


Review

A Survey of Open-Source Autonomous Driving Systems and Their Impact on Research

Nourdine Aliane 

Department of Engineering, Universidad Europea de Madrid, Villaviciosa de Odón, 28670 Madrid, Spain; nourdine.aliane@universidadeuropea.es

Abstract: Open-source autonomous driving systems (ADS) have become a cornerstone of autonomous vehicle development. By providing access to cutting-edge technology, fostering global collaboration, and accelerating innovation, these platforms are transforming the automated vehicle landscape. This survey conducts a comprehensive analysis of leading open-source ADS platforms, evaluating their functionalities, strengths, and limitations. Through an extensive literature review, the survey explores their adoption and utilization across key research domains. Additionally, it identifies emerging trends shaping the field. The main contributions of this survey include (1) a detailed overview of leading open-source platforms, highlighting their strengths and weaknesses; (2) an examination of their impact on research; and (3) a synthesis of current trends, particularly in interoperability with emerging technologies such as AI/ML solutions and edge computing. This study aims to provide researchers and practitioners with a holistic understanding of open-source ADS platforms, guiding them in selecting the right platforms for future innovation.

Keywords: autonomous vehicle; autonomous driving; self-driving; automated vehicle; connected vehicle; open-source autonomous driving systems



Academic Editors: Vasco N. G. J. Soares, Robin Haunschild, Giorgio Maria Di Nunzio, Paulo Quaresma, Luigi Laura and Marcin Paprzycki

Received: 2 March 2025

Revised: 3 April 2025

Accepted: 9 April 2025

Published: 17 April 2025

Citation: Aliane, N. A Survey of Open-Source Autonomous Driving Systems and Their Impact on Research. *Information* **2025**, *16*, 317. <https://doi.org/10.3390/info16040317>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous driving systems (ADS) are the result of a combination of several technologies, such as environmental perception and decision making, which enable vehicles to operate safely without human intervention. Their significance lies in their potential to enhance road safety, improve mobility, and provide better accessibility for individuals with disabilities [1]. As technology continues to advance, ADS are set to play a crucial role in shaping the future of smart cities and sustainable transportation ecosystems [2].

Autonomous driving is classified by the Society of Automotive Engineers (SAE) into six levels [3] based on the amount of required driver intervention and attentiveness. Level 0 refers to vehicles with no automation, and drivers are responsible for all aspects of driving. Level 1 introduces basic driver assistance, such as adaptive cruise control or lane-keeping assistance, but the driver must constantly monitor the environment. Level 2 combines some functions, such as steering, acceleration, and braking, into partial automation, though the driver is ready to take over at any moment. Level 3 marks the beginning of conditional automation, and the vehicle can manage all driving tasks under specific conditions, yet drivers must be prepared to intervene when requested. Level 4 represents high automation, enabling vehicles to operate in predefined environments and conditions, though their capabilities may be limited to certain scenarios. Finally, Level 5 signifies full automation, where vehicles can handle all driving tasks in any environment and conditions, eliminating total human involvement [4,5].

The evolution of ADS platforms has been driven by three key approaches: global research and development (R&D), proprietary systems, and open-source initiatives. Global R&D efforts have laid the groundwork for advancements in autonomy, with pivotal contributions from programs such as the DARPA Grand Challenge [6], which established critical benchmarks in the field. In Europe, institutions like the Technical University of Munich have made significant progress in AV verification and cooperative driving systems [7], while the PEGASUS project has advanced validation methods and safety standards for Level 3–4 autonomy [8]. Meanwhile, Japan's Autoware [9] and China's Baidu Apollo [10] have played instrumental roles in shaping urban mobility solutions and fostering open ecosystems for Autonomous Vehicle (AV) development.

Proprietary systems developed by automakers and technology companies have also been a major force in ADS innovation. Automakers such as Tesla (Autopilot), Cruise (backed by General Motors), and BMW (partnered with Mobileye) have introduced commercially viable autonomous technologies [4]. Similarly, IT companies like Waymo (Alphabet), Apple (Project Titan), and Zoox (Amazon) have leveraged cutting-edge research to push the boundaries of self-driving capabilities [5]. Focused on commercialization, these systems rely on closed-source software, and they are often inaccessible to the broader public, prioritizing exclusivity and market leadership over public accessibility.

In contrast, open-source ADS initiatives have cultivated a collaborative environment where developers can freely contribute to and refine autonomous technologies. Early projects such as Duckietown [11], DeepPicar [12], DonkeyCar, and Udacity's self-driving car initiative [13] were primarily designed for education and small-scale experimentation, though their real-world applicability was limited. Other efforts, including PolySync [14] and Ford's OpenXC [15], sought to standardize vehicle data and promote open-source design but were ultimately discontinued due to insufficient adoption.

Today, mature open-source ADS platforms like Autoware [9], Baidu Apollo [10], NVIDIA Drive [16], and OpenPilot [17] dominate the landscape, offering robust frameworks for AV development. These platforms have democratized access to ADS technology, enabling global collaboration and accelerating progress in the field.

While open-source ADS have accelerated the AV field, their heterogeneous architectures, varying technical capabilities, and specialized application domains present significant research challenges. Key issues requiring investigation include evaluating their adoption and impact on research and comparative strengths and weaknesses, particularly in interoperability with emerging technologies such as AI/ML solutions and edge computing. Thus, a comprehensive survey is needed to evaluate these ADS platforms regarding these challenges and ultimately provide researchers and developers with insights to guide their selection and utilization of open-source ADS for future innovation.

1.1. Related Work

Several recent survey studies have comprehensively explored advancements in autonomous driving technology, with a focus on different aspects, such as perception and decision making, simulation tools, or integration of Artificial Intelligence (AI) methodologies, to cite a few.

In the perception and decision-making domain, the survey in [18] summarizes the state of radar sensor modeling, presenting a classification based on modeling methodologies. It examines how these models integrate into the verification and validation (V&V) tasks across the development lifecycle. The survey in [19] analyzes LiDAR-based 3D object detection methods, proposing a specific taxonomy for point-cloud data to address complexity in sparse environments. Additionally, Ref. [20] benchmarks Visual SLAM techniques for localization accuracy and robustness, while Ref. [21] surveys planning tools (maps,

middleware) and identifies integration and standardization gaps. Furthermore, Ref. [22] outlines architectures for SAE Level 3+ systems, detailing perception and decision-making subsystems, and Ref. [23] reviews functions like perception and localization alongside real-world algorithm comparisons. Finally, Ref. [24] reviews advances in ADS technology related to road infrastructure, identifying critical gaps that must be addressed to enable the transition from human-driven to AV.

On the simulation front, Ref. [25] surveys 22 simulators, 35 datasets, and 10 virtual competitions, highlighting their role in addressing traditional road-testing limitations. The study in [26] compares 15 simulation platforms identifying CARLA [27], AirSim [28], LGSVL [29], and SUMO [30] as relevant open-source simulators for ADS testing. Similarly, Ref. [31] analyzes the evolution of simulators and explains how the functionalities and utilities have developed. Lastly, Ref. [32] underscores the simulation's cost-effectiveness despite its fidelity gaps.

The integration of AI techniques further bridges gaps across perception, decision making, and validation. In this regard, Ref. [33] highlights convolutional neural networks (CNNs) and multimodal sensor fusion for scene understanding, but the authors note scalability challenges. The work in [34] surveys imitation learning (e.g., behavioral cloning) to replicate expert driving behaviors, addressing dataset dependency in end-to-end systems. Ref. [35] analyzes AI's evolution across autonomy levels, emphasizing adaptive learning and ethical considerations. Lastly, Refs. [36,37] highlight a hybrid training strategy, where human-guided reinforcement learning (RL) models for ADS are first trained in virtual environments before real-world deployment, improving adaptability and safety.

All these surveys provide a nuanced understanding of the ADS technological landscape, balancing advancements in perception, decision making, simulation realism, and AI-driven innovation with persistent challenges in real-world validation.

1.2. Motivation and Contributions

To the best of our knowledge, existing survey papers lack a comprehensive focus on leading open-source ADS platforms. This survey addresses this gap by examining prominent open-source ADS, analyzing their functionalities, and evaluating their impact on research. Through a detailed literature review, the survey explores the adoption and application of these ADS platforms across key research topics. The contributions of this survey are summarized as follows:

- **Comprehensive Overview:** The survey thoroughly examines leading open-source ADS, highlighting their strengths and limitations.
- **Research Domain Identification:** The survey identifies the applicability and utilization of prominent ADS in key research areas.
- **Trend Insights:** The survey offers an in-depth understanding of current trends and developments, particularly in interoperability with emerging technologies such as AI/ML solutions and edge computing.

By doing so, the survey provides researchers and practitioners with insights into the potential of each open-source ADS in advancing research.

1.3. Survey Organization

Section 2 outlines the methodology used for conducting the present survey. Section 3 provides an overview of leading open-source ADS, focusing on their features and functionalities. Section 4 explores the use and application of open-source ADS platforms in key research areas through an in-depth discussion of academic scientific literature. Section 5 presents the main findings of the survey, assessing the impact of each ADS platform on specific research domains. Section 6 analyzes the strengths and limitations of each platform

and discusses their interoperability with next-generation technologies. Finally, Section 7 concludes the survey by summarizing its key insights.

2. Materials and Methods

The present survey provides a synthesis of the technical features of leading open-source ADS, drawing insights from their official websites, technical papers, and relevant literature. Additionally, it evaluates the impact of these ADS on research. This evaluation involves a search of scientific literature and a detailed analysis of the literature to identify the scope and impact of open-source ADS on research.

The search process was conducted across publicly available databases, including IEEE Xplore, ScienceDirect, MDPI, and Wiley Library. To ensure relevance, the survey focuses exclusively on contributions published between 2018 and 2025. The selection process employed specific inclusion criteria, such as keywords (Autoware, Apollo, “NVIDIA Drive”, OpenPilot), combined with alternative terms using OR conjunctions (OpenPilot OR Comma.ai, “NVIDIA Drive” OR DriveWorks). These were further refined by adding the filtering term “Autonomous” using AND conjunctions. The primary keywords searches are summarized in Table 1.

Table 1. Keywords used for search strategy.

ADS	Keywords Searches
Autoware	Autonomous AND Autoware
Baidu Apollo	Autonomous AND (Apollo OR “Baidu Apollo”)
NVIDIA Drive	Autonomous AND (“NVIDIA Drive” OR “Drive PX2” OR DriveWorks)
OpenPilot	Autonomous AND (OpenPilot OR Comma.ai)

The screened results primarily include journal articles and conference papers. From an initial pool of 225 papers, the filtering process resulted in a rejection rate of approximately 13%. The rejected publications were excluded because they only mentioned ADS platforms without utilizing them. The remaining papers were reviewed and included in the present survey. Table 2 provides a breakdown of the identified 199 publications across the open-source ADS and academic databases.

Table 2. Survey publications across platforms and academic databases.

Database	Autoware	Baidu Apollo	NVIDIA Drive	OpenPilot	Total
IEEE Xplore	38	47	24	15	124
MDPI	10	6	4	3	23
ScienceDirect	7	5	6	3	21
Wiley	2	5	4	1	12
Sage	2	1	0	1	4
Taylor	3	1	0	0	4
ACM	1	4	1	1	7
Others	1	0	3	0	4
Total	64	69	42	24	199

The subsequent literature analysis, identifying the impact of open-source ADS on research, is detailed in Section 4.

3. Overview of Prominent Open-Source ADS

Recently, Autoware, Baidu Apollo, NVIDIA Drive, and OpenPilot have emerged as leading open-source ADS, as evidenced by the significant volume of scientific contributions leveraging these platforms. This section provides an overview of their key features and capabilities, concluding with a comparative table highlighting their salient attributes.

3.1. Autoware

Autoware is an open-source software stack for autonomous driving, first released in 2015 in Japan. Initially developed by Nagoya University and Tier IV, it is now maintained by the Autoware Foundation. Through its development kit, Autoware offers a comprehensive suite of tools for AV development, including localization using LiDAR, GNSS, and IMU; perception through object detection and tracking with LiDAR, cameras, and radar; planning and control for path generation; and vehicle maneuvering. It also supports HD mapping and vehicle-to-infrastructure communication.

To operate vehicles with Autoware, they should be equipped with a DataSpeed Drive-By-Wire (DBW) interface that communicates directly with the CAN bus, enabling precise vehicle control [38]. It is highly scalable and can be customized to work with distributed or less-powerful hardware. The minimum hardware requirements include an 8-core CPU with 16 GB of RAM. However, advanced GPUs (e.g., NVIDIA GPUs) are mandatory for enabling neural network functions, such as LiDAR-based object detection and camera-based object detection, as well as traffic light detection and classification. It is typically installed on Ubuntu with a Robotics Operative System (ROS), allowing users to leverage a vast ecosystem of robotics tools and libraries. Autoware can be integrated with several simulators, including Gazebo for ROS-compatible simulations, CARLA for realistic environments [20], and the LGSVL Simulator for HD maps and sensor configurations [22]. These tools enable developers to test and validate autonomous driving systems in virtual environments before deploying them in real-world scenarios. They support C/C++ and Python for scripting.

Autoware has a strong open-source community, with contributions from researchers and developers worldwide. Its documentation is comprehensive, covering installation, configuration, and module-specific guides, though it can be challenging for beginners, and it also provides workshops [39] and training courses [40] to foster collaboration and knowledge sharing.

3.2. Baidu Apollo

Apollo, developed by Baidu (Beijing, China), a leading Chinese technology company, was launched in 2017. It provides a comprehensive ecosystem for developing self-driving vehicles, offering modules for perception, prediction, planning, control, and simulation. Apollo supports localization using LiDAR and GNSS, HD mapping, and DNN-based perception systems for object detection and tracking.

To operate a vehicle with Apollo, the vehicle must be modified with a Drive-by-Wire interface by a professional service company. The latest version, Apollo 10, requires robust hardware, such as Neousys's In-Vehicle Computer with GPUs for AI processing or NVIDIA Drive PX2 (or similar GPUs). Apollo is typically installed on Ubuntu Linux with ROS, enabling modular development. Apollo equips developers with two key tools: Dreamview and CyberRT. Dreamview is a web-based visualization and debugging tool designed to provide developers with an intuitive interface for monitoring and interacting with the ADS. CyberRT is a high-performance communication framework replacing the ROS middleware, offering lower latency, better scalability, and improved performance for real-time autonomous driving applications. It supports scripting in C++, Python, and Shell-

scripts. Additionally, Apollo provides a cloud service platform enabling data sharing and fleet management and serves as research and development infrastructure for autonomous driving development.

Apollo integrates several simulators, including its own Apollo Game Engine-Based Simulator [41], which offers realistic driving scenarios and sensor modeling. It also supports third-party simulators such as CARLA and LGSVL, enabling developers to test algorithms in virtual environments before real-world deployment.

To support developers, Apollo provides detailed documentation, including installation guides and Docker containers, for easier setup. It also maintains a dedicated developer portal to streamline research and development [42].

3.3. NVIDIA Drive

NVIDIA Drive (Santa Clara, CA, U.S.) was launched in 2015. It provides a comprehensive ecosystem for developing self-driving vehicles. It offers modules for perception, localization, path planning, decision making, and vehicle control. The platform supports several sensors, including LiDAR, GNSS, IMU, cameras and radar, HD maps, etc. NVIDIA Drive integrates several key components to enable advanced driving capabilities. At its core are DriveOS, a software stack acting as a real-time operating system (RTOS) hypervisor, NVIDIA Cuda libraries for parallel computing, NVIDIA TensorRT for optimizing deep learning inference, and additional modules for secure boot and firewall protection.

