Development Methodologies for Safety Critical Machine Learning Applications in the Automotive Domain: A Survey

Martin Rabe[1], Stefan Milz[1,2], Patrick Mäder[1]
[1]Software Engineering for Safety-Critical Systems, Technische Universität Ilmenau, Germany.*
[2]Spleenlab GmbH, Saalburg-Ebersdorf, Germany.
{martin.rabe,stefan.milz,patrick.maeder}@tu-ilmenau.de, stefan.milz@spleenlab.ai

Abstract

Enabled by recent advances in the field of machine learning, the automotive industry pushes towards automated driving. The development of traditional safety-critical automotive software is subject to rigorous processes, ensuring its dependability while decreasing the probability of failures. However, the development and training of machine learning applications substantially differs from traditional software development. The processes and methodologies traditionally prescribed are unfit to account for specifics like, e.g., the importance of datasets for a development. We perform a systematic mapping study surveying methodologies proposed for the development of machine learning applications in the automotive domain. We map the identified primary publications to a general machine learning–based development process and preliminary assess their maturity. The reviews's goal is providing a holistic view of current and previous research contributing to ML-aware development processes and identifying challenges that need more attention. Additionally, we list methods, network architectures, and datasets used within these publications. Our meta-study identifies that model training and model V&V received by far the most research attention accompanied by the most mature evaluations. The remaining development phases, concerning domain specification, data management, and model integration, appear underrepresented and in need of more thorough research. Additionally, we identify and aggregate typically methods applied when developing automated driving applications like models, datasets and simulators showing the state of practice in this field.

*The work was financially supported by BMBF (Bundesministeriums für Bildung und Forschung) under the project "reDesigN" (support code 01IS18074E).

1. Introduction

Automated driving is an active field in both research and industry alike. It promises to decrease road accidents and increase mobility for people who can not drive themself. On the software side the key enabler of all this are recent advances in machine learning like object recognition[115], decision making[17] and planning[47].

However the development of software used in safety-critical applications like automotive applications is subject to a rigorous process. This is necessary to ensure the level of dependability we come to expect from our cars. For example, the international ISO 26262 [48] standard prescribes the software development process applied in the automotive industry. However, this standard was created for the development of traditional software which shows fundamental differences to the training of machine learning applications. Publications found and argue that, without modification, machine learning applications cannot be developed with the prescribed process [41, 57, 87, 43, 98]. This is a problem for all regulated domains that aim to utilize software containing machine learning applications.

In this paper, we aim to identify the current state of research in the field of software development for automated driving and how this research can be classified into a consistent development process for machine learning applications used in safety-critical applications, i.e., automated driving. More specifically, we overview current research on machine learning development processes intended to be used in safety-critical applications and propose a general classification of steps within these processes into phases. We then perform a systematic mapping study identifying research within the field of automated driving summarizing proposed methods per development phase and evaluating their maturity. Additionally, we identify methods, tools and datasets that are used throughout the primary publications. Our study aims to answer the following research questions:

RQ1 Which are the most used publication venues for the field of automated driving?
RQ2 How much research was done for each phase of the development process?
RQ3 How mature is the research of each part of the development process?
RQ4 What are the technologies used in the field of automated driving?

The rest of this paper is structured as follows: First we discuss related work in the field of studies in automated driving in Sec. 2. We then present core phases within a general machine learning–based development process in Sec. 3 and provide an overview about development processes to argue its validity. In Sec. 4 we perform the systematic mapping study to identify current research within the domain of automated driving, classify the research within our proposed development phases and answer our research questions. Possible threats to validity are discussed in Sec. 5. Finally we will conclude in Sec. 6.

2. Related Work

In this section, we discuss related meta-studies on the development of safe machine learning applications. Previous studies mainly focus on specific problems or identify and discuss open challenges in the area. Schwarting et el. [91] review previous work on planning and decision-making for automated driving. Borg et al. [11] review publications on verification and validation methods and analyze challenges for the automotive industry. Grigorescu et al. [39] perform a survey of deep learning techniques for scene perception, path planning, behavior arbitration and motion control and identified seven challenges for automated driving. The impact of artificial intelligence on automated driving was reviewed in Nascimento et al. [71]. Tahir et al. [99] review literature concerning testing and safety assurance. All these meta-studies where focussed on specific activities within a development process, while ours is the first to classify and overview methods holistically across the entire development process of machine learning-base software used in cars.

3. Processes for ML Developments

The development of machine learning applications shows substantial differences to traditional software development [65, 110, 112]. While traditional software is requirements-driven, machine learning applications are data-driven [52]. A development methodology that shall be applicable to dependable machine learning applications consisting of both, programmed code and trained models, needs to unify their individual development steps. In this section, we propose and discuss four core phases that the development of a dependable machine learning application should consist of. These phases are: (1) *operational domain specification*, (2) *data orchestration and preparation*, (3) *model training*, and (4) *model integration* (cp. Fig. 1). The aim of this rather coarse-grained phase classification is having a description model that allows us to later group primary publications identified in our review. A development would typically follow these phases in an incremental and iterative manner and each phase would likely consist of V&V activities that we only emphasize where relevant for the discussion of machine learning applications. The following subsections discuss each phase in detail. To argue the validity of theses phases, we then briefly discuss existing development methodologies for such systems and exemplarily assign the prescribed development steps of one methodology to the four proposed phases.

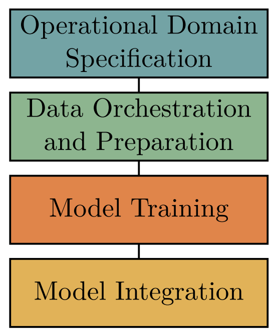


Figure 1: Core development phases of dependable machine learning applications.

3.1. Phase 1: Operational Domain Specification

The steps of this phase are performed to specify the operational design domain (ODD) of the target machine learning application. SAE J3016 [86] defines the ODD as:

> Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics.

The task of this phase is on one hand to specify all conditions under which the machine learning application is designed to function. These conditions contain, e.g., the road environment, behavior limitations of the vehicle like speed limit and state of the vehicle like load. On the other hand in this phase the final machine learning applications has to be verified to satisfy the specified behavior within this ODD.

