Safety Assurance of Machine Learning for Chassis Control Functions

Simon Burton[1], Iwo Kurzidem[1], Adrian Schwaiger[1], Philipp Schleiss[1], Michael Unterreiner[2], Torben Graeber[2], and Philipp Becker[2]

[1] Fraunhofer IKS, 80686 Munich, Germany
{firstname.lastname}@iks.fraunhofer.de

[2] Porsche AG, 71287 Weissach, Germany
{firstname.lastname}@porsche.de

Abstract. This paper describes the application of machine learning techniques and an associated assurance case for a safety-relevant chassis control system. The method applied during the assurance process is described including the sources of evidence and deviations from previous ISO 26262 based approaches. The paper highlights how the choice of machine learning approach supports the assurance case, especially regarding the inherent explainability of the algorithm and its robustness to minor input changes. In addition, the challenges that arise if applying more complex machine learning technique, for example in the domain of automated driving, are also discussed. The main contribution of the paper is the demonstration of an assurance approach for machine learning for a comparatively simple function. This allowed the authors to develop a convincing assurance case, whilst identifying pragmatic considerations in the application of machine learning for safety-relevant functions.

Keywords: Assurance Case · Safety Engineering · Machine Learning · Automotive Software.

1 Introduction

Recent advances in Machine Learning (ML) have demonstrated the potential for efficient and sophisticated classifications based on data-driven models [16]. This is especially visible in domains where conventional programming is difficult and computationally expensive. However with the increased application of ML techniques to safety-related tasks, concerns related to the probability of incorrect or inaccurate predictions have also increased. Current safety-related challenges in ML include, but are not limited to: explainability of decision-making, unreliable confidence information, inadequate approximations via limited data-sets, insufficient or incomplete definitions, and meaningful safety metrics [15]. These functional insufficiencies and safety concerns are especially important for ML in automated driving applications, as they may potentially impact the overall vehicle's safety goals [3]. As such, industry safety standards, such as, ISO 26262

(Road vehicles Functional safety) [6] and ISO/PAS 21448 (Safety of the Intended Functionality - SOTIF) [7] apply. However, while these ISO standards tackle operational safety and offer guidance for safety analyses, neither standard offers a complete and coherent safety assurance approach suited for ML [1]. These shortcomings are not just limited to automated driving functions alone, guidance in the standards is also lacking when applying ML to other classes of vehicle functions, such as powertrain and chassis control. Such functions can directly impact the stability of the vehicle and therefore contribute to vehicle-level safety goals. Hence, a comprehensive and tailored safety assurance is vital, before deployment of ML-based systems, to guarantee safety.

This paper demonstrates such an assurance approach for a road surface estimation based on sound patterns. Within the application, acoustic sensors are used to categorise road conditions between the classes of dry (*dry*) and not dry (*!dry*). This information is then used to adapt chassis control functions to the road surface traction. A misclassification of the surface condition could therefore lead to a hazardous control action.

The paper is organised as follows: first, an overview of related publications and ideas is given in Section 2, followed by a description of the case study in Section 3 and an outline of the proposed approach in Section 4. In Section 5 properties of the chosen ML technique are analysed with respect to their strengths and weaknesses for assuring safety. The insights of the analysis are discussed and summarised as lessons learned in Section 6, leading to a conclusion in Section 7.

2 Related Work

Safety standards already exist for automotive functionality. ISO 26262 focuses on functional safety and provides comprehensive guidelines for the analysis of conventional software and hardware failures and ISO/PAS 21448 addresses insufficiencies and potential exploits for (conventional) software and ML, such as performance limitations, impact from the environment and foreseeable misuse by third parties. However, both standards do not offer a general strategy or approach for validating safety of non-conventional SW, such as ML algorithms.

In [12] both ISO standards are combined to create a product development process for ML. The authors incorporate ISO/PAS 21448 into ISO 26262 work products and development phases. The proposed approach is heavily based on ISO 26262 definitions, such as the chapter enumeration, and workflow, e.g. V-model. The ML specific work products are handled as additional documentation within each development phase. However, open questions regarding applicability for complex systems, *semantic gaps* (cf. Section 4) and meaningful evidence acquisition still remain. Additionally, no examples or methods are given on how to generate these additional documents, as no case study is presented.

A different take on this matter is introduced by Picardi et al. in [11], with *argument patterns* to demonstrate the safety of ML. The presented safety assurance patterns are tailored for ML components and highlight *how* the collected evidence, assumptions, strategies and claims relate to overall system safety goals.

