Towards Monitoring of AI Models in Automated Driving Functions

Michael Langer*† , Lukas Hacker*, Steven Peters†

Abstract: Artificial Intelligence (AI) represents a key enabling technology for automated driving functions, particularly in perception and planning tasks. However, ensuring the safety of AI-based systems remains challenging. Key reasons include black-box characteristics, limitations in robustness and reliability, and possible deviations from the intended scope of application. Given the difficulties in exhaustively testing these systems, the need for monitoring during operation after deployment is of high importance. To address these challenges, the recently published ISO/PAS 8800 standard introduces a structured framework for the development and operation of AI-based systems in safety-related automotive contexts. While the standard defines an AI safety lifecycle and related measures, it leaves considerable freedom in their technical realization. Runtime monitoring is a key mechanism within this context, enabling the detection of safety-relevant deviations during operation. However, the diversity of existing monitoring approaches leads to challenges for their systematic comparison, selection, and integration. This paper provides a comprehensive overview of existing monitoring taxonomies and proposes a novel taxonomy for AI monitors aligned with ISO/PAS 8800. The proposed taxonomy aims to improve the traceability and comparability of monitoring strategies, thereby strengthening AI safety argumentation and aiding in the fulfillment of relevant safety standards.

Keywords: Automated Driving, Artificial Intelligence, Monitoring, Safety Argumentation.

1 Introduction

An important driving force and enabler of progress in the field of automated driving is the use of deep learning for perception and planning tasks [1]. However, demonstrating the safety of these AI functions is a challenge due to the black box properties and complex, content-dependent behavior of AI. Typical AI components for perception tasks are trained under the assumption of a closed world in which all inputs can be assigned to a known category. In the real world, scenarios may arise that were not accounted for during training, which may lead to unexpected system responses.

The recently published safety standard ISO/PAS 8800 [2] defines a comprehensive set of measures covering the entire AI lifecycle to ensure safe development and deployment of AI-based functions in road vehicles. Among these measures, runtime monitoring is

*† Mercedes-Benz AG, Research & Development, 71059 Sindelfingen, Germany and Institute of Automotive Engineering Darmstadt (FZD), TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Germany (e-mail: michael.langer@mercedes-benz.com)

* Mercedes-Benz AG, Research & Development, 71059 Sindelfingen, Germany (e-mail: lukas.l.hacker@mercedes-benz.com).

† Institute of Automotive Engineering Darmstadt (FZD), TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Germany (e-mail: steven.peters@tu-darmstadt.de)

identified as a key mechanism to detect safety-related deviations during operation. However, AI runtime monitoring comprises a diverse set of technical approaches, including uncertainty estimation, explainability methods, and out-of-distribution detection techniques. This diversity motivates the development of a common framework for their systematic classification, comparison, and selection in relation to ISO/PAS 8800. Building on this motivation, this paper presents a structured overview of existing runtime monitoring taxonomy approaches and introduces a novel monitoring taxonomy aligned with the safety-related properties and influencing factors defined in ISO/PAS 8800.
This paper is structured as follows. Section 2 discusses the need for monitoring, addressing in section 2.1 the various safety-related insufficiencies associated with AI, while section 2.2 provides an overview of different automotive-specific standards. In section 3, we then present, discuss, and compare various taxonomy approaches found in literature and propose in section 4, a novel structured ISO/PAS 8800-related monitoring taxonomy. Finally, section 5 concludes the paper and outlines perspectives for further research.

2 Motivation for Monitoring of AI Functions

Due to various safety-related insufficiencies with AI-based systems, the automotive-specific safety standards ISO/PAS 8800 and ISO 21448 highlight the need for runtime monitoring to detect AI related errors and to support the safety assurance during operation. This section provides an overview of different safety conerns (see section 2.1) and the standardization landscape (see section 2.2).

2.1 Potential Safety-related Insufficiencies of AI in Automated Driving Systems

The use of AI is attractive for automotive applications, because it can learn complex perception or planning tasks based on data, without being explicitly programmed [3]. However, this data-driven nature can result in various safety concerns during development and deployment in safety-critical functions [4].
First, it could lead to **safety concerns in terms of compliance with the intended scope of application**, described by the Operational Design Domain (ODD). Because AI models are learned from examples, their decision boundaries are not fully explicit. As a result, formal verification of all relevant safety conditions is often not feasible [5]. This can result in a divergence between the intended learning output and what the model actually learned [6]. In addition, the high complexity of the AI models and the environment can lead to a semantic gap between the intended and specified functionality. A semantic gap occurs when the range of implicit, ambiguous intentions within a system exceeds the scope of its explicit, well-defined specifications [7], [8], [9].
Since the model is trained solely based on data, there is also a lack of implementation transparency, and the model exhibits **black-box characteristics** [4], [6], [10]**.** This means that the decision-making process usually remains unclear, and it is difficult to trace the model predictions [4]. The models rely on learned features extracted from data representations instead of directly using semantic information in the input. The model may encode incorrect or unintended decision logic without being noticed [6], [11].
A third Insufficiency relates to the **robustness and performance** under real-world conditions [4], [6], [11]. Neural networks tend to generate overconfident outputs. Therefore, the output score cannot be trusted to detect errors [4]. In addition, neural networks can react

