segmentation results [32] from camera and LiDAR sensors are also utilized to detect anomalies. Approaches from the field of open-set recognition are used in the automotive context to detect unknown objects [30], [31]. Further, in [33] and [43], the output of camera- and LiDAR-based segmentation networks is monitored to detect anomalous samples.

Common approaches at detection level include autoencoders and cross-sensor consistency checking, leveraging the availability of independent sensor detections for the same scene, enabling early detection of perception failures.

3) Object State Level: This level represents the output of the tracking component: trajectories representing objects over time in the automotive object list. The object state level represents a key interface within the architecture of AVs, as many subsequent components depend on it, e. g., trajectory generation [44] or trajectory prediction [45], [46]. In [14], a JEPA-based embedding method encodes the object states from an arbitrary sensor into a latent embedding space used for anomaly detection. Further, representative approaches utilize independent object states of different sensor modalities and compare their inconsistencies to detect unusual behavior. In

[23], an autoencoder with a *k*-means friendly embedding procedure, was used to embed object states from camera and LiDAR sensors into a latent representation space to identify unknown object behavior. Further, in [34], spoofing attacks are detected by the comparison of camera and radar data.

Frequently used detection methods at object state level include cross-sensor consistency checking and embedding-based approaches in the combination with anomaly detection and clustering techniques. The temporal nature of object states (trajectories) enables detection methods that analyze motion patterns over time. The level of object states represents a critical monitoring point, as it serves as a key interface between perception and planning, with many downstream components depending on its accuracy. Trajectory representations are lower-dimensional than input sensor data, enabling more sophisticated detection methods within computational budgets.

B. Prediction-Level Monitoring

1) Scenario-Level: Within a traffic scenario representation, all relevant information (objects detected by the perception system, ego-data, infrastructure data) is combined.

Representative approaches utilize ML-based encoders to embed traffic scenarios into a latent representation space for clustering or anomaly detection. For this approach, representation learning methods are primarily used, such as triplet learning [35], quadruplet learning [36], Self-Supervised Learning (SSL) [37], [38], or autoencoder variations [39]–[41].

Detection methods on scenario level often use metric learning, SSL or autoencoder approaches. These methods embed high-dimensional scenario representations into latent spaces enabling efficient clustering and anomaly detection. Scenario-level monitoring enables the assessment of traffic scenarios, capturing relationships between multiple objects and infrastructure. This abstraction level is suitable for identifying complex, multi-agent interactions that may not be apparent at lower levels. However, defining appropriate scenario representations remains challenging due to real-world diversity.

2) Trajectory Prediction Level: At this level, monitoring frameworks assess the predicted future trajectories of other traffic participants. Predictions are made based on the scenario representation and serve as a critical input to the subsequent planning stage. In [13], a performance monitoring model is proposed to evaluate black-box trajectory prediction models. The monitoring model evaluates predicted trajectories and identifies insufficient predictions to support safe and informed decision-making. The model is designed for online application and operates as a standalone system.

C. Planning-Level Approaches

At this level, monitoring frameworks supervise the planned trajectory of the ego vehicle. The planning module integrates the output of the perception and prediction stages to compute a safe and feasible trajectory for the ego vehicle. Monitoring at this level evaluates the planner's output before execution, serving as a final safety check prior to actuation. Approaches include checking predicted trajectories against safety criteria such as collision avoidance, friction limits, and vehicle dynamics [12], [17], [18].

D. Cross-Level Approaches

1) End-to-End Level: End-to-end approaches such as the Alpamayo Reasoning-VLA-Model [47] represent a recent development. These approaches omit intermediate steps and