The patterns are graphically represented using GSN [14] and show how performance evidence is (indirectly) connected to specific ML safety requirements. The result is to coherently and unambiguously represent a compelling safety argumentation. Additionally, the authors show how the argumentation patterns can be applied within different stages and activities of a complete ML assurance process. In a further work, Picardi et al. utilises the argument patterns to develop an assurance case specifically for a ML component within medical diagnostics [10].

Outside the automotive domain, audio interpretation via ML, for instance in form of speech recognition, achieves impressive performance. However, most considerations, analyses and evaluations of ML actually exclude safety as a major desideratum [2].

In this paper, we apply similar argument patterns to [11] for an assurance case of a chassis control function based on ML. We point out similarities to and extensions of the ISO standards within our assurance case. Furthermore, we highlight how a suitable selection of a ML paradigm can support the assurance case claims.

3 Case Study

This paper describes the development of an assurance case for a Tyre Noise Recognition (TNR) component that is used to improve multiple vehicle-level functions. The TNR makes use of microphones positioned within the wheel housing to measure road surface noise in order to determine, in real time, whether or not the road is dry. Here, dryness is defined as a road surface without any materials between tyre and road surface. This classification is, in turn, used as an additional source of information by chassis control and powertrain systems to determine the current surface traction and thereupon adapt control parameters accordingly, i.e. a *!dry* surface requires adaptations for a consistent traction. An overview of the architecture is depicted in Figure 1.

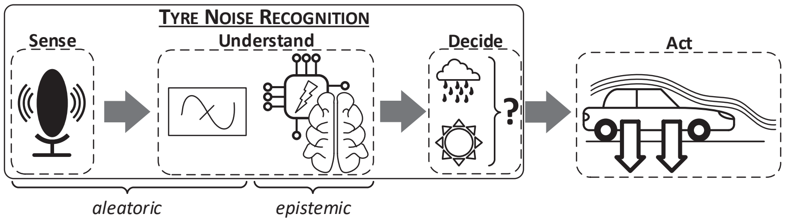

Fig. 1. TNR within system context and its sources of uncertainty.

In order to provide accurate information to the chassis control system, the TNR must process the audio signal with strict real-time requirements and be able to filter sampling anomalies caused by conditions such as the impact of loose gravel. Due to the runtime properties as well as the ability to process a wide range of signal patterns based on available data, a ML technique was chosen to implement the classification function of the TNR (cf. Section 5). Previous versions of the TNR were used to optimise chassis control performance. Through limits imposed within the vehicle-level function, the remaining safety concerns regarding the ML-based classification were low enough to assign only *Quality Management* (QM) requirements to the TNR after completing the hazard and risk analysis according to ISO 26262. However, in order to increase the functional benefits of the vehicle-level function through usage of TNR information, it was decided to evaluate the impact of reducing the limits imposed within the vehicle-level function. This in turn placed an increased safety load onto the TNR and hence led to the following *functional safety requirement* (FSR) allocated to the TNR:

- **FSR x:** The TNR shall not provide the result *dry* in case of a non-dry road surface (ASIL B).

The objective of the project was to develop an assurance case to argue that this level of integrity can be achieved for the TNR even though its output depends on a ML-based classification function. This includes ensuring that the hardware and software components were developed and verified according the *ASIL B* relevant guidelines of ISO 26262 to ensure the integrity of the execution with respect to hardware failures and software errors. In addition, SOTIF-like safety concerns regarding uncertainty in the domain understanding as well as accuracy of perception functions must also be considered when developing such novel ML-based perception systems. This work describes the underlying approach for ensuring a sufficient level of accuracy of the ML-based road surface classification across all target operational scenarios, providing a crucial building block for assuring the safety of ML-based systems for vehicle control systems.

4 Assurance Approach

ML as an implementation paradigm is increasingly used in automotive use cases where the characteristics of the environment can not be adequately specified for the purposes of an algorithmic implementation or where such an implementation may be too computationally intensive, as was the case in the TNR. This, however can come at the price of introducing uncertainty into the system, which in turn can manifest itself in the form of functional insufficiencies as defined by ISO/PAS 21448. These uncertainties manifest themselves in various components within the logical architecture of the system. Of particular interest for this work were the aleatoric uncertainty inherent in the environment in terms of the manifold factors that can impact the acoustic signal as well as the epistemic uncertainty introduced by the ML models themselves. Safety assurance must demonstrate that the system performance is able to satisfy the safety goals, despite these potential sources of inadequacies. Therefore, principles from the ISO/PAS 21448 were adapted to extend the safety lifecycle based on ISO 26262.

