Making the Case for Safety of Machine Learning in Highly Automated Driving

Simon Burton(✉), Lydia Gauerhof, and Christian Heinzemann

Corporate Research, Robert Bosch GmbH, Renningen, Germany
{Simon.Burton,Lydia.Gauerhof,Christian.Heinzemann}@de.bosch.com

Abstract. This paper describes the challenges involved in arguing the safety of highly automated driving functions which make use of machine learning techniques. An assurance case structure is used to highlight the systems engineering and validation considerations when applying machine learning methods for highly automated driving. Particular focus is placed on addressing functional insufficiencies in the perception functions based on convolutional neural networks and possible types of evidence that can be used to mitigate against such risks.

1 Introduction

The transition from *hands-on* (Levels 1–2 of [25]) driver assistance to *hands-off* automated driving (Levels 3–5) requires a number of changes to system safety approaches. For example, a higher level of availability is required as the system cannot be simply deactivated upon detection of a component hardware fault (fail operational vs. fail safe) [20]. At a functional level, an approach to interpreting the current driving situation including environmental conditions and making judgements regarding the subsequent actions is required in order to ensure critical driving situations are avoided under all possible circumstances [26]. The use of machine learning (e.g. for perception tasks [19]) is seen as a promising answer to some of the functional challenges of highly automated driving (HAD) based on the ability to extract relevant features within an unstructured input space. However, systems based on machine learning can only be released into the public domain if they can be argued to be acceptably safe.

The conditions for being acceptably safe with respect to functional safety are set by ISO 26262 [16]. Adherence to this standard remains a necessary prerequisite for demonstrating the safety of HAD in order to ensure a reliable and fault tolerant implementation of the system with respect to random hardware and systematic failures. Nevertheless, in a number of areas the standard does not transfer well to the application of machine learning for open context systems, i.e., systems where the operational context and the environmental conditions for operation cannot be clearly defined at design time. As an example, the development and verification methods contained within part 6 of the standard do not address the problem that when applying machine learning approaches, the functionality itself is essentially embedded in highly dimensional data matrices

S. Tonetta et al. (Eds.): SAFECOMP 2017 Workshops, LNCS 10489, pp. 5–16, 2017.
DOI: 10.1007/978-3-319-66284-8_1

[11] for which verification techniques such as coding guidelines and white-box code coverage [22] provide no relevant insights. In addition, the issue of the insufficiency of the system to meet the safety goals, due to inherent restrictions in sensors, actuators or the inadequacy of the target function itself is also not well addressed by ISO 26262. Extensions to the standard, such as the "Safety of the Intended Function" (SOTIF) approach currently under development aim to address some of these issues, but are mainly focused on driver assistance rather than HAD systems [3]. As a result, alternative methods must be developed and the ability of the system to meet its safety goals must be systematically argued based on "first principles" where adherence to a standard is only one part of the overall argument. An assurance case [15] provides a convincing and valid argument that a set of claims regarding the safety of a system is justified for a given function based on a set of assumptions over its operational context.

In this paper we will explore how assurance case approaches can be applied to the problem of arguing the safety of machine learning within the scope of HAD. As a basis, we analyse and discuss different uses of machine learning and their impact on arguing system safety (Sect. 2). Using a systematic analysis of claims, context and assumptions, we demonstrate that the question of safety for a machine learning function cannot be answered without a detailed understanding of the system context (Sect. 3). The argumentation path proposed in this paper will focus on mitigating the main causes of functional insufficiencies in machine learning based functions (Sect. 4). Finally, we summarise techniques that could be used to create the evidence required to support the assurance case claims (Sect. 5). We use the Goal Structuring Notation [30] to illustrate main lines of the argumentation but our example should not be seen as a comprehensive safety case. We conclude the paper with a brief examination of future work required in this area.