3.2. Phase 2: Data Orchestration and Preparation

In the steps of this phase the main concern is training data. For machine learning applications training data represent the source code of traditional software components [52]. Which means it needs to satisfy the defined specification for a given task. As such, the performance of the trained model depends strongly on the quality of the utilized training data. For that reason training data has to be evaluated ensuring that it satisfies the required quality. Metrics for the quality of a dataset can, e.g., be coverage of the specified ODD, coverage of out-of-distribution data, coverage of corner-cases. Other important considerations are the modality of data to be analyzed, whether sufficient real data is available and whether synthetic data support the training, which data augmentations are advisable and permissible, and how to split meaningful test data.

3.3. Phase 3: Model Training

This phase is centered around the machine learning model. There are two main tasks in this phase: (1) training the model and (2) evaluating the model with the application of verification and validation (V&V) methods. The steps to create a model can again be divided into: (1.a) specifying the model's architecture, (1.b) implementing model and training process, and (1.c) training the model. During model specification, a data scientist typically not only decides on the model architecture but also on manifold structural parameters, such as the number of hidden layers or the chosen activation function. Furthermore, depending on the task to solve and available data the type of training has to be chosen, i.e., supervised, unsupervised, reinforcement or a combination thereof, as well as an appropriate objective function. Considerations regarding the training refer to choosing an optimization technique and making decisions on its hyper-parameters, such as learning rate, momentum, number of training epochs, batch size, parameter initialization and regularization techniques. Once the model is trained, verification and validation of the model is an important part of the development process utilizing formal verification and validation as well as testing approaches. Especially these V&V activities are an integral part of the development of dependable software and their realization for machine learning components is of great interest.

3.4. Phase 4: Model Integration

In the integration phase, model independent steps are taken to increase the dependability of the machine learning application. This is especially important for safety-critical systems. For such systems a malfunction during runtime can lead to disastrous outcomes. So measures for increasing the dependability of machine learning applications need to be applied that typically also necessitate accompanying model architectures that facilitate the desired integration of the trained model with traditional software and hardware.

3.5. Existing Development Processes

Various processes aiming to support the development of dependable machine learning applications, have been introduced in the last decades. In 1993, Peterson [79] proposed a process for developing machine learning applications proposing explicit V&V steps for data gathering and model design. In 1999, Rodvold [83] proposed a process inspired by the prominent waterfall model that comprises explicit development steps for data gathering and preprocessing; training and testing of the model; as well as for deploying the model. In 2019, Amershi et al. [5] derived a development process for machine learning applications in general, i.e., without explicit focus on dependability. The most recent process for the development of safety-critical components utilizing machine learning applications we could identify is EASA AI Task Force's and Daedalus AG's evolved W-Model published last year [22]. The W-Model prescribes nine development steps: requirement management, data management, learning management, model training, learning process verification, model implementation, inference model verification, data verification, and requirements verification (cp. Fig. 2). In order to argue the validity of our proposed four core development phases, Fig. 2 also shows how we map them to the W's development steps. We classified the requirements management and verification step within the phase of *operational domain specification*. The data management and verification step within the phase *data orchestration and preparation*. The steps classified within the phase *model training* are learning process management and verification as well as model training. The inference model verification step was classified within the phase of *model integration*. The step model implementation contains aspects of both the *model training* and *model integration* phase.

4. Method Study

In this section we present the systematic mapping study we performed to identify relevant research in the field of automated driving. First, we briefly describe the review process (cp. Sec. 4.1) and then, we present and discuss our findings in (cp. Sec. 4.2).

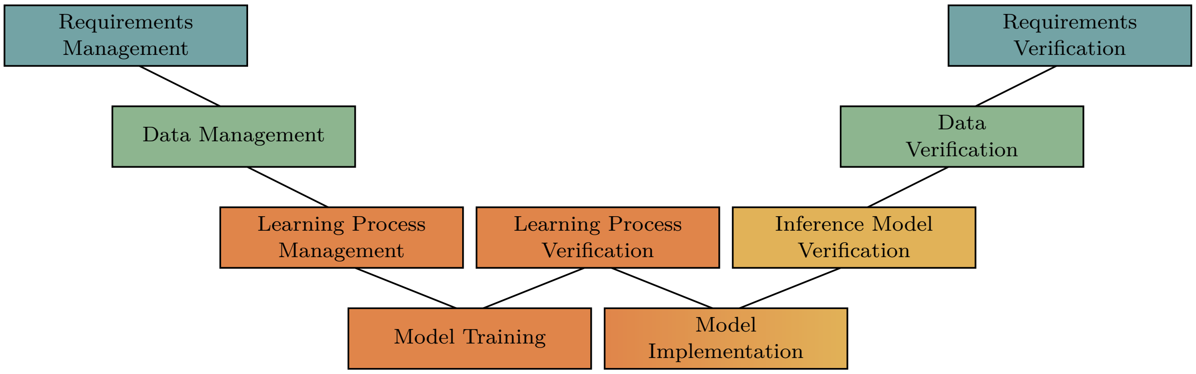


Figure 2: Classification of W model's process steps into the four development phases.

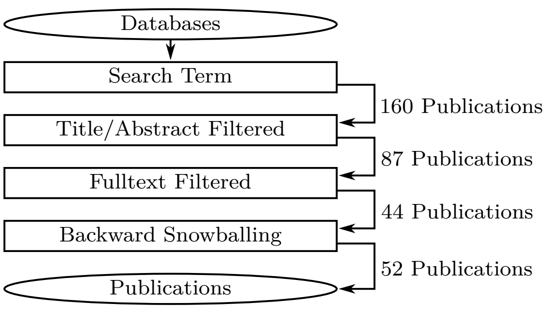


Figure 3: Review process illustrating the search and pruning of primary publications.