An additional factor increasing the difficulty of assuring safety is the issue of the *semantic gap* [4] by which the lack of a precise definition of the functional and performance requirements leads to an inadequate definition of safety requirements in relation to the intended or expected behaviour of the system. These considerations led to the identification of the following additional requirements on the safety lifecycle:

- A *domain analysis* as an extension of the *item definition* phase is required in order to include a thorough investigation of the operational domain and understanding of aspects of the environment that can lead to misclassifications. This phase led to an improved understanding of the system's safety requirements and the identification of a domain model, which in turn was used when reasoning about the completeness of training data and tests.
- The *design phase* refined these system-level requirements into technical safety requirements allocated to either primary functions or diagnostic and monitoring mechanisms. An analysis of potential *failure modes*, in terms of insufficiencies of the ML technique and model was required in order to identify performance improvements and diagnostic methods.
- Measures to *validate* the completeness of the specification and to determine whether a sufficient coverage of environmental conditions that lead to known insufficiencies has been reached and to minimise the residual risk of unknown triggering events.
- Due to the lack of specific guidance from the relevant safety standards, an *assurance case* approach [8] is required in order to reason about the adequacy of the safety approach. GSN [14] was applied in order to document, evaluate and argue the sufficiency of the safety measures within the project.

The phases of the assurance process (cf. Figure 2) were implemented as an iterative process. For example, technical system design choices, such as the selection of sensor types, impact properties of the environment that must be analysed as part of the domain analysis. The discovery of unsupported assumptions in the assurance case may require a restriction of the functionality in order to ensure that the system safety requirements are fulfilled.

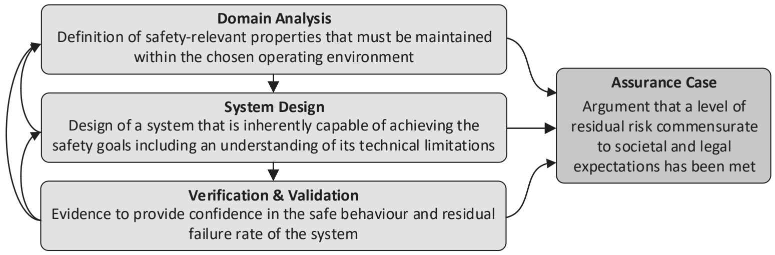
Domain Analysis
Definition of safety-relevant properties that must be maintained within the chosen operating environment
System Design
Design of a system that is inherently capable of achieving the safety goals including an understanding of its technical limitations
Verification & Validation
Evidence to provide confidence in the safe behaviour and residual failure rate of the system
Assurance Case
Argument that a level of residual risk commensurate to societal and legal expectations has been met

Fig. 2. Summary of the assurance process.

4.1 Domain Analysis

During the domain analysis phase, the open context environment was systematically investigated in order to understand factors influencing the sound profile and hence lead to unintended classifications. In order to focus on factors affecting safety, the following relationship between classified and actual prevailing road surface condition was established:

- True-Positive (TP) Predicted *dry* while actually *dry*,
- True-Negative (TN) Predicted *!dry* while actually *!dry*,
- False-Negative (FN) Predicted *!dry* while actually *dry*,
- False-Positive (FP) Predicted *dry* while actually *!dry*.

The misclassification *FN* only results in an overly conservative control strategy as higher traction is not actually needed but still activated, thereby not violating any safety goals. Hence, only the misclassification *FP*, which corresponds to *FSR x* (cf. Section 3), is safety-relevant.

Next, the concept of identifying *triggering events* as described in ISO/PAS 21448 was applied in order to develop an understanding of environmental conditions that could lead to a *FP* classification. This analysis was based on a thorough technical understanding of the sensing and signal processing principles involved as well as experience gained during the development and test of the previous QM-rated version of the TNR. The three main influences on the acoustic sensing that were identified from the environment are: tyres, road surface and the transmission medium of sound. These factors were then decomposed into their fundamental properties, e.g. tyres into rubber mixture, tyre pressure, tyre dimensions and others. The granularity and definition of each property has been selected according to physical realisability, for instance, tyre sizes only within actual produced dimensions. The resulting domain model consists of all feasible combinations of these properties and can be used to identify *known triggering events* describing known performance limitations of the systems [9]. The domain model can also be used to determine coverage criteria for test cases. However, even for this relatively simple application, the procedure created an unmanageably large amount of combinations. Too many, in fact, to be practically feasible. To reduce the amount of test cases, while still arguing coverage of the operational domain, each property and their individual impact was evaluated using expert knowledge. This assessment included considerations about safety with special attention to the physical sensing principle in detail, possible dependence between properties, as well as their overall significance for the classification. For instance, test cases regarding tyre dimensions only included min and max sizes and other combinations of parameters were considered irrelevant as no correlation between the parameters could be determined that would have an impact on the performance beyond the individual impact of the parameters themselves.