2 Machine Learning and Safety

The issues involved in arguing the safety of machine learning depend very heavily on the types of techniques applied and their application within the overall system context. The following perspectives are useful in evaluating the impact of machine learning on the overall safety case:

– **Scope of the Function:** Machine learning can be applied for different tasks within a HAD functional architecture. The more restricted the task, the more specific the performance criteria on the function can be defined, allowing focused validation and verification activities. Attempts have also been made at applying machine learning techniques for end-to-end learning, the scope of which covers the data fusion of various sensor inputs (e.g. camera and radar data), trajectory planning and decision making and eventual vehicle control (braking, acceleration, steering) [6]. Safety requirements for end-to-end learning approaches are by necessity more abstract due to the scope of the function (e.g. avoid collisions with other vehicles) thus making the task of formulating and validating measurable performance criteria significantly more difficult [28].

– **Learning technique:** Probabilistic inference and statistical learning techniques include methods such as Support Vector Machines [7], Gaussian Processes [23], Markov decision process [9] and Bayesian Networks [10]. These approaches build statistical models based on training data and typically exhibit a continuous behaviour with increasing accuracy of results, the more they are trained. Deep learning algorithms apply multiple processing layers, composed of multiple linear and non-linear transformations to model high-level abstractions within the input data [11]. Examples of which are Convolutional Neural Networks[1] (CNNs, [11, ch. 9]) which show significant potential for processing images [19]. These algorithms add additional challenges due to the lack of transparency and explainability [5] in the features learned at deeper layers and vulnerability to unpredictable "adversarial examples" [12].
– **Learning time:** Machine learning functions that are trained during the development phase (for example using offline supervised learning) can be validated as part of a system release strategy. Online training techniques such as reinforcement learning involve continuously adapting the function during execution (i.e. while driving). Furthermore approaches can be distinguished between centralized training (functions within different vehicles are training in the same way with the same input data) or decentralized training (in this case individual vehicles learn on their own and differ from each other in their learning progress). Techniques which continue their training online must be embedded within a context that ensures that the function remains within a safe envelope as it adapts to its environment in the field. Therefore, unless a sufficiently complete set of invariants for a safe operating envelope can be deterministically defined, decentralized, online training imposes the greatest challenges for safety validation.
– **Distance between critical events:** Events exist that must be handled correctly by the system to ensure safety goals are met (e.g., children appearing suddenly on the road or the car being close to loosing traction) but that are typically under-represented in the training data for the machine learning function. This is either because they occur rarely in reality or because the collection of sufficient training data itself represents an unacceptable risk. As a result, analysis is required to determine how such situations can be adequately covered during training and validation to ensure that they are equally well handled by the system as commonly occurring situations.

The challenges involved in providing a convincing assurance case for the system will heavily depend on the scope of the function as defined by the above dimensions. It is expected that, in practice, the initial applications of machine learning in series development of HAD systems will be for offline, centrally trained functions, implementing well specified detection tasks which can be supported by plausibility checks based on alternative channels within the system context. One such example application, which shall be referenced in the rest of this paper, is the application of CNNs to detect (i.e. classify and localize) objects based on camera images as part of a collision avoidance system for self driving vehicles.

[1] See https://www.youtube.com/watch?v=u6aEYuemt0M for an introduction.

3 Application Context

The *Goal G1* that forms the top-level claim of the assurance case scope addressed in this paper will focus on the contribution of the machine learning function to *functional insufficiencies* in the system and can therefore be defined as follows:

– **G1: The residual risk associated with functional insufficiencies in the object detection function is acceptable.**

Arguing the contribution to the *residual risk* due to functional insufficiencies requires a detailed understanding of the functional and performance requirements on the object detection function within the overall system context. These requirements are referenced by the following *Context element C1* of the argument:

– **C1: Definition of functional and performance requirements on the object detection function.**

An assurance case structure at the system level is required that leads to a definition of the performance requirements and risk contribution allocated to the machine learning function. An argument is also required that the contributions of systematic failures and random hardware faults are also adequately covered, for example based on an ISO 26262 related argument. The development of these arguments also lead to significant unsolved challenges in the engineering and validation of HAD systems, but are outside the scope of this paper. The derivation of performance (Safety) requirements within the system context is one of the key contributions to ensuring overall system safety and requires deep domain and system knowledge. Deriving a suitable set of requirements for open context systems is a non-trivial task in itself, systematic approaches to systems engineering are therefore indispensable.