4.1. Study Process

To identify relevant research in the field of automated driving, we perform a systematic mapping study [78]. We restrict our focus to peer-reviewed, english publications published at workshops, conferences or journals. The study's goal is identifying which parts of the development process described in Sec. 3 are covered in current research and to what extend as well as identifying methods and datasets utilized across the primary publications. As such, we include publications which cover at least one of the phases introduced above in combination with a focus on automotive machine learning applications and, e.g., cover the following topics:

- hazard and risk analysis, safety case creation, and safety requirements elicitation [Phase 1];
- dataset generation and evaluation [Phase 2];
- model training and V&V [Phase 3];
- model integration architectures and runtime safety measures [Phase 4].

Based on this focus, we derived the following search term structure:

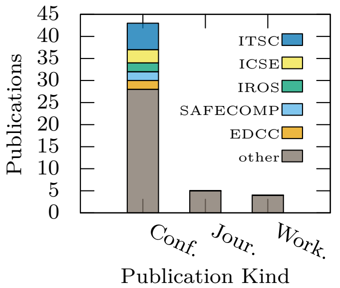
- `technology` $\wedge$ `property` $\wedge$ `domain`, where
  - `technology`: *machine learning* $\vee$ *deep learning* $\vee$ *artificial intelligence* $\vee$ *neural network* $\vee$ *reinforcement learning* $\vee$ *convolutional network* $\vee$ *data* $\vee$ *simulation* $\vee$ *verification* $\vee$ *testing* $\vee$ *requirements*
  - `property`: *safety* $\vee$ *assurance* $\vee$ *reliable* $\vee$ *dependable*
  - `domain`: *iso 26262* $\vee$ *autonomous driving* $\vee$ *autonomous vehicle* $\vee$ *automated driving* $\vee$ *driverless* $\vee$ *self-driving car*

We iteratively searched four digital libraries with the derived search term structure: ACM Digital Library, IEEExplore, Springer Link, and Google Scholar (cp. Fig. 3). In the first iteration, we entered the search term into the database and selected the first 40 publication retrieved by each digital library resulting in a total of 160 publications. In the second iteration, we removed duplicates and retained only relevant publications based on their title and abstract resulting in a set of 87. In the third iteration, we excluded publications based on their content. We exclude publications that propose methods focussed solely on hardware development rather than software; those that focus on cooperative driving, which we deem out of our scope; and those that are only applicable to lower automation levels (≤ 2) (cp. SAE J3016 [86]) focusing on collaborative interaction with the driver. This iteration yielded 44 publications conforming to all study criteria. As a final step, we performed backward snowballing [113] on the references of all identified publications. The goal is identifying further research relevant to our study. With this step we identified an additional 8 publications to a final total of 52 primary publications further discussed in this paper.

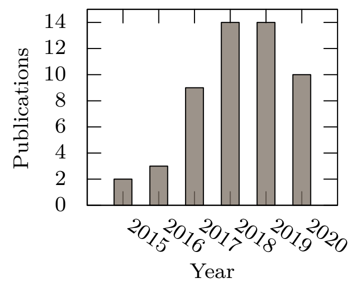
4.2. Results and Discussion

In this section, we discuss the 52 primary publications identified and classified into the development phases introduced in Sec. 3. Publications that propose methods contributing to multiple phases are represented multiple times in the respective discussions. Below, we discuss findings along our four research questions.

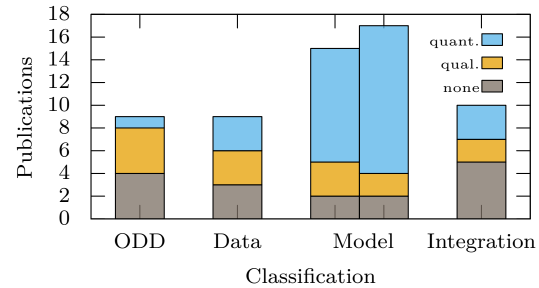
RQ1: publication venues. Fig. 4a and 4b show



(a) Publication kind and venue.

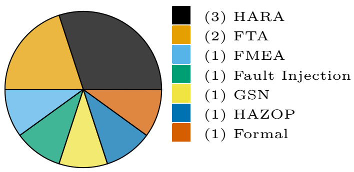


(b) Publications depending on year.

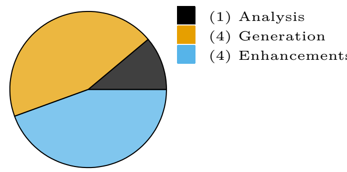


(c) Classification of the reviewed publications and their evaluation state.

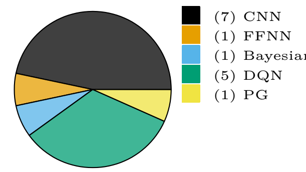
Figure 4: Overview of the reviewed primary publication identified within our study.



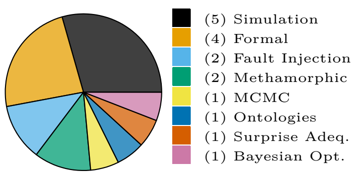
(a) Methodologies used for operational domain specification.



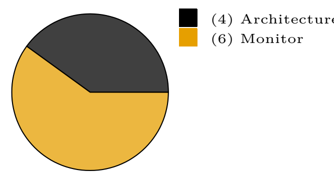
(b) Methodologies used for data orchestration and preparation.



(c) Methodologies used for model training.



(d) Methodologies used for model V&V.



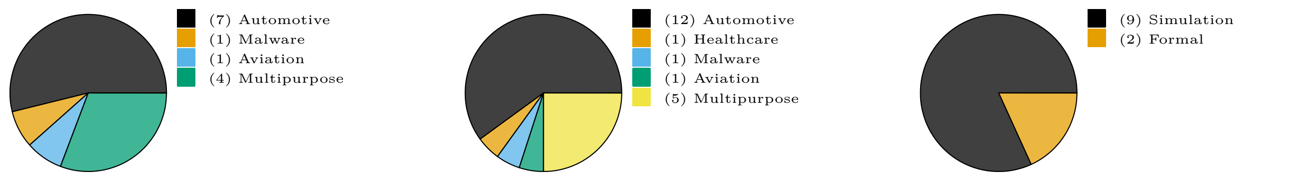
(e) Methodologies used for model integration.