Nevertheless, uncertainty in the completeness of the domain model and adequacy of the abstractions required to reduce combinatorial explosion leads to the possibility of *unknown triggering events* and must still be accounted for in the assurance process. Therefore additional measures were defined in the V&V phase in order to validate the domain model.

4.2 System Design

To analyse the design of systems based on the TNR, the architecture presented in Figure 1 was decomposed into the logical component groups: *Sense, Understand, Decide* and *Act*. This allowed for a clear separation of the concerns identified in the *Domain Analysis* and an analysis of each component's contribution to overall performance insufficiencies in the system.

The *sensing* part of the system includes the microphones inside the wheel arches and their task of measuring the sound waves. The sound waves are recoded within certain frequency boundaries and compressed for data transmission (cf. Figure 1). Potential sources of aleatoric uncertainty are the lack of information, meaning the recorded frequency range does not cover the complete frequency spectrum sufficiently, measurement uncertainty, defining an imperfect measurement process by technical devices, and numerical approximations within the data compression algorithm. As all of these uncertainties can potentially lead to insufficient performance of the TNR, they have been addressed within the assurance case along with supporting evidence, e.g., mathematical analysis of data compression losses. The *understanding* portion of the logical architecture is accomplished by signal pre-processing and an ML-based classifier within the TNR (cf. Figure 1). The classification exploits the fact that different road conditions are differentiable through acoustic properties. Potential causes of uncertainties are ambiguous sound patterns or epistemic uncertainty arising from the selected ML technique and model. A pessimistic decision strategy was used. In particular, the TNR will select the safer option *!dry* in case of conflicting predictions. Samples from multiple sensors are combined and aggregated over multiple sampling steps before providing a *dry* classification. The components corresponding to the *decide* and *act* function groups contain the chassis control logic and actuator components, respectively. According to the prediction the driving performance is optimised, for instance by adapting the suspension or spoiler.

4.3 Verification and Validation

Within the *verification and validation* (V&V) phase, performance requirements allocated to the system and its components were confirmed. In addition, assumptions regarding the performance potential of the design, as well as the environment operating conditions were confirmed in order to argue the safety of the system for its chosen context. This led to the identification of the following additional objectives within the V&V strategy, with respect to the QM version of the TNR function:

- **Confirmation of assumptions made during system design and safety assurance:** This included, for example, evaluating field data to assess whether the operating condfitions matched the assumptions in the domain model (cf. Section 4.1) and confirming that pre-processing of the audio signal did not reduce the dimensionality of the input data in such a way that *dry* and *!dry*

signals could be mapped to similar feature vectors. Other assumptions include the influence of signal noise (aleatoric uncertainty) on the performance of the classifier.
- **Evaluation of the function with regard to *known triggering events*:** These include combinations of environmental conditions discovered during the domain analysis as well as specific corner cases discovered during field testing.
- **Evaluation of the potential for *unknown triggering events*:** This objective includes confirms that the domain model covers a sufficient range of conditions that can impact the performance of the function and that all relevant usage scenarios have been considered.
- **Evaluation of the resilience of the function with regard to residual *unknown triggering events*:** This objective relates to the ability of the system to respond to signal patterns not considered in the domain model and for which either a valid response must nevertheless be given, or no value at all, resulting in a conservative action from the chassis control system.

A number of analyses, simulations and tests had previously been performed for the QM-rated version of the TNR. However, these measures were not necessarily aligned to the objectives described above, resulting in some gaps in the argumentation structure. Therefore, the following methods were identified in order to provide explicit evidence corresponding to the V&V objectives. In some cases, existing evidence could be aligned with the V&V objectives, in other cases, additional tests and associated documentation were required.