Figure 1 illustrates how a machine learning function could be embedded within its system context. Typical requirements that might be derived at the level of the machine learning function could include for example: *Locate objects of class person from a distance of* $X1$*, with a lateral accuracy of* $X2$*, a false negative rate of* $X3$ *and false positive rate of* $X4$. The parameters $X1, X2, X3, X4$ can be functions that depend on the velocity of the ego vehicle or time to collision (TTC), respectively. The distance to an object $X1$ and the accuracy $X2$ might be also limited by sensor range and resolution and by estimated relevant minimum width of objects (e.g. width of infant legs), while a false negative rate of $X3$ and false positive rate of $X4$ can be tuned by training, evaluation parameters or system measures (e.g. data fusion). Such requirements provide a clear measure of performance for the machine learning environment, but also imply a number of assumptions on the system context. These assumptions might include that the braking distance and speed are sufficient to react when detecting persons for example as close as 10 m ahead on the planned trajectory of the vehicle and that other system measures are available to decrease the overall false negative and false positive rate to a sufficiently safe level, etc.

As indicated in Fig. 1, a design by contract approach is recommended whereby each functional component of the system is defined by a set of assumptions that needs to hold in order to ensure the specified behaviour. The following list contains typical assumptions that are relevant for the assurance case:

- **A1: Assumptions on the operational profile of the system's environment.** For example, the types and occurrence distribution of objects in the environment.
- **A2: Assumptions on attributes of inputs to the machine learning function.** For example, the camera resolution is sufficient to be able to detect persons from a distance of 100 m with the required accuracy.
- **A3: Assumptions on the performance potential of machine learning.** For example, the chosen CNN approach has the intrinsic potential, given the right parameterization and set of learning data to fulfil the allocated performance requirements.

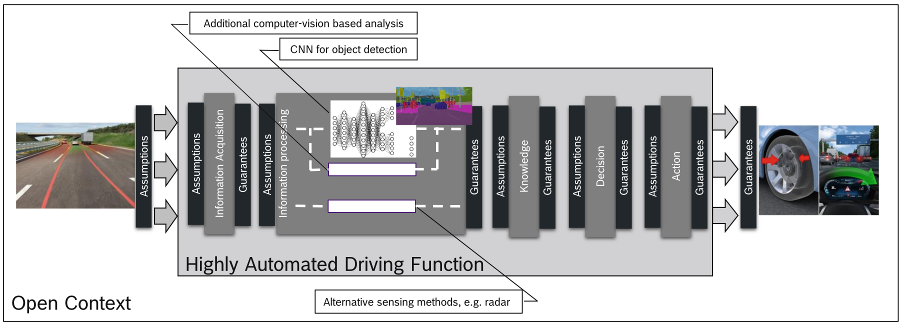

Fig. 1. Functional architecture of a HAD system

4 Causes of Functional Insufficiencies in Machine Learning

The inherent uncertainty associated with machine learning techniques coupled with the open context environment lead to different causes of hazards compared to traditional, algorithmic and control law approaches to vehicle control. In order to argue the claim that functional insufficiencies within the machine learning function (here: camera based object detection, supervised training) are minimised, it is important to understand the causes of the insufficiencies. The assurance case strategy S1 described here can thus be described as follows:

- **S1:** Argument over causes of functional insufficiencies in machine learning.

As interest in machine learning safety has grown, a number of authors [1,27,31] have investigated different causes of insufficiencies in machine learning functions. A number of causes that are applicable to the HAD use case described here are summarised below and are positively formulated as sub-goals G2-G6 within the assurance case, which is summarised in Fig. 2:

– **G2: The environmental context is well defined and reflected in training data.** One of the key differences in machine learning techniques compared to algorithmic approaches is the lack of a detailed specification of the target function. Instead, the functional specification can be seen to be encoded within the set of training data. Therefore, if the training data does not reflect the target operating context, then there is a strong likelihood that the learned function will exhibit insufficiencies. Critical or ambiguous situations, within which the system must react in a predictably safe manner, may occur rarely or may be so dangerous that they are not adequately represented in the training data. This leads to the effect that critical situations remain undertrained in the final function. Additional potential for insufficiencies comes from overfitting that results from lack of diversity in the training data, eroding the generalisation properties of the CNN.
– **G3: The function is robust against distributional shift in the environment.** The system should continue to perform accurately even if the operational environment differs from the training environment [1]. This effectively can be formulated as the robustness of the system to react in a shift of distribution between its training and operational environment. Distributional shift will be inevitable in most open context systems, as the environment constantly changes and can adapt to the behaviour of actors within the system. For example, car drivers will adjust their behaviour within an environment in which autonomous vehicles are present, vehicle and pedestrian appearances change over time, etc.
– **G4: The function exhibits a uniform behaviour over critical classes of situations.** An often cited problem associated with neural networks, is the possibility of adversarial examples [8,12,21]. An adversarial example is an input sample that is similar (at least to the human eye) to other samples but that leads to a completely different categorisation with a high confidence value. It has been shown that such examples can be automatically generated and used to "trick" the network. Although it is still unclear to what extent adversarial examples could occur naturally or whether they would be exploited for malicious purposes, from a safety validation perspective, they are useful for demonstrating that features can be learnt by the network and assigned an incorrect relevance [12]. Therefore an argument should be found to minimise the probability of such behaviour especially in critical driving situations.
– **G5: The function is robust against differences between its training and execution platforms.** Machine learning functions can be sensitive to subtle changes in the input data. When using machine learning to represent a function that is embedded as part of a wider system as described here

(see Fig. 1), the input to the neural network will have typically been processed by a number of elements already, such as image filters and buffering mechanisms. These elements may vary between the training and operation environments leading to the trained function becoming dependent on hidden features of the training environment. It is therefore necessary to understand the differences between training and operational environment, including any potential weaknesses or faults in the training environment (e.g. software defects in open source training libraries) and data leakage resulting from hidden, unnoticed correlations in the environmental context of the provided training samples [17].

– **G6: The function is robust against changes in its system context.** Vehicle systems are typically developed and deployed in a wide number of system variants which may include different combinations of sensors as well as different positioning of sensors within the vehicle. In addition, over time changes are made to the system design as part of continuous improvement or cost reduction measures. However, hidden data dependencies [27] may exist by training and validating the machine learning function within a given context. Subtle differences between the original context (e.g. a particular camera lens, or installation height) for which the function was trained and the re-use context can lead to functional insufficiencies that may not appear during development and regression testing but may lead to degraded performance in the field. Furthermore, when updating the system it is also necessary to ensure a monotonic performance improvement, i.e. situations that are safely covered in a previous version of the system must also be covered in the new version, even if the overall performance is improved.

5 Sources of Evidence for the Assurance Case

Dedicated methods for validating machine learning functions, and in particular neural networks, to the level of integrity required by safety-critical systems is currently an emerging field of research. It is expected that, analogous to traditional algorithmic-based software approaches, a diverse set of complementary evidence based on constructive measures, formal analyses and test methods will be required to make a robust assurance case. In this section we discuss different categories of potential evidence and how it can support the sub-claims of the assurance case described in the previous section. Note that each sub-claim in Fig. 2 will depend on more than one source of evidence.

– **Training data coverage:** Open context systems such as HAD are defined by the fact that it is not possible to specify a priori all possible operational scenarios. Applied to the training of neural networks for image processing in HAD, this relates to the infinite dimensionality and variations in the input images. Criteria therefore need to be applied to define how much training data is required for a particular application and which data will lead to the most accurate approximation of the (unspecifiable) target function. It is expected