Figure 5: Overview of used methodologies in the core development phases.

where and when primary publications have been published respectively. The *International Conference on Intelligent Transportation Systems (ITSC)* is the most popular publication venue (six primary publications); followed by the *International Conference on Software Engineering (ICSE)* (three primary publications); and the *International Conference on Intelligent Robots and Systems (IROS)*, the *International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, and the *European Dependable Computing Conference (EDCC)* (two primary publications each). Furthermore, we find that publication activity increased from about 2017. From these results, we derive two interesting observations: (1) research in this area span multiple research communities in the area of computer science, and (2) the majority of research is still mainly published at conferences indicating a fast pacing but not yet mature research topic.

RQ2: process phase focus. We find that 9 of the 52 primary publications focus on the domain specification phase, 9 focus on data orchestration and preparation, 32 focus on the machine learning model, and 10 focus on trained models' integration (cp. Fig. 4c). This distribution shows that a primary research focus is the training of the model and its validation being roughly three times more prominent than research into the other phases. Taking a closer look at this phase, we find that 15 primary publications are concerned with models and their training, while 17 publications focus on model V&V.

RQ3: research maturity. Of 9 primary publications that propose methods for the domain specification phase, 4 are proposals without evaluation, 4 are comprised by a qualitative evaluation, and merely 1 is supported by a quantitative evaluation. Of the 9 primary publications with focus on data orchestration



(a) Models used within primary publications.

(b) Datasets used within primary publications.

(c) Simulators used within primary publications.

Figure 6: Overview of tools used within all primary publications.

and preparation, 3 are proposals without evaluation, 3 are comprised by qualitative evaluation, and 3 are supported by a quantitative evaluation. Of the 32 primary publications focussing on model training and validation, 4 are proposals without evaluation, 5 are comprised by a qualitative evaluation, and 23 are supported by a quantitative evaluation. Of the 10 primary publications with focus on model integration, 5 are proposals without evaluation, 2 are comprised by a qualitative evaluation, and 3 are supported by a quantitative evaluation.

RQ4: proposed methods per phase. Below, we discuss prominent primary publications and their proposed methods grouped per development phase. Tables 1 to 2 show all primary publications along with a brief summary of their methodological contribution.

A prominent share of primary publications that we associate to the operational domain specification phase, propose evolved and adjusted methods to the needs of machine learning development. Fig. 5a shows a coarse-grain classification of proposed methods. Paul et al. [77], e.g., apply existing methodologies like FMEA and HAZOP to identify safety requirements of an experimental autonomous vehicle. Juez et al. [49] propose to leverage fault injection already within the early concept phase to derive safety goals and requirements, while Sini et al. [95] propose simulations for the initial specification. The applicability of the goal structuring notation (GSN) to support safety argumentation is demonstrated by Schmid et al. [90]. Girard-Satabin et al. [37] present formalisms to verify safety properties.

Fig. 5b groups publications of the data orchestration and preparation phase into analysis, generation, and enhancements. For example, publications propose methods to improve dataset quality with fuzzy inference to identify specific scenarios [75], or to automatically label lane markers in images [9]. Furthermore, research focusses on extending existing datasets by adding synthetic or simulated data derived from existing data, e.g., by using GANs [117] or by applying image transformations [16, 101]. We observe that these methods are often not the main focus of a publication but a by-product that is often not individually evaluated for its effects.

Primary publications focusing on the model training phase can be divided into those mainly concerning supervised and those mainly concerning reinforcement learning. No publication proposed an unsupervised learning approach. The dominant model architecture for supervised learning are CNNs, for reinforcement learning they are DQNs (cp. Fig. 5c). Holen et al. [44] propose a reinforcement learning approach which uses a CNN-based lane detection as reward function. McAllister et al. [68] propose and study Bayesian deep learning greatly improving both uncertainty estimation and interpretability of models. Other authors propose the use of attention architectures to reduce the complexity and to increase the performance of the training process [88, 17, 115]. Another distinction we identified within this phase is that some of the publications consider the process from sensor input to model prediction (aka driving decision) as an end-to-end process [31, 47], while others separate this into layers, e.g., sensing [3], perception [93] and decision making [17], thereby, gaining potentially more control over the inference process.

Fig. 5d gives a brief overview of methods proposed for the model V&V development phase. Examples of these methods are: fault injection [58, 18], simulation-based black-box testing [26], Markov Chain Monte Carlo algorithms [92], and formal methods-based forward reachability analysis [103]. Various other publications contribute methods to model testing, e.g., by systematically generating specific test cases extracted from real world data [109], by leveraging DNNs [117], by using neuron coverage [101], by analyzing parameter correlations [100], by agent-based modeling [14], or by considering ontologies [56].

Fig. 5e groups publications of the model integration phase into those that focus on model deployment and those that propose techniques for runtime monitoring of models aiming to increase the dependability of machine learning applications by runtime safety measures. Examples of the earlier are, e.g., layered architecture-based observe, orient, decide, act [8] or sensing, perception,