Reference	Application Level	Detection Method	Sensor(s)	Dataset	RT
Griebel et al. [24]	Input Data	PointNet++ MFG	Radar	Custom	Yes
Bogdoll et al. [25]	Input Data	CL-VAE	Camera	Cityscapes, Lost&Found	-
Wurst et al. [26]	Input Data (Infra.)	Entropy-based Outlier	-	Custom	-
Piroli et al. [27]	Detection	LS-VOS	LiDAR	Custom	-
Rivera et al. [28]	Detection	Consistency (Mirror)	LiDAR	KITTI	-
Rivera et al. [29]	Detection	Cross-sensor	Camera + LiDAR	KITTI, nuScenes	-
Sekaran et al. [30]	Detection	Prototype Mining	Camera	MNIST, CiFAR-10, TinyImageNet	Yes
Sekaran et al. [31]	Detection	Multi-Modal Usage	Camera + LiDAR	TUMTraf-I, Custom	Yes
Bogdoll et al. [32]	Detection (Seg.)	Multi-Modal AD	Camera + LiDAR	Custom	-
Shoeb et al. [33]	Detection (Seg.)	Encoding + GMM	Camera	Cityscapes, StreetObstacleSequences	-
Fertig et al. [14]	Object State	JEPA Embedding	Single Sensor	nuScenes	Yes
Fertig et al. [23]	Object State	AE + k-means	Camera + LiDAR	nuScenes	-
Alkanat et al. [34]	Object State	Consistency Check	Camera + Radar	RADIal dataset	Yes
Wurst et al. [35]	Scenario	Triplet Learning	-	Custom	-
Wurst et al. [36]	Scenario	Quadruplet Learning	-	Custom, SUMO	-
Balasubramanian et al. [37], [38]	Scenario	Self-supervised	-	highD / Argoverse	-
Sonntag et al. [39]	Scenario	Autoencoder	-	exiD	-
Roßberg et al. [40], [41]	Scenario	CVQ-VAE	-	highD	-
Shao et al. [13]	Prediction	PMBP	-	SIND, INTERACTION	Yes
Stahl et al. [12]	Planning	The Supervisor	-	Custom	Yes
Grieser et al. [17]	End-to-End / Planning	Dependability Cage	LiDAR	Custom	Yes
Wang et al. [18]	End-to-End / Planning	ARGUS	-	Bench2Drive, CARLA	Yes
Hashemi et al. [16], [42]	End-to-End	Gaussian Activations	Camera	MNIST, F-MNIST and CIFAR-10	Yes

TABLE I: Overview of reviewed methods and key characteristics. RT represents if the authors stated the run-time capability.

interfaces, complicating interpretation of the models. Consequently, monitoring systems are imperative for safety assurance. In [17], a monitoring framework for a miniature AV is introduced that monitors LiDAR data along a predicted trajectory and triggers emergency braking if required. The hazard monitor from [18] uses bird's eye view data and motion prediction to asynchronously identify potential collisions, stop signal violations, and stalling hazards. Further, there are approaches to monitor the neuron activation patterns of neural networks using Gaussian distributions in order to identify OOD samples [16], [42]. Monitoring end-to-end approaches is challenging, as the absence of interpretable intermediate steps hinders fault localization, complicates safety certification and increases the computational burden for run-time validation.

2) Behavioral-Level: In contrast to monitoring frameworks tied to specific processing chain components, behavioral-level frameworks evaluate the entire system. For instance, in [19], a Scene Graph Generator abstracts sensor data into semantic scene graphs and evaluates them against driving rules (such as traffic laws) using a logic-based language and temporal logic to identify behavioral violations in real-time.

This review provides an overview of detection methods for automotive monitoring frameworks. For comprehensive surveys on automotive anomaly detection, see [2]–[4].

IV. Challenges and Future Directions

Onboard monitoring frameworks operate on AV edge hardware with limited computational budgets. Minimizing resource usage while maintaining adequate performance remains a key challenge. For passive systems, real-time computational constraints are relaxed, as post hoc detection is sufficient. Offboard frameworks do not operate on edge devices; however, given AV data volumes, data transfer is challenging.

Monitoring frameworks must balance multiple performance criteria. A trade-off must be found between achieving a high True Positive Rate (TPR) (detection rate) and minimizing the False Positive Rate (FPR) (false alarm rate). For post hoc evaluation, limited false-alarm budgets must be considered, as excessive false alarms overwhelm evaluation capacities. In-vehicle deployment requires careful trade-offs between detection capability and resource usage, as computational efficiency and latency are essential considerations.