unpredictably to minor disturbances in the sensor setup, small environmental changes or noisy inputs [11]. As a result, the ability to generalize unknown inputs may be disrupted and the model may provide incorrect outputs with a high level of confidence [12].
A fourth safety concern relates to **testing and verification** of AI-based automotive systems. Due to the high dimensionality of the data used for training and operation, it is infeasible to exhaustively test all possible input combinations or to fully validate the complete ODD [4], [6]. In addition, some AI models rely on probabilistic inference and may exhibit non-deterministic behavior. Consequently, identical inputs do not always yield identical outputs. This contrasts with deterministic systems, where the system behavior is fully defined by the initial conditions and can be precisely predicted. As a consequence, traditional verification and validation methods are often insufficient for AI systems, increasing the risk that unsafe behaviors remain undetected during development [13].

2.2 Automotive-specific Safety Standards

The safety standard **ISO 21448** - *Safety of the Intended Functionality (SOTIF)* [14] provides a framework that addresses "unreasonable risks due to hazards caused by functional insufficiencies" [14]. As real-world operation is characterized by open-world conditions and inherent uncertainty, not all safety-relevant situations can be anticipated during development and validation. The monitoring methods described in section 13 of the standard therefore complement the SOTIF measures by supporting the detection of potentially unsafe situations that may emerge during operation [14]. The standard also defines a hazardous event model that describes the causal chain from triggering conditions to hazardous events, as illustrated in Figure 1. In this model, triggering conditions lead to hazardous behavior of the system which can result in a hazard. Combined with a scenario in which the hazard can lead to harm, this results in a hazardous event [14]. The triggering conditions out of SOTIF can be related to the influencing factors defined in ISO/PAS 8800.

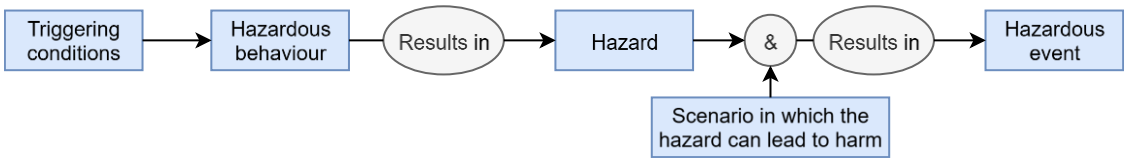

Figure 1: SOTIF related hazardous event model [14]

The recently published AI safety standard **ISO/PAS 8800** - *Road vehicles - Safety and AI* [2] provides domain-specific guidance for the application of AI within safety-related systems, expanding the established methodologies of ISO 26262 and ISO 21448 [2]. The standard outlines various AI safety measures for systematically addressing AI specific safety-related properties (e.g. robustness, explainability) throughout the AI lifecycle. Chapter 14 of ISO/PAS 8800 outlines measures during the operation phase, including an overview of different monitoring possibilities for deployed AI systems.
As described above, a central concept introduced by the standard is the notion of **Safety-Related Properties (SRPs)**, defined as inherent characteristics of an AI system. The absences of SRPs potentially lead to insufficiencies and make it difficult to argue the safety of the system [2]. Annex D provides a list of SRPs, which have to be addressed by specific safety measures from a safety perspective. The applicability of individual SRPs depends on the specific use case, system architecture, and model type. The standard recommends

reviewing the fulfillment and consistency of SRPs used to define AI safety requirements and allows them to be used as a checklist to derive refined qualitative safety requirements. Since SRPs follow an inductive approach, exhaustiveness is not guaranteed, and they are intended as a complementary element within the overall safety argumentation [2].
To systematically address the root causes of AI insufficiencies, the standard further introduces **Influencing Factors**, defined as "factors contributing to the achievement or absence of a safety-related property" [2]. These are categorized into four classes: (1) *Observation* factors, related to data representing the input space in training and test datasets; (2) *Label* factors, related to the quality and correctness of data annotations; (3) *Model* factors, related to the ML model itself; and (4) *Operation* factors, related to changes within the input space during deployment [2]. By mapping AI safety requirements to SRPs and their associated influencing factors, SRPs can be systematically considered throughout the entire lifecycle, i.e. from requirements to monitoring. As a result, the standard can provide a structured argumentation framework for the safety assurance of AI-based automated driving systems.