Ref	Contribution
[42]	Proposes changes to safety assessment.
[58]	Proposes a phased development by progressively extending the operational domain, to help with identifying corner cases.
[49]	Proposes Sabotage, a framework for using fault injection in concept phase for hazard identification, safety goal refinement, definition of Fault Tolerant Time Interval and establishing safe state.
[1]	Proposes methodology for conducting safety and risk assessment.
[95]	Perform HARA through use of vehicle-level simulators to test initial specification.
[90]	Safety argumentation for fail-operational driving systems and provides a continuous example for the proposed safety argumentation discussed in context of ISO 26262.
[37]	Presents formalism for formally describing and verifying safety properties based on simulation data.
[38]	Propose adaptation of hazard identification for ADS.
[77]	Proposes assessment of safety requirements by applying FMEA and HAZOP. *
[75]	Proposes the analysis of datasets using fuzzy inference system. Identified scenarios in dataset in which 2 second rule was violated.
[9]	Proposes methodology to automatically label lane markers in images.
[84]	Presents framework based on statistical learning and model-based analysis to assess safety impact of AD functions and to identify relevant scenarios.
[82]	Discusses data set criticality and proposes usage of synthetic and simulation data to enhance existing datasets.
[117]	Proposes using a GAN for synthesizing driving scenes with various weather conditions.
[51]	Presents a methodology to model GPS errors with the help of auto-regressive models with Gaussian mixture model distribution.
[101]	Increase dataset coverage by synthetic images. Utilize nine different image transformations. *
[16]	Applied random majority under sampling, normalization of input data and augmentation (flipping). *
[97]	Provides labeled scenarios. *
[68]	Proposes use of Bayesian deep learning to solve problems in uncertainty and interpretability,
[88]	Propose framework for deep reinforcement learning with attention model.
[108]	Proposes hybrid CNN/SVM system for object recognition.
[82]	Discusses CNNs' interpretability via visualisations using Picasso framework.
[31]	Deep reinforcement learning using DQN for urban navigation.
[47]	Use DQN for navigating partially occluded intersections.
[33]	Proposes implementation to capture uncertainty in 3D LiDAR point clouds.
[16]	Use a deep CNN-LSTM for characterizing driving environment and controls movement.
[17]	Proposes lane change via reinforcement learning. Leverages CNN (Recognition), and attention for better Interpretability.
[115]	Improve object recognition for sparse LiDAR point clouds via ground-aware attention model.
[55]	Proposes methodology to adaptively restrict action space with DFSM according to current driving situation and combine this with DQN.
[62]	Proposes end-to-end implementation of multi-objective deep reinforcement learning.
[93]	Proposes single shot multi-box detector (SSD) to detect non-vehicles and vehicles, with an embedded model for lane detection.
[3]	Proposes a DNN for sensor-less brake state estimation.
[44]	Proposes reinforcement learning application with lane detection as reward function.
[58]	Proposes fault injection as a useful tool for validation.
[92]	Validate machine learning applications via Markov Chain Monte Carlo algorithm with subsampling.
[109]	Proposes to test and validate AD components with two step methodology: (1) feed real world data into machine learning system which will classify different functionality (2) deduce test scenarios from available data.
[56]	Proposes test framework based on ontologies.
[117]	Unsupervised DNN-based framework utilizing metamorphic testing for testing ADS and online validation.
[101]	Systematic testing utilizing neuron coverage and test case generation.
[111]	Proposes formal verification of safety properties in two steps: (1) symbolic linear relaxation (2) direct constraint refinement.
[116]	Propose formal operational verification approach based on stochastic hybrid automata (SHA).
[18]	Binary-search like fault injection for finding safety-critical parts in ML applications to measure resilience.
[53]	Evaluate test quality with surprise adequacy for deep learning (SADL).
[35]	Test case identification with Bayesian optimization to drive the system to violate its safe boundaries.
[103]	Safety verification via forward reachability analysis applicable for reinforcement learning.
[19]	Proposes verification framework for direct perception neural networks.
[72]	Introduces a sensitivity analysis approach for developing and validating radar simulation. Goal is to identify radar sensor effect with the greatest impact for system under test.
[26]	Simulation-based black-box testing. Simulate Oracle vehicle and EGO vehicle to test the decisions of the EGO vehicles behavior.
[14]	Present methodology of guided testing via agent-based modeling.
[100]	Structured test case generation via parameter correlation.

Table 1: Overview of publications contributing to the operational domain specification, data orchestration and preparation and model training and V&V phase. Contributions marked with * are not the main focus of the publication, but a by-product that is not individually evaluated.

Ref	Contribution
[8]	Proposes functional architecture based on Observe, Orient, Decide, Act (OODA) loop.
[58]	Proposes failover strategies as redundant paths for machine learning based components.
[2]	Proposes safety supervisor to avoid specified critical combinations of vehicle behavior and runtime situation.
[64]	Proposes architecture pattern for automated driving: Safety Channel. The goal is to provide a strategy to ensure safety in presence of functional errors. ASIL decomposition for components.
[70]	Proposes architecture based on Sensing, Perception, Decision, Planning and Action.
[104]	Proposes architecture for dynamic safety management during runtime.
[34]	Proposes a CNN for risk assessment during runtime.
[77]	Proposes a safety-bag architecture to reduce risk during runtime.
[54]	Proposes wrapper for uncertainty estimates during runtime.
[97]	SafeOracle: Detect dangerous situations at runtime via confidence.

Table 2: Overview of publications contributing to the model integration phase.

decision, planning, acting [70], while examples of the later group are safety-bag concepts [77], CNN-base dynamic risk assessments [34], uncertainty wrapper to assess decisions (aka predictions) of the model [54], and a SafeOracle termed technology that aims to detect dangerous driving situations [97].

A majority of primary publications evaluates their proposed methods on existing datasets. We divide these into datasets specific for a domain and more general ones (cp. Fig. 6b). The majority of these datasets comprises domain-specific data, twelve focus on automated driving [66, 4, 13, 23, 28, 45, 96, 36, 73, 74, 105, 25], one is aviation-specific [50], one is healthcare-specific [40], and one originates from malware detection [6]. Additionally, five mostly well-known multipurpose datasets are used [63, 12, 20, 24, 69]. In addition to these benchmark datasets, publications often apply their methods to driving scenarios (cp. Fig. 6c). Five publications perform this evaluation in real-world implementations of their method [31, 19, 64, 70, 77], while all others providing implementations utilize simulations. These simulations mostly refer to driving environments [27, 29, 46, 76, 89, 59, 102, 106, 80] and only two publications use formal simulations [67, 107].

We also found that primary publications typically benchmark their work against previously published prominent models (cp. Fig. 6a). These models can be divided into those specific to a certain domain and those that are general, e.g., for computer vision tasks [60, 61, 114, 94]. We identified a total of seven models specifically used for automated driving [7, 15, 21, 10, 30, 81, 85], one focussed on the aviation domain [50], and one focussed on malware detection on the android platform [6].

5. Threats to Validity

In this section we will discuss threats to the validity of our study and structure our discussion into commonly accepted categories [32]. **Internal Validity.** Our method study followed a well-defined and accepted study procedure [78]. We applied justified inclusion and exclusion criteria for primary publications and the classification of publications was discussed among the authors in case of uncertainty. **External Validity.** We included four databases and peer-reviewed publications to increase generalizability of our results. However, since publications considered grey literature were excluded from our meta-study there is the possibility that we miss proposed methods or do not correctly report the distribution of applied methods among all research.