- **Analysis:** An understanding of the strengths and weaknesses of the chosen ML technique and model provided evidence for the inherent properties regarding robustness and generalisation. In addition, the prototypes generated by the algorithm (cf. Section 5) were amenable to examination by subject matter experts to confirm that they corresponded to known properties of the *dry* and *!dry* signals.
- **Simulation:** A simulation environment based on synthetic and recorded data was used for a focused verification of ML properties. Here, signal noise can also be simulated in order to verify the robustness of the classifier.
- **Structured testing:** The domain model was used to determine a set of test cases which cover all known properties which could influence the performance of the function. In addition, the test cases also included specific corner cases discovered during field tests and added to the regression test set.
- **Field tests:** Field tests, where the function was tested on real roads (both test track and public roads) were performed according to selected properties of the domain model (cf. Section 4.1). This allowed the coverage of conditions to be evaluated. Anomalies which could not be explained by the parameters of the domain model were used to iteratively refine the domain model.

4.4 Assurance Case

The objective of the assurance case was to develop a structured and convincing argument that the classifier fulfilled its technical requirements, in particular with respect to functional insufficiencies that could lead to *FP* identifications of dry road surface conditions. The assurance case was described using GSN and applied the principles from [8]. During the project it was primarily used as a means of communicating and evaluating the safety assurance approach within the team but was also developed with future external safety assessors in mind.

The top level structure of the assurance case is shown in Figure 3. The assurance case focused on claims regarding a sufficient understanding of the domain and subsequent completeness of the technical safety requirements, the intrinsic performance potential of the chosen ML technique, the sufficiency of the training data to cover critical conditions of the domain, and the performance of the trained function itself. In addition, arguments were developed that the ML-based classifier was robust against changes in the operating environment as well as differences between the development and test environment and future deployment scenarios (e.g. different vehicle configurations).

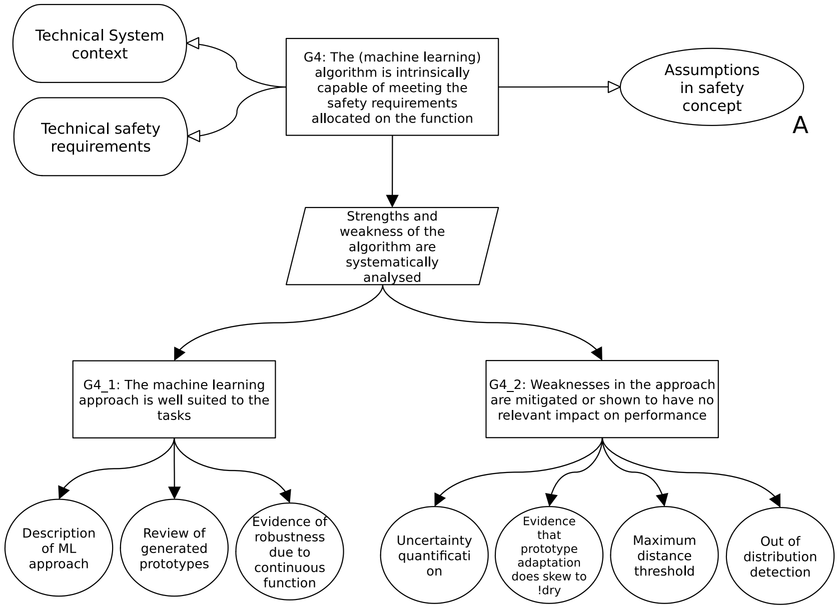
Technical System context
Technical safety requirements
G4: The (machine learning) algorithm is intrinsically capable of meeting the safety requirements allocated on the function
Assumptions in safety concept
A
Strengths and weakness of the algorithm are systematically analysed
G4_1: The machine learning approach is well suited to the tasks
G4_2: Weaknesses in the approach are mitigated or shown to have no relevant impact on performance
Description of ML approach
Review of generated prototypes
Evidence of robustness due to continuous function
Uncertainty quantificati on
Evidence that prototype adaptation does skew to !dry
Maximum distance threshold
Out of distribution detection

Fig. 3. Top-level assurance case structure.

5 Detailed Analysis of the Machine Learning Function

A detailed analysis of the applied ML-based classifier with respect to the technical safety requirements allocated to it and its general suitability regarding the target task was performed.

In the case of the TNR, *Adaptive Generalised Learning Vector Quantisation* (AGLVQ), an extension to GLVQ [13], was used. Figure 4 shows the operating principle of this algorithm.