3 Taxonomy Approaches of AI-Monitors

The literature presents a variety of approaches for classifying and clustering AI monitoring methods, reflecting the complex and multifaceted nature of AI systems and their operational environments. This subsection provides an overview of various published taxonomy approaches.

3.1.1 Definitions and requirements

Runtime verification is a field within computer science dedicated to developing and applying methods that enable verification of whether a system behaves as intended during execution in accordance with its specified correct properties or if it is violating them [15]. The verification of an execution compliance with a correctness property is usually performed using a monitor [5]. Leucker and Schallhart define a monitor as "a device that reads a finite trace and yields a certain verdict [15]".
The successful and reliable implementation of monitors depends on meeting several requirements. Goodloe et al. [16] identified four key requirements: Functionality, meaning the monitor only intervenes if the monitored system violates its specified behavior; Schedule-ability, ensuring the monitor doesn't hinder the system's real-time performance; Reliability, requiring the monitored system's reliability to be maintained or improved; and Certifiability, meaning the monitor's design shouldn't demand excessive code changes.
Ferreira et al. [4] further emphasize the importance of ensuring with the monitor that the system avoids a safety-critical state, guarantees high system availability, and consistently meeting the system's runtime conditions.

3.1.2 System Definition

A fundamental aspect of any monitoring strategy is defining the scope of observation. This involves clarifying what is to be monitored. This can range from the isolated **AI model** (e.g. single DNN) itself to the broader **AI system** encompassing pre- and post-processing components, or even the entire **overall system** in which the AI operates. Furthermore, the kind of AI being monitored has a significant impact on monitor design. This can be categorized according to AI functionalities such as **perception** (e.g., classification,

segmentation, object detection, tracking), **planning**, or **acting**. Each of these AI types presents unique monitoring challenges and requirements.

3.1.3 Architectural measures

Architectural considerations dictate how monitoring is implemented and integrated within a system. A primary distinction, as highlighted by Leucker et al. [15], is between **online monitoring** and **offline monitoring**. Online monitors continuously check the current execution of a system in real-time, designed for incremental and efficient processing of data streams. In contrast, offline monitoring operates on a finite set of recorded executions, often involving post-analysis of logs or stored data.

Another architectural dimension, particularly relevant for automated systems, is the distinction between **on-board** and **off-board monitoring** [2]. Onboard monitoring is running in the vehicle. Off-board monitoring is running outside the vehicle, often employing cloud-based solutions to monitor the general behaviors of multiple AI-equipped systems, such as a fleet of vehicles and typically leverages higher computational power and more precise models.

Rahman et al. [17] further categorize monitors based on their operational principles, including those relying on **past experiences**, those detecting **inconsistencies during inference**, and those utilizing **uncertainty estimation and confidence** levels.

Rahman et al. [17] also differentiate between the nature of knowledge acquisition for monitoring. **Implicit** monitors do not require training with examples of successful or incorrect classifications. **Explicit** monitors, conversely, leverage examples of successful or incorrect classifications derived during the design phase.

Ferreira [6], building on Rahman's [18] work, distinguishes between **internal (integrated)** mechanisms and **external (parallel)** mechanisms. Internal mechanisms are deeply embedded within the AI system, learning through rejection or uncertainty estimation and incorporating domain knowledge. External mechanisms, in contrast, operate parallel and monitoring the **inputs**, **outputs**, or **internal layers** of the AI system.

Temporally, monitors can be classified as **predictive**, anticipating future issues; **descriptive**, explaining past events; or **diagnostic**, elucidating the causes of observed occurrences.

3.1.4 Addressed Influencing Factors, Threats and Triggering Conditions

An important aspect of AI monitoring is identifying and addressing potential hazards and their triggering conditions. The ISO/PAS 8800 [2] standard identifies several **influencing factors** that can impact AI system behavior, including operation-, model-, observations-, and label-related influencing factors.

Ferreira [4] categorizes monitors based on the threats they address (**in-distribution errors**, **novelty threats**, and **distributional shift threats**)**.** This is closely related to the triggering condition categorization approach by Hacker et al**.** [19], which differentiates between monitors for **in-distribution** triggers, **out-of-distribution (OOD)** triggers, and **adversarial attacks**.

In addition, the ISO/PAS 8800 [2] distinguishes monitors according to the specific events they can report. This includes **input-related events** such as the detection of OOD data, distributional shifts, and concept drifts (e.g., new objects or changed rules). **Output-related events** encompass the detection of abnormal outputs, output bias, and outputs with low confidence. **Model behavior-related events** involve identifying abnormal behaviors, and

incident/accident analysis focuses on investigating accidents in which the AI system was involved.

3.1.5 AI Insufficiencies and Safety-related Properties

AI monitors are crucial for strengthening the safety argumentation of AI systems. Monitors can be differentiated depending on which **safety-related properties** they contribute to or which existing insufficiencies they address. Safety-related properties, as outlined in ISO/PAS 8800, include aspects such as explainability, robustness, predictability, generalization capability or distributional shifts over time [2].