6. Conclusion

In this paper we contribute to a continuous development process for machine learning applications used in safety-critical applications, i.e. automated driving. We proposed a general machine learning–based development process with the core phases of: *operational domain specification*, *data orchestration and preparation*, *model training* and *model integration*. To argued the validity of these phases we gave an overview of existing machine learning–based development processes with a focus on safety-critical applications. We furthermore presented our findings gathered through a systematic mapping study in the field of automated driving. Our findings are: that the phase concerning the model training is by far the most researched one. Phases concerning domain specification, data and integration are still in development. We identified that not many publications are published in journals which indicates that the field is still in a state of flux. However we could identify a number of tools for the specific applications in the domain of automated driving like models, datasets and simulators. In future work we want to extend our survey to include publications developed for other safety-critical areas such as health care and aviation since we did find a number of cross-references to publications focusing on these domains and also take a closer look at findings within grey literature.

References

[1] M. Adedjouma, G. Pedroza, and B. Bannour. Representative safety assessment of autonomous vehicle for public transportation. In *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*, pages 124–129, 2018. 7

[2] R. Adler, P. Feth, and D. Schneider. Safety engineering for autonomous vehicles. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 200–205, 2016. 8

[3] M. Al-Sharman, D. Murdoch, D. Cao, C. Lv, Y. Zweiri, D. Rayside, and W. Melek. A sensorless state estimation for a safety-oriented cyber-physical system in urban driving: Deep learning approach. *IEEE/CAA Journal of Automatica Sinica*, 8(1):169–178, 2021. 6, 7

[4] M. Aly. Real time detection of lane markers in urban streets. In *2008 IEEE Intelligent Vehicles Symposium*, pages 7–12, 2008. 8

[5] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: A case study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '19, page 291–300. IEEE Press, 2019. 3

[6] Daniel Arp, Michael Spreitzenbarth, Hugo Gascon, and Konrad Rieck. Drebin: Effective and explainable detection of android malware in your pocket, 2014. 8

[7] Autumn Model. `https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/autumn`. 8

[8] Sagar Behere and Martin Törngren. A functional architecture for autonomous driving. In *Proceedings of the First International Workshop on Automotive Software Architecture*, WASA '15, page 3–10, New York, NY, USA, 2015. Association for Computing Machinery. 6, 8

[9] K. Behrendt and J. Witt. Deep learning lane marker segmentation from automatically generated labels. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 777–782, 2017. 6, 7

[10] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016. 8

[11] Markus Borg, Cristofer Englund, Krzysztof Wnuk, Boris Duran, Christoffer Levandowski, Shenjian Gao, Yanwen Tan, Henrik Kaijser, Henrik Lönn, and Jonas Törnqvist. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. *Journal of Automotive Software Engineering*, 1:1–19, 2019. 2

[12] Caltech-101 Dataset. `http://www.vision.caltech.edu/Image_Datasets/Caltech101/`. 8

[13] Caltech Pedestrian Dataset. `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/`. 8

[14] G. Chance, A. Ghobrial, S. Lemaignan, T. Pipe, and K. Eder. An agency-directed approach to test generation for simulation-based autonomous vehicle verification. In *2020 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 31–38, 2020. 6, 7

[15] Chauffeur Model. `https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/chauffeur`, 2016. 8

[16] Sikai Chen, Yue Leng, and Samuel Labi. A deep learning algorithm for simulating autonomous driving considering prior knowledge and temporal information. *Computer-Aided Civil and Infrastructure Engineering*, 35(4):305–321, 2020. 6, 7

[17] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3697–3703, 2019. 1, 6, 7

[18] Zitao Chen, Guanpeng Li, Karthik Pattabiraman, and Nathan DeBardeleben. Binfi: An efficient fault injector for safety-critical machine learning systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19, New York, NY, USA, 2019. Association for Computing Machinery. 6, 7

[19] C. H. Cheng, C. H. Huang, T. Brunner, and V. Hashemi. Towards safety verification of direct perception neural networks. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1640–1643, 2020. 7, 8

[20] CIFAR Dataset. `https://www.cs.toronto.edu/~kriz/cifar.html`. 8

[21] comma.ai's Steering Model. `https://github.com/commaai/research`, 2016. 8

[22] Concepts of Design Assurance for Neural Networks, 2020. 3

[23] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016. 8

[24] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 8

[25] DF-3D Dataset. `https://datafountain.cn/competitions/314/details/rank?sch=1367&page=1&type=a.` 8

[26] A. Djoudi, L. Coquelin, and R. Régnier. A simulation-based framework for functional testing of automated driving controllers. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020. 6, 7

[27] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR, 13–15 Nov 2017. 8

[28] Driving Dataset. https://github.com/SullyChen/driving-datasets. 8

[29] Dynacar Simulator. http://dynacar.es/en/home.php. 8

[30] Epoch Model. https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/cg23, 2016. 8

[31] A. R. Fayjie, S. Hossain, D. Oualid, and D. Lee. Driverless car: Autonomous driving using deep reinforcement learning in urban environment. In *2018 15th International Conference on Ubiquitous Robots (UR)*, pages 896–901, 2018. 6, 7, 8

[32] Robert Feldt and Ana Magazinius. Validity threats in empirical software engineering research - an initial survey. In *Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering*, pages 374–379, 01 2010. 8

[33] D. Feng, L. Rosenbaum, and K. Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273, 2018. 7

[34] Patrik Feth, Mohammed Naveed Akram, René Schuster, and Oliver Wasenmüller. Dynamic risk assessment for vehicles of higher automation levels by deep learning. In Barbara Gallina, Amund Skavhaug, Erwin Schoitsch, and Friedemann Bitsch, editors, *Computer Safety, Reliability, and Security*, pages 535–547, Cham, 2018. Springer International Publishing. 8

[35] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings. Identification of test cases for automated driving systems using bayesian optimization. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1961–1967, 2019. 7

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. pages 3354–3361, 05 2012. 8

[37] Julien Girard-Satabin, Guillaume Charpiat, Zakaria Chihani, and Marc Schoenauer. CAMUS: A Framework to Build Formal Specifications for Deep Perception Systems Using Simulators. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain, June 2020. 6, 7