To operate a vehicle, it is necessary to use Vehicle IO, a device acting as an interface, enabling efficient data flow between vehicle sensors and NVIDIA Drive Software. The main hardware components include advanced GPUs such as NVIDIA Drive AGX (PX2, Xavier, and Orin). The heart of the platform is DriveWorks SDK, the AV software stack that provides essential services (perception, localization, path planning, and decision making), enabling the vehicle to navigate and operate autonomously. Furthermore, DriveWorks' deep neural network (DNN) harnesses NVIDIA's expertise in AI and machine learning, empowering the deployment of DNN models for critical tasks like object detection, sensor fusion, and image segmentation. Furthermore, it provides Drive Sim, a simulation platform. This leverages advanced AI, ray tracing, and sensor modeling to create lifelike simulations of cameras, LiDAR, radar, and other sensors, enabling comprehensive testing and validation of ADS. It also supports C/C++ scripting.

NVIDIA Drive ADS benefits from NVIDIA, its matrix company, providing extensive documentation, developer guides, and technical support, as well as providing an AI training infrastructure [43].

3.4. OpenPilot

OpenPilot was developed by Comma.ai, (San Diego, CA, USA) founded by George Hotz, and was released in 2016 as an open-source driver-assistance system (ADAS). Unlike full ADS, OpenPilot focuses on Level 2 autonomy, aiming to improve safety and convenience for drivers and to enhance existing ADAS in consumer vehicles, providing features like adaptive cruise control, lane-keeping, and automated lane changes.

The hardware for OpenPilot centers on Comma-2/3/3X devices, which are dash-mounted units equipped with three cameras, a Snapdragon processor, and connectivity modules. These devices interface with supported vehicles via CAN bus using the Panda device, requiring minimal installation effort, and are compatible with a growing list of automakers. The core autonomous driving logic processes forward road images through a neural network for real-time lane detection and process data from the sensors, offering a more advanced and customizable alternative to factory-installed ADAS. Unlike HD map-dependent systems, OpenPilot supplements its vision-based approach with Mapbox for

turn-by-turn navigation and OpenStreetMap (OSM) for speed limit data and road geometry. This combination enhances driver assistance capabilities while preserving the system's map-independent architecture. The software runs on Ubuntu, allowing developers to contribute to and customize the code using Python and Shell-scripts. It also supports CARLA and MetaDrive simulators, enabling developers to test algorithms in virtual environments before real deployment.

OpenPilot has a community of developers supported by GitHub repositories. The project documentation is less formal compared to other platforms, and the Comma.ai company emphasizes the Do-It-Yourself approach.

The key features and capabilities of the prominent open-source ADS are summarized in Table 3.

Table 3. Features and key capabilities of prominent open-source ADS.

Feature	Autoware	Baidu Apollo	NVIDIA Drive	OpenPilot
Hardware				
Interface	DataSpeed	Drive-by-Wire	Vehicle IO	Panda
Sensor	Lidar, Radar GNSS/IMU Camera	Lidar, Radar GNSS/IMU Camera	Lidar, Radar GNSS/IMU Camera	Car's built-in Radar GNSS/IMU HDR Camara
Minimal Hdware	Ava/Autocore/NVIDIA Agx/ Nxp Bluebox	NVIDIA Gpu/multi-core Cpu	DRIVE Agx Orin/Thor/ Hyperion	Comma 3, 3X
Software				
Programing Lang	C/C++	C/C++	C/C++	C++, Python
Scripting Lang	C++, Python, Bash	C++, Python, Bash	C/C++, Bash	Python, Bash
UI	Rviz	Dreamview	DriveWorks SDK	Android-UI
ROS Integration	Yes	Yes	No	No
Maps	HD Maps	HD maps	HD maps	Mapbox/OSM
Cloud Integration	Limited	Extensive	Extensive	Limited
Security	Community-driven updates	Encryption, secure boot	Encryption, secure boot	Community-driven updates
Simulators	Gazebo/ Carla/Lgsvl	Apollo Sim/ Carla/Lgsvl/	Drive Sim	Carla/ MetaDrive
Others				
Documentation	Comp. Doc Workshops	Comp. Doc Annual Summit	Comp.Doc/ Guides/Support	Less Formal Doc
Comment	Research-focused	Full autonomy, cloud-integrated	AI-driven, scalable	Consumer-focused, Level 2 Adas

3.5. Openness of Prominent Open-Source ADS

The openness of autonomous driving platforms varies significantly across the different ADS. Autoware stands out as the most transparent, offering its fully open-source Autoware. Auto software stack under community-driven governance, though it faces limitations with proprietary hardware drivers. Apollo follows a partially open approach, releasing core perception and planning modules while retaining proprietary control over critical components like HD maps and cloud services. NVIDIA Drive adopts a similar mixed approach, providing open elements such as the DRIVE OS SDK, CUDA/TensorRT libraries for AI acceleration, pre-trained models for perception tasks, the NVIDIA TAO (train,

adapt, optimize) toolkit for model customization, and Omniverse Replicator's APIs for synthetic data generation. However, NVIDIA restricts access to key components like DriveWorks middleware and HD Maps & Localization, enforcing hardware dependency on its platforms. In contrast, OpenPilot embraces complete openness with its community-centric development, though its scope is restricted to ADAS applications and requires Comma hardware (e.g., Comma 2/3/3X).

4. Use of Open-Source ADS Platforms in Academic Research

The primary objective of this survey is to assess the scope and research impact of four prominent open-source ADS, namely, Autoware, Apollo, NVIDIA Drive, and OpenPilot. To achieve this, the study begins by categorizing the collected literature according to key research topics within the autonomous driving domain. These topics are raised by the 2023 update of the "Connected, Cooperative, and Automated Mobility" (CCAM) framework and the "Strategic Research and Innovation Agenda" (SRIA) report [44], which outline critical research challenges for connected and automated vehicles. While this report addresses a wide range of multidisciplinary challenges, this survey focuses on the technical dimensions of ADS platforms, excluding non-technical aspects such as legal frameworks, sustainability, and societal implications. Specifically, the study investigates how the leading open-source ADS platforms are utilized across the following key research areas:

1. Sensing and Perception.
2. Localization and Mapping.
3. Decision Making and Planning.
4. Connectivity and Communication.
5. Safety, Testing, and Validation.
6. Real-Time Aspects.
7. Software Quality and Cybersecurity Risks.
8. Application of Artificial Intelligence.

As a result, this survey analyzes 199 papers, categorizing them into eight key research topics. The categorization process involved reading the abstracts and, when necessary, the full text of papers. Although some articles could fit into multiple categories due to overlapping themes, each was assigned to the most relevant category to avoid duplication. By concentrating on these challenges, the survey aims to provide an insightful analysis of the role and impact of each open-source ADS in each key research area.

4.1. Sensing and Perception

Sensing and perception involve processing data from sensors such as LiDAR, radar, and cameras to perceive the environment under various conditions, including adverse weather and low-light scenarios. Sensor fusion techniques play a critical role by integrating data from multiple sensors, enabling a comprehensive understanding of the surroundings. As a result, perception is typically the first task in an ADS platform.

The Autoware platform is widely utilized for sensing and perception tasks. For instance, Ref. [45] presents the design of a vision control unit capable of real-time processing for both camera and LiDAR data, while Ref. [46] showcases an Autoware-based system that employs depth sensing utilizing off-the-shelf depth cameras, validated on a Jetson platform. Concerning sensor fusion, Ref. [47] introduces a modular multi-sensor fusion framework that leverages Autoware's LiDAR detection capabilities. Similarly, Ref. [48] describes a multimodal sensor dataset collection framework using IseAuto, a platform built on Autoware, and Ref. [49] proposes an extrinsic calibration method for camera and LiDAR, benchmarking its performance. For object detection, Ref. [50] presents an intelligent camera system for object detection and position estimation, while Ref. [51] proposes a method

using a monocular camera and 3D point cloud maps for detecting non-fixed obstacles. Finally, Ref. [52] implements a traffic light recognition system, and Ref. [53] describes a lane-finding prototype.

In comparison, fewer references were identified as relevant to sensing and perception using Apollo. The study outlined in [54] evaluates sensor performance, particularly for lateral distance perception, by analyzing pavement measurements under different environmental factors. Ref. [55] develops a LiDAR-based 3D object detection technique leveraging Apollo's PointPillars-HSL algorithm, while Ref. [56] provides the adaptation of the Apollo self-driving stack to a research vehicle, outlining the hardware configurations. Additionally, Ref. [57] enhances perception at obstructed intersections through a LiDAR infrastructure system and vehicle-to-everything (V2X) collaboration, and Ref. [58] presents a data acquisition system that employs Apollo's algorithms for 3D point cloud segmentation. Finally, Ref. [59] proposes a virtual simulation component used to analyze the impact of perception errors on AV safety without the need to model the sensors themselves, evaluated on both the Apollo platform and LGSVL simulator.

NVIDIA Drive platforms are also widely used for perception tasks across a range of applications. For instance, Ref. [60] introduces light-enhanced depth, a cost-effective approach to improve night-time depth estimation using high-definition headlights, validated using NVIDIA Drive Sim. In the realm of perception, Ref. [61] introduces HydraFusion, a dynamic sensor fusion framework using built-in DriveWorks algorithms. Similarly, Ref. [62] proposes a parking slot detection system based on YOLOv4, deployed on Drive AGX, while Ref. [63] introduces NVAutoNet, an accurate 360° 3D perception network deployed on the NVIDIA Drive Orin platform. In the field of stereo vision, Ref. [64] presents a framework for object and lane detection using DriveWorks SDK. Additionally, Ref. [65] proposes a model-based methodology to design an onboard surround vision system with SysML language, which is then deployed on the Drive PX2 platform. Lastly, Ref. [66] proposes the design of a stereo vision system developed using the NVIDIA DriveWorks framework deployed on the NVIDIA Drive PX2 platform.

Regarding the OpenPilot platform, the sole identified study, Ref. [67], focuses on the development of Strym, a Python tool for real-time CAN data logging, analysis, and visualization using the Comma.ai Panda device. This tool is designed to decode CAN messages for vehicle-agnostic analysis, highlighting OpenPilot's niche focus in this area.

4.2. Localization and Mapping

Localization and mapping are processes aimed at determining a vehicle's position and identifying the locations of surrounding objects. These tasks are typically accomplished by leveraging pre-existing environmental maps and employing Simultaneous Localization and Mapping (SLAM) techniques, enabling the estimation of the vehicle's relative position and the detection of nearby objects by analyzing data from sources such as LiDAR point clouds, image sequences, or depth images.

In this regard, few contributions using the Apollo platform were found in this domain. For instance, Ref. [68] proposes a self-localization framework that leverages onboard sensors and a hierarchical map, achieving centimeter-level accuracy without dependence on GPS or wireless signals, while Ref. [69] introduces a binocular object localization method utilizing stereo cameras and traditional vision techniques. Furthermore, Ref. [70] proposes a data-driven prediction architecture to automate model training and deployment across various geo-fenced areas. Lastly, Ref. [71] proposes CROUTE, a route coverage testing method for AVs that employs map modeling, with its efficiency validated on both Apollo and LGSVL simulators.

Autoware is also utilized for localization and mapping. Ref. [72] generates lane-level HD maps using Autoware's Lanelet2 format, while Ref. [73] designs an edge–fog–cloud system for HD map generation from LiDAR data, utilizing Autoware's NDT-Mapping algorithms. Similarly, Ref. [74] proposes fast point cloud feature extraction methods for SLAM, improving both accuracy and computational efficiency over Autoware's NDT-Mapping. Further contributions include [75], which develops a tool for generating vector maps, and [76], which provides a tutorial on HD map generation for autonomous driving, enabling level 4 autonomy for environmental-monitoring vehicles. Additionally, Ref. [77] presents an outdoor mobile robot using low-cost LiDAR with localization and mapping capabilities. Lastly, Ref. [78] describes a LiDAR-based SLAM method that improves pose estimation and map accuracy for ground vehicles in rough environments, with LiDAR-camera calibration performed using Autoware's tools.

Regarding the use of the NVIDIA platforms, four publications addressing localization and mapping have been identified. The study outlined in [79] introduces a cost-effective mobile mapping system for 3D color point cloud reconstruction, while Ref. [80] introduces a mobile mapping system designed for the automatic extraction of 3D geodetic coordinates for traffic signs utilizing enhanced point cloud reconstruction techniques using NVIDIA DriveWorks SDK. Further, Ref. [81] develops a robust architecture for vehicle dynamics state estimation and localization in high-performance race cars, utilizing LIDAR data. Finally, Ref. [82] presents a method for geolocating road objects using a monocular camera and the inverse Haversine formula and provides detailed guidelines for calibrating cameras using NVIDIA DriveWorks.

4.3. Decision Making and Planning

Decision making and planning are fundamental components of autonomous driving (AV), as they determine how the vehicle should navigate by converting target waypoints into actionable commands such as steering, throttle, and brake inputs. This module focuses on developing algorithms for real-time control in dynamic traffic environments. It encompasses both lateral and longitudinal control, as well as dynamic path planning, especially in challenging edge cases or complex scenarios where the environment demands advanced adaptability and precision.

In this context, the Apollo platform is widely utilized for experimentation and development. For instance, Refs. [83,84] propose Polsa and PoSa, respectively, which are control policy-based systems designed to analyze driving safety, identify hazardous scenarios, and optimize control behaviors. The work outlined in [85] presents an optimal path-planning approach using quadratic programming to ensure kinematically feasible and collision-free paths in cluttered environments, while Ref. [86] integrates a mixed-integer optimization-based planning algorithm, demonstrating smooth trajectory tracking in real-world experiments. For lateral control optimization, Ref. [87] refines the linear quadratic regulator (LQR) by employing artificial bee colony optimization (ABC), genetic algorithm (GA), and particle swarm optimization (PSO) algorithms. Further contributions include [88], which introduces a personalized motion planning method incorporating driver characteristics for highway driving, and Ref. [89] explores the parameterization of trajectory planning in lane-change scenarios. Additionally, Ref. [90] introduces a motion-planning framework for last-mile delivery vehicles, reusing Apollo's implementation, and Ref. [91] develops an experimental testbed based on Apollo's architecture, integrating multiple sensors to test parallel parking in narrow environments. Lastly, Ref. [92] presents a C++ Library for responsibility-sensitive safety integrated with Apollo for the planning module.

Concerning the Autoware platform, numerous contributions have explored decision making and planning. For instance, Ref. [93] conducts a comparative analysis of Autoware

and Apollo platforms, evaluating their algorithms for scene understanding, path planning, and vehicle control. Ref. [94] introduces a tracing framework that generates timing models for Autoware's LiDAR-based motion planning, and Ref. [95] proposes a cooperative path-planning model using future path sharing, improving efficiency at intersections. Further contributions include [96], which introduces AUDRIC2, a modular framework for automated decision making and control, validated in simulated environments, and [97], which presents a lightweight long-term vehicular motion prediction method using spatial databases and kinematic trajectory data. For specialized applications, Ref. [98] presents an autonomous delivery robot leveraging Autoware to enhance navigation and obstacle detection, Ref. [99] proposes a fuzzy predictive control system for self-driving electric wheelchairs in crowded environments, and Ref. [100] develops a sigmoid-based path planning algorithm for smooth and safe overtaking maneuvers, validated through simulations and real-world experiments using the iseAuto shuttle running Autoware's planning algorithms. Lastly, Ref. [101] proposes an enhanced A*-based path planning algorithm for autonomous land vehicles, comparing its performance against an Autoware-based vehicle.

Concerning NVIDIA Drive, several contributions highlight their use in decision-making and planning tasks. For example, Ref. [102] presents a collision-free autonomous navigation module leveraging edge computing developed with NVIDIA's DriveNet, and Ref. [103] implements dynamic path planning and model predictive control (MPC) for self-driving electric vehicles, demonstrating its effectiveness in parking and obstacle avoidance scenarios. Further, Ref. [104] presents an image-based control system concept for a lane-keeping application validated through hardware-in-the-loop experiments, and Ref. [105] introduces a framework for vision-based lateral control. Lastly, Ref. [106] proposes a planning procedure that considers obstacles, contrasting with graph-based algorithms.