Key challenges include defining *normal* behavior in complex systems and handling distribution shifts as environments evolve. Further, a significant challenge is the lack of standardized evaluation metrics and datasets with labeled real-world automotive anomaly data (specifically on object state and scenario level) [3], [4], [6]. Different approaches use different datasets, making direct comparison difficult. Additionally, validating developed methods is challenging given the rarity of anomalies in real-world environments [48]. Lastly, the recent trend toward end-to-end approaches is particularly challenging to monitor, as clear intermediate steps are omitted.

The complexity of AV systems necessitates monitoring frameworks at multiple levels for comprehensive run-time validation. Key future research directions include:

1) Multi-level integration: Aligned monitoring across the processing chain for cross-level consistency evaluation.
2) Standardized evaluation: Development of benchmark datasets and protocols to enable fair comparison.
3) Adaptive monitoring: Integration with continual and active learning to handle distribution shifts.
4) Resource-efficient methods: Optimization of ML-based approaches for edge deployment while maintaining detection performance.
5) End-to-end system supervision: Novel monitoring strategies for black-box models, including activation analysis and uncertainty quantification.

V. Conclusion

This work presents a focused review on run-time monitoring frameworks for the validation of AVs, explaining why monitoring frameworks are critical for AV safety assurance and discussing application opportunities and key characteristics.

The main contribution is organizing monitoring frameworks and methods by their application level in the automotive signal processing chain. This taxonomy spans from input data level monitoring (raw sensor inputs, infrastructure data) through perception-level (object detection, object state tracking) to prediction-level (scenario representation, trajectory prediction) and end-to-end approaches. For each level, representative detection methods, common techniques, and specific challenges are identified. While few works present complete monitoring frameworks, numerous anomaly, novelty, and outlier detection approaches can serve as detection components within such frameworks. This structured view enables practitioners to identify suitable methods for their specific application level and integration requirements.

Key challenges include computational constraints, detection accuracy trade-offs, and lack of standardized benchmarks. Future priorities include resource-efficient architectures, evaluation standardization, and solutions for end-to-end approaches. Monitoring frameworks will enable safe AV deployment at scale, providing ongoing validation as prescribed by SOTIF.

Acknowledgment

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] "ISO 21448:2019, Road vehicles – Safety of the intended functionality," ISO, Std., 2019.
[2] D. Bogdoll, M. Nitsche, and J. M. Zöllner, "Anomaly Detection in Autonomous Driving: A Survey," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. abs/2204.07974, 2022.
[3] S. Baccari, M. Hadded *et al.*, "Anomaly Detection in Connected and Autonomous Vehicles: A Survey, Analysis, and Research Challenges," *IEEE Access*, vol. 12, pp. 19 250–19 276, 2024.
[4] J. R. V. Solaas, E. Mariconti, and N. Tuptuk, "Systematic Literature Review: Anomaly Detection in Connected and Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 43–58, Jan. 2025.

[5] H. Naveed, S. Barnett *et al.*, "Monitoring Machine Learning Systems: A Multivocal Literature Review," *arXiv preprint arXiv:2509.14294*, 2025.

[6] F. Caetano, P. Carvalho, and J. Cardoso, "Deep Anomaly Detection for In-Vehicle Monitoring—An Application-Oriented Review," *Applied Sciences*, vol. 12, no. 19, p. 10011, Oct. 2022.

[7] CERTAIN, "Resilient and Continuous Safety Assurance Methodology for CCAM and its HMI Components," https://www.certainproject.eu/, EU Horizon Europe project, 2025.

[8] M. C. Galassi, A. Lagrange *et al.*, "Safety Approval of Automated Vehicles in the EU: Moving Beyond Highway Applications," *Transportation Research Procedia*, vol. 72, pp. 4396–4403, 2023.

[9] O. O. D. Camp and E. De Gelder, "Operationalization of Scenario-Based Safety Assessment of Automated Driving Systems," in *2025 IEEE International Automated Vehicle Validation Conference (IAVVC)*. Baden-Baden, Germany: IEEE, Sep. 2025, pp. 1–6.

[10] E. De Gelder, T. Singh *et al.*, "Scenario Metrics for the Safety Assurance Framework of Automated Vehicles: A Review of Its Application," *Vehicles*, vol. 7, no. 3, p. 100, Sep. 2025.