[38] R. Graubohm, T. Stolte, G. Bagschik, and M. Maurer. Towards efficient hazard identification in the concept phase of driverless vehicle development. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1297–1304, 2020. 7

[39] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. 2

[40] Haberman's Survival Dataset. https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival. 8

[41] Karl Heckemann, Manuel Gesell, Thomas Pfister, Karsten Berns, Klaus Schneider, and Mario Trapp. Safe automotive software. In Andreas König, Andreas Dengel, Knut Hinkelmann, Koichi Kise, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 167–176, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 1

[42] T. Helmer, L. Wang, K. Kompass, and R. Kates. Safety performance assessment of assisted and automated driving by virtual experiments: Stochastic microscopic traffic simulation as knowledge synthesis. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2019–2023, 2015. 7

[43] Jens Henriksson, Markus Borg, and Cristofer Englund. Automotive safety and machine learning: Initial results from a study on how to adapt the iso 26262 safety standard. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, SEFAIS '18, page 47–49, New York, NY, USA, 2018. Association for Computing Machinery. 1

[44] Martin Holen, Rupsa Saha, Morten Goodwin, Christian W. Omlin, and Knut Eivind Sandsmark. Road detection for reinforcement learning based autonomous car. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System*, ICISS 2020, page 67–71, New York, NY, USA, 2020. Association for Computing Machinery. 6, 7

[45] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013. 8

[46] IPG Carmaker Simulator. https://ipg-automotive.com/de/simulationsolutions/carmaker/. 8

[47] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2034–2039, 2018. 1, 6, 7

[48] ISO 26262-6 Road vehicles — Functional safety — Part 6: Product development at the software level, 2018. 1

[49] G. Juez, E. Amparan, R. Lattarulo, J. P. Rastelli, A. Ruiz, and H. Espinoza. Safety assessment of automated vehicle functions by simulation-based fault injection. In *2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 214–219, 2017. 6, 7

[50] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer. Policy compression for aircraft collision avoidance systems. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–10, 2016. 8

[51] Erik Karlsson and Nasser Mohammadiha. A data-driven generative model for gps sensors for autonomous driving. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, SEFAIS '18, page 1–5, New York, NY, USA, 2018. Association for Computing Machinery. 7

[52] F. Khomh, B. Adams, J. Cheng, M. Fokaefs, and G. Antoniol. Software engineering for machine-learning applications: The road ahead. *IEEE Software*, 35(5):81–84, 2018. 2, 3

[53] J. Kim, R. Feldt, and S. Yoo. Guiding deep learning system testing using surprise adequacy. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1039–1049, 2019. 7

[54] Michael Kläs and Lena Sembach. Uncertainty wrappers for data-driven models. In Alexander Romanovsky, Elena Troubitsyna, Ilir Gashi, Erwin Schoitsch, and Friedemann Bitsch, editors, *Computer Safety, Reliability, and Security*, pages 358–364, Cham, 2019. Springer International Publishing. 8

[55] Patrick Klose and Rudolf Mester. Simulated autonomous driving in a realistic driving environment using deep reinforcement learning and a deterministic finite state machine. In *Proceedings of the 2nd International Conference on Applications of Intelligent Systems*, APPIS '19, New York, NY, USA, 2019. Association for Computing Machinery. 7

[56] F. Klueck, Y. Li, M. Nica, J. Tao, and F. Wotawa. Using ontologies for test suites generation for automated and autonomous driving functions. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 118–123, 2018. 6, 7

[57] A. Knauss, J. Schroder, C. Berger, and H. Eriksson. Software-related challenges of testing automated vehicles. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 328–330, 2017. 1

[58] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016. 6, 7, 8

[59] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo - simulation of urban mobility. 2012. 8

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. 8

[61] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1990. 8

[62] Changjian Li and Krzysztof Czarnecki. Urban driving with multi-objective deep reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 359–367, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. 7

[63] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 8

[64] Y. Luo, A. K. Saberi, T. Bijlsma, J. J. Lukkien, and M. van den Brand. An architecture pattern for safety critical automated driving applications: Design and analysis. In *2017 Annual IEEE International Systems Conference (SysCon)*, pages 1–7, 2017. 8

[65] Lucy Ellen Lwakatare, Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In *International Conference on Agile Software Development*, pages 227–243. Springer, Cham, 2019. 2

[66] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition — how far are we from the solution? pages 1–8, 08 2013. 8

[67] MATLAB Simulator. https://www.mathworks.com/. 8

[68] Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4745–4753, 2017. 6, 7

[69] MNIST Dataset. https://deepai.org/dataset/mnist. 8

[70] Farzeen Munir, Shoaib Azam, Muhammad Ishfaq Hussain, Ahmed Muqeem Sheri, and Moongu Jeon. Au-

tonomous vehicle: The architecture aspect of self driving car. In *Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing*, SSIP 2018, page 1–5, New York, NY, USA, 2018. Association for Computing Machinery. 8

[71] A. M. Nascimento, L. F. Vismari, C. B. S. T. Molina, P. S. Cugnasca, J. B. Camargo, J. R. d. Almeida, R. Inam, E. Fersman, M. V. Marquezini, and A. Y. Hata. A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. *IEEE Transactions on Intelligent Transportation Systems*, 21(12):4928–4946, 2020. 2

[72] A. Ngo, M. P. Bauer, and M. Resch. A sensitivity analysis approach for evaluating a radar simulation for virtual testing of autonomous driving functions. In *2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 122–128, 2020. 7

[73] NGSIM I-80 Dataset. https://github.com/cemsaz/NGSIM-trajectories. 8

[74] NGSIM US-101 Dataset. https://github.com/Rim-El-Ballouli/NGSIM-US-101-trajectory-dataset-smoothing. 8

[75] M. O'Brien, K. Neubauer, J. Van Brummelen, and H. Najjaran. Analysis of driving data for autonomous vehicle applications. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3677–3682, 2017. 6, 7

[76] OpenAI carracing-v0 Simulator. https://gym.openai.com/envs/CarRacing-v0/. 8

[77] C. Paul, L. Benjamin, S. Walter, and M. Brini. Validation of safety necessities for a safety-bag component in experimental autonomous vehicles. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 33–40, 2018. 6, 7, 8

[78] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015. 4, 8

[79] Gerald E. Peterson. Foundation for neural network verification and validation. In Dennis W. Ruck, editor, *Science of Artificial Neural Networks II*, volume 1966, pages 196 – 207. International Society for Optics and Photonics, SPIE, 1993. 3

[80] PTV Vissim Simulator. https://www.ptvgroup.com/en/solutions/products/ptv-vissim. 8

[81] Rambo Model. https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/rambo, 2016. 8

[82] Qing Rao and Jelena Frtunikj. Deep learning for self-driving cars: Chances and challenges. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, SEFAIS '18, page 35–38, New York, NY, USA, 2018. Association for Computing Machinery. 7

[83] D. M. Rodvold. A software development process model for artificial neural networks in critical applications. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, volume 5, pages 3317–3322 vol.5, 1999. 3

[84] C. Roesener, J. Hiller, H. Weber, and L. Eckstein. How safe is automated driving? human driver models for safety performance assessment. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2017. 7

[85] rwightman Model. https://github.com/udacity/self-driving-car/tree/master/steering-models/evaluation. 8

[86] SAE J3016, 2018. 2, 4

[87] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An analysis of iso 26262: Machine learning and safety in automotive software. In *SAE Technical Paper*. SAE International, 04 2018. 1

[88] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017. 6, 7

[89] SCANeR Studio Simulator. https://www.avsimulation.com/solutions/. 8

[90] T. Schmid, S. Schraufstetter, S. Wagner, and D. Hellhake. A safety argumentation for fail-operational automotive systems in compliance with iso 26262. In *2019 4th International Conference on System Reliability and Safety (ICSRS)*, pages 484–493, 2019. 6, 7

[91] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018. 2

[92] Weijing Shi, Mohamed Baker Alawieh, Xin Li, Huafeng Yu, Nikos Arechiga, and Nobuyuki Tomatsu. Efficient statistical validation of machine learning systems for autonomous driving. In *Proceedings of the 35th International Conference on Computer-Aided Design*, ICCAD '16, New York, NY, USA, 2016. Association for Computing Machinery. 6, 7

[93] Shubham, M. Reza, S. Choudhury, J. K. Dash, and D. S. Roy. An ai-based real-time roadway-environment perception for autonomous driving. In *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, pages 1–2, 2020. 6, 7

[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 8

[95] J. Sini, M. Violante, V. Dodde, R. Gnaniah, and L. Pecorella. A novel simulation-based approach for iso 26262 hazard analysis and risk assessment. In *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 253–254, 2019. 6, 7

[96] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. Selected Papers from IJCNN 2011. 8

[97] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. Misbehaviour prediction for autonomous driving systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, page 359–371, New York, NY, USA, 2020. Association for Computing Machinery. 7, 8

[98] Hamid Tabani, Leonidas Kosmidis, Jaume Abella, Francisco J. Cazorla, and Guillem Bernat. Assessing the adherence of an industrial autonomous driving framework to iso 26262 software guidelines. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, New York, NY, USA, 2019. Association for Computing Machinery. 1

[99] Z. Tahir and R. Alexander. Coverage based testing for v v and safety assurance of self-driving autonomous vehicles: A systematic literature review. In *2020 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 23–30, 2020. 2

[100] S. Thal, H. Znamiec, R. Henze, H. Nakamura, H. Imanaga, J. Antona-Makoshi, N. Uchida, and S. Taniguchi. Incorporating safety relevance and realistic parameter combinations in test-case generation for automated driving safety assessment. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020. 6, 7

[101] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, page 303–314, New York, NY, USA, 2018. Association for Computing Machinery. 6, 7

[102] Torcs Simulator. `https://sourceforge.net/projects/torcs/`. 8

[103] Hoang-Dung Tran, Feiyang Cai, Manzanas Lopez Diego, Patrick Musau, Taylor T. Johnson, and Xenofon Koutsoukos. Safety verification of cyber-physical systems with reinforcement learning control. *ACM Trans. Embed. Comput. Syst.*, 18(5s), Oct. 2019. 6, 7

[104] M. Trapp, D. Schneider, and G. Weiss. Towards safety-awareness and dynamic safety management. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 107–111, 2018. 8

[105] Udacity Dataset. `https://github.com/udacity/self-driving-car/tree/master/datasets`. 8

[106] Udacity Simulator. `https://github.com/udacity/self-driving-car-sim`. 8

[107] Uppaal Simulator. `https://uppaal.org/`. 8

[108] Ayşegül Uçar, Yakup Demir, and Cüneyt Güzeliş. Object recognition and detection with deep learning for autonomous driving applications. *SIMULATION*, 93(9):759–769, 2017. 7

[109] H. J. Vishnukumar, B. Butting, C. Müller, and E. Sax. Machine learning and deep neural network — artificial intelligence core for lab and real-world test and validation for adas and autonomous vehicles: Ai for efficient and quality test and validation. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 714–721, 2017. 6, 7

[110] Z. Wan, X. Xia, D. Lo, and G. C. Murphy. How does machine learning change software development practices? *IEEE Transactions on Software Engineering*, pages 1–1, 2019. 2

[111] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 7

[112] H. Washizaki, H. Uchida, F. Khomh, and Y. Guéhéneuc. Studying software engineering patterns for designing machine learning systems. In *2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP)*, pages 49–495, 2019. 2

[113] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, New York, NY, USA, 2014. Association for Computing Machinery. 4

[114] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018. 8

[115] Jian Wu, Jianbo Jiao, Qingxiong Yang, Zheng-Jun Zha, and Xuejin Chen. Ground-aware point cloud semantic segmentation for autonomous driving. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 971–979, New York, NY, USA, 2019. Association for Computing Machinery. 1, 6, 7

[116] B. Xu, Q. Li, T. Guo, Y. Ao, and D. Du. A quantitative safety verification approach for the decision-making process of autonomous driving. In *2019 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 128–135, 2019. 7

[117] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 132–142, 2018. 6, 7