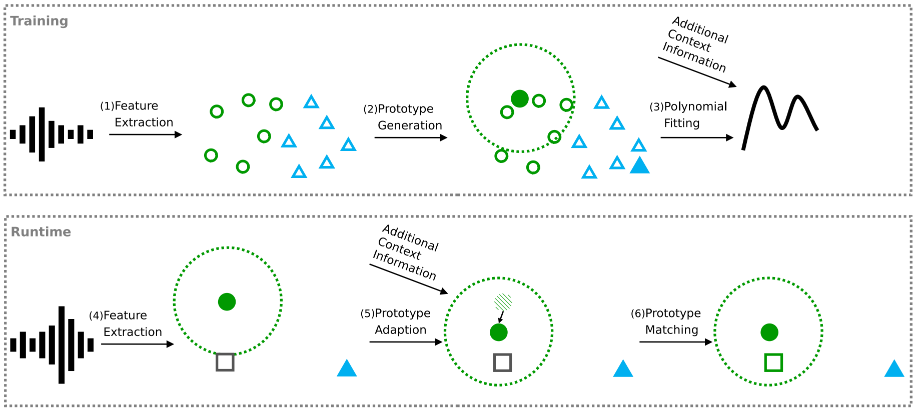

Fig. 4. Overview of the employed AGLVQ algorithm. In the training phase, meaningful features are extracted from the audio signal (1) that are subsequently used to generate prototypes (2) for both classes, *dry* (circles) and *!dry* (triangles), maximising the distance between them. A threshold for the *dry* prototype is defined, forming the decision space for this class. Additionally, a polynomial is fitted (3) that, using additional context information, adapts the *dry* prototype to the current situation. At runtime, features are again extracted from the audio signal (4), mapping the current sample (rectangle) to the feature space. After that, the learned polynomial is used to adapt (5) the *dry* prototype to the current situation. Finally, the current sample is matched to the prototypes (6) based on the Euclidean distance in the feature space. If the current sample is within the decision space of the *dry* prototype it is classified as such else a *!dry* road surface is assumed.

The use of AGLVQ had several advantages over other ML approaches from the perspective of safety assurance. The learned prototypes have been represented in the same form as the feature engineered audio signals and allowed the engineers to verify their plausibility. Additionally, in combination with the straightforward prototype matching used for runtime predictions and the interpretable adaption polynomial, it allowed for a detailed analysis of the decision space and uncovered potential error patterns. This also outlined another strength of this approach, the robustness to small input perturbations. Compared to, e.g., neural networks, there has been no feature subspace in which small changes of individual features could be amplified in a way that causes drastic and unexpected changes in the output. While not explicitly investigated, this may also significantly reduce the susceptibility to adversarial attacks. Due to this absence of discontinuities, the sensitivity to individual factors, e.g., tread depth, the burden of proof on the empirical tests was significantly lower compared to other discontinuous functions. Figure 5 shows an extract of the GSN regarding the choice of the ML technique. Here, the strengths of AGLVQ have been reflected in G4_1.

Several limitations of AGLVQ were also identified and used to derive additional *Technical Safety Requirements* (TSR). One such limitation was a low level of generalisation. As only a single prototype for the relevant class was generated there was a noticeable trade-off between safety and execution performance. A prototype adaption function mitigated this to some extent by incorporating additional knowledge about the context of the present situation. However, this was not sufficient for complex generalisations such as completely new types of road surfaces. Another limitation was that the prototype adaption in certain situations transformed the *dry* prototype slightly towards the decision space for non-dry road conditions, increasing the risk of incorrect classifications. Furthermore, the approach did not have an explicit way to quantify the uncertainty for the learning of, adapting, and matching to the prototype apart from the Euclidean distance, which does not fully account for the relation between the features. The known weaknesses and the evidence associated with the effectiveness of the counter-measures to these have been reflected in the GSN under G4_2. The analyses of these inherent weaknesses in the approach led to the proposal to develop self-assessment methods, specifically uncertainty quantification [5] and out-of-distribution detection to be applied at run-time.

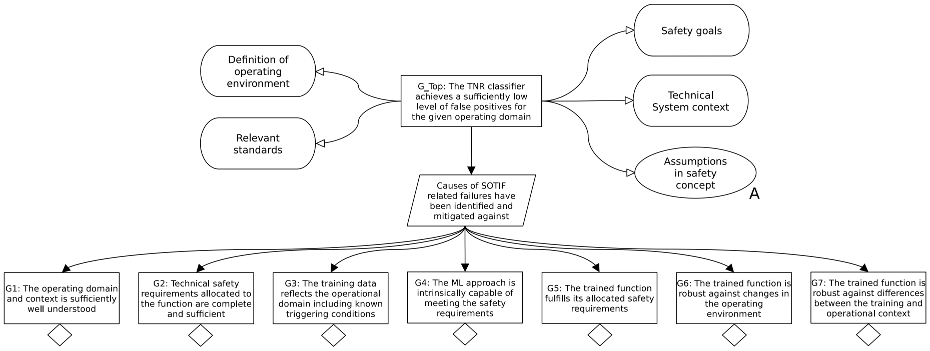
Definition of operating environment
Relevant standards
G_Top: The TNR classifier achieves a sufficiently low level of false positives for the given operating domain
Safety goals
Technical System context
Assumptions in safety concept
A
Causes of SOTIF related failures have been identified and mitigated against
G1: The operating domain and context is sufficiently well understood
G2: Technical safety requirements allocated to the function are complete and sufficient
G3: The training data reflects the operational domain including known triggering conditions
G4: The ML approach is intrinsically capable of meeting the safety requirements
G5: The trained function fulfills its allocated safety requirements
G6: The trained function is robust against changes in the operating environment
G7: The trained function is robust against differences between the training and operational context