In addition, monitors can be designed to address various **AI insufficiencies** [19]. These include a lack of data completeness, poor data quality, insufficient model interpretability, model incorrectness, inadequate model robustness, and insufficient uncertainty representation.

3.1.6 Output and Reaction

The utility of an AI monitor lies in its output and the subsequent reactions it enables. Output adaptation mechanisms, as described by Lukina et al. [20], can involve the **adaptation of new classes**, where an automatic framework with monitoring and adaptation modes works in parallel to the original neural network, recognizing and adding new input classes during prediction time. Another form is the **adaptation of reliability scores**, which involves calculating a quantitative metric for the confidence of each observed input-output pair.

The output of the monitor itself can take various forms, such as a **failure/uncertainty score** or an **indication of OOD** input. Crucially, the reaction to the monitoring input dictates the system's response. This can range from a **control system switch** to **immediate prediction enhancement** or even **long-term system enhancement**, demonstrating the monitor's role in both immediate risk mitigation and continuous improvement of AI system safety and performance [4].

3.1.7 Technical Approaches

Various approaches for monitoring deep neural networks at runtime have been proposed in recent literature. Lust et al. [21] or Mohseni et al. [12] identify three principal research fields for runtime monitoring: predictive uncertainty estimation, out-of-distribution detection, and adversarial detection.

Predictive **uncertainty estimation** aims to quantify the reliability of a model's predictions beyond the raw softmax output. Henne et al. [22] and Arnez et al. [23] provide comparisons of different uncertainty estimation methods. These can be categorized into sampling-based and sampling-free approaches. Sampling-based methods include Monte Carlo Dropout (MCDO) and Deep Ensembles (DE). MCDO introduced by Gal and Ghahramani [24], approximates Bayesian inference through multiple stochastic forward passes with active dropout at inference time. DE aggregate predictions from independently trained models where disagreement among members indicates uncertainty [25]. Sampling-free alternatives, such as Learned Confidence Estimation by DeVries and Taylor [26], train an auxiliary branch to directly predict prediction confidence without requiring multiple forward passes.

Out-of-distribution (OOD) detection identifies inputs that deviate significantly from the training data distribution, under the assumption that model predictions on such samples are unreliable [21]. Yang et al. [27] provide a comprehensive survey of OOD detection methods. Cheng et al. [28] introduce neuron activation pattern (NAP) monitoring, where the activation patterns of the network are recorded during training to establish a reference

set of known patterns. During operation, each classification decision is reviewed by comparing the observed activation pattern with the stored reference values using the Hamming distance. Lee et al. [29] propose a framework that uses the Mahalanobis distance computed on intermediate feature representations of pretrained networks, enabling the detection of both out-of-distribution samples and adversarial attacks without requiring model retraining.

Adversarial detection addresses the vulnerability of DNNs to adversarial imperceptible input perturbations that cause erroneous model outputs during runtime. Adversarial detection methods can be grouped in metric, inconsistency and generative methods [21]. A generative method by Meng et al. [30] introduced an autoencoder-based detector that is trained on clean input data to learn compressed representations. During operation, input samples are passed through the autoencoder, which reconstructs the input from its latent representation. Adversarial inputs are then detected by measuring the reconstruction loss between the original and reconstructed input, where a high discrepancy indicates a potentially adversarial perturbation.

A further substantial body of research exists in the field of **explainable** AI, which investigates methods for making the internal decision-making processes of deep neural networks transparent and interpretable [31]. In the context of monitoring, Kuznietsov et al. [32] provide an overview of the visual explanation techniques that can be used to assess the reliability of perception or prediction at the time of inference. Saliency maps such as Grad-CAM use the gradients of all target features that feed into the final convolutional layer to generate a heatmap and highlight input pixels that contribute most to the models output [33]. This allows verification of whether the model focuses on semantically significant image regions [34].

In addition to pixel-level explanations, Heidemann et al. [10] investigate concept-based models in which a CNN generates binary concept predictions from the input image that indicate the presence or absence of predefined semantic concepts, and analyze how concept correlations affect the reliability of such concept-based explanations.

Another technical approach for monitoring is to perform rule-based **plausibility checks** during operation to identify potentially erroneous model outputs [14].

4 AI Monitoring in Context of ISO/PAS 8800

Based on safety standards discussed in section 2.2 and the taxonomy dimensions summarized in section 3, we propose an ISO/PAS 8800-related monitoring taxonomy for AI-based systems. In section 4.1, we present the combination of influencing factors, SRPs and monitoring methods from ISO/PAS 8800 with the SOTIF-related hazardous event model. Using this model as a basis, we then present our ISO/PAS 8800-related monitoring taxonomy in section 4.2, which we have developed by analyzing, combining, and expanding existing monitoring taxonomies. The benefits of this new taxonomy are then discussed in section 4.3.