[11] M. Oldoni and S. Khastgir, "Introducing ODD-SAF: An Operational Design Domain Safety Assurance Framework for Automated Driving Systems," in *Road Vehicle Automation 10*, G. Meyer and S. Beiker, Eds. Cham: Springer Nature Switzerland, 2023, pp. 133–151.

[12] T. Stahl, M. Eicher *et al.*, "Online Verification Concept for Autonomous Vehicles – Illustrative Study for a Trajectory Planning Module," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Rhodes, Greece: IEEE, Sep. 2020, pp. 1–7.

[13] W. Shao, B. Li *et al.*, "When Is It Likely to Fail? Performance Monitor for Black-Box Trajectory Prediction Model," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 17 355–17 369, 2025.

[14] A. Fertig, K. C. Sekaran *et al.*, "Online Monitoring Framework for Automotive Time Series Data using JEPA Embeddings," *arXiv preprint arXiv:2602.09985*, Feb. 2026.

[15] M. H. Osman, S. Kugele, and S. Shafaei, "Run-Time Safety Monitoring Framework for AI-Based Systems: Automated Driving Cases," in *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, Dec. 2019, pp. 442–449.

[16] V. Hashemi, J. Křetínský *et al.*, "Gaussian-Based Runtime Detection of Out-of-distribution Inputs for Neural Networks," in *Runtime Verification*. Springer International Publishing, 2021, pp. 254–264.

[17] J. Grieser, M. Zhang *et al.*, "Assuring the Safety of End-to-End Learning-Based Autonomous Driving through Runtime Monitoring," in *2020 23rd Euromicro Conference on Digital System Design (DSD)*. Kranj, Slovenia: IEEE, Aug. 2020, pp. 476–483.

[18] D. Wang, Y. Lu *et al.*, "Argus: Resilience-Oriented Safety Assurance Framework for End-to-End ADSs," in *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Seoul, Korea, Republic of: IEEE, Nov. 2025, pp. 13–25.

[19] F. Toledo, T. Woodlief *et al.*, "Specifying and Monitoring Safe Driving Properties with Scene Graphs," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 15 577–15 584.

[20] Y. Hu, J. Yang *et al.*, "Planning-oriented Autonomous Driving," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 17 853–17 862.

[21] C. Segler, S. Kugele *et al.*, "Anomaly Detection for Advanced Driver Assistance Systems Using Online Feature Selection," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2019, pp. 578–585.

[22] L. Balasubramanian, J. Wurst *et al.*, "Open-World Learning for Traffic Scenarios Categorisation," *IEEE Transactions on Intelligent Vehicles*, pp. 1–16, 2023.

[23] A. Fertig, L. Balasubramanian, and M. Botsch, "Clustering and Anomaly Detection in Embedding Spaces for the Validation of Automotive Sensors," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[24] T. Griebel, D. Authaler *et al.*, "Anomaly Detection in Radar Data Using PointNets," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. Indianapolis, IN, USA: IEEE, Sep. 2021.

[25] D. Bogdoll, S. Pavlitska *et al.*, "Conditioning Latent-Space Clusters for Real-World Anomaly Classification," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Dec. 2023.

[26] J. Wurst, A. F. Fernández *et al.*, "An Entropy Based Outlier Score and its Application to Novelty Detection for Road Infrastructure Images," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1436–1443.

[27] A. Piroli, V. Dallabetta *et al.*, "LS-VOS: Identifying Outliers in 3D Object Detections Using Latent Space Virtual Outlier Synthesis," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. Bilbao, Spain: IEEE, Sep. 2023, pp. 1242–1248.

[28] E. Rivera, L. Stratil, and M. Lienkamp, "Inconsistency-Based Active Learning for LiDAR Object Detection," in *2025 IEEE Intelligent Vehicles Symposium (IV)*. Cluj-Napoca, Romania: IEEE, Jun. 2025.

[29] E. Rivera, A. C. S. do Nascimento, and M. Lienkamp, "Camera-LiDAR Inconsistency Analysis for ActiveLearning in Object Detection," in *2024 IEEE Intelligent Vehicles Symposium (IV)*. Jeju, Korea: IEEE, 2024.

[30] K. C. Sekaran, L. Balasubramanian *et al.*, "Metric Learning Based Class Specific Experts for Open-Set Recognition of Traffic Participants in Urban Areas Using Infrastructure Sensors," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2023.

[31] K. C. Sekaran, L. Balasubramanian *et al.*, "Open-Set Object Detection for the Identification and Localization of Dissimilar Novel Classes by means of Infrastructure Sensors," in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2024.

[32] D. Bogdoll, E. Eisen *et al.*, "Multimodal Detection of Unknown Objects on Roads for Autonomous Driving," *arXiv preprint arXiv:2205.01414*, 2022.

[33] Y. Shoeb, A. Nowzad, and H. Gottschalk, "Adaptive Neural Networks for Intelligent Data-Driven Development," in *2025 IEEE Intelligent Vehicles Symposium (IV)*. Cluj-Napoca, Romania: IEEE, Jun. 2025.

[34] T. Alkanat and A. Pandharipande, "Deep Sensor Fusion for Detection and Localization of Automotive Radar Spoofing Attacks," in *2025 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2025, pp. 1–6.

[35] J. Wurst, L. Balasubramanian *et al.*, "Novelty Detection and Analysis of Traffic Scenario Infrastructures in the Latent Space of a Vision Transformer-Based Triplet Autoencoder," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 1304–1311.

[36] J. Wurst, L. Balasubramanian *et al.*, "Expert-LaSTS: Expert-Knowledge Guided Latent Space for Traffic Scenarios," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 484–491.

[37] L. Balasubramanian, J. Wurst *et al.*, "Traffic Scenario Clustering by Iterative Optimisation of Self-Supervised Networks Using a Random Forest Activation Pattern Similarity," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 682–689.

[38] L. Balasubramanian, J. Wurst *et al.*, "ExAgt: Expert-guided augmentation for representation learning of traffic scenarios," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 1471–1478.

[39] M. Sonntag, L. Vater *et al.*, "Detecting Edge Cases from Trajectory Datasets Using Deep Learning Based Outlier Detection," in *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*. SciTePress, May 2024.

[40] N. Roßberg, M. Neumeier *et al.*, "Assessing the Completeness of Traffic Scenario Categories for Automated Highway Driving Functions via Cluster-Based Analysis," in *2025 IEEE Intelligent Vehicles Symposium (IV)*. Cluj-Napoca, Romania: IEEE, Jun. 2025, pp. 1787–1794.

[41] N. Roßberg, S. Hasirlioglu *et al.*, "Behavior-Centric Extraction of Scenarios from Highway Traffic Data and their Domain-Knowledge-Guided Clustering using CVQ-VAE," *arXiv preprint arXiv:2603.16964*, 2026.

[42] V. Hashemi, J. Křetínský *et al.*, "Gaussian-Based and Outside-the-Box Runtime Monitoring Join Forces," in *Runtime Verification*. Springer Nature Switzerland, Oct. 2024, pp. 218–228.

[43] D. Bogdoll, F. Sartoris *et al.*, "Label-Free Model Failure Detection for Lidar-based Point Cloud Segmentation," in *2025 IEEE Intelligent Vehicles Symposium (IV)*. Cluj-Napoca, Romania: IEEE, Jun. 2025.

[44] R. Egolf, A. Fertig, and M. Botsch, "Conditioned Trajectory Generation for Realistic Driving Scenarios via a Hybrid Machine Learning Architecture," in *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Nov. 2025, pp. 3297–3304.

[45] M. Neumeier, S. Dorn *et al.*, "Reliable Trajectory Prediction and Uncertainty Quantification with Conditioned Diffusion Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2024, pp. 3461–3470.

[46] A. Fertig, L. Balasubramanian, and M. Botsch, "Hybrid Machine Learning Model with a Constrained Action Space for Trajectory Prediction," in *2025 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2025, pp. 38–44.

[47] NVIDIA, Y. Wang *et al.*, "Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail," *arXiv preprint arXiv:2511.00088*, Jan. 2026.

[48] D. Bogdoll, "Anomaly Detection for Autonomous Driving," Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2026.