Fig. 5. Assurance case structure for choice of ML technique.

In order to demonstrate the performance of the trained function itself (subgoal G5 of Figure 5) performance metrics were defined and related to the TSRs. Since the TNR is a classification task, appropriate metrics were, among others, *accuracy*, *precision/recall* and *confusion matrices*. These helped to measure the overall performance and aided the investigation of error patterns. For instance, class-wise precision and recall allowed appropriate distance thresholds for the prototype matching to be identified and validated using the available test data. Additionally, the metrics helped with finding variances in the test data. The causes of these variances were iteratively analysed in more depth, either by gathering additional data or by qualitatively assessing the function with respect to the properties of the variance causing data. Lastly, the metrics were used to define acceptance criteria for the TSRs, e.g. that a certain class-wise accuracy on all validation datasets shall be achieved.

Based on a combination of the measures described above, the fundamental capability of the classifier for the target task was argued. Regarding the TNR, AGLVQ was found to be generally suitable, especially as the high degree of explainability allowed for a thorough analysis of the function and its behaviour. However, the known limitations and their consequences still left a burden of proof on the training data and the validation results, which were argued in sub-goals G3 and G5 of the assurance case.

6 Lessons Learned

The evaluation of the TNR with respect to its application for a safety-critical system (*ASIL B*) led to a number of lessons learned that could be applied in future projects as well as open questions that still remain to be resolved. The nature of the system level safety goals associated with the chassis control functions allowed for a safe state to be achieved if the road surface could be considered as *!dry*, thereby leading to a conservative traction control strategy. This allowed the function to be designed to indicate an invalid output in the case of ambiguous inputs as well as a skewing of the audio signals towards the *!dry* prototype if required.

The robustness and explainability of the approach helped with the in-depth analysis of the machine learning component. The ability to analyse the generated prototypes, their adaption to the current situation at runtime, and the respective decision space allowed the incorporation of expert knowledge in the quality assessment. In addition, the robustness due to the continuity of the function substantially facilitated the investigation of the influence of factors such as tread depth. This allowed for a significant reduction in the amount of in-field tests due to a reduction in the dimensions considered during coverage analysis.

Open questions nevertheless remain on the required level of granularity in the domain model used to evaluate the completeness of selected training data as well as quantitative test stopping criteria related to statistical performance metrics. Inevitably, an iterative approach to system development and assurance will be required (cf. Figure 2) where field-based validation is required to confirm that sufficient detail in the domain model was achieved and that assumptions made during analysis, simulation and test were valid. These questions, however, are currently not addressed by existing safety standards such as ISO 26262, which assumes behaviour of software that can be evaluated through qualitative measures or ISO/PAS 21448 which requires a function-specific allocation of quantitative performance targets. The approach used within the project was to use qualitative arguments to argue the robustness of the ML function with respect to a broad range of operating conditions as defined by the domain model, whilst applying a range of measures (including extensive structured field tests) to confirm the assumptions behind the domain model. In addition, the TNR is embedded within a vehicle chassis control system, which in turn is developed and released according to a set of established development and homologation guidelines. Nevertheless, an external evaluation of the assurance approach by a qualified third party is recommended to examine the strength of the provided arguments.

7 Conclusion

The work described within this paper has demonstrated the feasibility of an assurance case for the application of ML for chassis control systems. The assurance approach made use of a systematic domain analysis to define properties of the environment relevant to the performance, dedicated measures in the system architecture to reduce the safety requirements on the ML function itself, the choice of an ML technique that enhanced robustness and explainability combined with a systematic validation plan to argue the absence of unknown triggering events. However, questions remain relating to the statistical level of performance that should be demonstrated by the ML algorithm. This type of evidence, would go above and beyond the forms of V&V proposed by the ISO 26262 standard for software but is required due to the inherent uncertainties when applying ML compared to conventional non data-driven algorithms.