4.1 Cause-effect Chain: From Influencing Factor to Hazardous Event

Figure 2 illustrates on the example of an AI-based perception system, a cause-and-effect chain combining, SOTIF, ISO/PAS 8800 and monitoring concepts.

A sensor converts physical characteristics of the environment into measurement data (e.g., raw image data), which serve as operational inputs to the AI system. The operation input is

passed to the AI model, which performs AI-based perception tasks such as classification, object detection, or segmentation. Different influencing factors can affect the training data observations and labels, the operation input, or the AI-model itself. This can result in the absence of safety-related properties, which could potentially lead to influenced outputs such as wrong labels or incorrectly placed bounding boxes. These erroneous outputs could propagate through the planning and acting parts, potentially resulting in hazardous behavior. In certain scenarios, hazardous behavior can lead to hazardous events.
In accordance with ISO/PAS 8800 Chapter 14 [2], monitoring measures need to be implemented during runtime to detect erroneous outputs and to mitigate the propagation of them in the hazardous event chain. As described in section 3.1.3, monitors with different architectures can be used to inspect and evaluate the input, activation, or output of the AI model, depending on its structure. The monitors then report different kinds of events related to the input, behavior or output or conduct an incident/accident analysis after a hazardous event has occurred (see section 3.1.4). The monitors operate at the interface between influencing factors and hazardous behavior, aiming to detect states in which the AI system's output is influenced.

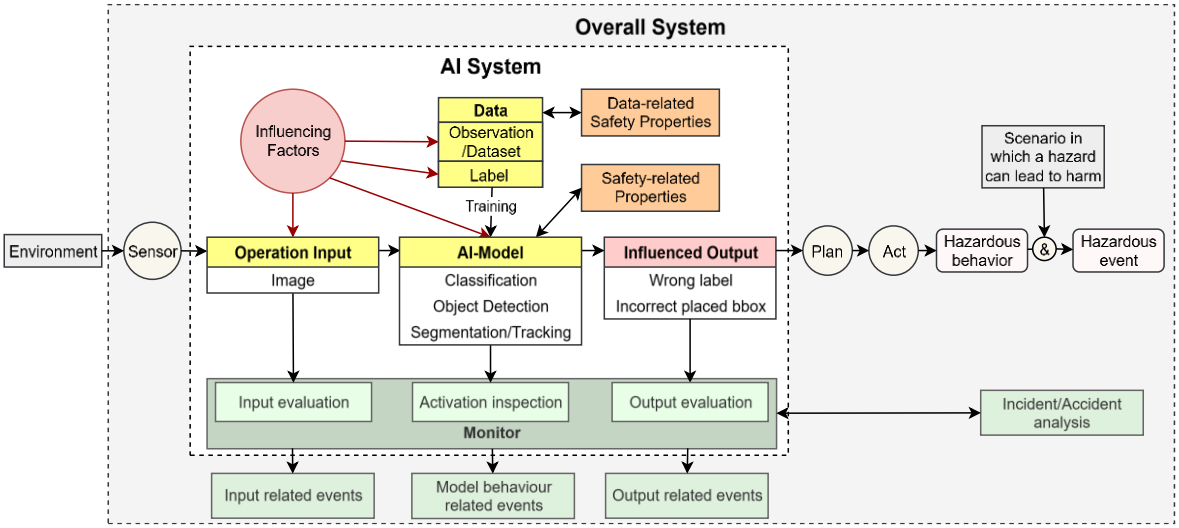

Figure 2: ISO/PAS 8800-related influencing factors in combination with SOTIF-related hazardous event model and AI monitoring

4.2 Monitor Taxonomy in the Context of ISO/PAS 8800

Based on Figure 2 above and the taxonomy dimensions discussed in section 3.1.1 through section 3.1.7, Figure 3 summarizes the identified classification criteria into a comprehensive ISO/PAS 8800-related taxonomy for AI-based perception monitors. The taxonomy is structured as a top-down hierarchy. At the top, the influencing factors from ISO/PAS 8800 (related to: operation input, model, observation, and label) define the root causes that monitors must address. Five different technical monitoring approaches can be used for this purpose: uncertainty monitoring, explainability monitoring, OOD monitoring, adversarial monitoring, and plausibility monitoring [19]. The taxonomy further classifies monitors along three input dimensions (input evaluation, activation inspection, and output evaluation [18]) and three architectural dimensions: (A-1) parallel or integrated deployment, (A-2) explicit or implicit knowledge acquisition, and (A-3) on-board or off-board execution. The

monitor output is differentiated between image-level assessments (e.g., Out of Operational Design Domain (ODD) detection or image perturbation detection) or object-level assessments (e.g., confidence scores for individual false positive or false negative detections). These monitoring outputs are related to input-, model behavior-, or output-events, which trigger one of three reaction types: control system switch, immediate prediction enhancement, or long-term system enhancement [4]. At the bottom of the taxonomy, each monitoring approach can be linked to the safety-related properties it addresses. ISO/PAS 8800 specifies a range of AI safety measures for the systematically addressing of SRPs throughout the AI lifecycle. A key aspect here is architecture [2]. By embedding different types of monitors, distinct SRPs can be addressed explicitly. Thus, the taxonomy provides a direct link to the safety argumentation framework of ISO/PAS 8800.

