# Run-time Safety Monitoring Framework for AI-based Systems: Automated Driving Cases

Mohd Hafeez Osman
*Department of Software Engineering & Information Systems*
*University Putra Malaysia*
Serdang, Malaysia
*Department of Computer Science*
*Technical University of Munich*
Munich, Germany
hafeez.osman@upm.edu.my

Stefan Kugele, Sina Shafaei
*Department of Computer Science*
*Technical University of Munich*
Munich, Germany
stefan.kugele@tum.de, sina.shafaei@tum.de

***Abstract*—Intelligent systems based on artificial intelligence techniques are increasing and are recently being accepted in the automotive domain. In the competition of automobile makers to provide fully automated vehicles, it is perceived that artificial intelligence will profoundly influence the automotive electric and electronic architecture in the future. However, while such systems provide highly advanced functions, safety risk increases as AI-based systems may produce uncertain output and behaviour. In this paper, we devise a run-time safety monitoring framework for AI-based intelligence systems focusing on autonomous driving functions. In detail, this paper describes (i) the characteristics of a safety monitoring framework; (ii) the safety monitoring framework itself, and (iii) we develop a prototype and implement the framework for two critical driving functions: *Lane detection* and *object detection*. Through an implementation of the framework to a prototypic control environment, we show the possibility of this framework in the real context. Finally, we discuss the techniques used in developing the safety monitoring framework and describes the encountered challenges.**

***Index Terms*—System Safety, Run-time Safety Monitoring, Autonomous Driving, Artificial Intelligence**

## I. Introduction

*Context:* During the last decades, a lot of software-controlled functions running on a large number of electronic control units were included in modern car electric/electronic (E/E) architectures. Driving forces are (i) safety requirements, (ii) customer demand for more comfort and the newest infotainment systems, and (iii) advanced driver assistance systems allowing to reach hitherto unrivalled levels of driving automation (cf. [1]).

Machine intelligence is an emerging and cross-disciplinary application area which is advancing with great strides also in in-vehicle functions. Examples are computer vision for obstacle detection, traffic sign recognition, driver's emotion detection, and of course, *automated driving*. Besides typically non-safety-critical well-being functions such as the HVAC (heating, ventilation, air conditioning) system, more and more functions that are at least partially (or even fully) based on machine learning algorithms are being embedded into modern vehicles to either (i) assist (current or future Advanced driver-assistance system (ADAS) functions) or (ii) take over control (future *digital chauffeur*) by performing a complex series of perception and control tasks in both normal and emergency situations.

From a safety point of view, those functions are particularly challenging for OEMs and their supplier network. Safety refers "to the extent to which such a system is free of hazards, that is, of risks of physical harm for humans, the environment, the system itself, and other usually physical assets." [2]. To ensure functional safety, companies apply comprehensive quality and safety assurance activities demanded by regulatory authorities and pertinent standard such as the ISO 26262 [3].

*Problem Statement:* For higher levels of driving automation, e.g., levels 3-5 according to [1], perception functions are based on machine learning algorithms such as neural networks (NN). These inductive learning approaches require a huge training set in order to—in the best case—generalise to all possible *operational situations* such as road, traffic, and weather conditions. The generalisation is the primary driving force to use neural networks since in practice it is infeasible to state *all requirements* (e.g. functional, safety, etc.) a highly automated driving system needs to fulfil. However, commonly used development processes are aligned along the V-model, which is the reference model in ISO 26262. The V-model assumes that all requirements are known and thus *complete*, *correct*, and *unambiguous*, which is in contrast to the motivation to use generalising NNs. The activities within the safety lifecycle include the identification and assessment of hazards, derivation of safety requirements, and management and tracking them to guarantee that risk diminishment is acknowledged in the final product.