The project highlighted the need for better industry-specific standards regarding the use of ML for safety-relevant functions, including outside of the domain of automated driving. These standards should include specific guidelines for determining coverage and selection criteria for training data, as well as for determining quantitative performance targets and testing criteria. These aspects would become even more relevant by alternative choices of ML technique, such as *Deep Neural Networks*, where qualitative arguments relating to the robustness and generalisation properties of the trained functions are more difficult to generate based on the complexity and opaqueness of the calculations involved. As such, any future standardisation should also include a differentiation of measures based on the intrinsic characteristics of the ML algorithms.

Acknowledgments

On side of Fraunhofer IKS, the research for developing the theoretical safety foundations utilisied in this work was partially funded by the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

References

1. Bagschik, G., Reschka, A., Stolte, T., Maurer, M.: Identification of Potential Hazardous Events for an Unmanned Protective Vehicle. In: Proc. IEEE Intelligent Vehicles Symp. (IV). pp. 691–697. Gotenburg (2016)

2. Belinkov, Y.: On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition. Ph.D. thesis, Massachusetts Institute of Technology (MIT) (2018)
3. Burton, S., Gauerhof, L., Heinzemann, C.: Making the Case for Safety of Machine Learning in Highly Automated Driving. In: Proc. 36th Int. Conf. on Comp. Safety, Reliability, and Security (SafeComp). pp. 5–16. Trento (2017)
4. Burton, S., Habli, I., Lawton, T., McDermid, J., Morgan, P., Porter, Z.: Mind the Gaps: Assuring the Safety of Autonomous Systems from an Engineering, Ethical, and Legal Perspective. Artificial Intelligence **279** (2020)
5. Fischer, L., Hammer, B., Wersing, H.: Efficient Rejection Strategies for Prototype-based Classification. Neurocomputing **169**, 334–342 (2015)
6. International Organization for Standardization: Road Vehicles — Functional Safety (ISO 26262) (2018)
7. International Organization for Standardization: Safety Of The Intended Functionality - SOTIF (ISO/PAS 21448) (2019)
8. International Organization for Standardization: Systems and Software Engineering - ISO/IEC/IEEE 15026-1:2019 (2019)
9. Kurzidem, I., Saad, A., Schleiss, P.: A Systematic Approach to Analyzing Perception Architectures in Autonomous Vehicles. In: Proc. 7th Int. Symp. on Model-Based Safety and Assessment (IMBSA). pp. 149–162. Lisbon (2020)
10. Picardi, C., Hawkins, R., Paterson, C., Habli, I.: A Pattern for Arguing the Assurance of Machine Learning in Medical Diagnosis Systems. In: Proc. 38th Int. Conf. on Comp. Safety, Reliability, and Security (SafeComp). pp. 165–179. Turku (2019)
11. Picardi, C., Paterson, C., Hawkins, R.D., Calinescu, R., Habli, I.: Assurance Argument Patterns and Processes for Machine Learning in Safety-Related Systems. In: Proc. Workshop on Artificial Intelligence Safety (SafeAI). New York (2020)
12. Radlak, K., Szczepankiewicz, M., Jones, T., Serwa, P.: Organization of Machine Learning based Product Development as per ISO 26262 and ISO/PAS 21448. In: Proc. 25th IEEE Pacific Rim Int. Symp. on Dependable Computing (PRDC). pp. 110–119. Perth (2020)
13. Sato, A., Yamada, K.: Generalized Learning Vector Quantization. In: Proc. 8th Int. Conf. on Neural Information Processing Systems (NIPS). pp. 423–429 (1995)
14. The Assurance Case Working Group (ACWG): Goal Structuring Notation - Community Standard. No. 2, Safety Critical Systems Club (SCSC) (2018)
15. Willers, O., Sudholt, S., Raffatnia, S., Abrecht, S.: Safety Concerns and Mitigation Approaches Regarding the Use of Deep Learning in Safety-Critical Perception Tasks. In: Proc. 39th Int. Conf. on Comp. Safety, Reliability, and Security (SafeComp). Lisbon (2020)
16. Ye, H., Liang, L., Li, G.Y., Kim, J., Lu, L., Wu, M.: Machine Learning for Vehicular Networks: Recent Advances and Application Examples. IEEE Vehicular Technology Magazine **13**(2), 94–101 (2018)