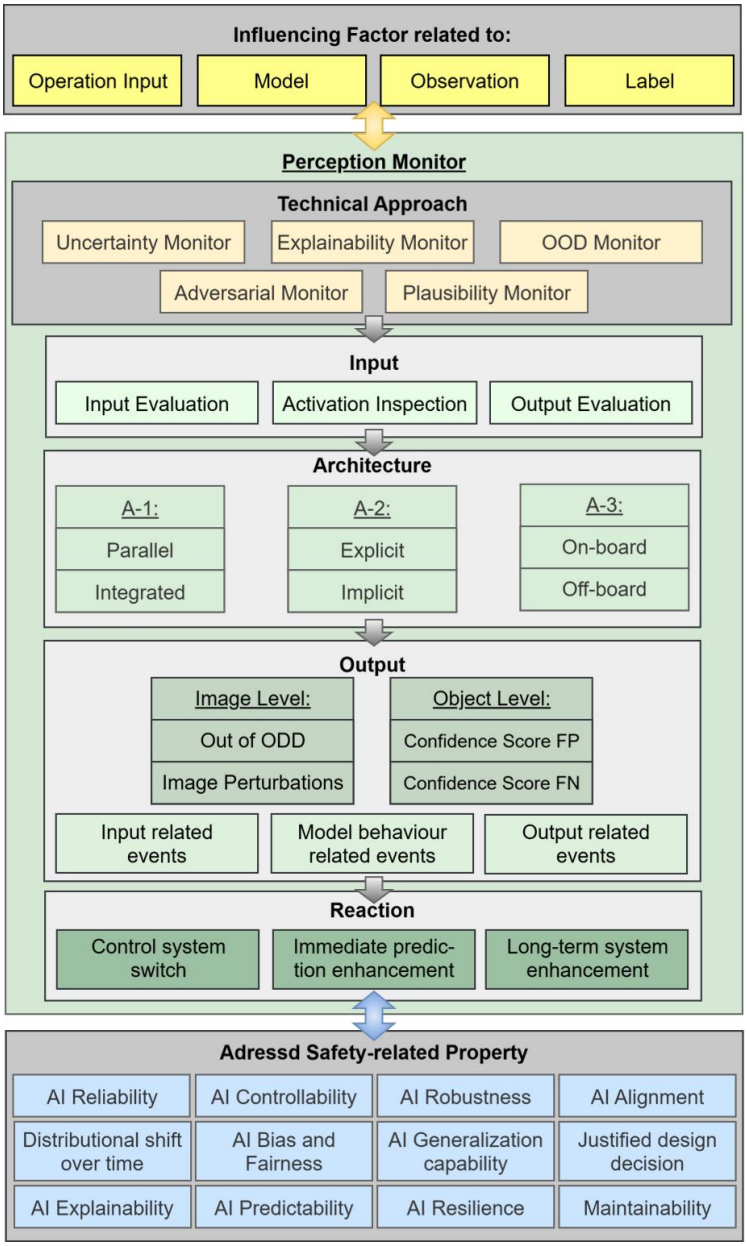

Figure 3: ISO/PAS 8800-related AI-based perception monitor taxonomy

4.3 Purpose of the Taxonomy for Safety

The proposed taxonomy enables the systematic classification of runtime monitoring approaches in safety-related contexts, which leads to several benefits. First, it enables a structured completeness argument by explicitly documenting which influencing factors and safety-related properties (SRPs) are covered and where gaps remain. Second, it improves bidirectional traceability: bottom-up, starting from identified SRPs, it documents which monitors contribute to their fulfillment; top-down, starting from identified influencing factors, it documents which monitors are deployed to detect them. Third, it simplifies the comparison and selection of monitors by providing a unified framework across fundamentally different technical approaches. Fourth, it supports the application of ISO/PAS 8800 by providing a transparent view of how relevant safety-related properties are covered by monitoring functions. Furthermore, the inclusion of diverse monitoring approaches can contribute to more robust monitoring architecture.