Following the highly recommended methods for "Software unit design and implementation" (ISO 26262, Part 6, §8), and for instance software unit test (cf. §9.4.2), it becomes evident that the standard is highly source code-oriented and that it is tested against given requirements. Unfortunately, it does not accommodate new software technologies such as machine learning and thus does not provide any guidance. As we see, assurance activities are performed during *development-time*,

2640-0715/19/$31.00 ©2019 IEEE
DOI 10.1109/APSEC48747.2019.00066

which was perfectly fine for well-defined functions so far. However, for a function that is built using machine learning, the output is highly dynamic.

*Goal:* Therefore, we propose to combine traditional and well-established safety assurance processes with activities performed after sales during system operation. We introduce the *Safety Monitoring Framework* (SMF) allowing to detect *safety violations* and *anomalies* at run-time. Moreover, any safety-related findings are fed back to the OEM, facilitating roundtrip engineering and improving the fleet's safety and new products.

*Contribution:* Elaborating on previous work in [4], we contribute:

(i) a *Safety Monitoring Framework* to monitor AI-based functions, and
(ii) a prototypic implementation in a controlled environment and discuss our experience and challenges.

*Outline:* The remainder of this paper is structured as follows. Section II summarises background and related work. This is followed by the overview of the proposed *Safety Monitoring Framework* in Section III and its detail explanation in Section IV. Section V describes the prototype and example cases. The paper concludes with a discussion and conclusion in Sections VI and VII.

## II. Background & Related Work

In this section, we explain the anomaly detection approach and sketch the related work.

### A. Anomaly Detection

Vehicle functions need to be reliable enough to ensure the safety of the passengers, as well as the entities in the surrounding environment. This certainly requires high accuracy and data availability from all sensors along with the functions of the car to behave appropriately in all situations, considering the traffic rules, the comfort of the passengers and driving ethics. To tackle this issue, anomaly detection is one of the promising approaches that are employed in car functions and operations to improve and maintain safety. Anomalies are generally classified into three types of *"point"* anomalies in which anomalous data points can be described individually, *"contextual"* in which the anomalous data points are observed in some defined context and *"collective"* anomalies which applies for a collection of anomalous data points. Offline anomaly detectors work on a static dataset, with the assumption that it contains enough information to learn the required characteristics of anomalous data points, whereas online anomaly detectors process data incrementally while making predictions and learning from each new data point [5].

Anomaly detection in vehicles contains a wide range of applications from sensory data and detecting the faulty data source, to complex functions of motion planning, driver drowsiness or object detection. Most of the known algorithms in object detection perform feature extraction on the images. Classical approaches like HOG [6], SURF [7] or deep neural networks [8] can be used for this purpose. The easiest way to detect anomaly for point-based anomalies is to calculate the distance between each sample and in case of having multidimensional data calculating Mahalanobis distance [9] will be more effective than just Euclidean. Another effective approach in facing multidimensional data is to employ prediction based approaches that make it possible to learn from the data based on the previous predictions. The application of anomaly itself does not need to be necessarily online but can be used in an online learning context.

In [10], Richter and Roy learn a visual collision prediction model online by self-generating data from visual SLAM module. Anomaly detection was used to classify an image as novel or not-novel during operation. If an image is classified as novel, the prediction of the collision prediction model will be considered as "unreliable." In that case, the system reverts back to a safe prior behaviour. Data is generated both offline and online using a SLAM system, which gives a geometric map of the environment. Labels are generated by taking random actions from the recorded configurations in the geometric map and evaluating the outcome.

A neural network is trained on the image and action data, which outputs "collision" or "non-collision." This forms the collision prediction model. It is improved continuously during operation. To detect novelty in the images, an autoencoder is used to learn a compact representation of images from a training dataset. The assumption here is that if a novel image is encountered, the reconstruction cost for the autoencoder will be high as that image is unlikely to be in the training dataset and there will be only needed to threshold the reconstruction error to classify novel images.

### B. Related Work

From our best knowledge, researchers and practitioner still finding the best solution to address the safety concern of Automated Driving Vehicles. As of now, several studies have outlined the safety challenges and proposed several recommended practice.