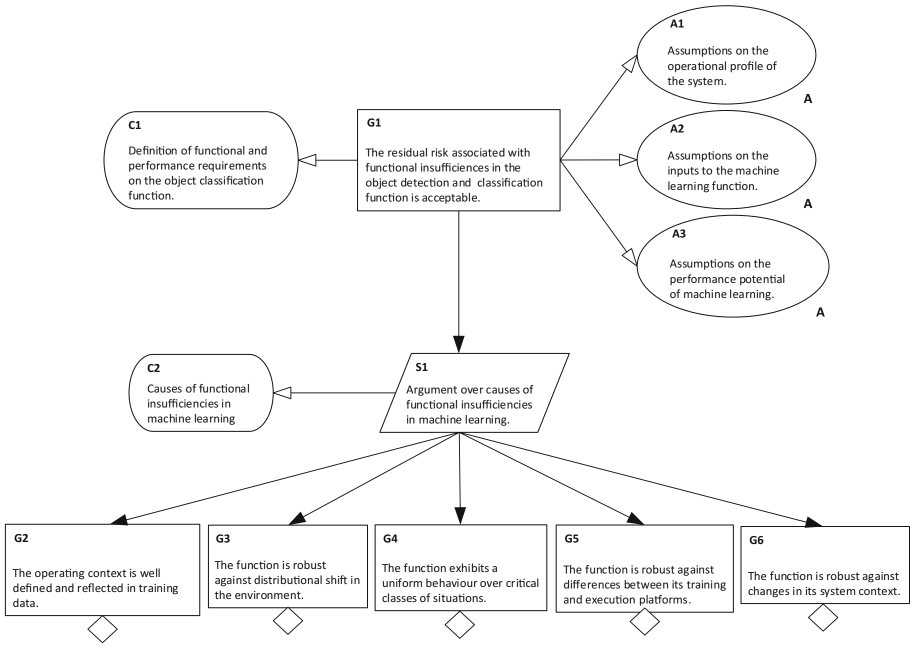
A1
Assumptions on the operational profile of the system.
A
A2
Assumptions on the inputs to the machine learning function.
A
A3
Assumptions on the performance potential of machine learning.
A
C1
Definition of functional and performance requirements on the object classification function.
G1
The residual risk associated with functional insufficiences in the object detection and classification function is acceptable.
C2
Causes of functional insufficiencies in machine learning
S1
Argument over causes of functional insufficiencies in machine learning.
G2
The operating context is well defined and reflected in training data.
G3
The function is robust against distributional shift in the environment.
G4
The function exhibits a uniform behaviour over critical classes of situations.
G5
The function is robust against differences between its training and execution platforms.
G6
The function is robust against changes in its system context.

Fig. 2. Assurance Case Structure

that a combination of complementary criteria applied in parallel will be most effective and may include:

- *Training data volume*: A sufficient amount of training data is used to provide a statistically relevant spread of scenarios and to ensure a stabilization of a strong coverage of weightings in the neural network.
- *Coverage of known, critical scenarios*: Domain experience based on well understood physical properties of the system and environment as well as previous validation exercises leads to the identification of classes of scenarios that should exhibit similar behaviour in the function. Although some classes may be obvious and simple to reproduce (day and night driving, weather conditions, traffic situations), other classes may occur rarely and will require targeted data acquisition and possibly synthetic generation to ensure sufficient coverage during training.
- *Minimisation of unknown, critical scenarios*: Other critical combinations of classes will not even be known during system design [2]. A combination of systematic identification of equivalence classes in the training data and statistical coverage during training and validation will therefore be essential to adequately minimise the risk of insufficiencies due to inadequate training data.

– **Explainability of the learned function:** A key component of demonstrating the correctness of traditional safety-critical software are white-box techniques that include manual code review, static analysis, code coverage and formal verification. These techniques allow for an argument to be formulated on the detailed algorithmic design and implementation but cannot be easily transferred to the machine learning paradigms. Other arguments must therefore be found that make use of knowledge of the internal behaviour of the neural networks. Saliency maps [4,29], based on the back propagation of results in the network to highlight those portions of an image that have greatest influence on classification results, can be used to provide a manual plausibility check of results as well as to determine potential causes of failed tests, e.g., resulting from data leakage. Another line of research tries to generate a natural language explanation referring in human understandable terms to the contents of the input image to explain which features were relevant for the classification. A recent approach in this direction has been presented by Hendricks et al. [14]. It is also conceivable that other metrics could be found for neural networks that can be used in the assurance argumentation, e.g. the extent by which weightings are affected by the training data or how well computations of the neural network have been covered by the performed tests.