5 Conclusion and Outlook

AI-based automated driving functions introduce several safety-related considerations, including black-box characteristics, performance and robustness under varying operating conditions, and adherence to the intended scope of application. To address these safety-related considerations, ISO/PAS 8800 includes runtime monitoring of AI systems as one of several recommended measures. This paper provides an overview of various monitoring taxonomy approaches and the standardization landscape. Subsequently, a novel monitoring taxonomy for AI-based systems was presented, offering a structured classification framework aligned with ISO/PAS 8800. The taxonomy introduces an effect chain that links influencing factors through technical monitoring approaches, architectural integration patterns, monitoring results, reaction strategies, and safety-related properties. This structured representation is intended to support the systematic selection, comparison, and assessment of monitoring approaches in the context of safety architecture decisions and safety assurance activities.

In future work, we plan to apply this taxonomy by implementing various technical monitoring approaches to evaluate its practical benefits for safety argumentation.

6 Acknowledgement

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “Safe AI Engineering – Sicherheitsargumentation befähigendes AI Engineering über den gesamten Lebenszyklus einer KI-Funktion”. The authors would like to thank the consortium for the successful cooperation.

References

[1] X. Xie, „Artificial Intelligence-Driven Autonomous Vehicles: Current Developments and the Future Prospects“, *Comput. Artif. Intell.*, Bd. 2, Nr. 3, S. 75–78, Okt. 2025, doi: 10.70267/ajp73f40.

[2] *ISO/PAS 8800:2024: Road vehicles - Safety and artificial intelligence*, Edition 1, Dezember 2024. [Online]. Verfügbar unter: https://www.iso.org/standard/83303.html

[3] J. M. Faria, „Machine learning safety: An overview“, in *Proceedings of the 26th Safety-Critical Systems Symposium, York, UK*, 2018.
[4] R. Sena Ferreira, „Runtime safety monitoring of ML-based perception functions in autonomous systems“, Theses, Université Paul Sabatier - Toulouse III, 2023. [Online]. Verfügbar unter: https://theses.hal.science/tel-04342588
[5] M. Kläs und L. Jöckel, „A Framework for Building Uncertainty Wrappers for AI/ML-Based Data-Driven Components“, in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, Bd. 12235, A. Casimiro, F. Ortmeier, E. Schoitsch, F. Bitsch, und P. Ferreira, Hrsg., in Lecture Notes in Computer Science, vol. 12235. , Cham: Springer International Publishing, 2020, S. 315–327. doi: 10.1007/978-3-030-55583-2_23.
[6] S. Mohseni, M. Pitale, V. Singh, und Z. Wang, „Practical Solutions for Machine Learning Safety in Autonomous Vehicles“, 20. Dezember 2019, *arXiv*: arXiv:1912.09630. doi: 10.48550/arXiv.1912.09630.
[7] C. Bergenhem *u. a.*, „How to reach complete safety requirement refinement for autonomous vehicles“, in *CARS 2015-Critical Automotive applications: Robustness & Safety*, 2015.
[8] S. Burton, I. Habli, T. Lawton, J. McDermid, P. Morgan, und Z. Porter, „Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective“, *Artif. Intell.*, Bd. 279, S. 103201, Feb. 2020, doi: 10.1016/j.artint.2019.103201.
[9] S. Burton und B. Herd, „Addressing uncertainty in the safety assurance of machine-learning“, *Front. Comput. Sci.*, Bd. 5, S. 1132580, Apr. 2023, doi: 10.3389/fcomp.2023.1132580.
[10] L. Heidemann, M. Monnet, und K. Roscher, „Concept Correlation and Its Effects on Concept-Based Models“, in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA: IEEE, Jan. 2023, S. 4769–4777. doi: 10.1109/WACV56688.2023.00476.
[11] G. Schwalbe *u. a.*, „Structuring the Safety Argumentation for Deep Neural Network Based Perception in Automotive Applications“, in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, Bd. 12235, A. Casimiro, F. Ortmeier, E. Schoitsch, F. Bitsch, und P. Ferreira, Hrsg., in Lecture Notes in Computer Science, vol. 12235. , Cham: Springer International Publishing, 2020, S. 383–394. doi: 10.1007/978-3-030-55583-2_29.
[12] S. Mohseni, H. Wang, C. Xiao, Z. Yu, Z. Wang, und J. Yadawa, „Taxonomy of Machine Learning Safety: A Survey and Primer“, *ACM Comput. Surv.*, Bd. 55, Nr. 8, S. 1–38, Aug. 2023, doi: 10.1145/3551385.
[13] D. Marijan, A. Gotlieb, und M. Kumar Ahuja, „Challenges of Testing Machine Learning Based Systems“, in *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, Newark, CA, USA: IEEE, Apr. 2019, S. 101–102. doi: 10.1109/AITest.2019.00010.
[14] *ISO: Road Vehicles – Safety of the Intended Function ISO/PAS 21448:2022*.
[15] M. Leucker und C. Schallhart, „A brief account of runtime verification“, *J. Log. Algebr. Program.*, Bd. 78, Nr. 5, S. 293–303, Mai 2009, doi: 10.1016/j.jlap.2008.08.004.
[16] A. E. Goodloe und L. Pike, „Monitoring Distributed Real-Time Systems: A Survey and Future Directions“, NASA Center for AeroSpace Information, Juli 2010.