Regarding the OpenPilot platform, fewer contributions addressing decision-making aspects have been found. Ref. [107] presents a new open architecture for AV control combining OpenPilot with ROS, validated through real-world tests. Ref. [108] develops MPC models for adaptive cruise control (ACC) to tackle traffic congestion and enhance string stability, leveraging leader history and driver relaxation. Additionally, Ref. [109] examines the impact of low-level control on string stability in commercial ACC systems, validated through on-road experiments. Lastly, Ref. [110] incorporates driver relaxation into factory ACC systems to reduce lane-change disruptions, enhancing both comfort and traffic flow stability, with validation conducted through simulations and road tests.

4.4. Connectivity and Communication

Connectivity and communication are also critical since vehicles are equipped with advanced technologies that enable them to communicate with other vehicles, infrastructure, and external systems.

Regarding the use of the Apollo platform, two contributions have been identified. The first, Ref. [111], presents a cooperative control scheme for connected automated vehicle platoons using V2X communication, validated through both simulations and real-world experiments. The second, Ref. [112], introduces an IoT-based framework for autonomous driving centered on platooning, leveraging Apollo's sensor architecture to enhance connectivity and coordination among vehicles.

In the Autoware context, several studies have addressed V2X and cooperative perception. For instance, Ref. [113] proposes a V2X-integrated microservices architecture enabling real-time control and vulnerable road users' safety. Similarly, Ref. [114] develops a system for real-time 3D map distribution using roadside edges. Concerning cooperative perception, Ref. [115] introduces AutoC2X, a software solution for V2X cooperative perception, achieving low-latency communication. Ref. [116] proposes networked roadside cooperative

perception units, while Ref. [117] evaluates a system integrating infrastructure and vehicle sensors for CAV navigation and safety, reusing Autoware's clustering-based 3D object detection. Furthermore, Ref. [118] proposes a cooperative perception framework tested on the Autoware co-simulation platform. Concerning cloud services, Ref. [119] implements an adaptive big data management system integrating Autoware with cloud services for real-time data processing. Finally, Ref. [120] presents ASIRA, an architecture combining ROS2 and adaptive automotive open system architecture (AUTOSAR), validated through interoperability with Autoware.

On the NVIDIA front, Ref. [121] develops a cooperative adaptive cruise control tested using LTE-V for V2V communication for cooperative driving. Similarly, Ref. [122] proposes a cooperative driving system that reuses DriveWorks sub-models for real-time object and lane detection in platooning scenarios, and Ref. [123] proposes a human-lead-platooning cooperative adaptive cruise control system for connected vehicles. Finally, Ref. [124] proposes F-Cooper, a feature-based cooperative perception framework for CAVs using 3D point clouds, using NVIDIA Drive PX2.

On the OpenPilot side, a unique study, Ref. [125], focuses on the design and implementation of a framework for remote car driving using 4G/5G networks.

4.5. Real-Time Aspects

Systems are engineered to process and respond to inputs or events within a defined time frame. Latency, in this context, refers to the delay between an input and its corresponding response. Minimizing latency is crucial in real-time applications to ensure optimal system performance and reliability, as even minor delays can result in failures.

In the context of Autoware, several contributions focus on latency analysis. For instance, Ref. [126] evaluates scheduling performance by examining executor deadline misses, while Ref. [127] investigates end-to-end latency. Additionally, Ref. [128] introduces CARET, a tool for measuring end-to-end latency and visualizing bottlenecks in Autoware. Similarly, Ref. [129] proposes Autoware_Perf, a framework for tracing and performance analysis to measure end-to-end latency and evaluate system performance. Ref. [130] explores power and performance bottlenecks in Autoware, identifying sources of latency and energy consumption. Furthermore, Ref. [131] presents ResCue, a predictable data-driven resource management system for autonomous embedded systems, ensuring low-overhead temporal and spatial data availability. Finally, Ref. [132] introduces Autoware on Board, a profile for embedded systems, demonstrating its real-time performance.

In the NVIDIA ecosystem, Ref. [133] measures the latency and runtime behavior of algorithms on NVIDIA Drive PX2, providing insights into real-time capabilities and execution times. For instance, Ref. [134] proposes a centralized vehicle management system integrating strict real-time and non-critical functions on a low-cost embedded platform, demonstrated using DriveOS with a hardware-in-the-loop simulation. Ref. [135] proposes a safety-aware energy optimization framework for multi-sensor neural controllers. Lastly, Ref. [136] analyzes and optimizes the performance of NVIDIA automotive GPUs (TX2 and AGX Xavier) for ADAS and autonomous driving workloads, identifying opportunities to enhance efficiency.

Despite the importance of real-time issues, no contributions have been identified that address real-time issues using Apollo and OpenPilot platforms.

4.6. Safety, Testing, and Validation

Ensuring the safety and reliability of AVs requires extensive testing across a wide range of scenarios, including edge cases and rare events. Developing standardized testing protocols and validation frameworks requires using sophisticated simulation environments

that incorporate Hardware-in-the-Loop (HIL) and Software-in-the-Loop (SIL) frameworks. The literature on testing and validation using open-source platforms highlights a diverse array of methodologies and tools for evaluating the safety of autonomous driving systems.

Numerous contributions have focused on testing and validation using Apollo platforms. For example, Ref. [137] presents a tool for generating road-generalizable scenarios from accident reports, while Ref. [138] reports on testing of Apollo's automated parking system. Additionally, Ref. [139] assesses AV safety in real-world crash scenarios, Ref. [140] examines Apollo's safety performance in high-risk scenarios, and Ref. [141] investigates the limitations of traditional road-based testing tools regarding time and cost-effectiveness and proposes virtual validation based on the vehicle-in-the-loop (VIL) approach as an efficient testing methodology that reduces the need for road testing. Further, Ref. [142] proposes accelerated verification of AV based on adaptive subset simulation for safety verification, and Ref. [143] develops a complexity evaluation framework for urban intersection scenarios. The study outlined in [144] demonstrates the effectiveness of a vehicle-in-the-loop (ViL) testbed in conducting dynamic tests of vehicles equipped with highly automated driving functions, and Ref. [145] optimizes simulation test environments using Apollo and CARLA. In the realm of safety and mapping, Ref. [146] develops collision-avoidance testing for AV based on complete maps, while Ref. [147] introduces ATLAS, systematic testing using the map topology-based scenario classification method. Ref. [148] introduces SALVO, a fully automated framework designed to identify critical driving scenarios that can be instantiated on existing high-definition HD maps. To streamline testing processes, Ref. [149] proposes a genetic algorithm-based technique to generate testing scenarios for AV, demonstrating its effectiveness using the LGSVL and Apollo framework. Similarly, Ref. [150] proposes AV-FUZZER, a genetic algorithm-based framework for identifying safety violations, and Ref. [151] proposes AVChecker, a framework for AV developers to systematically find violations of complex scenario-dependent driving rules in AV software using formal methods. Other contributions include [152,153], which focus on evolutionary test generation and rule violation identification, respectively. Ref. [154] proposes a reinforcement learning-based method for scenario generation, while Ref. [155] proposes RBML, a refined behavior modeling language for safety-critical systems. Efforts to automate AV testing are also evident in [156], which explores metamorphic testing for perception-camera modules, and [157], which introduces in-place metamorphic testing for video streams, identifying failures in the camera perception module. Lastly, Ref. [158] proposes a run-time fault detection in perception systems, and Ref. [159] presents a mathematical model for runtime monitoring and fault detection in perception systems, validated on both the Apollo platform and LGSVL simulator.

For the Autoware platform, several studies have also contributed to advancing the safety of AV technologies. For example, Ref. [160] underscores the importance of training engineers to prioritize safety and understand the limitations of AV systems. Ref. [161] evaluates collision warning systems using ISO standards, while Ref. [162] validates several ADAS applications for various driving scenarios. Ref. [163] shares real-world experiences and provides recommendations for improving Autoware. Additionally, Ref. [164] proposes a technique for testing AVs across diverse geographical features. Other contributions include [165], which introduces SITAR, a testing framework for evaluating the adversarial robustness of traffic light recognition. Refs. [166,167] present frameworks for scenario conversion and scenario-driven development. Ref. [168] introduces a generic risk assessment methodology and run-time monitoring. Specialized applications include [169,170], which describes a MATLAB/Simulink benchmark toolbox designed for risk assessment and runtime monitoring of Autoware-based automated vehicles, and [171], which proposes a methodology for creating digital twins, validated using Autoware and the LGSVL

simulator. Finally, Ref. [172] summarizes experiments conducted with a shuttle powered by Autoware.

Regarding NVIDIA platforms, Ref. [173] presents an in-system-test (IST) architecture for NVIDIA Drive-AGX platforms, which is designed to ensure functional safety and achieve fail-safe states in autonomous driving systems. Ref. [135] proposes a safety-aware energy optimization framework for multi-sensor neural controllers tested on CARLA and on the NVIDIA Drive PX2 platform. Ref. [174] proposes methods to generate and characterize scenarios for AV safety testing, ranking scenarios by complexity metrics to evaluate accident avoidance. Finally, Ref. [175] introduces Suraksha, a framework used to analyze the safety of perception design choices in AVs, quantifying trade-offs between accuracy, processing rate, and power savings.

The OpenPilot platform [176] presents the use of Grand Theft Auto (GTA) as a simulation environment for testing edge cases. Ref. [177] develops environmental risk profiles tested on OpenPilot, revealing risk trends comparable to those observed in human drivers. Ref. [178] proposes a scenario-sampling-based safety evaluation framework for ADAS validated on OpenPilot. Lastly, Ref. [179] introduces CarFASE, a CARLA-based fault and attack simulation engine for testing AVs, which is used to assess OpenPilot's behavior under fault injections, such as changes in brightness.

4.7. Software Quality and Cybersecurity Risks

Software quality (SQ) in AVs refers to software reliability, its adherence to industry standards, and its fault tolerance. Measuring SQ involves techniques such as static analysis, integration testing, and fault injection to assess the system's robustness. Cybersecurity risks in AVs arise from vulnerabilities in their interconnected systems, making them susceptible to threats like hacking, spoofing, and data breaches. Ensuring software quality and avoiding threats is crucial for maintaining the integrity and safety of AVs.

The Apollo platform has been the subject of numerous studies addressing software and cybersecurity issues. For instance, Ref. [180] evaluates Apollo's compliance with ISO 26262 software safety guidelines, offering both quantitative and qualitative metrics to assess adherence in software design and testing. Similarly, Ref. [181] proposes the assessment of the safety of Apollo's perception system, observing that requirements relating to traditional software are fulfilled, while most requirements specific to machine learning are not. Regarding software bug identification, Ref. [182] identifies 499 bugs in Apollo software, classifying them by root causes, symptoms, and affected components. Ref. [183] introduces DoppelTest, a genetic algorithm-based test generation approach that uncovers 123 bug-revealing violations in Apollo. Fault tolerance has also been a key focus, with [184] proposing a distributed safety mechanism using hypervisors for fault detection validated on both Apollo and LGSVL simulators. Ref. [185] proposes a method to test AI-based integrated modules in autonomous driving systems by isolating them into standalone modules, simplifying their validation. Further contributions include [186], which introduces a Bayesian fault injection method to accelerate fault tolerance testing AV software, and [187], which introduces DriveFI, a machine learning-based fault injection engine designed to identify scenarios and faults that have the greatest impact on AV safety. Ref. [188] applies metamorphic testing for Apollo, highlighting its potential for software quality assurance, and [189], which introduces metamorphic testing to evaluate untestable software, validated on Apollo platform, detailing several identified software failures. Additionally, Ref. [190] proposes MT-Nod, a method for detecting non-optimal decisions in Apollo through interactive scenario generation, and Ref. [191] proposes a Just-in-Time (JIT) defect prediction method to assist developers in reducing code inspection and testing efforts, thereby improving the quality of software development. On the cybersecurity front, Ref. [192] introduces

AVGuardian, a system designed to detect publish–subscribe over-privilege in Apollo by enforcing fine-grained access control policies. Lastly, Refs. [193,194] propose transformer-based one-class classification models to detect GPS spoofing, traffic sign recognition attacks, and lane detection attacks.

Significant research has also focused on software quality and cybersecurity within the Autoware framework. For instance, Ref. [182] investigates bugs in AV software, classifying their root causes, symptoms, and affected components. Similarly, Ref. [195] analyzes bug characteristics, identifying symptoms, fixing strategies, and their correlation with modular architecture to improve software quality and reliability. Complementing these studies, Refs. [196,197] propose metamorphic testing frameworks for understanding AV decisions and evaluating non-deterministic behaviors in complex scenarios. On the cybersecurity front, Ref. [198] analyzes and protects the AV software stack using fault injection, proposing a dynamic protection system to reduce error rates. Similarly, Ref. [199] introduces Guardauto, a decentralized runtime protection system for AVs, which mitigates runtime failures and attacks with minimal performance overhead. Additionally, Ref. [200] presents a secure pose estimation method for AVs under cyberattacks, utilizing the Extended Kalman Filter and cumulative sum techniques for sensor attack detection. Lastly, Ref. [201] investigates function interaction risks in robot applications, proposing security policies and coordination nodes, with validation conducted on both Apollo and Autoware.

Contributions addressing software quality using NVIDIA platforms appear limited, with only two identified publications. Ref. [202] conducts a case study on fault injection techniques for assessing the safety of AV, leveraging both the Apollo and NVIDIA Drive platforms to assess their resilience. Ref. [203] proposes the analysis of the software of cooperative automotive architectures applied to a platooning prototype.

Numerous contributions explore a variety of approaches to assess OpenPilot’s robustness. For instance, Ref. [204] presents a systems-theoretic process analysis-based fault injection framework to evaluate software resilience under different conditions and faults. Similarly, Ref. [205] investigates its robustness against affecting sensor data, which are significant contributors to the disengagement of the ADAS. Complementing these efforts, Ref. [206] introduces VulFuzz, a vulnerability-oriented fuzz testing framework that uncovers 335 crashes in OpenPilot, highlighting several critical areas for improvement. Additionally, Ref. [207] proposes security vulnerability metrics for connected vehicles to prioritize testing efforts on vulnerable components. Empirical studies also contribute to understanding AV performance, with [208] analyzing OpenPilot’s issues and pull requests to categorize strengths, weaknesses, and future directions for the Level 2 ADAS systems. On the cybersecurity front, Ref. [209] proposes an end-to-end approach to mitigate adversarial attacks on OpenPilot’s automated lane-centering system. Ref. [210] explores network falsification to detect safety property violations in OpenPilot’s DNN. Finally, Ref. [211] analyzes the security of self-driving technologies, including OpenPilot, within the context of Saudi Arabia, emphasizing the importance of infrastructure compatibility for widespread adoption.

4.8. Application of Artificial Intelligence

The application of AI is essential to improve the performance and adaptability of AV systems in terms of scene interpretation and decision-making capabilities. Handling edge cases and rare scenarios, where training data may be limited, is particularly challenging.

The Apollo framework has seen extensive integration of AI techniques. For example, Refs. [212,213] propose a real-time safety monitoring system that combines machine learning (ML) and fuzzing techniques to detect and mitigate safety violations. Ref. [214] investigates the integration of ML models into the intricate framework of autonomous

driving systems, using the Apollo platform as a case study. To enhance fault tolerance, Ref. [215] presents perception simplex, a fault-tolerant architecture for collision avoidance, combining LiDAR and DNNs to ensure safety in perception faults. On the hardware front, Ref. [216] explores the shift from GPU-based to FPGA-based acceleration for DNNs, evaluating performance improvements in terms of accuracy. Concerning vehicle dynamics, Ref. [217] introduces a learning-based vehicle dynamics modeling procedure for large-scale, cross-vehicle applications. Ref. [218] proposes a multi-agent reinforcement learning framework for discovering hazardous scenarios, while Ref. [219] proposes a car-following model using gantry control unit networks. Finally, Ref. [220] leverages Wasserstein generative adversarial networks to generate high-risk scenarios for AV testing.