- **Uncertainty calculation:** Although machine learning approaches offer a promising performance on perception tasks, false negatives (not detected objects), false positives (ghost objects), misclassification and mislocalisation may have safety critical consequences within the overall functional architecture of the system. One way to reduce the impact of these errors is considering uncertainties. Thereby, two types of uncertainties can be distinguished. Aleatoric uncertainty covers noise that is inherent in the observation (e.g. sensor or motion noise) and that cannot be reduced by increasing training data. In contrast, epistemic uncertainty captures uncertainty within the model (e.g. uncertainty of parameters) [18] and emphasises that assumptions on the model or the model itself may not represent the reality accurately enough. This has the effect that for a given input class (e.g. a particular vehicle, under similar environmental conditions), the system performs inconsistently within a particular range of error. A high classification uncertainty for a specific input class, belonging to epistemic uncertainty, could indicate inadequacies in the training data or sub-optimal parametrisation fo the neural network, etc. Uncertainty quantification can provide information that is used in plausibility and sensor fusion algorithms [18], thus improving the overall robustness and reliability of the subsequent trajectory planning tasks [26].
- **Black-box testing:** Due to the inherent restrictions of white-box approaches to verification of the trained function, a strong emphasis will need to be placed on black-box testing techniques. These techniques will include targeted testing of the software component containing the learned function including the use of systematically selected test data based on equivalence classes (see discussion of training data selection above). Based on advances in computer graphics realism as well as the possibility to generate labelled data with specific properties, the use of synthetically generated data may also play a role as demonstrated by Richter et al. [24]. This would imply the introduction of an additional assumption into the assurance case, that the synthetic data

would lead to test results that are representative of the operational environment. A combination of synthetic, real and manipulated real data, including a detailed analysis of the impact within the network due to the difference input types would be required. System-level vehicle integration tests will also need to form an essential part of the assurance case in order to validate all assumptions made during system development including whether or not sufficient understanding of critical scenarios have lead to an adequate training of the function. In reality, it will not be viable to provide assurance of the required level of system performance through driving test hours alone during the development phase. Therefore evidence will need to be provided for scenario coverage of the tests combined with statistical extrapolation techniques, field-based observations, component and integration tests (including simulation) as well as constructive safety measures.
- **Run-time measures:** An additional source of evidence for minimising the impact of functional insufficiencies will be run-time measures which make use of secondary channels, not used by the machine learning function.
  - *Run-time plausibility checks*: Plausibility checks on the outputs of the neural network could involve tracking results over time (e.g. objects detected in one frame should appear in contiguous frames, until out of view) or by comparing against alternative sensor inputs (e.g. radar or lidar reflections). Such plausibility checks may mitigate against insufficiencies that occur spontaneously for individual frames.
  - *Run-time monitoring of assumptions*: If certain assumptions regarding the operational distribution are determined to be critical, then they could also be monitored during run-time. Discrepancies between the distribution of objects detected at run-time and the assumptions could indicate either errors in the trained function or that the system is operating within a context for which it was not adequately trained. If such a situation is detected, appropriate actions for mitigating the effect of the discrepancy can be initiated. One approach in this direction are software safety cages [13].

6 Conclusions and Future Work

The main challenge for using machine learning algorithms for HAD is arguing an adequately low level of residual risk associated with functional insufficiencies resulting from imperfections of the used machine learning algorithms and the impossibility of testing all possible driving situations at design time. Such arguments are not currently supported by the relevant safety norms, most prominently ISO 26262. This paper proposed applying an assurance case approach to determine how such an argument could be formed based on "first principles" by decomposing the safety goals of the system into technical performance requirements on the machine learning function under explicit consideration of assumptions on the system (and components within the system context) environment. The assurance case would be completed by providing systematically derived and diverse evidence to support the claim that various causes of insufficiencies have been sufficiently mitigated against.

The assurance case structure presented in this paper raises several issues that require substantial future research activities. First, providing the necessary evidence for the assurance case requires the development of entirely new verification techniques including a demonstration of their effectiveness. This work will require advances in theoretical insights into machine learning as well as large scale experimental research to confirm the effectiveness of proposed measures. Further research will also include the application of dynamic safety cases for systems that apply decentralized, online reinforcement learning techniques, i.e., systems that continue to adapt their behaviour at runtime. These activities have to be integrated into a holistic system engineering approach that supports the structure of the assurance case. This technical research work needs to be complemented by activities within industry to form a consensus on risk evaluation and acceptable argumentation structures that would feed into future standards.