Koopman and Wagner [11] identified five major challenges from the testing perspective (based on ISO 26262 V-Model) for autonomous vehicles. Driver out of the loop, complex requirements, non-deterministic algorithms, inductive learning algorithms and, fail-operational systems are the challenges that they identified and discussed. By assuming the V-model as the software development process, they proposed phased development, monitor/actuator architecture and fault injection as a promising solution for testing autonomous vehicle. Also in [12], Koopman and Wagner proposed several approaches to enhance Highly Automated Vehicles (HAV) validation efficiency, improve effectiveness, and towards more justifiable safety argument.

On the other hand, Salay et al. [13] analysed the impacts that the use of Machine Learning (ML) as an implementation

approach has on ISO 26262 safety lifecycle. They identified five safety-related issues on implementing ML-based approach in the automotive domain (i.e., identifying hazards, fault and failure modes, usage of training sets, the level of ML usage, and required software techniques). They proposed the recommendation to address the identified issues from the perspective of safety standard (such as: extending the definition of hazard, different safety requirement of ML systems) and outlined future research.

Shalev-Shwartz et al. [14] introduce "Responsibility Sensitive Safety" (RSS) – a formal model based on the interpretation of "Duty of Care" defined in Tort Law that may lead to a driving policy for self-driving cars. Simon et al. [15] indicated that the primary challenge of utilising ML approach for HAV is (i) arguing a satisfactorily low level of residual risk related to functional insufficiently as the result of imperfections of the employed machine learning algorithms, and (ii) the difficulty of arguing testing all possible driving scenario at design time. They explored the possibility of assurance case approaches to address the problem of arguing the safety of machine learning from the scope of HAV.

Meanwhile, Monkhouse et al. [16] investigate the gap between the automotive industry goal on HAV and its safety assurance. An empirical study was conducted to explore the safety concerns that the safety engineers have around assured the safety of HAV. They highlighted the need for new risk and safety concept, interactive complexity and changes to legal and standards landscape as a high-level direction to address some issues on the safety of HAV.

Regarding safety and monitoring approaches, Machin et al. [17] proposed Safety Monitoring Framework (SMOF) to generate safety rules based on the safety margin concept. Formal verification techniques are used to synthesise the safety rule during the hazard analysis stage.

Motivated by the increasing complexity of the E/E systems, and to enhance the flexibility of E/E systems, Jalena [18] proposed a model-based approach for generating safety configuration and fault-management layer that facilitates error detection and fault-handling mechanisms to preserves the safety properties of the system at run-time.

Jaradat and Punnekkat [19] proposed a technique to detect inconsistency failure rate between run-time and design-time. They recommended the use of a safety contract (derived from Fault Tree Analysis (FTA)) to monitor the failure rate during run-time. The safety monitor provides essential feedback on safety confidence. Meanwhile, Kane et al. [20] develop an online monitoring approach that monitors an autonomous research vehicle. They built a run-time monitoring algorithm that checks the state trace of violations.

Most of the works mentioned above endeavour to come out with a safety monitoring mechanism that is based on the safety rule derived from hazard analysis activity; motivated by either to cope with the complexity of systems or to get the run-time safety information for violation analysis. This work proposes a Safety Monitoring Framework (SMF) that aim at providing a mechanism to monitor the safety of AI-based systems. The framework is built based on the Safety Executive Pattern or Safety Kernel Pattern [21] that is as a suitable safety pattern for complex and highly safety-critical systems [22].

## III. Framework Overview

This section describes the characteristics of the SMF and the overview of the SMF.

### A. Characteristics

Since AI-based software is always be seen as a black-box system and also motivated by the automated driving cases presented in [4] and safety challenges in [23], [24], we perceived that the SMF should incorporate the following criteria:

(i) monitoring and control the input and output,
(ii) alert safety violations,
(iii) dynamic safety threshold/limiters, and
(iv) dynamic logging mechanism,

From the safety tactics perspective, we observed that the following safety tactics (based on [25]) are essential for AI-based systems:

(i) *Failure Detection* (Condition Monitoring) and
(ii) *Failure Containment* (Redundancy – Functional Redundancy, Recovery – Degradation).