NVIDIA's platforms have demonstrated exceptional effectiveness in implementing AI techniques, as evidenced by a broad spectrum of contributions. Ref. [221] provides a comprehensive survey of deep neural network (DNN) techniques applied to autonomous driving, highlighting the critical role of NVIDIA's hardware in tasks such as perception, planning, and control. Concerning object detection, Ref. [222] implements DNN-based object detection methods, while Ref. [223] focuses on evaluating the reliability of a DNN designed for object detection and classification. Further contributions include [224], which presents a faster R-CNN for real-time traffic sign detection. Similarly, Ref. [225] proposes a DNN model for moving object detection for ADAS applications. Complementing these efforts, Ref. [226] optimizes CNN frameworks for time-sensitive applications, ensuring robust performance in dynamic environments. For lane detection, Ref. [227] introduces PointLaneNet, an efficient end-to-end CNN for real-time lane detection. Ref. [228] outlines PilotNet, a lane-keeping system powered by a single DNN that processes pixel inputs to generate a desired vehicle trajectory as its output. Additionally, Ref. [229] presents an efficient encoder–decoder CNN for real-time multilane detection. Ref. [230] proposes an integrated framework for AV based on the NVIDIA DNN for multi-class object detection, lane detection, and free space detection, with experimental validation on the Drive PX2 platform. Specialized applications include [231], which introduces a brain-inspired deep imitation learning (DIL) algorithm to enable driving systems to perform better in various scenarios; [232], which designs a lightweight CNN for steering angle prediction, and [233], which proposes a system with several end-to-end DNNs co-trained with a generative world model to mitigate the imitation learning covariate shift problem, which was validated on NVIDIA Drive Sim platform.

AI techniques have also been integrated into the OpenPilot framework. For instance, Ref. [234] explores neural network-based functional degradation to synthesize redundancy models. Regarding navigation challenges, Ref. [235] introduces an auxiliary task network leveraging semantic segmentation and temporal information to improve navigation performance. Complementing this, Ref. [236] presents an end-to-end CNN model for lane keeping trained using the comma.ai dataset, while Ref. [237] proposes an end-to-end controller using visual attention and RNNs for better control of speed and steering. Additionally, Ref. [238] assesses human driving behavior using AI techniques validated on OpenPilot. Finally, Ref. [239] develops a deep learning framework for optimizing lane detection and steering, validated on real-world Comma.ai footage.

Contributions addressing the application of AI using Autoware are relatively limited, with only three publications identified. Ref. [240] proposes a self-adaptive approach to service deployment under mobile edge computing, utilizing reinforcement learning for resource optimization. Ref. [241] proposes a method to estimate vehicle localization using a DNN fed by LiDAR 3D point cloud and HD maps inspired by the MinkLoc3D-SI method. Lastly, Ref. [242] describes an optimized single-stage deep convolutional neural network to detect objects in urban environments using nothing more than point cloud data.

5. Impact of Prominent Open-Source ADS Platforms on Research

The significant number of contributions leveraging open-source ADS platforms underscores their widespread adoption for research and their pivotal role in advancing ADS technology. However, the adoption of these platforms is not evenly distributed, as some platforms are more dominant in specific research areas than others. Table 4 presents a detailed breakdown of the collected publications, organized by platforms and research areas of research.

Table 4. Breakdown of the survey publications across platforms and key research areas.

Key Research	Autoware	Apollo	N. Drive	OpenPilot	Total
Sensing and Perception	9	6	7	1	23
Localization and Mapping	7	4	4	0	15
Decision Making and Planning	9	10	5	4	28
Connectivity and communications	8	2	4	1	15
Safety, Testing, and Validation	12	24	4	4	44
Real-Time Aspects	8	0	3	0	11
Software Q. and Cybersecurity	8	14	2	8	32
App. Artificial Intelligence	3	9	13	6	31
Total	64	69	42	24	199

The quantitative analysis of data highlights, with respect to ADS platforms use, that Apollo emerges as the most commonly used platform, with 69 publications (34.7% of total), closely followed by Autoware with 64 (32.2%). NVIDIA Drive accounts for 42 publications (21.1%), while OpenPilot has the fewest, at 24 (12.0%). This distribution reflects the dominant role of Apollo and Autoware in ADS research, likely due to their maturity, industry support, and open-source nature, whereas OpenPilot's limited presence aligns with its narrower focus on Level 2 ADAS rather than full autonomy.

Examining the research areas, safety, testing and validation stands out as the most commonly studied category, with 44 publications, highlighting its critical importance for real-world deployment, followed by the software quality and cybersecurity (32) area. The application of Artificial Intelligence (31 publications) and decision-making and planning (28) also receive substantial attention, indicating their central roles in advancing ADS technology. In contrast, localization and mapping (15 contributions) and real-time aspects (11) are relatively understudied, suggesting that real-time optimizations may be handled through proprietary solutions, while localization might be considered a more mature field requiring less active research.

On the other hand, the figures shown in Table 4 serve as a basis for qualitative analysis that explores the "why" and "how" behind the observed trends, since the ADS platforms exhibit distinct research strengths and focus areas. In this aspect, Apollo demonstrates a clear emphasis on testing and validation (24 publications) and software quality and cybersecurity (14), likely driven by Baidu's commercial deployment priorities. Notably, Apollo shows no publications on real-time aspects, raising questions about whether this reflects the proprietary handling of real-time challenges or a gap in research. Autoware presents a more balanced distribution, excelling in sensing and perception (nine publications) and localization and mapping (seven), which may stem from its modular ROS-based architecture. Its strong showing in decision making (nine) and real-time aspects (eight) positions it as a preferred platform for academic and prototyping work. NVIDIA Drive displays a strong research profile, dominating in the application of AI with (13 publications). This

dominance stems from the NVIDIA hardware-driven and AI business model, while safety and real-time aspects are either handled internally or hardware-accelerated. OpenPilot, with its smaller footprint, concentrates primarily on decision making (four) and software quality (8), with no contributions to localization and mapping or real-time aspects, a finding consistent with its Level 2 ADAS orientation and more limited scope compared to full autonomy platforms.

In summary, the analysis shows the relevance of open-source platforms to ADS research while revealing variations in platform adoption and research focus. The dominance of Apollo and Autoware reflects their established positions in the field, while the specialized profiles of NVIDIA Drive and OpenPilot illustrate alternative approaches to autonomous driving challenges. The identified research gaps, particularly in real-time aspects, point to important opportunities for future investigations that could further advance the development and deployment of AV technologies.

6. Strengths and Weaknesses of Leading Open-Source ADS Platforms

Each of the prominent open-source ADS presents strengths and weaknesses, particularly when considering installation complexity, expertise requirements, cost-effectiveness, documentation quality, community support, and suitability for small or large research groups.

Autoware is a moderately complex platform to install due to its modular architecture and reliance on the Robot Operating System (ROS). While it offers detailed documentation, setting up the environment can be challenging for beginners. It requires expertise in robotics, AI, and software development, making it more suitable for teams with technical depth. As an open-source platform, Autoware is cost-effective for research, though its high computational demands can increase hardware costs. It benefits from extensive documentation and a large, active community, making it a strong choice for both small and large research groups focused on prototyping and experimentation. However, its complexity and real-time performance limitations may pose challenges for smaller teams with limited resources.

Apollo, in contrast, is highly complex to install due to its comprehensive ecosystem and dependencies, though it provides detailed guides and scripts to ease the process. It demands expertise in AI, machine learning, and autonomous systems, particularly for advanced features like HD mapping and cloud integration. While open-source, Apollo's advanced tools and hardware requirements can increase costs, making it less accessible for small teams. Its robust documentation and strong community support, backed by Baidu's industry partnerships, make it ideal for large research groups and commercial projects. However, its complexity and proprietary elements may be overwhelming for smaller teams without significant resources.

NVIDIA Drive stands out for its relatively straightforward installation process, thanks to developer tools, though it requires compatible NVIDIA hardware. It requires expertise in AI, DNN, and GPU programming to fully leverage its capabilities. The platform is expensive due to its reliance on high-end GPUs and proprietary software, making it less cost-effective for small teams. NVIDIA provides excellent documentation and developer support, though its community is more industry-focused than open-source platforms. NVIDIA Drive is best suited for large research groups and commercial projects with substantial budgets, offering real-time performance and seamless hardware integration. However, its high costs and vendor lock-in may limit accessibility for smaller teams.

OpenPilot, on the other hand, is the most accessible platform, with a straightforward installation process and minimal dependencies. It requires basic knowledge of software development and ADAS systems, making it ideal for hobbyists and small teams. OpenPilot is highly cost-effective and designed to work with affordable, consumer-grade hardware.

It has good documentation and a strong community of enthusiasts, though it may lack the depth of more professional platforms. OpenPilot is best suited for small research groups focused on ADAS development. However, its limitations on ADAS applications and hardware constraints make it less suitable for large-scale or fully autonomous projects.

Table 5 compares the strengths and limitations of leading open-source ADS platforms.

Table 5. Pros and cons of prominent open-source ADS.

	Baidu Apollo	Autoware	NVIDIA Drive	OpenPilot
Installation	High	Moderate	Moderate	Low
Expertise	AI, Cloud Comp.	ROS, AI	AI, GPU Prog.	ADAS
Cost	High	Moderate	Expensive Hardware	Low
Documentation	Extensive	Extensive, Active community	Extensive, Industry-Focused	Less Formal
Research Teams	Large, Comm. Projects	Small & Large Prototyping	Large, Comm. Projects	Small
Weakness	Steep Learning Curve	Limited Clouds Support	Expensive hardware	Level 2

Interoperability with New Technologies and Future Directions

The interoperability of open-source ADS is a critical factor in advancing innovation in next-generation mobility. Open-source ADS platforms such as Autoware, Apollo, NVIDIA Drive, and OpenPilot, each with distinct architectures and strengths, must seamlessly integrate to enable scalable and adaptable autonomous driving ecosystems. One approach to achieving interoperability is through standardized communication protocols and modular frameworks that allow different systems to exchange data efficiently. For instance, leveraging middleware like ROS2 can bridge gaps between disparate ADS platforms, enabling them to share sensor data, perception outputs, and control commands. Additionally, adopting common interfaces for vehicle-to-everything (V2X) communication ensures that open-source ADS can interact with infrastructure, other vehicles, and cloud-based services, enhancing cooperative driving and traffic management.

Integrating open-source ADS platforms with next-generation technologies such as AI and ML can be leveraged using modular AI pipelines that incorporate state-of-the-art models for perception, planning, and decision making. For instance, Autoware and Apollo support plug-and-play AI frameworks, enabling researchers to implement models like YOLOv8 for object detection or Transformer-based architectures for path prediction. Federated learning also presents a promising approach, allowing collaborative model training across vehicle fleets without centralized data aggregation, a technique already being explored by OpenPilot.

Edge computing further enhances the capabilities of open-source ADS platforms by enabling distributed processing and reducing reliance on cloud-based systems. Platforms like Autoware and Apollo can offload computationally intensive tasks, such as sensor fusion and localization, to edge devices, including NVIDIA Orin or Qualcomm Snapdragon Ride processors. For example, coupling NVIDIA Drive’s high-performance edge AI capabilities with Autoware’s modular autonomy stack can lead to more efficient and responsive autonomous operations. However, optimizing these platforms for heterogeneous edge hardware, such as TPUs and FPGAs, remains a challenge.

Advancements in 5G and beyond will enhance vehicle-to-everything (V2X) communication, enabling open-source ADS to leverage distributed computing and swarm

intelligence. They will also open new possibilities for real-time coordination between vehicles and infrastructure, as demonstrated by Apollo's cloud-edge integration.

Finally, open-source platforms are well-positioned to drive the next wave of innovation. Strategic partnerships, such as Autoware's collaboration with NVIDIA or Apollo's integration with 5G networks, can help bridge these gaps, while industry-wide standardization efforts will be key to delivering interoperable, production-ready ADS solutions. A cross-platform consortium, potentially under the Linux Foundation's LF Edge initiative, could play a central role in fostering this evolution, ensuring that open-source ADS remains at the forefront of autonomous driving technology.

7. Conclusions

This survey highlights the diverse strengths and weaknesses of each platform, reflecting their unique priorities and technological capabilities. Autoware and Apollo emerge as the most comprehensive, with broad coverage across multiple research areas, while NVIDIA Drive demonstrates specialized focuses on AI. Further, OpenPilot highlights its narrower focus in several key areas, such as localization and mapping and real-time aspects, compared to the other platforms. Furthermore, the survey underscores the importance of safety, testing and validation, software quality, and cybersecurity issues, as well as the application of AI methods, which constitute the most active research domain in the autonomous driving field. This analysis provides a snapshot of the research landscape for autonomous driving platforms and highlights areas for future development.

Further considerations also highlight that Apollo excels in scalability and advanced features but is complex and costly, ideal for large teams and commercial applications, while Autoware is a versatile platform for research and prototyping but requires significant expertise and resources, making it better suited for medium to large groups. NVIDIA Drive offers high performance and ease of installation but is expensive and requires specialized expertise, making it best for well-funded projects. OpenPilot is the most accessible and cost-effective option, suitable for small teams, though limited to ADAS applications and level 2 AVs. The choice of platform depends on the team's size, budget, expertise, and project goals, as each platform caters to different needs in the autonomous driving ecosystem.

To improve efficiency and innovation, the open-source community should address fragmentation in tools, protocols, and frameworks. Encouraging shared standards, interoperable tools, and better documentation can enhance collaboration between researchers and developers. Participation in working groups and open forums can align efforts, reduce redundancy, and accelerate progress. A unified approach to standardization will strengthen the ecosystem and support impactful development.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

ACC	Adaptive cruise control.
ADAS	Advanced Driver-Assistance System.
ADS	Autonomous driving system.
AI	Artificial Intelligence.
AV	Autonomous vehicle.
CAN	Controller Area Network.

CARLA	Car Learning to Act.
CAV	Connected automated vehicle.
CNN	Convolutional neural network.
DBW	Drive-By-Wire.
DNN	Deep Neural Network.
GNSS	Global Navigation Satellite System.
GPU	Graphics Processing Unit.
HD maps	High-Definition Maps.
IMU	Inertial Measurement Unit.
LGSVL	LG Silicon Valley Lab Simulator.
LiDAR	Light Detection and Ranging.
ML	Machine learning.
R-CNN	Recursive Convolutional Neural Network.
RL	Reinforcement learning.
RNN	Recurrent Neural Network.
ROS	Robot Operating System.
SAE	Society of Automotive Engineers.
SLAM	Simultaneous Localization and Mapping.
V2X	Vehicle-to-Everything.