References

1. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in AI safety. arXiv e-prints, June 2016
2. Attenberg, J., Ipeirotis, P., Provost, F.: Beat the machine: challenging humans to find a predictive model's "unknown unknowns". ACM J. Data Inf. Qual. **1**(1), 1–17 (2014)
3. Bergenhem, C., Johansson, R., Söderberg, A., Nilsson, J., Tryggvesson, J., Törngren, M., Ursing, S.: How to reach complete safety requirement refinement for autonomous vehicles. Technical report, CARS 2015 - Critical Automotive applications: Robustness & Safety, September 2015, Paris, France (2015)
4. Binder, A., Bach, S., Montavon, G., Müller, K.R., Samek, W.: Layer-wise relevance propagation for deep neural network architectures. In: Kim, K., Joukov, N. (eds.) Information Science and Applications (ICISA). LNEE, vol. 376, pp. 913–922. Springer, Singapore (2016). doi:10.1007/978-981-10-0557-2_87
5. Castelvecchi, D.: Can we open the black box of AI? Nature **538**(7623), 20–23 (2016). http://www.nature.com/news/can-we-open-the-black-box-of-ai-1.20731
6. Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: learning affordance for direct perception in autonomous driving. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2722–2730. IEEE (2015)
7. Christmann, A., Steinwart, I.: Support Vector Machines. Springer, Heidelberg (2008)
8. Fawzi, A., Fawzi, O., Frossard, P.: Analysis of classifiers' robustness to adversarial perturbations. arXiv:1502.02590 (2015)
9. Feinberg, E.A., Shwartz, A. (eds.): International Series in Operations Research & Management Science, vol. 40. Springer, Heidelberg (2002)
10. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**(2), 131–163 (1997)
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
12. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv:1412.6572 (2015)

13. Heckemann, K., Gesell, M., Pfister, T., Berns, K., Schneider, K., Trapp, M.: Safe automotive software. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011. LNCS, vol. 6884, pp. 167–176. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23866-6_18
14. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 3–19. Springer, Cham (2016). doi:10.1007/978-3-319-46493-0_1. http://arxiv.org/abs/1603.08507
15. IEEE: IEEE standard adoption of ISO/IEC 15026–1 - systems and software engineering - systems and software assurance (2014)
16. ISO: ISO 26262: Road vehicles - functional safety (2011)
17. Kaufman, S., Rosset, S., Perlich, C.: Leakage in data mining: Formulation, detection, and avoidance. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 556–563. ACM (2011)
18. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? CoRR abs/1703.04977 (2017). http://arxiv.org/abs/1703.04977
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
20. Mohan, N., Törngren, M., Izosimov, V., Kaznov, V., Roos, P., Svahn, J., Gustavsson, J., Nesic, D.: Challenges in architecting fully automated driving; with an emphasis on heavy commercial vehicles. In: 2016 Workshop on Automotive Systems/Software Architectures (2016)
21. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, pp. 427–436 (2015)
22. Piziali, A.: Functional Verification Coverage Measurement and Analysis. Springer, Heidelberg (2008)
23. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
24. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. arXiv:1608.02192 (2016)
25. SAE: J3016, taxonomy and definitions for terms related to on-road motor vehicle automated driving systems (2013)
26. Tas, Ö.S., Kuhnt, F., Zöllner, J.M., Stiller, C.: Functional system architectures towards fully automated driving. In: 2016 IEEE Intelligent Vehicles Symposium. IEEE (2016)
27. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F. Dennison, D.: Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 28 (NIPS 2015) (2015)
28. Shalev-Shwartz, S., Shashua, A.: On the sample complexity of end-to-end training vs. semantic abstraction training. In: arXiv:1604.06915 (2016)
29. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: arXiv:1312.6034 (2014)
30. Kelly, T., Weaver, R.: The goal structuring notation - a safety argument notation. In: Proceedings of the DSN 2004 Workshop on Assurance Cases (2004)
31. Varshney, K.: Engineering safety in machine learning. ArXiv e-prints, January 2016