Hence, we incorporate these safety tactics in our framework.

### B. Overall View

The SMF is endeavour to increase the controllability and to address the uncertainty of AI-based systems. It is partly built based on the Safety Executive Pattern [22], [23]. The SMF expands the Safety Executive Pattern to include the following:

(i) safety analysis mechanism,
(ii) safety profile (consists of a safety contract and safety measure [22]),
(iii) centralise in-vehicle safety control, and
(iv) use anomaly detection technique for safety monitor and control.

Even though the framework emphasis on safety in the run-time phase, the safety information needs to be collected from the design phase (i.e., Concept Phase and Development Phase in [3]). Fig. 1 shows the overall view of the SMF that explains the relationship between *Design-Time* and *Run-time* stage.

## IV. The Safety Monitoring Framework

In this section, we describes the *Design-Time* and *Run-Time* activity in more detail.

### A. Design-Time

During this stage, the common hazard analysis technique such as Fault Tree Analysis (FTA), Failure Mode Effect Analysis (FMEA), Hazard and operability study (HAZOP), Systematic Theoretic Process Analysis (STPA) or other hazard analysis techniques must be performed to determine (i) the risk level (ASIL) and (ii) the hazard mitigation process and procedure.

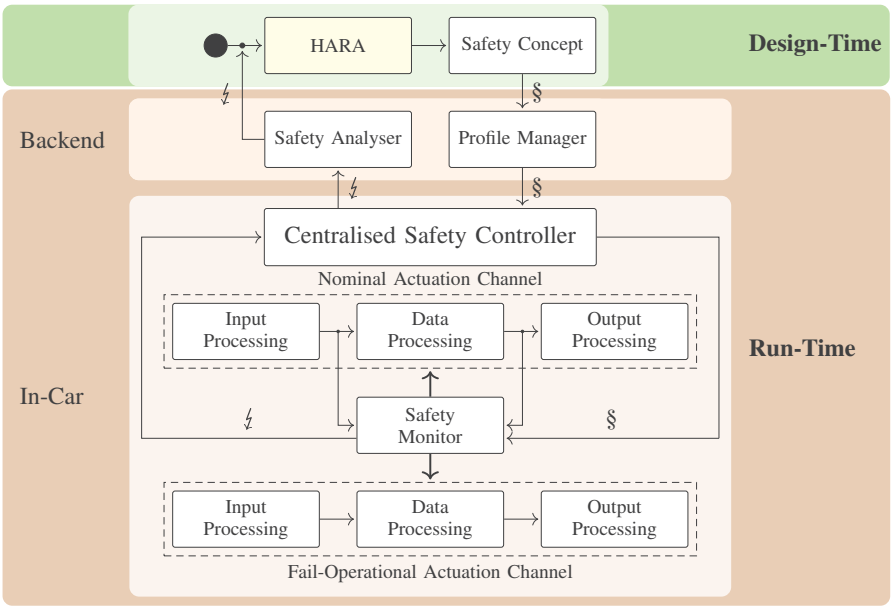



Fig. 1: Safety Monitoring Framework using the Safety Executive Pattern [21].

The hazard analysis techniques mentioned above are normally performed for traditional systems where the output or system behaviour are predictable. However, for AI-based systems (especially data-driven ML-based systems), the system's behaviour is not easy to predict. This type of system is a common data-dependent and the process of producing the output is not transparent or sometimes unexplainable. Hence, in this work, we propose pattern recognition of the following items should be explored (as a complement to hazard analysis of standard/regular automotive system):

(1) *the training and testing data*: If the training and testing data of the ML-based system is provided and accessible, the data exploration should be accompanied by a suitable anomaly detection model. The purpose of developing the anomaly detection model is to provide a mechanism to exclude data input that is not nominal. Filtering the anomaly input will reduce the uncertainty risk of ML-based system output. Prior to the anomaly detection model development, the feature and the anomaly detection algorithm should be selected based on the system context and characteristics. For example, an object detection system consumes a lot of resources and may require a longer response time. Adding anomaly detection may decrease the system's performance. In this case, we propose to use the minimal feature and a simple anomaly detection algorithm (e.g. Simple statistical methods).

(2) *the AI/ML-based systems input and output:* In case the training and testing data for ML-based system are provided (e.g., System as Commercial off the shelf (COTS)), the anomaly detection model needs to be developed by simulating the input and output on the nominal environment. The anomaly detection model will be built by using this nominal input and output data. By using this model, the abnormal input and output can be detected and excluded.

For each anomaly detection model, at least one fail-over operation should be assigned to it. The fail-over operation should be identified from the hazard analysis and risk assessment (HARA) activity. However, in this paper, we do not discuss the fail-over operation in detail.

Next, the Functional Safety Concept defines the safety mechanism (e.g., failure mitigation, safe state transition) for the system. The information gathered from the HARA activity and the Function Safety Concept allows the creation of a *Safety Profile* (§). The *Safety Profile* is a safety configuration of each AI-based software. It covers the safety contract and safety measure [22] and consists of the following information:

(i) system description
(ii) system ID,
(iii) safety limiters,
(iv) anomaly detection function,
(v) safety violation identifier,
(vi) fail-over operation, and
(vii) safety log configuration (indicate the list of information that shall be logged for safety purposes (e.g., forensic, accountability)).

### B. Run-time.

The run-time monitoring framework is divided into two main components, i.e., Safety Backend and In-Car Safety Monitoring.

(1) **Safety Backend (SBK)**. The SBK's primary functions are: (i) managing the safety profile, (ii) analysing logs (ϟ), and (iii) alert violations. After a safety profile is created

for an AI-based system (during *design-time*), the *profile manager* registers the safety profile in the Safety Backend system. The *profile manager* performs the configuration management of safety profiles which controls the deletion, modification, and installation of a new profile. Once an AI-based system is running, the *Safety Analyser* will collect the log from the vehicle for the safety analysis. It is advisable that the HARA is conducted periodically to determine the level of security of the system. Therefore, the collected logs can be a valuable input to Hazard Analysis.

(2) **In-Car Safety Management (ICSM)**. The main functionality of ICSM is to manage the (In-Car) safety of the AI-based systems centrally. It consists of two main sub-components:
*Centralised Safety Controller* (CSC): The CSC communicates with the SBK to get updated safety profiles. Once CSC has received an updated safety profile, it will store it locally and forward it to the Safety Monitor. The CSC will periodically check for updates. In terms of safety log, CSC collects the safety logs (from the *Safety Monitor*), stores it internally (temporary) and sends the logs to the SBK component.
*Safety Monitor* (SM): The main functionality of the SM is to (i) hold the current anomaly detection function and safety limiters value, (ii) detect the safety violation, (iii) activate the fail-over operation, and (iv) send safety violation log to CSC. During the AI-based system operation, the SM
- benchmarks the input and the output from the input and output limiters, and
- detects the anomaly of input and output using anomaly detection technique.

If a safety violation occurs (abnormal input and output), the SM will send an instruction to move from *Nominal Actuation Channel* to *Fail-Operational Actuation Channel*. Then, a violation alert will be logged and transmitted to the CSC.

## V. Prototype and Example Cases

This section describes the prototypic implementation and the (selected) example cases.

### A. Prototypic Implementation

In this paper, we intend to propose a pragmatic solution for the safety of AI-based system (focusing on the autonomous driving domain). Thus, based on the SMF that has been described in the previous section, we develop a prototype purposely (i) to present how the framework can be implemented, and (ii) to observe the challenges of implementing it.

We conduct a (master-level) practical course to implement the SMF. Four groups (consist of 2 members per group) has been assigned to develop the SMF. The students were given a detailed requirement to develop the SMF. As part of the requirement, the autonomous car prototype should be built

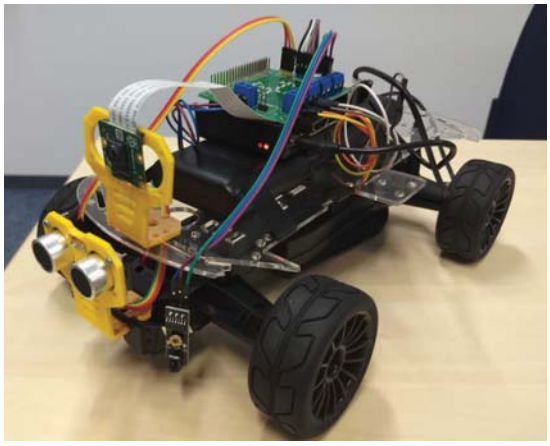


Fig. 2: Autonomous car prototype

using the following setup (an example prototype car is shown in Figure 1):

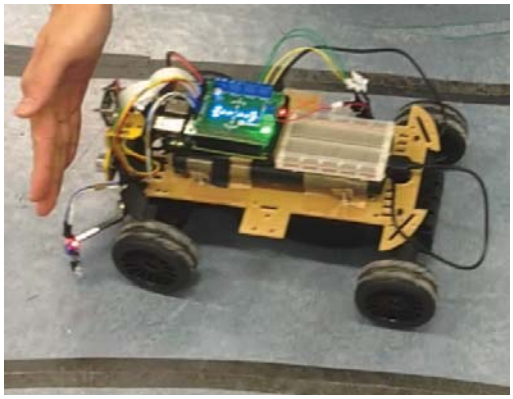
(i) Remote control car,
(ii) Motor shield,
(iii) Raspberry Pi 3,
(iv) Raspberry Pi Camera,
(v) Ultrasonic Sensor, and
(vi) Infrared Sensor.

The students are free to use any development environment and they are recommended to use Tensorflow object detection API [26] for object detection solution.

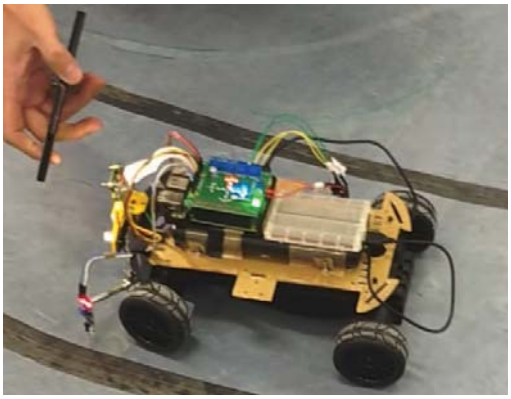
### B. Example Cases

In this paper, we selected two systems that are essential in the autonomous driving functionality as our example cases i.e. Lane detection and Obstacle detection. These systems are currently the core functions to support the Advanced Driver Assistance Systems and in future will be the critical systems of highly automated or fully autonomous driving systems.

1) *Lane Detection:* The lane detection system is a component which identifies the driving path limits and gauges the geometry position of the lane (our case, in 2D), concerning the vehicle position. In this work, we use two different lane detection systems that are (i) lane detection based on camera images, and (ii) lane detection based on infrared sensors. We use camera images as the primary lane detection while Infrared sensor-based lane detection is used to simulate the backup function.
2) *Obstacle Detection:* An obstacle detection system recognises obstacles in front of the vehicle and finds the dynamic relativity between the vehicle and each of the obstacles. We use two obstacle detection systems that are: (i) obstacle detection based on camera images and, (ii) obstacle detection based on the ultrasonic sensor. Obstacle detection based on camera is used as the primary obstacle detection system while another system act as the backup function.



(a) Blocking the camera vision



(b) Glare effect on camera

Fig. 3: Example on safety-related scenario

Fig. 4: Demonstration Lane

Since the primary input of both systems are from the camera, the students were asked to develop a safety profile that can mitigate or tolerate the anomalous images. To simulate this safety-related scenario, we asked the students to develop the safety profile for the following scenario:

(i) Camera blocked by an object (see Figure 3a)

(ii) Corrupted Image (e.g. glare effect on the camera or an extensive amount of light (see Figure 3b)

The students demonstrate the working prototype using the test circuit as shown in Figure 4. Based on the prototype development, we assessed the challenge of implementing this safety monitoring framework. The major challenges are discussed in the next section.

## VI. Discussion

In this section, we summarise the challenges and experience after implementing the SMP to a prototype and discuss the threats to the validity of this work.

### A. Challenges

After implementing this framework to two example cases, we outline the following practical challenges and possible solutions:

**Challenge 1: Anomaly detection feature set.** In machine learning, a feature is a quantifiable property or characteristics of the observed event. A high number of features for anomaly detection will generally increase the processing time and use more resources. In the automotive domain, resources and performance are limited. *Possible Solution* – A machine learning feature selection should be conducted to find the most influential features of the AI model. This information allows the selection of suitable features for the anomaly detection model. Reducing the features may decrease the coverage of anomaly detection; however, there should be a trade-off between performance, resource, and safety. The usage of prominent features in the anomaly detection model may provide the model that can only focus on critical situations.

**Challenge 2: Multiple input and output.** The detailed information (e.g., algorithms, data) of the AI model is not provided in most cases. For example, we use the object detection model that is provided by TensorFlow for the

obstacle detection system. We were not able to access the detailed information on how the object detection model was constructed as well as the training data. Hence, we decided to develop an anomaly detection model based on the input of the system (e.g., capture images) and the output of the system (e.g., processing time and quality of detection). Developing the anomaly detection model for every input and output may decrease system performance. Most of the inputs and outputs are critical that need to be monitored for safety purposes. *Possible solution* – Many algorithms can be used to develop the anomaly detection model. To reduce the performance degradation risk, we propose to use the simple anomaly detection technique such as statistics outlier detection or lightweight machine learning algorithms (e.g., $k$-means). The amount of safety mechanism to be included in AI-based systems should be suitable for the ASIL of the function.

**Challenge 3: Safety Limiters and Anomaly Detection limit the self-adaptive capability of AI-based system.** Self-adaptive is one of the key benefits of using AI techniques. Self-adaptive is the capability of the system to reconfigure itself in response to changes in its environment automatically—however, employing safety limiters and anomaly detection mechanism prevent the system from learning new data. *Possible solution* – The AI model should not be static. From our Safety Monitoring Framework, new data can be derived from the violation logs. Not all data in violation logs are safety-related data. It is merely an "outlier" data that is excluded to prevent abnormal input processing and to avoid unusual output. This information can be used as the input to update the AI model.

### B. Threats to Validity

This section presents the threat to internal validity and external validity.

**Internal threats to validity.** The prototype setup and the selected example cases may not represent the real situation of the AI-based system in the automotive domain. However, this setup is suitable as early evaluation of the proposed SMF since evaluation in a real environment is expensive and risky.

**External threats to validity.** We suggested the use of anomaly detection and safety limiters as the principal method to ensure the safety of AI-based systems during run-time. However, the diversity of AI-based systems prevents us from claiming that this method can be generalised or apply to all AI-based system. Nevertheless, we perceived that this method is beneficial for the AI-based system that provides detailed information about the AI-model construction such as algorithms and used training data.

## VII. Conclusion and Future Work

In this paper, we presented a Run-time Safety Monitoring Framework to address the uncertainty issue of AI-based systems for Autonomous Driving Functions. We used anomaly detection techniques to prevent the abnormal input and the undesired output of an AI-based system. The anomaly is not equal to noise, but rather a specific pattern in data that deviates from the other observations. Unsupervised can be useful tools to deal with this problem, which we employed in the proposed framework. By implementing the framework to a prototype in a controlled environment, we showed that in principle, the proposed Run-Time Safety Monitoring Framework is a suitable mechanism to address the uncertainty issues for AI-based systems such as in Autonomous Driving domain. Furthermore, we outline several practical challenges in implementing this framework.

Our primary goal is to come out with a possible solution to address the uncertainty issue of AI-based systems. This research is conducted in an iterative manner where we develop and test the current solution to a prototype. For the next iteration, we would like to explore the usage of online anomaly detection technique to provide a more dynamic safety approach to the framework. We also would like to expand this framework to incorporate the creation of safety profile during the design-time phase.

## Acknowledgment

This work was supported by a German automotive industry. We thank our project partners for putting our research into an innovative practical context.

## References

[1] SAE Standard J3016, *Taxonomy and Definitions for Terms Related to On-road Motor Vehicle Automated Driving Systems, SAE Standard J3016*, 2014.

[2] M. Gleirscher and S. Kugele, "Assurance of System Safety: A Survey of Design and Argument Patterns," *CoRR*, vol. abs/1802.08327, 2019. [Online]. Available: http://arxiv.org/abs/1902.05537

[3] ISO/IEC 26262, "ISO 26262:2011 Road vehicles – Functional safety," 2011.

[4] S. Shafaei, S. Kugele, M. H. Osman, and A. Knoll, "Uncertainty in machine learning: A safety perspective on autonomous driving," in *SAFECOMP 2018 Workshop Proceedings*, ser. LNCS, vol. 11094. Springer, 2018, pp. 458–464.

[5] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on computer vision & Pattern Recognition (CVPR'05)*, vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[8] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, 2013, pp. 2553–2561.

[9] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[10] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems XIII, MIT*. Robotics: Science and Systems Foundation, 2017.

[11] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

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

[13] R. Salay, R. Queiroz, and K. Czarnecki, "An analysis of iso 26262: Using machine learning safely in automotive software," *arXiv preprint arXiv:1709.02435*, 2017.

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

[15] S. Burton, L. Gauerhof, and C. Heinzemann, "Making the case for safety of machine learning in highly automated driving," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2017, pp. 5–16.

[16] H. Monkhouse, I. Habli, J. McDermid, S. Khastgir, and G. Dhadyalla, "Why functional safety experts worry about automotive systems having increasing autonomy," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2017, pp. 1–6.

[17] M. Machin, J. Guiochet, H. Waeselynck, J.-P. Blanquart, M. Roy, and L. Masson, "Smof: A safety monitoring framework for autonomous systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 702–715, 2018.

[18] J. Frtunikj, "Safety framework and platform for functions of future automotive e/e systems," *Automotive and Engine Technology*, vol. 1, no. 1-4, pp. 93–105, 2016.

[19] O. Jaradat and S. Punnekkat, "Using safety contracts to verify design assumptions during runtime," in *23rd International Conference on Reliable Software Technologies - Ada-Europe 2018*, vol. 10873, June 2018. [Online]. Available: http://www.es.mdh.se/publications/5058-

[20] A. Kane, O. Chowdhury, A. Datta, and P. Koopman, "A case study on runtime monitoring of an autonomous research vehicle (arv) system," in *Runtime Verification*. Springer, 2015, pp. 102–117.

[21] B. P. Douglass, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[22] A. Armoush, "Design patterns for safety-critical embedded systems." Ph.D. dissertation, RWTH Aachen University, 2010.

[23] M. Borg, C. Englund, K. Wnuk, B. Duran, C. Levandowski, S. Gao, Y. Tan, H. Kaijser, H. Lönn, and J. Törnqvist, "Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry," *CoRR*, vol. abs/1812.05389, 2018. [Online]. Available: http://arxiv.org/abs/1812.05389

[24] S. Nair, S. Shafaei, S. Kugele, M. H. Osman, and A. Knoll, "Monitoring safety of autonomous vehicles with crash prediction networks," in *Workshop on Artificial Intelligence Safety 2019*, ser. CEUR Workshop Proceedings, vol. 2301. CEUR-WS.org, 2019. [Online]. Available: http://ceur-ws.org/Vol-2301/paper_12.pdf

[25] W. Wu and T. Kelly, "Safety tactics for software architecture design," in *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.* IEEE, 2004, pp. 368–375.

[26] "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/