References

1. Autonomous Vehicle Market Size & Share Analysis—Trends & Forecasts (2025–2030). Mordor Intelligence. Available online: <https://www.mordorintelligence.com/industry-reports/autonomous-driverless-cars-market-potential-estimation> (accessed on 15 February 2025).
2. Anderson, J.M.; Kalra, N.; Stanley, K.D.; Sorensen, P.; Samaras, C.; Oluwatola, O.A. *Autonomous Vehicle Technology: A Guide for Policymakers*; RAND Corp.: Santa Monica, CA, USA, 2016; ISBN 978-0-8330-8398-2.
3. SAE International. Levels of Driving. Available online: <https://www.sae.org/site/blog/sae-j3016-update> (accessed on 15 February 2025).
4. Cusumano, M.A. Self-driving vehicle technology: Progress and promises. *Commun. ACM* **2020**, *63*, 20–22. [CrossRef]
5. Chai, L. Self-driving cars closer to reality. *RoboGlobal White Paper*. 2020, pp. 1–13. Available online: <https://f.hubspotusercontent40.net/hubfs/7764048/White%20Papers/Autonomous%20Vehicles.pdf> (accessed on 15 February 2025).
6. DARPA Grand Challenge. Available online: https://en.wikipedia.org/wiki/DARPA_Grand_Challenge (accessed on 15 February 2025).
7. TUM-FTM Autonomous Vehicle Lab. Available online: <https://www.mos.ed.tum.de/en/ftm/labs/autonomous-vehicle/> (accessed on 15 February 2025).
8. PEGASUS Project. Available online: <https://pegasus-family.de/> (accessed on 15 February 2025).
9. Autoware Foundation. Available online: <https://www.autoware.org/> (accessed on 15 February 2025).
10. Baidu Apollo. Available online: <https://github.com/ApolloAuto/apollo> (accessed on 15 February 2025).
11. Skruch, P.; Długosz, M.; Szelest, M.; Morys-Magiera, A. Duckietown project pros and cons. In Proceedings of the 2023 21st International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 26–27 October 2023; pp. 446–450. [CrossRef]
12. Bechtel, M.G.; McEllhiney, E.; Kim, M.; Yun, H. DeepPicar: A low-cost deep neural network-based autonomous car. In Proceedings of the 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hakodate, Japan, 28–31 August 2018; pp. 11–21. [CrossRef]
13. Alsherif, M.; Daowd, M.; Bassiuny, A.M.; Metered, H.A. Utilizing transfer learning in the Udacity simulator to train a self-driving car for steering angle prediction. In Proceedings of the 2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 25–27 November 2023; pp. 134–139. [CrossRef]
14. PolySync. Available online: <https://polysync-xrcc.squarespace.com/home> (accessed on 15 February 2025).
15. OpenXC. Available online: <https://openxcplatform.com/> (accessed on 15 February 2025).
16. Nvidia-Drive. Available online: <https://www.nvidia.com/en-us/self-driving-cars/> (accessed on 15 February 2025).
17. Comma.ai OpenPilot. Available online: <https://github.com/commaai/openpilot> (accessed on 15 February 2025).
18. Magosi, Z.F.; Li, H.; Rosenberger, P.; Wan, L.; Eichberger, A. A Survey on Modelling of Automotive Radar Sensors for Virtual Test and Validation of Automated Driving. *Sensors* **2022**, *22*, 5693. [CrossRef]

19. Fernandes, D.; Silva, A.; Névoa, R.; Simões, C.; Gonzalez, D.; Guevara, M.; Novais, P.; Monteiro, J.; Melo-Pinto, P. Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Inf. Fusion* **2021**, *68*, 161–191. [[CrossRef](#)]
20. Sharafutdinov, D.; Griguletskii, M.; Kopanev, P.; Kurenkov, M.; Ferrer, G.; Burkov, A.; Gonnochenko, A.; Tsetserukou, D. Comparison of modern open-source Visual SLAM approaches. *J. Intell. Robot. Syst.* **2023**, *107*, 43. [[CrossRef](#)]
21. Tong, K.; Ajanovic, Z.; Stettinger, G. Overview of tools supporting planning for automated driving. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–8. [[CrossRef](#)]
22. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.M.; Mutz, F.; et al. Self-driving cars: A survey. *Expert. Syst. Appl.* **2021**, *165*, 113816. [[CrossRef](#)]
23. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [[CrossRef](#)]
24. Mihalj, T.; Li, H.; Babić, D.; Lex, C.; Jeudy, M.; Zovak, G.; Babić, D.; Eichberger, A. Road Infrastructure Challenges Faced by Automated Driving: A Review. *Appl. Sci.* **2022**, *12*, 3477. [[CrossRef](#)]
25. Zhang, T.; Liu, H.; Wang, W.; Wang, X. Virtual tools for testing autonomous driving: A survey and benchmark of simulators, datasets, and competitions. *Electronics* **2024**, *13*, 3486. [[CrossRef](#)]
26. Silva, I.; Silva, H.; Botelho, F.; Pendão, C. Realistic 3D simulators for automotive: A review of main applications and features. *Sensors* **2024**, *24*, 5880. [[CrossRef](#)]
27. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16. Available online: <https://proceedings.mlr.press/v78/dosovitskiy17a.html> (accessed on 15 February 2025).
28. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. *Springer Proc. Adv. Robot.* **2017**, *5*, 621–635. [[CrossRef](#)]
29. Rong, G.; Shin, B.H.; Tabatabaee, H.; Lu, Q.; Lemke, S.; Možeiko, M.; Boise, E.; Uhm, G.; Gerow, M.; Mehta, S.; et al. LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020. [[CrossRef](#)]
30. Kusari, A.; Li, P.; Yang, H.; Punshi, N.; Rasulis, M.; Bogard, S.; LeBlanc, D.J. Enhancing SUMO Simulator for Simulation-Based Testing and Validation of Autonomous Vehicles. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; pp. 829–835. [[CrossRef](#)]
31. Li, Y.; Yuan, W.; Zhang, S.; Yan, W.; Shen, Q.; Wang, C.; Yang, M. Choose your simulator wisely: A review on open-source simulators for autonomous driving. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4861–4876. [[CrossRef](#)]
32. Kaur, P.; Taghavi, S.; Tian, Z.; Shi, W. A survey on simulators for testing self-driving cars. In Proceedings of the 2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD), Detroit, MI, USA, 28–29 April 2021; pp. 62–70. [[CrossRef](#)]
33. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **2021**, *10*, 100057. [[CrossRef](#)]
34. Le Mero, L.; Yi, D.; Dianati, M.; Mouzakitis, A. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14128–14147. [[CrossRef](#)]
35. Garikapati, D.; Shetiya, S.S. Autonomous vehicles: Evolution of artificial intelligence and the current industry landscape. *Big Data Cogn. Comput.* **2024**, *8*, 42. [[CrossRef](#)]
36. Yang, H.; Zhou, Y.; Wu, J.; Liu, H.; Yang, L.; Lv, C. Human-Guided Continual Learning for Personalized Decision-Making of Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 5435–5447. [[CrossRef](#)]
37. Wu, J.; Zhou, Y.; Yang, H.; Huang, Z.; Lv, C. Human-Guided Reinforcement Learning with Sim-to-Real Transfer for Autonomous Navigation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14745–14759. [[CrossRef](#)]
38. Guo, H.; Li, J.; Saravanan, N.K.; Wishart, J.; Zhao, J. Developing an automated vehicle research platform by integrating Autoware with the DataSpeed drive-by-wire system. *SAE Tech. Pap.* **2024**. [[CrossRef](#)]
39. Autoware Workshops. Available online: <https://autoware.org/iv2025/> (accessed on 15 February 2025).
40. Autoware NIT Academy. Available online: <https://autoware.org/nit-academy-joins-the-autoware-foundation/> (accessed on 15 February 2025).
41. Apollo Game Engine Sim. Available online: <https://developer.apollo.auto/gamesim.html> (accessed on 15 February 2025).
42. Apollo DevCenter. Available online: <https://developer.apollo.auto/devcenter/devcenter.html> (accessed on 15 February 2025).
43. NVIDIA AI Training Platform. Available online: <https://www.nvidia.com/en-us/self-driving-cars/ai-training/> (accessed on 15 February 2025).
44. CCAM-SIRIA Roadmap. Available online: <https://www.ccam.eu/wp-content/uploads/2023/11/CCAM-SRIA-Update-2023.pdf> (accessed on 15 February 2025).

45. Chakaravarthy, R.V.; Kwon, H.; Jiang, H. Vision control unit in fully self-driving vehicles using Xilinx MPSoC and open-source stack. In Proceedings of the 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, 18–21 January 2021; pp. 311–317.
46. Xu, J.; Yao, Y. Poster: A demo of Autoware-based autonomous driving using depth sensing. In Proceedings of the 2023 IEEE/ACM Symposium on Edge Computing (SEC), Wilmington, DE, USA, 6–9 December 2023; pp. 261–263. [\[CrossRef\]](#)
47. Senel, N.; Kefferpütz, K.; Doycheva, K.; Elger, G. Multi-Sensor Data Fusion for Real-Time Multi-Object Tracking. *Processes* **2023**, *11*, 501. [\[CrossRef\]](#)
48. Gu, J.; Lind, A.; Chhetri, T.R.; Bellone, M.; Sell, R. End-to-End Multimodal Sensor Dataset Collection Framework for Autonomous Vehicles. *Sensors* **2023**, *23*, 6783. [\[CrossRef\]](#)
49. Yin, L.; Luo, B.; Wang, W.; Yu, H.; Wang, C.; Li, C. CoMask: Corresponding Mask-Based End-to-End Extrinsic Calibration of the Camera and LiDAR. *Remote Sens.* **2020**, *12*, 1925. [\[CrossRef\]](#)
50. Genser, S.; Muckenhuber, S.; Solmaz, S.; Reckenzaun, J. Development and Experimental Validation of an Intelligent Camera Model for Automated driving. *Sensors* **2021**, *21*, 7583. [\[CrossRef\]](#)
51. Shikishima, J.; Urasaki, K.; Tasaki, T. PMOD-Net: Point-cloud-map-based metric scale obstacle detection by using a monocular camera. *Adv. Robot.* **2022**, *37*, 458–466. [\[CrossRef\]](#)
52. Hirabayashi, M.; Sujiwo, A.; Monrroy, A.; Kato, S.; Edahiro, M. Traffic light recognition using high-definition map features. *Robot. Auton. Syst.* **2019**, *111*, 62–72. [\[CrossRef\]](#)
53. Dragojević, M.; Stević, S.; Krunić, M.; Lukić, N. Advanced lane finding prototype based on Autoware platform. In Proceedings of the 2020 Zooming Innovation in Consumer Technologies Conference (ZINC), Online, 26–27 May 2020; pp. 169–173. [\[CrossRef\]](#)
54. Li, H.; Bamminger, N.; Magosi, Z.F.; Feichtinger, C.; Zhao, Y.; Mihalj, T.; Orucevic, F.; Eichberger, A. The Effect of Rainfall and Illumination on Automotive Sensors Detection Performance. *Sustainability* **2023**, *15*, 7260. [\[CrossRef\]](#)
55. Li, C.; Yao, G.; Long, T.; Yuan, X.; Li, P. A Novel Method for 3D Object Detection in Open-Pit Mine Based on Hybrid Solid-State LiDAR Point Cloud. *J. Sens.* **2024**, *2024*, 5854745. [\[CrossRef\]](#)
56. Kessler, T.; Bernhard, J.; Buechel, M.; Esterle, K.; Hart, P.; Malovetz, D.; Le, M.T.; Diehl, F.; Brunner, T.; Knoll, A. Bridging the gap between open source software and vehicle hardware for autonomous driving. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1612–1619. [\[CrossRef\]](#)
57. Mo, Y.; Vijay, R.; Rufus, R.; de Boer, N.; Kim, J.; Yu, M. Enhanced Perception for Autonomous Vehicles at Obstructed Intersections: An Implementation of Vehicle to Infrastructure (V2I) Collaboration. *Sensors* **2024**, *24*, 936. [\[CrossRef\]](#)
58. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zheng, Z.; Ma, J. An automated driving systems data acquisition and analytics platform. *Transp. Res. Part C Emerg. Technol.* **2023**, *151*, 104120. [\[CrossRef\]](#)
59. Piazzoni, A.; Cherian, J.; Dauwels, J.; Chau, L. PEM: Perception Error Model for Virtual Testing of Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 670–681. [\[CrossRef\]](#)
60. De Moreau, S.; Almelio, Y.; Bursuc, A.; El-Idrissi, H.; Stanculescu, B.; Moutarde, F. LED: Light enhanced depth estimation at night. *arXiv* **2024**, arXiv:2409.08031. [\[CrossRef\]](#)
61. Malawade, A.V.; Mortlock, T.; Al Faruque, M.A. HydraFusion: Context-Aware Selective Sensor Fusion for Robust and Efficient Autonomous Vehicle Perception. In Proceedings of the 2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPs), Milano, Italy, 4–6 May 2022; pp. 68–79. [\[CrossRef\]](#)
62. Wang, L.; Musabini, A.; Leonet, C.; Benmokhtar, R.; Breheret, A.; Yedes, C.; Bürger, F.; Boulay, T.; Perrotton, X. Holistic Parking Slot Detection with Polygon-Shaped Representations. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 5797–5803. [\[CrossRef\]](#)
63. Pham, T.; Maghoumi, M.; Jiang, W.; Jujjavarapu, B.S.S.; Sajjadi, M.; Liu, X.; Lin, H.-C.; Chen, B.-J.; Truong, G.; Fang, C.; et al. NVAutoNet: Fast and Accurate 360° 3D Visual Perception for Self Driving. In Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2024; pp. 7361–7370. [\[CrossRef\]](#)
64. Kemsaram, N.; Das, A.; Dubbelman, G. A Stereo Perception Framework for Autonomous Vehicles. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–6. [\[CrossRef\]](#)
65. Kemsaram, N.; Das, A.; Dubbelman, G. Model-Based Systems Engineering to Design an Onboard Surround Vision System for Cooperative Automated Vehicles. In Proceedings of the 2021 2nd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey, 16–17 December 2021; pp. 1–6. [\[CrossRef\]](#)
66. Kemsaram, N.; Das, A.; Dubbelman, G. A model-based design of an onboard stereo vision system: Obstacle motion estimation for cooperative automated vehicles. *SN Appl. Sci.* **2022**, *4*, 199. [\[CrossRef\]](#)
67. Bhadani, R.; Bunting, M.; Nice, M.; Tran, N.M.; Elmadani, S.; Work, D.; Sprinkle, J. Strym: A Python Package for Real-time CAN Data Logging, Analysis and Visualization to Work with USB-CAN Interface. In Proceedings of the 2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS), Milan, Italy, 3–6 May 2022; pp. 14–23. [\[CrossRef\]](#)

68. Xia, C.; Shen, Y.; Yang, Y.; Deng, X.; Chen, S.; Xin, J.; Zheng, N. Onboard Sensors-Based Self-Localization for Autonomous Vehicle with Hierarchical Map. *IEEE Trans. Cybern.* **2023**, *53*, 4218–4231. [[CrossRef](#)] [[PubMed](#)]
69. Jiang, F.; Hua, W.; Huang, Z.; Zha, C.; Ma, Y.; Gao, H.; Zhang, S. Online objects localization using stereo camera. In Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), Jinghong, China, 5–9 December 2022; pp. 2122–2127. [[CrossRef](#)]
70. Xu, K.; Xiao, X.; Miao, J.; Luo, Q. Data Driven Prediction Architecture for Autonomous Driving and its Application on Apollo Platform. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 175–181. [[CrossRef](#)]
71. Tang, Y.; Zhou, Y.; Wu, F.; Liu, Y.; Sun, J.; Huang, W.; Wang, G. Route Coverage Testing for Autonomous Vehicles via Map Modeling. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11450–11456. [[CrossRef](#)]
72. Bellusci, M.; Cudrano, P.; Mentasti, S.; Cortelazzo, R.E.; Matteucci, M. Semantic interpretation of raw survey vehicle sensory data for lane-level HD map generation. *Robot. Auton. Syst.* **2024**, *172*, 104513. [[CrossRef](#)]
73. Lee, J.; Lee, K.; Yoo, A.; Moon, C. Design and Implementation of Edge-Fog-Cloud System through HD Map Generation from LiDAR Data of Autonomous Vehicles. *Electronics* **2020**, *9*, 2084. [[CrossRef](#)]
74. Lee, S.W.; Hsu, C.M.; Lee, M.C.; Fu, Y.T.; Atas, F.; Tsai, A. Fast point cloud feature extraction for real-time SLAM. In Proceedings of the 2019 International Automatic Control Conference (CACCS), Keelung, Taiwan, 13–16 November 2019; pp. 1–6. [[CrossRef](#)]
75. Tun, W.N.; Kim, S.; Lee, J.W.; Darweesh, H. Open-source tool of vector map for path planning in Autoware autonomous driving software. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; pp. 1–3. [[CrossRef](#)]
76. Jeong, J.; Yoon, J.Y.; Lee, H.; Darweesh, H.; Sung, W. Tutorial on High-Definition Map Generation for Automated Driving in Urban Environments. *Sensors* **2022**, *22*, 7056. [[CrossRef](#)]
77. Kim, M.; Lee, S.; Ha, J.; Lee, H. Make your autonomous mobile robot on the sidewalk using the open-source LiDAR SLAM and Autoware. *IEEE Trans. Intell. Veh.* **2024**, 1–12. [[CrossRef](#)]
78. Wang, Y.; Ren, B.; Zhang, X.; Wang, P.; Wang, C.; Song, R.; Li, Y.; Meng, M.Q.H. ROLO-SLAM: Rotation-Optimized LiDAR-Only SLAM in Uneven Terrain with Ground Vehicle. *J. Field Robot.* **2025**; 1–23, *early view*. [[CrossRef](#)]
79. Peng, C.W.; Hsu, C.C.; Wang, W.Y. Cost Effective Mobile Mapping System for Color Point Cloud Reconstruction. *Sensors* **2020**, *20*, 6536. [[CrossRef](#)] [[PubMed](#)]
80. Peng, C.W.; Hsu, C.C.; Wang, W.Y. Mobile mapping system for automatic extraction of geodetic coordinates for traffic signs based on enhanced point cloud reconstruction. *IEEE Access* **2022**, *10*, 117374–117384. [[CrossRef](#)]
81. Wischnewski, A.; Stahl, T.; Betz, J.; Lohmann, B. Vehicle Dynamics State Estimation and Localization for High Performance Race Cars. *IFAC-PapersOnLine* **2019**, *52*, 154–161. [[CrossRef](#)]
82. Shami, M.B.; Kiss, G.; Haakonsen, T.A.; Lindseth, F. Geo-locating road objects using inverse Haversine formula with NVIDIA Driveworks. *arXiv* **2024**, arXiv:2401.07582. [[CrossRef](#)]
83. Kang, L.; Shen, H. A Control Policy based Driving Safety System for Autonomous Vehicles. In Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 4–7 October 2021; pp. 464–472. [[CrossRef](#)]
84. Kang, L.; Shen, H.; Li, Y.; Xu, S. A Data-Driven Control-Policy-Based Driving Safety Analysis System for Autonomous Vehicles. *IEEE Internet Things J.* **2023**, *10*, 14058–14070. [[CrossRef](#)]
85. Zhang, Y.; Sun, H.; Zhou, J.; Pan, J.; Hu, J.; Miao, J. Optimal Vehicle Path Planning Using Quadratic Optimization for Baidu Apollo Open Platform. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 978–984. [[CrossRef](#)]
86. Kessler, T.; Esterle, K.; Knoll, A. Mixed-Integer Motion Planning on German Roads Within the Apollo Driving Stack. *IEEE Trans. Intell. Veh.* **2023**, *8*, 851–867. [[CrossRef](#)]
87. Bui, D.T.; Li, H.; De Cristofaro, F.; Eichberger, A. Lateral Control Calibration and Testing in a Co-Simulation Framework for Automated Vehicles. *Appl. Sci.* **2023**, *13*, 12898. [[CrossRef](#)]
88. Zeng, D.; Zheng, L.; Li, Y.; Zeng, J.; Wang, K. A Personalized Motion Planning Method with Driver Characteristics in Longitudinal and Lateral Directions. *Electronics* **2023**, *12*, 5021. [[CrossRef](#)]
89. Li, H.; De Cristofaro, F.; Orucevic, F.; Gu, Z.; Eichberger, A. Quantitative Analysis of the Impact of Baidu Apollo Parameterization on Trajectory Planning in a Critical Scenario. *Transp. Res. Procedia* **2023**, *73*, 102–109. [[CrossRef](#)]
90. Wang, H.; Zhang, L.; Kong, Q.; Zhu, W.; Zheng, J.; Zhuang, L.; Xu, X. Motion planning in complex urban environments: An industrial application on autonomous last-mile delivery vehicles. *J. Field Robot.* **2022**, *39*, 1258–1285. [[CrossRef](#)]
91. Zhang, Z.; Cheng, G.; Guo, Z.; Karimi, H.R.; Lu, Y. Parallel Parking Path Planning and Trajectory Tracking in Narrow Environments for Autonomous Unmanned Vehicles. *Optim. Control Appl. Methods*, **2025**; *early view*. [[CrossRef](#)]

92. Gassmann, B.; Oboril, F.; Buerkle, C.; Liu, S.; Yan, S.; Elli, M.S.; Alvarez, I.; Aerrabotu, N.; Jaber, S.; van Beek, P.; et al. Towards Standardization of AV Safety: C++ Library for Responsibility Sensitive Safety. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2265–2271. [[CrossRef](#)]
93. Raju, V.M.; Gupta, V.; Lomate, S. Performance of open autonomous vehicle platforms: Autoware and Apollo. In Proceedings of the 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 29–31 March 2019; pp. 1–5. [[CrossRef](#)]
94. Abaza, H.; Roy, D.; Fan, S.; Saidi, S.; Motakis, A. Trace-enabled timing model synthesis for ROS2-based autonomous applications. In Proceedings of the 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE), Valencia, Spain, 25–27 March 2024; pp. 1–6. [[CrossRef](#)]
95. Hirata, M.; Tsukada, M.; Okumura, K.; Tamura, Y.; Ochiai, H.; Défago, X. Roadside-assisted cooperative planning using future path sharing for autonomous driving. In Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 27–30 September 2021; pp. 1–7. [[CrossRef](#)]
96. Lattarulo, R.; Hidalgo, C.; Arizala, A.; Perez, J. AUDRIC2: A modular and highly interconnected automated driving framework focus on decision making and vehicle control. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 763–769. [[CrossRef](#)]
97. Tao, L.; Watanabe, Y.; Takada, H. A Lightweight Long-Term Vehicular Motion Prediction Method Leveraging Spatial Database and Kinematic Trajectory Data. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 463. [[CrossRef](#)]
98. Gao, F.; Cheng, Y.; Gao, M.; Ma, C.; Liu, H.; Ren, Q.; Zhao, Z. Design and implementation of an autonomous driving delivery robot. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 3832–3837. [[CrossRef](#)]
99. Kawaguchi, E.; Sekiguchi, K.; Nonaka, K. Self-driving Electric Wheelchair in Crowded Environments Using a Fuzzy Potential Model Predictive Control. *IFAC-PapersOnLine* **2023**, *56*, 11827–11833. [[CrossRef](#)]
100. Malayjerdi, E.; Sell, R.; Malayjerdi, M.; Udal, A.; Bellone, M. Practical path planning techniques in overtaking for autonomous shuttles. *J. Field Robot.* **2022**, *39*, 410–425. [[CrossRef](#)]
101. Erke, S.; Bin, D.; Yiming, N.; Qi, Z.; Liang, X.; Dawei, Z. An improved A-Star based path planning algorithm for autonomous land vehicles. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420962263. [[CrossRef](#)]
102. Dutta, T.; Reddy, D.S.; Rajalakshmi, P. Real-Time Deep Learning Based Safe Autonomous Navigation. In Proceedings of the 2024 8th International Conference on Robotics, Control and Automation (ICRCA), Shanghai, China, 12–14 January 2024; pp. 400–406. [[CrossRef](#)]
103. Chung, Y.; Yang, Y.P. Hardware-in-the-Loop Simulation of Self-Driving Electric Vehicles by Dynamic Path Planning and Model Predictive Control. *Electronics* **2021**, *10*, 2447. [[CrossRef](#)]
104. Mohamed, S.; Goswami, D.; Nathan, V.; Rajappa, R.; Basten, T. A scenario- and platform-aware design flow for image-based control systems. *Microprocess. Microsyst.* **2020**, *75*, 103037. [[CrossRef](#)]
105. Mohamed, S.; De, S.; Bimpisidis, K.; Nathan, V.; Goswami, D.; Corporaal, H.; Basten, T. IMACS: A Framework for Performance Evaluation of Image Approximation in a Closed-loop System. In Proceedings of the 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2019; pp. 1–4. [[CrossRef](#)]
106. Reiter, R.; Kirchengast, M.; Watzenig, D.; Diehl, M. Mixed-integer optimization-based planning for autonomous racing with obstacles and rewards. *IFAC-PapersOnLine* **2021**, *54*, 99–106. [[CrossRef](#)]
107. Barrio, A.V.; Alvarez, W.M.; Olaverri-Monreal, C.; Hernández, J.E.N. Development and Validation of an Open Architecture for Autonomous Vehicle Control. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 4–7 June 2023; pp. 1–6. [[CrossRef](#)]
108. Zhou, H.; Zhou, A.; Li, T.; Chen, D.; Peeta, S.; Laval, J. Congestion-mitigating MPC design for adaptive cruise control based on Newell’s car following model: History outperforms prediction. *Transp. Res. Part C Emerg. Technol.* **2022**, *142*, 103801. [[CrossRef](#)]
109. Zhou, H.; Zhou, A.; Li, T.; Chen, D.; Peeta, S.; Laval, J. Significance of low-level control to string stability under adaptive cruise control: Algorithms, theory and experiments. *Transp. Res. Part C Emerg. Technol.* **2022**, *140*, 103697. [[CrossRef](#)]
110. Zhou, H.; Zhou, A.; Laval, J.; Liu, Y.; Peeta, S. Incorporating Driver Relaxation into Factory Adaptive Cruise Control to Reduce Lane-Change Disruptions. *Transp. Res. Rec.* **2022**, *2676*, 13–27. [[CrossRef](#)]
111. Wang, X.; Li, Y.; Huang, L.; Huang, X.; Zhao, H.; Han, X.; Xiang, J.; Wang, H. Cooperative Control for Connected Automated Vehicle Platoon with C-V2X PC5 Interface. In Proceedings of the 2023 IEEE International Conference on Unmanned Systems (ICUS), Hefei, China, 13–15 October 2023; pp. 635–640. [[CrossRef](#)]
112. Zhou, Z.; Akhtar, Z.; Man, K.L.; Siddique, K. A deep learning platooning-based video information-sharing Internet of Things framework for autonomous driving systems. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719883133. [[CrossRef](#)]
113. Ramos, J.; Figueiredo, A.; Almeida, P.; Aston, T.; Campos, A.; Perna, G.; Mendes, M.; Rito, P.; Sargento, S. Enhancing autonomous vehicles control: Distributed microservices with V2X integration and perception modules. In Proceedings of the 2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST), Dallas, TX, USA, 1–3 May 2024; pp. 143–153. [[CrossRef](#)]

114. Mizutani, M.; Tsukada, M.; Iida, Y.; Esaki, H. 3D maps distribution of self-driving vehicles using roadside edges. In Proceedings of the 2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW), Naha, Japan, 24–27 November 2020; pp. 40–45. [\[CrossRef\]](#)
115. Tsukada, M.; Oi, T.; Ito, A.; Hirata, M.; Esaki, H. AutoC2X: Open-source software to realize V2X cooperative perception among autonomous vehicles. In Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 18 November–16 December 2020; pp. 1–6. [\[CrossRef\]](#)
116. Tsukada, M.; Oi, T.; Kitazawa, M.; Esaki, H. Networked Roadside Perception Units for Autonomous Driving. *Sensors* **2020**, *20*, 5320. [\[CrossRef\]](#) [\[PubMed\]](#)
117. Chen, H.; Bandaru, V.K.; Wang, Y.; Romero, M.A.; Tarko, A.; Feng, Y. Cooperative Perception System for Aiding Connected and Automated Vehicle Navigation and Improving Safety. *Transp. Res. Rec.* **2024**, *2678*, 1498–1510. [\[CrossRef\]](#)
118. Chen, C.; Tang, Q.; Hu, X.; Huang, Z. Infrastructure sensor-based cooperative perception for early stage connected and automated vehicle deployment. *J. Intell. Transp. Syst.* **2023**, *28*, 956–970. [\[CrossRef\]](#)
119. Donzia, S.K.Y.; Kim, H.K.; Geum, Y.P. Implementation of Autoware application to real-world services based adaptive big data management system for autonomous driving. In Proceedings of the 2021 21st International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 13–16 September 2021; pp. 251–257. [\[CrossRef\]](#)
120. Hong, D.; Moon, C. Autonomous Driving System Architecture with Integrated ROS2 and Adaptive AUTOSAR. *Electronics* **2024**, *13*, 1303. [\[CrossRef\]](#)
121. Wang, P.; Chen, Y.; Wang, C.; Liu, F.; Hu, J.; Van, N.N.N. Development and verification of cooperative adaptive cruise control via LTE-V. *IET Intell. Transp. Syst.* **2019**, *13*, 991–1000. [\[CrossRef\]](#)
122. Kemsaram, N.; Das, A.; Dubbelman, G. Architecture Design and Development of an On-board Stereo Vision System for Cooperative Automated Vehicles. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–8. [\[CrossRef\]](#)
123. Zhang, Y.; Wu, Z.; Zhang, Y.; Shang, Z.; Wang, P.; Zou, Q.; Zhang, X.; Hu, J. Human-Lead-Platooning Cooperative Adaptive Cruise Control. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18253–18272. [\[CrossRef\]](#)
124. Chen, Q.; Ma, X.; Tang, S.; Guo, J.; Yang, Q.; Fu, S. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (SEC '19), Arlington, Virginia, 7–9 November 2019; pp. 88–100. [\[CrossRef\]](#)
125. Saez-Perez, J.; Wang, Q.; Alcaraz-Calero, J.M.; Garcia-Rodriguez, J. Design, Implementation, and Empirical Validation of a Framework for Remote Car Driving Using a Commercial Mobile Network. *Sensors* **2023**, *23*, 1671. [\[CrossRef\]](#) [\[PubMed\]](#)
126. Peng, B.; Hasegawa, A.; Azumi, T. Scheduling performance evaluation framework for ROS 2 applications. In Proceedings of the 2022 IEEE 24th International Conference on High Performance Computing & Communications; 8th International Conference on Data Science & Systems, Chengdu, China, 18–21 December 2022; pp. 2031–2038. [\[CrossRef\]](#)
127. Lee, H.; Choi, Y.; Han, T.; Kim, K. Probabilistically guaranteeing end-to-end latencies in autonomous vehicle computing systems. *IEEE Trans. Comput.* **2022**, *71*, 3361–3374. [\[CrossRef\]](#)
128. Kuboichi, T.; Hasegawa, A.; Peng, B.; Miura, K.; Funaoka, K.; Kato, S.; Azumi, T. CARET: Chain-aware ROS 2 evaluation tool. In Proceedings of the 2022 IEEE 20th International Conference on Embedded and Ubiquitous Computing (EUC), Wuhan, China, 28–30 October 2022; pp. 1–8. [\[CrossRef\]](#)
129. Li, Z.; Hasegawa, A.; Azumi, T. Autoware_Perf: A tracing and performance analysis framework for ROS 2 applications. *J. Syst. Archit.* **2022**, *123*, 102341. [\[CrossRef\]](#)
130. Becker, P.H.E.; Arnau, J.M.; González, A. Demystifying power and performance bottlenecks in autonomous driving systems. In Proceedings of the 2020 IEEE International Symposium on Workload Characterization (IISWC), Beijing, China, 27–29 October 2020; pp. 205–215. [\[CrossRef\]](#)
131. Bateni, S.; Liu, C. Predictable data-driven resource management: An implementation using Autoware on autonomous platforms. In Proceedings of the 2019 IEEE Real-Time Systems Symposium (RTSS), Hong Kong, China, 3–6 December 2019; pp. 339–352. [\[CrossRef\]](#)
132. Kato, S.; Tokunaga, S.; Maruyama, Y.; Maeda, S.; Hirabayashi, M.; Kitsukawa, Y.; Monrroy, A.; Ando, T.; Fujii, Y.; Azumi, T. Autoware on board: Enabling autonomous vehicles with embedded systems. In Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPs), Porto, Portugal, 11–13 April 2018; pp. 287–296. [\[CrossRef\]](#)
133. Widerspick, C.; Bauer, W.; Fey, D. Latency Measurements for an Emulation Platform on Autonomous Driving Platform NVIDIA Drive PX2. In Proceedings of the ARCS Workshop 2018; 31st International Conference on Architecture of Computing Systems, Braunschweig, Germany, 9–12 April 2018; pp. 1–8.
134. Sinha, S.; West, R. Towards an integrated vehicle management system in DriveOS. *ACM Trans. Embed. Comput. Syst.* **2021**, *20*, 82. [\[CrossRef\]](#)

135. Odema, M.; Ferlez, J.; Shoukry, Y.; Al Faruque, M.A. SEO: Safety-Aware Energy Optimization Framework for Multi-Sensor Neural Controllers at the Edge. In Proceedings of the 2023 60th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 9–13 July 2023; pp. 1–6. [[CrossRef](#)]
136. Tabani, H.; Mazzocchetti, F.; Benedicte, P.; Abella, J.; Cazorla, F.J. Performance Analysis and Optimization Opportunities for NVIDIA Automotive GPUs. *J. Parallel Distrib. Comput.* **2021**, *152*, 21–32. [[CrossRef](#)]
137. Guo, A.; Zhou, Y.; Tian, H.; Fang, C.; Sun, Y.; Sun, W.; Gao, X.; Luu, A.T.; Liu, Y.; Chen, Z. SoVAR: Building Generalizable Scenarios from Accident Reports for Autonomous Driving Testing. In Proceedings of the 2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 27 October–1 November 2024; pp. 268–280.
138. Towey, D.; Luo, Z.; Zheng, Z.; Zhou, P.; Yang, J.; Ingkasit, P.; Lao, C.; Pike, M.; Zhang, Y. Metamorphic Testing of an Automated Parking System: An Experience Report. In Proceedings of the 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 26–30 June 2023; pp. 1774–1779. [[CrossRef](#)]
139. Zhou, R.; Zhang, G.; Huang, H.; Wei, Z.; Zhou, H.; Jin, J.; Chang, F.; Chen, J. How would autonomous vehicles behave in real-world crash scenarios? *Accid. Anal. Prev.* **2024**, *202*, 107572. [[CrossRef](#)]
140. Zhou, R.; Lin, Z.; Zhang, G.; Huang, H.; Zhou, H.; Chen, J. Evaluating Autonomous Vehicle Safety Performance Through Analysis of Pre-Crash Trajectories of Powered Two-Wheelers. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 13560–13572. [[CrossRef](#)]
141. Li, H.; Nalic, D.; Makkapati, V.; Eichberger, A.; Fang, X.; Tettamanti, T. A Real-Time Co-Simulation Framework for Virtual Test and Validation on a High Dynamic Vehicle Test Bed. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 1132–1137. [[CrossRef](#)]
142. Tian, Y.; Fu, A.; Zhang, H.; Tang, L.; Sun, J. Accelerated Verification of Autonomous Driving Systems based on Adaptive Subset Simulation. *IEEE Trans. Intell. Veh.* **2024**, 1–11. [[CrossRef](#)]
143. Li, J.; Zong, R.; Wang, Y.; Deng, W. Complexity Evaluation for Urban Intersection Scenarios in Autonomous Driving Tests: Method and Validation. *Appl. Sci.* **2024**, *14*, 10451. [[CrossRef](#)]
144. Li, H.; Makkapati, V.P.; Wan, L.; Tomasch, E.; Hoschopf, H.; Eichberger, A. Validation of Automated Driving Function Based on the Apollo Platform: A Milestone for Simulation with Vehicle-in-the-Loop Testbed. *Vehicles* **2023**, *5*, 718–731. [[CrossRef](#)]
145. Luan, W.; Ding, Q.; Wu, Y. Research on Integrated Environment of Autonomous Vehicle Simulation Based on Apollo. In Proceedings of the 2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI), Hangzhou, China, 1–3 December 2023; pp. 397–401. [[CrossRef](#)]
146. Tang, Y.; Zhou, Y.; Liu, Y.; Sun, J.; Wang, G. Collision Avoidance Testing for Autonomous Driving Systems on Complete Maps. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 179–185. [[CrossRef](#)]
147. Tang, Y.; Zhou, Y.; Zhang, T.; Wu, F.; Liu, Y.; Wang, G. Systematic Testing of Autonomous Driving Systems Using Map Topology-Based Scenario Classification. In Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 15–19 November 2021; pp. 1342–1346. [[CrossRef](#)]
148. Nguyen, V.; Huber, S.; Gambi, A. SALVO: Automated Generation of Diversified Tests for Self-driving Cars from Existing Maps. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence Testing (AITest), Oxford, UK, 23–26 August 2021; pp. 128–135. [[CrossRef](#)]
149. Sun, L.; Huang, S.; Zheng, C.; Bai, T.; Hu, Z. Test Case Generation for Autonomous Driving Based on Improved Genetic Algorithm. In Proceedings of the 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS), Chiang Mai, Thailand, 22–26 October 2023; pp. 272–278. [[CrossRef](#)]
150. Li, G.; Li, Y.; Jha, S.; Tsai, T.; Sullivan, M.; Hari, S.K.S.; Kalbarczyk, Z.; Iyer, R. AV-FUZZER: Finding Safety Violations in Autonomous Driving Systems. In Proceedings of the 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), Coimbra, Portugal, 12–15 October 2020; pp. 25–36. [[CrossRef](#)]
151. Zhang, Q.; Hong, D.K.; Zhang, Z.; Chen, Q.A.; Mahlke, S.; Mao, Z.M. A systematic framework to identify violations of scenario-dependent driving rules in autonomous vehicle software. *ACM Meas. Anal. Comput. Syst.* **2021**, *5*, 15.
152. Ebadi, H.; Moghadam, M.H.; Borg, M.; Gay, G.; Fontes, A.; Socha, K. Efficient and Effective Generation of Test Cases for Pedestrian Detection—Search-based Software Testing of Baidu Apollo in SVL. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence Testing (AITest), Oxford, UK, 23–26 August 2021; pp. 103–110. [[CrossRef](#)]
153. Zhou, J.; Tang, S.; Guo, Y.; Li, Y.F.; Xue, Y. From Collision to Verdict: Responsibility Attribution for Autonomous Driving Systems Testing. In Proceedings of the 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), Florence, Italy, 9–12 October 2023; pp. 321–332. [[CrossRef](#)]
154. Wei, Z.; Huang, H.; Zhang, G.; Zhou, R.; Luo, X.; Li, S.; Zhou, H. Interactive Critical Scenario Generation for Autonomous Vehicles Testing Based on In-depth Crash Data Using Reinforcement Learning. *IEEE Trans. Intell. Veh.* **2024**, 1–12. [[CrossRef](#)]
155. Chen, Z.; Liu, J.; Ding, X.; Zhang, M. RBML: A Refined Behavior Modeling Language for Safety-Critical Hybrid Systems. In Proceedings of the 2019 26th Asia-Pacific Software Engineering Conference (APSEC), Putrajaya, Malaysia, 2–5 December 2019; pp. 339–346. [[CrossRef](#)]

156. Zhang, Y.; Towey, D.; Pike, M.; Han, J.C.; Zhou, G.; Yin, C.; Wang, Q.; Xie, C. Metamorphic Testing Harness for the Baidu Apollo Perception-Camera Module. In Proceedings of the 2023 IEEE/ACM 8th International Workshop on Metamorphic Testing (MET), Melbourne, Australia, 14 May 2023; pp. 9–16. [[CrossRef](#)]
157. Zhou, Z.Q.; Zhu, J.; Chen, T.Y.; Towey, D. In-Place Metamorphic Testing and Exploration. In Proceedings of the 2022 IEEE/ACM 7th International Workshop on Metamorphic Testing (MET), Pittsburgh, PA, USA, 9 May 2022; pp. 1–6. [[CrossRef](#)]
158. Antonante, P.; Nilsen, H.G.; Carlone, L. Monitoring of perception systems: Deterministic, probabilistic, and learning-based fault detection and identification. *Artif. Intell.* **2023**, *325*, 103998. [[CrossRef](#)]
159. Antonante, P.; Spivak, D.I.; Carlone, L. Monitoring and Diagnosability of Perception Systems. In Proceedings of the 2021 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 168–175. [[CrossRef](#)]
160. Carballo, A.; Wong, D.; Ninomiya, Y.; Kato, S.; Takeda, K. Training engineers in autonomous driving technologies using Autoware. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3347–3354. [[CrossRef](#)]
161. Reddy, D.S.; Charan, K.S.; Kayam, S.K.; Rajalakshmi, P. Robust obstacle detection and collision warning for autonomous vehicles using Autoware Universe. In Proceedings of the 2024 16th International Conference on Computer and Automation Engineering (ICCAE), Melbourne, Australia, 14–16 March 2024; pp. 378–384. [[CrossRef](#)]
162. Stević, S.; Krunić, M.; Dragojević, M.; Kaprocki, N. Development and validation of ADAS perception application in ROS environment integrated with CARLA simulator. In Proceedings of the 2019 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, 26–27 November 2019; pp. 1–4. [[CrossRef](#)]
163. Malayjerdi, M.; Sell, R.; Malayjerdi, E.; Akbaş, M.I.; Razdan, R. Real-Life Experiences in Using Open Source for Autonomy Applications. *Eng. Proc.* **2024**, *79*, 19. [[CrossRef](#)]
164. Seo, S.; Lee, J.; Kim, M. Testing diverse geographical features of autonomous driving systems. In Proceedings of the 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE), Tsukuba, Japan, 28–31 October 2024; pp. 439–450. [[CrossRef](#)]
165. Yang, B.; Yang, J. SITAR: Evaluating the Adversarial Robustness of Traffic Light Recognition in Level-4 Autonomous Driving. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Republic of Korea, 2–5 June 2024; pp. 1068–1075. [[CrossRef](#)]
166. Miura, K.; Azumi, T. Converting driving scenario framework for testing self-driving systems. In Proceedings of the 2020 IEEE 18th International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 29 December–1 January 2020; pp. 56–63. [[CrossRef](#)]
167. Gulzar, M.; Matiisen, T.; Muhammad, N. Scenario driven development for open source autonomous driving stack. In Proceedings of the 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA), Padova, Italy, 10–13 September 2024; pp. 1–8. [[CrossRef](#)]
168. Tong, K.; Solmaz, S.; Sikic, H.; Reckenzaun, J. A Generic Risk Assessment Methodology and its Implementation as a Run-time Monitoring Device for Automated Vehicles. *Transp. Res. Procedia* **2023**, *72*, 303–310. [[CrossRef](#)]
169. Tokunaga, S.; Miura, K.; Azumi, T. MATLAB/Simulink benchmark suite for ROS-based self-driving software platform. In Proceedings of the 2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC), Valencia, Spain, 7–9 May 2019; pp. 83–84. [[CrossRef](#)]
170. Miura, K.; Tokunaga, S.; Ota, N.; Tange, Y.; Azumi, T. Autoware toolbox: Matlab/simulink benchmark suite for ros-based self-driving software platform. In Proceedings of the 30th International Workshop on Rapid System Prototyping (RSP'19), New York, NY, USA, 17–18 October 2019; pp. 8–14. [[CrossRef](#)]
171. Wang, S.H.; Tu, C.H.; Juang, J.C. Automatic traffic modelling for creating digital twins to facilitate autonomous vehicle development. *Connect. Sci.* **2021**, *34*, 1018–1037. [[CrossRef](#)]
172. Sell, R.; Malayjerdi, M.; Pikner, H.; Razdan, R.; Malayjerdi, E.; Bellone, M. Open-source level 4 autonomous shuttle for last-mile mobility. In Proceedings of the 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA), Padova, Italy, 10–13 September 2024; pp. 1–6. [[CrossRef](#)]
173. Datla Jagannadha, P.K.; Yilmaz, M.; Sonawane, M.; Chadalavada, S.; Sarangi, S.; Bhaskaran, B.; Bajpai, S.; Reddy, V.A.; Pandey, J.; Jiang, S. Special Session: In-System-Test (IST) Architecture for NVIDIA Drive-AGX Platforms. In Proceedings of the 2019 IEEE 37th VLSI Test Symposium (VTS), Monterey, CA, USA, 23–25 April 2019; pp. 1–8. [[CrossRef](#)]
174. Ghodsi, Z.; Hari, S.K.S.; Frosio, I.; Tsai, T.; Troccoli, A.; Keckler, S.W.; Garg, S.; Anandkumar, A. Generating and characterizing scenarios for safety testing of autonomous vehicles. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 157–164. [[CrossRef](#)]
175. Zhao, H.; Hari, S.K.S.; Tsai, T.; Sullivan, M.B.; Keckler, S.W.; Zhao, J. Suraksha: A framework to analyze the safety implications of perception design choices in AVs. In Proceedings of the 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE), Wuhan, China, 25–28 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 434–445. [[CrossRef](#)]

176. Maaradji, A.; Bouchemal, N.; Smadhi, I.; Bouhraoua, A.; Ghanine, A. Beyond Traditional Simulators: Adopting Videogames for Autonomous Vehicle Testing. In Proceedings of the 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Abu Dhabi, United Arab Emirates, 13–17 November 2023; pp. 570–575. [\[CrossRef\]](#)
177. Anih, J.; Kolekar, S.; Dargahi, T.; Babaie, M.; Saraee, M.; Wetherell, J. Deriving Environmental Risk Profiles for Autonomous Vehicles From Simulated Trips. *IEEE Access* **2023**, *11*, 38385–38398. [\[CrossRef\]](#)
178. Weng, B.; Zhu, M.; Redmill, K. A Formal Safety Characterization of Advanced Driver Assist Systems in the Car-Following Regime with Scenario-Sampling. *IFAC-PapersOnLine* **2022**, *55*, 266–272. [\[CrossRef\]](#)
179. Maleki, M.; Farooqui, A.; Sangchoolie, B. CarFASE: A Carla-based Tool for Evaluating the Effects of Faults and Attacks on Autonomous Driving Stacks. In Proceedings of the 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Porto, Portugal, 27–30 June 2023; pp. 92–99. [\[CrossRef\]](#)
180. Tabani, H.; Kosmidis, L.; Abella, J.; Cazorla, F.J.; Bernat, G. Assessing the Adherence of an Industrial Autonomous Driving Framework to ISO 26262 Software Guidelines. In Proceedings of the 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2–6 June 2019; pp. 1–6. [\[CrossRef\]](#)
181. Kochanthara, S.; Singh, T.; Forrai, A.; Cleophas, L. Safety of Perception Systems for Automated Driving: A Case Study on Apollo. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 64. [\[CrossRef\]](#)
182. Garcia, J.; Feng, Y.; Shen, J.; Almanee, S.; Xia, Y.; Chen, Q.A. A Comprehensive Study of Autonomous Vehicle Bugs. In Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), Seoul, Republic of Korea, 27 June–19 July 2020; pp. 385–396. [\[CrossRef\]](#)
183. Huai, Y.; Chen, Y.; Almanee, S.; Ngo, T.; Liao, X.; Wan, Z.; Chen, Q.A.; Garcia, J. Doppelgänger Test Generation for Revealing Bugs in Autonomous Driving Software. In Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 14–20 May 2023; pp. 2591–2603. [\[CrossRef\]](#)
184. Bijlsma, T.; Buriachevskiy, A.; Frigerio, A.; Fu, Y.; Goossens, K.; Ors, A.O.; van der Perk, P.J.; Terechko, A.; Vermeulen, B. A Distributed Safety Mechanism using Middleware and Hypervisors for Autonomous Vehicles. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 1175–1180. [\[CrossRef\]](#)
185. Alcon, M.; Tabani, H.; Abella, J.; Cazorla, F.J. Enabling unit testing of already-integrated AI software systems: The case of Apollo for autonomous driving. In Proceedings of the 2021 24th Euromicro Conference on Digital System Design (DSD), Palermo, Italy, 1–3 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 426–433. [\[CrossRef\]](#)
186. Mei, Y.; Nie, T.; Sun, J.; Tian, Y. Bayesian Fault Injection Safety Testing for Highly Automated Vehicles with Uncertainty. *IEEE Trans. Intell. Veh.* **2024**, 1–15. [\[CrossRef\]](#)
187. Jha, S.; Banerjee, S.S.; Tsai, T.; Hari, S.K.; Sullivan, M.B.; Kalbarczyk, Z.T.; Keckler, S.W.; Iyer, R.K. ML-based fault injection for autonomous vehicles: A case for Bayesian fault injection. In Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Portland, OR, USA, 24–27 June 2019; pp. 112–124. [\[CrossRef\]](#)
188. Zhang, Y.; Pike, M.; Towey, D.; Han, J.C.; Zhou, Z.Q. Preparing Future SQA Professionals: An Experience Report of Metamorphic Exploration of an Autonomous Driving System. In Proceedings of the 2022 IEEE Global Engineering Education Conference (EDUCON), Tunis, Tunisia, 28–31 March 2022; pp. 2121–2126. [\[CrossRef\]](#)
189. Zhou, Z.Q.; Sun, L. Metamorphic testing of driverless cars. *Commun. ACM* **2019**, *62*, 61–67. [\[CrossRef\]](#)
190. Yang, Z.; Huang, S.; Wang, X.; Bai, T.; Wang, Y. MT-Nod: Metamorphic testing for detecting non-optimal decisions of autonomous driving systems in interactive scenarios. *Inf. Softw. Technol.* **2025**, *180*, 107659. [\[CrossRef\]](#)
191. Choi, J.; Kim, T.; Ryu, D.; Baik, J.; Kim, S. Just-in-time defect prediction for self-driving software via a deep learning model. *J. Web Eng.* **2023**, *22*, 303–326. [\[CrossRef\]](#)
192. Hong, D.K.; Kloosterman, J.; Jin, Y.; Cao, Y.; Chen, Q.A.; Mahlke, S.; Mao, Z.M. AVGuardian: Detecting and Mitigating Publish-Subscribe Overprivilege for Autonomous Vehicle Systems. In Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P), Genoa, Italy, 16–18 June 2020; pp. 445–459. [\[CrossRef\]](#)
193. Han, X.; Chen, K.; Zhou, Y.; Qiu, M.; Fan, C.; Liu, Y.; Zhang, T. A Unified Anomaly Detection Methodology for Lane-Following of Autonomous Driving Systems. In Proceedings of the 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), New York City, NY, USA, 30 September–3 October 2021; pp. 836–844. [\[CrossRef\]](#)
194. Han, X.; Zhou, Y.; Chen, K.; Qiu, H.; Qiu, M.; Liu, Y.; Zhang, T. ADS-Lead: Lifelong Anomaly Detection in Autonomous Driving Systems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 1039–1051. [\[CrossRef\]](#)
195. Jiang, Y.; Mo, R.; Zhan, W.; Wang, D.; Li, Z.; Ma, Y. Leveraging modular architecture for bug characterization and analysis in automated driving software. *ACM Trans. Softw. Eng. Methodol.* **2024**. [\[CrossRef\]](#)
196. Cheng, K.; Zhou, Y.; Chen, B.; Wang, R.; Bai, Y.; Liu, Y. Guardauto: A decentralized runtime protection system for autonomous driving. *IEEE Trans. Comput.* **2021**, *70*, 1569–1581. [\[CrossRef\]](#)

197. Luu, Q.H.; Liu, H.; Chen, T.Y.; Vu, H.L. A sequential metamorphic testing framework for understanding autonomous vehicle's decisions. *IEEE Trans. Intell. Veh.* **2024**, 1–13. [[CrossRef](#)]
198. Liu, Q.; Mo, Y.; Mo, X.; Lv, C.; Mihankhah, E.; Wang, D. Secure pose estimation for autonomous vehicles under cyber-attacks. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1583–1588. [[CrossRef](#)]
199. Underwood, R.; Luu, Q.H.; Liu, H. A metamorphic testing framework and toolkit for modular automated driving systems. In Proceedings of the 2023 IEEE/ACM 8th International Workshop on Metamorphic Testing (MET), Melbourne, Australia, 14 May 2023; pp. 17–24. [[CrossRef](#)]
200. Gan, Y.; Whatmough, P.; Leng, J.; Yu, B.; Liu, S.; Zhu, Y. Braum: Analyzing and protecting autonomous machine software stack. In Proceedings of the 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE), Charlotte, NC, USA, 31 October–3 November 2022; pp. 85–96. [[CrossRef](#)]
201. Xu, Y.; Bao, Y.; Wang, S.; Zhang, T. Function interaction risks in robot apps: Analysis and policy-based solution. *IEEE Trans. Dependable Secure Comput.* **2024**, *21*, 4236–4253. [[CrossRef](#)]
202. Iyer, R.K.; Kalbarczyk, Z.T.; Nakka, N.M. Internals of Fault Injection Techniques. In *Dependable Computing*; Iyer, R.K., Kalbarczyk, Z.T., Nakka, N.M., Eds.; IEEE: Piscataway, NJ, USA, 2024. [[CrossRef](#)]
203. Kochanthara, S.; Rood, N.; Khabbaz Saberi, A.; Cleophas, L.; Dajsuren, Y.; van den Brand, M. A functional safety assessment method for cooperative automotive architecture. *J. Syst. Softw.* **2021**, *179*, 110991. [[CrossRef](#)]
204. Rubaiyat, A.H.M.; Qin, Y.; Alemzadeh, H. Experimental Resilience Assessment of an Open-Source Driving Agent. In Proceedings of the 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), Taipei, Taiwan, 4–7 December 2018; pp. 54–63. [[CrossRef](#)]
205. Ali, K.; Jammal, M.; Sharkh, M.A. A Software QA Framework for Autonomous Vehicle Open Source Application: OpenPilot. In Proceedings of the 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), Aveiro, Portugal, 12–27 October 2023; pp. 1–6. [[CrossRef](#)]
206. Moukahal, L.J.; Zulkernine, M.; Soukup, M. Vulnerability-Oriented Fuzz Testing for Connected Autonomous Vehicle Systems. *IEEE Trans. Reliab.* **2021**, *70*, 1422–1437. [[CrossRef](#)]
207. Moukahal, L.; Zulkernine, M. Security Vulnerability Metrics for Connected Vehicles. In Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 22–26 July 2019; pp. 17–23. [[CrossRef](#)]
208. Jiao, R.; Liang, H.; Sato, T.; Shen, J.; Chen, Q.A.; Zhu, Q. End-to-end Uncertainty-based Mitigation of Adversarial Attacks to Automated Lane Centering. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 266–273. [[CrossRef](#)]
209. Zhang, T.; Guan, W.; Miao, H.; Guan, Q.; Liu, Z.; Wang, C.; Huang, X.; Fang, L.; Duan, Z. VSRQ: Quantitative Assessment Method for Safety Risk of Vehicle Intelligent Connected System. *IEEE Trans. Veh. Technol.* **2024**, *74*, 2635–2651. [[CrossRef](#)]
210. von Stein, M.; Elbaum, S. Finding property violations through network falsification: Challenges, adaptations and lessons learned from OpenPilot. In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE '22), Rochester, MI, USA, 10–14 October 2023; ACM: New York, NY, USA, 2023; p. 136. [[CrossRef](#)]
211. Alsubaei, F.S. Reliability and Security Analysis of Artificial Intelligence-Based Self-Driving Technologies in Saudi Arabia: A Case Study of Openpilot. *J. Adv. Transp.* **2022**, *2022*, 2085225. [[CrossRef](#)]
212. Zhang, B.; Huang, Y.; Li, G. Salus: A Novel Data-Driven Monitor that Enables Real-Time Safety in Autonomous Driving Systems. In Proceedings of the 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), Guangzhou, China, 5–9 December 2022; pp. 85–94. [[CrossRef](#)]
213. Zhang, B.; Huang, Y.; Chen, R.; Li, G. D2MoN: Detecting and Mitigating Real-Time Safety Violations in Autonomous Driving Systems. In Proceedings of the 2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Charlotte, NC, USA, 31 October–3 November 2022; pp. 262–267. [[CrossRef](#)]
214. Peng, Z.; Yang, J.; Chen, T.H.; Ma, L. A first look at the integration of machine learning models in complex autonomous driving systems: A case study on Apollo. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020), New York, NY, USA, 8–13 November 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1240–1250. [[CrossRef](#)]
215. Bansal, A.; Kim, H.; Yu, S.; Li, B.; Hovakimyan, N.; Caccamo, M.; Sha, L. Perception simplex: Verifiable collision avoidance in autonomous vehicles amidst obstacle detection faults. *Softw. Test. Verif. Reliab.* **2024**, *34*, e1879. [[CrossRef](#)]
216. Sciangula, G.; Restuccia, F.; Biondi, A.; Buttazzo, G. Hardware Acceleration of Deep Neural Networks for Autonomous Driving on FPGA-based SoC. In Proceedings of the 2022 25th Euromicro Conference on Digital System Design (DSD), Maspalomas, Spain, 31 August–2 September 2022; pp. 406–414. [[CrossRef](#)]
217. Xu, J.; Luo, Q.; Xu, K.; Xiao, X.; Yu, S.; Hu, J.; Miao, J.; Wang, J. An Automated Learning-Based Procedure for Large-scale Vehicle Dynamics Modeling on Baidu Apollo Platform. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5049–5056. [[CrossRef](#)]

218. Mu, Y.; Liu, W.; Yu, C.; Ning, X.; Cao, Z.; Xu, Z.; Liang, S.; Yang, H.; Wang, Y. Multi-Agent Vulnerability Discovery for Autonomous Driving Policy by Finding AV-Responsible Scenarios. In Proceedings of the 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, 28 August–1 September 2024; pp. 2320–2327. [\[CrossRef\]](#)
219. Fei, R.; Li, S.; Hei, X.; Xu, Q.; Liu, F.; Hu, B. The Driver Time Memory Car-Following Model Simulating in Apollo Platform with GRU and Real Road Traffic Data. *Math. Probl. Eng.* **2020**, *2020*, 4726763. [\[CrossRef\]](#)
220. Luo, X.; Wei, Z.; Zhang, G.; Huang, H.; Zhou, R. High-risk powered two-wheelers scenarios generation for autonomous vehicle testing using WGAN. *Traffic Inj. Prev.* **2024**, *26*, 243–251. [\[CrossRef\]](#)
221. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [\[CrossRef\]](#)
222. Tran, D.N.; Nguyen, H.H.; Pham, L.H.; Jeon, J.W. Object Detection with Deep Learning on Drive PX2. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics—Asia (ICCE-Asia), Seoul, Republic of Korea, 1–3 November 2020; pp. 1–4. [\[CrossRef\]](#)
223. Lotfi, A.; Hukerikar, S.; Balasubramanian, K.; Racunas, P.; Saxena, N.; Bramley, R.; Huang, Y. Resiliency of automotive object detection networks on GPU architectures. In Proceedings of the 2019 IEEE International Test Conference (ITC), Washington, DC, USA, 9–15 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–9. [\[CrossRef\]](#)
224. Ravindran, R.; Santora, M.J.; Faied, M.; Fanaei, M. Traffic Sign Identification Using Deep Learning. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 318–323. [\[CrossRef\]](#)
225. Lin, G.T.; Shivanna, V.M.; Guo, J.I. A Deep-Learning Model with Task-Specific Bounding Box Regressors and Conditional Back-Propagation for Moving Object Detection in ADAS Applications. *Sensors* **2020**, *20*, 5269. [\[CrossRef\]](#)
226. Yang, M.; Wang, S.; Bakita, J.; Vu, T.; Smith, F.D.; Anderson, J.H.; Frahm, J.-M. Re-Thinking CNN Frameworks for Time-Sensitive Autonomous-Driving Applications: Addressing an Industrial Challenge. In Proceedings of the 2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Montreal, QC, Canada, 16–18 April 2019; pp. 305–317. [\[CrossRef\]](#)
227. Chen, Z.; Liu, Q.; Lian, C. PointLaneNet: Efficient end-to-end CNNs for accurate real-time lane detection. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2563–2568. [\[CrossRef\]](#)
228. Bojarski, M.; Chen, C.; Daw, J.; Degirmenci, A.; Deri, J.; Firner, B.; Flepp, B.; Gogri, S.; Hong, J.; Jackel, L.; et al. The nvidia pilotnet experiments. *arXiv* **2020**, arXiv:2010.08776. [\[CrossRef\]](#)
229. Chougule, S.; Ismail, A.; Soni, A.; Kozonek, N.; Narayan, V.; Schulze, M. An efficient encoder-decoder CNN architecture for reliable multilane detection in real time. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1444–1451. [\[CrossRef\]](#)
230. Kemsaram, N.; Das, A.; Dubbelman, G. An integrated framework for autonomous driving: Object detection, lane detection, and free space detection. In Proceedings of the 2019 Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), London, UK, 30–31 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 260–265. [\[CrossRef\]](#)
231. Ahmedov, H.B.; Yi, D.; Sui, J. Application of a brain-inspired deep imitation learning algorithm in autonomous driving. *Softw. Impacts* **2021**, *10*, 100165. [\[CrossRef\]](#)
232. Hassan, I.U.; Zia, H.; Fatima, H.S.; Yusuf, S.A.; Khurram, M. A Lightweight Convolutional Neural Network to Predict Steering Angle for Autonomous Driving Using CARLA Simulator. *Model. Simul. Eng.* **2022**, *2022*, 5716820. [\[CrossRef\]](#)
233. Popov, A.; Degirmenci, A.; Wehr, D.; Hegde, S.; Oldja, R.; Kamenev, A.; Douillard, B.; Nistér, D.; Muller, U.; Bhargava, R.; et al. Mitigating covariate shift in imitation learning for autonomous vehicles using latent space generative world models. *arXiv* **2024**, arXiv:2409.16663. [\[CrossRef\]](#)
234. Huang, Z.H.; Wu, Y.S.; Lin, Y.D.; Yu, C.M.; Lee, W.B. Neural Network-based Functional Degradation for Cyber-Physical Systems. In Proceedings of the 2024 IEEE 24th International Conference on Software Quality, Reliability and Security (QRS), Cambridge, UK, 1–5 July 2024; pp. 425–434. [\[CrossRef\]](#)
235. Chen, Y.; Praveen, P.; Priyantha, M.; Muelling, K.; Dolan, J. Learning On-Road Visual Control for Self-Driving Vehicles with Auxiliary Tasks. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 331–338. [\[CrossRef\]](#)
236. Chen, Z.; Huang, X. End-to-end learning for lane keeping of self-driving cars. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1856–1860. [\[CrossRef\]](#)
237. Liu, Z.; Wang, K.; Yu, J.; He, J. End-to-end control of autonomous vehicles based on deep learning with visual attention. In Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020; pp. 584–589. [\[CrossRef\]](#)
238. Driessen, T.; Siebinga, O.; de Boer, T.; Dodou, D.; de Waard, D.; de Winter, J. How AI from Automated Driving Systems Can Contribute to the Assessment of Human Driving Behavior. *Robotics* **2024**, *13*, 169. [\[CrossRef\]](#)

239. Yordanov, D.; Chakraborty, A.; Hasan, M.M.; Cirstea, S. A Framework for Optimizing Deep Learning-Based Lane Detection and Steering for Autonomous Driving. *Sensors* **2024**, *24*, 8099. [[CrossRef](#)]
240. Xiong, W.; Lu, Z.; Li, B.; Wu, Z.; Hang, B.; Wu, J.; Xuan, X. A self-adaptive approach to service deployment under mobile edge computing for autonomous driving. *Eng. Appl. Artif. Intell.* **2019**, *81*, 397–407. [[CrossRef](#)]
241. Matsumoto, K.; Javanmardi, E.; Nakazato, J.; Tsukada, M. Localizability estimation for autonomous driving: A deep learning-based place recognition approach. In Proceedings of the 2023 Seventh IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, USA, 11–13 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 280–283. [[CrossRef](#)]
242. Karl, Y.D.S.; Kim, H.K.; Geum, Y.P. A study on deep learning model autonomous driving based on big data. *Int. J. Softw. Innov.* **2021**, *9*, 143–157. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.