[17]Q. M. Rahman, P. Corke, und F. Dayoub, „Run-Time Monitoring of Machine Learning for Robotic Perception: A Survey of Emerging Trends“, *IEEE Access*, Bd. 9, S. 20067–20075, 2021, doi: 10.1109/ACCESS.2021.3055015.
[18]Q. M. Rahman, N. Sünderhauf, und F. Dayoub, „Online Monitoring of Object Detection Performance During Deployment“, 2020, *arXiv*. doi: 10.48550/ARXIV.2011.07750.
[19]L. Hacker und J. Seewig, „Insufficiency-Driven DNN Error Detection in the Context of SOTIF on Traffic Sign Recognition Use Case“, *IEEE Open J. Intell. Transp. Syst.*, Bd. 4, S. 58–70, 2023, doi: 10.1109/OJITS.2023.3236531.
[20]A. Lukina, C. Schilling, und T. A. Henzinger, „Into the Unknown: Active Monitoring of Neural Networks“, in *Runtime Verification*, Bd. 12974, L. Feng und D. Fisman, Hrsg., in Lecture Notes in Computer Science, vol. 12974. , Cham: Springer International Publishing, 2021, S. 42–61. doi: 10.1007/978-3-030-88494-9_3.
[21]J. Lust und A. P. Condurache, „A Survey on Assessing the Generalization Envelope of Deep Neural Networks: Predictive Uncertainty, Out-of-distribution and Adversarial Samples“, 6. September 2021, *arXiv*: arXiv:2008.09381. doi: 10.48550/arXiv.2008.09381.
[22]M. Henne, A. Schwaiger, K. Roscher, und G. Weiß, „Benchmarking Uncertainty Estimation Methods for Deep Learning with Safety-Related Metrics“, doi: 10.24406/PUBLICA-FHG-407174.
[23]F. Arnez, H. Espinoza, A. Radermacher, und F. Terrier, „A Comparison of Uncertainty Estimation Approaches in Deep Learning Components for Autonomous Vehicle Applications“, 2. Juli 2020, *arXiv*: arXiv:2006.15172. doi: 10.48550/arXiv.2006.15172.
[24]Y. Gal und Z. Ghahramani, „Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning“, 2015, *arXiv*. doi: 10.48550/ARXIV.1506.02142.
[25]B. Lakshminarayanan, A. Pritzel, und C. Blundell, „Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles“, 2016, *arXiv*. doi: 10.48550/ARXIV.1612.01474.
[26]T. DeVries und G. W. Taylor, „Learning Confidence for Out-of-Distribution Detection in Neural Networks“, 13. Februar 2018, *arXiv*: arXiv:1802.04865. doi: 10.48550/arXiv.1802.04865.
[27]J. Yang, K. Zhou, Y. Li, und Z. Liu, „Generalized Out-of-Distribution Detection: A Survey“, 23. Januar 2024, *arXiv*: arXiv:2110.11334. Zugegriffen: 13. Juni 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2110.11334
[28]C.-H. Cheng, G. Nührenberg, und H. Yasuoka, „Runtime Monitoring Neuron Activation Patterns“, 19. September 2018, *arXiv*: arXiv:1809.06573. Zugegriffen: 28. Mai 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1809.06573
[29]K. Lee, K. Lee, H. Lee, und J. Shin, „A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks“, 27. Oktober 2018, *arXiv*: arXiv:1807.03888. doi: 10.48550/arXiv.1807.03888.
[30]D. Meng und H. Chen, „MagNet: a Two-Pronged Defense against Adversarial Examples“, 11. September 2017, *arXiv*: arXiv:1705.09064. doi: 10.48550/arXiv.1705.09064.
[31]T. Speith, „A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods“, in *2022 ACM Conference on Fairness Accountability and Transparency*,

Seoul Republic of Korea: ACM, Juni 2022, S. 2239–2250. doi: 10.1145/3531146.3534639.
[32]A. Kuznietsov, B. Gyevnar, C. Wang, S. Peters, und S. V. Albrecht, „Explainable AI for Safe and Trustworthy Autonomous Driving: A Systematic Review“, 8. Februar 2024, *arXiv*: arXiv:2402.10086. Zugegriffen: 19. April 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2402.10086
[33]R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, und D. Batra, „Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization“, 2016, doi: 10.48550/ARXIV.1610.02391.
[34]J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, und B. Kim, „Sanity Checks for Saliency Maps“, in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, und R. Garnett, Hrsg., Curran Associates, Inc., 2018. [Online]. Verfügbar unter: https://proceedings.neurips.cc/paper_files/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf