2026 Volume 4, Issue 1
Lecture Notes in Education, Arts, Management and Social Science
ISSN(Online): 2705-053X

Research Review on Testing Technologies for Autonomous Driving Software

Hao Liu*, Hashimah Ismail, Nazlin Hanie binti Abdullah
Faculty of Engineering and Life Sciences, Universiti Selangor (UNISEL), Selangor 40000, Malaysia

**Author to whom correspondence should be addressed.*

Abstract: Based on an extensive review of domestic and foreign literature, this paper makes a systematic analysis and exploration of the autonomous driving software test technology. Combined with the architecture characteristics and system characteristics of autonomous driving software, this paper deeply discusses the simulation test and real scene test methods for the autonomous driving system, as well as the test technology for software components. In terms of simulation test, this paper analyzes software simulation, semi-solid simulation, and ring simulation in detail, and discusses the simulation objects of static environment simulation, dynamic scene simulation, sensor simulation, and vehicle dynamics simulation. For the test technology of autonomous driving software components, this paper focuses on the latest progress of data-driven technology in the testing of sensing components, decision planning components, and control components. Finally, this paper summarizes and analyzes the current challenges of autonomous driving software testing technology, and prospects the direction and focus of future research, aiming to provide a useful reference for the further development of autonomous driving technology.

Keywords: Self-driving software; Overview; Simulation test; Data-driven test; Software test

Online publication: January 26, 2026

1. Introduction

Autonomous driving systems (ADS) rely fundamentally on sophisticated software to execute core functionalities, including environmental perception, intelligent decision-making, path planning, and vehicle control. However, like all software, ADS is susceptible to defects. Given its operation in safety-critical contexts, these flaws can precipitate catastrophic consequences [1]. High-profile incidents, such as a Tesla vehicle's failure to distinguish a white truck from a bright sky in 2019 and an Uber test vehicle's inability to detect a pedestrian at night in 2018, starkly illustrate the profound safety challenges inherent in this technology [2].

Consequently, the testing and verification of autonomous driving software has become a paramount research and industrial priority. The critical nature of its application demands rigorous quality assurance before deployment, driving significant advancements in relevant methodologies. This paper provides a comprehensive review of the current state of autonomous driving software testing, analyzing its architectural foundations, surveying established testing paradigms, and outlining future research trajectories.

2. Introduction to the autonomous driving software system

The Autonomous Driving Software System is a complex on-board computer that integrates diverse advanced technologies, including artificial intelligence (AI), computer vision, state machines, high-definition mapping, radar, and precise positioning systems [3]. Its architecture supports a pipeline of tasks: map localization, sensor data acquisition and fusion, environmental understanding, behavior prediction, path planning, decision-making, and vehicle control. This architecture exhibits three primary characteristics that distinguish it from traditional software and pose unique challenges for quality assurance.

First, it is a highly complex, data-driven system. Its core functions, particularly perception and recognition, are critically dependent on heterogeneous, high-dimensional data streams from LiDAR, cameras, radar, and vehicle state sensors. Unlike traditional driver-assistance systems that present raw data to a human, autonomous software must autonomously process, fuse, and abstract this data to make real-time driving decisions. This deep data dependency and intricate functional logic result in a system of immense scale and complexity, necessitating quality assurance methods that address data diversity, effectiveness, and functional completeness.

Second, the system exhibits highly uncertain behavior. The “intelligence” derived from machine learning (ML) and deep learning (DL) models is often a black box. The theoretical uninterpretability of these models makes it difficult to define their internal states or understand the root causes of erroneous outputs. This opacity presents a formidable challenge for traditional verification and validation, as it is arduous to formally guarantee that the system will behave correctly across the infinite space of potential driving scenarios.

3. Test technology for the autonomous driving software system

Traditional software testing methods, such as code coverage or MC/DC, are often inadequate for autonomous driving software due to its unique code structure and its reliance on data-driven AI components rather than deterministic logic [4]. A holistic, system-level perspective is essential, as testing individual modules in isolation fails to capture the emergent behaviors of the integrated system. Current mainstream approaches focus on scenario-based testing, where the system’s ability to handle standardized or critical driving scenarios is evaluated through simulation, closed-course testing, or on public roads. These methods are inherently data-driven, reflecting the system’s core dependency on data.

3.1. Simulation test

Real-world road testing is costly, time-consuming, and ethically fraught with safety risks. It also struggles to achieve sufficient coverage of rare but critical edge cases. Simulation testing has thus emerged as a vital complementary approach, offering a safe, scalable, and reproducible environment for development and validation.

3.1.1. Simulation methods

Simulation methods vary by their level of physical fidelity. Software Simulation (SIL): This method creates a fully virtual environment, modeling the vehicle, sensors, traffic, and world physics. It is ideal for early-stage algorithm design and validation before hardware prototyping, significantly reducing development costs and time. Platforms like AirSim built on Unreal Engine 4 provide high-fidelity environments for testing complete software stacks [5].

Hardware-in-the-Loop (HIL) / Semi-Physical Simulation: This technique integrates real hardware components into the virtual simulation. It addresses the limitations of pure software models by accounting for real-world hardware latencies, noise, and interface complexities, providing a more realistic testbed for functional prototypes [6].

Vehicle-in-the-Loop (VIL): This is the most physically grounded simulation, where a real vehicle is placed on a chassis dynamometer or in a controlled lab while interacting with a simulated virtual world [7]. It forms a complete closed-loop system, enabling rapid iteration and validation of the entire integrated system under a wide array of virtual conditions [8].

3.1.2. Simulation object segmentation

Effective simulation requires the accurate modeling of four key elements: Static Environment: This involves creating the fixed world geometry using 3D modeling tools, game engines, or by processing real-world data from drones, high-definition maps, and LiDAR point clouds to create photorealistic and geometrically accurate virtual worlds [9].

Dynamic Scene: Realistic traffic flow, pedestrian movement, and traffic light logic are simulated using statistical models derived from real-world data or physics-based agents within the engine [10]. Data-driven frameworks like AADS enhance realism by augmenting real street-view images with synthesized, realistic traffic flows.

Sensor Simulation: To be valid, simulated sensor data (camera, LiDAR, radar) must adhere to the laws of physics. Physics-based rendering engines are used to generate camera images with correct lighting, materials, and optical properties [11]. LiDAR simulation involves casting virtual rays to generate point clouds with accurate reflection intensities, while radar simulation models the transmission and reception of frequency-modulated continuous waves [12].

Vehicle Dynamics: Accurate vehicle models simulate the physical response of the car to control inputs, including steering, tire forces, and suspension dynamics [13]. This ensures that the virtual vehicle's behavior closely mirrors its real-world counterpart.

3.2. Real vehicle site test

Despite the power of simulation, real-world testing remains irreplaceable [14]. It validates the system against the full complexity and unpredictability of the physical world, compensating for any gaps in the simulated models [15].

Preset Scenario Tests: Conducted in controlled proving grounds that replicate specific road features to evaluate performance against known, repeatable challenges [16].

Real Road Tests: The ultimate validation, performed on public roads with prototype vehicles [17]. While offering the highest fidelity, it is constrained by cost, safety regulations, and the difficulty of encountering statistically rare events. Historical collision data can also be used to reconstruct and test against known failure modes [18].

3.3. Exploration of the test technology of autonomous driving software components

Component-level testing focuses on the core functional modules—perception, prediction, planning, and control—providing a more granular and theoretically grounded analysis than system-level tests [19].

Perception Component Testing: This is dominated by data-driven methods for evaluating ML/DL models. Techniques include generating adversarial examples, applying image transformations, and using neuron coverage metrics to guide the creation of test inputs that expose model weaknesses [20]. Other work focuses on validating the underlying algorithms for point cloud processing and multi-sensor fusion.

Decision and Planning Component Testing: As planning increasingly incorporates learning-based methods alongside traditional rule-based systems, new verification challenges arise [21]. Researchers are exploring formal methods, such as using theorem provers to verify logical correctness and numerical soundness of planners, and game-theoretic frameworks to evaluate strategic decision-making.

Control Component Testing: This involves validating the low-level algorithms that translate a planned trajectory into actuator commands. While traditional control logic can be verified formally, the integration of learned controllers requires new approaches. Frameworks using Signal Temporal Logic (STL) or its probabilistic extensions (e.g., C^2TL) are being developed to specify and verify safety properties of these hybrid controllers under uncertainty [22].

4. Challenges and future prospects

Despite significant progress, autonomous driving software testing faces two major, intertwined challenges.

4.1. Challenges of automated testing technology based on multi-source data

The ADS is a quintessential cyber-physical system whose state is a function of software logic, user input, and a dynamic physical environment sensed through multiple, expensive data sources. While open datasets exist, they cannot possibly cover the vast, continuous input domain of real-world driving [23]. A key future direction is the development of resource-efficient, automated methods for test data generation and augmentation. This includes techniques to synthesize diverse, realistic sensor data under cost constraints, thereby maximizing test coverage and adequacy without prohibitive expense [24].

4.2. Challenges of testing technologies for AI models

The core decision-making in modern ADS is increasingly delegated to AI models, primarily deep neural networks. Their inherent lack of interpretability (the "black box" problem) makes it extremely difficult to reason about their internal logic, predict their failure modes, or define what constitutes a "complete" test [25]. Furthermore, these models are vulnerable to adversarial attacks, where subtle, imperceptible perturbations to input data can cause catastrophic misclassifications [26]. Future research must therefore focus on developing robust, interpretable, and attack-resilient testing frameworks specifically designed for these intelligent components. This includes advancing formal verification for neural networks, creating more powerful metamorphic and coverage-guided testing strategies, and establishing standardized benchmarks for AI model robustness in safety-critical settings [27].

5. Conclusion

Autonomous driving software is a complex, data-driven, and safety-critical cyber-physical system whose reliability is paramount. This paper has reviewed the dominant testing paradigms—simulation testing, real-world site testing, and component-oriented testing—highlighting their respective strengths and limitations. The field is characterized by a strong shift towards data-driven and scenario-based methodologies, driven by the fundamental nature of the software itself. However, significant challenges remain, particularly concerning the automated generation of comprehensive multi-source test data and the effective verification of opaque AI models. Addressing these challenges through interdisciplinary research is crucial to building a robust theoretical and practical foundation for autonomous driving software testing, ultimately enabling its safe and widespread societal deployment.

Disclosure statement

The author declares no conflict of interest.

References

[1] Garcia J, Feng Y, Shen JJ, et al., 2020, A Comprehensive Study of Autonomous Vehicle Bugs. Proceedings of the 42nd ACM/IEEE International Conference on Software Engineering, 385–396.

[2] Lee TB, 2019, Autopilot Was Active When a Tesla Crashed into a Truck, Killing Driver. Accessed on December 30, 2025, https://arstechnica.com/cars/2019/05/feds-autopilot-was-active-during-deadly-march-tesla-crash/

[3] Levinson J, Askeland J, Becker J, et al., 2011, Towards Fully Autonomous Driving: Systems and Algorithms. *2011 IEEE Intelligent Vehicles Symposium*, 163–168.

[4] Koopman P, Wagner M, 2016, Challenges in Autonomous Vehicle Testing and Validation. *SAE International Journal of Transportation Safety*, 4(1): 15–24.

[5] Gietelink O, Ploeg J, De Schutter B, et al., 2006, Development of Advanced Driver Assistance Systems with Vehicle Hardware-in-the-loop Simulations. *Vehicle System Dynamics*, 44(7): 569–590.

[6] Araujo H, Mousavi MR, Varshosaz M, 2023, Testing, Validation, and Verification of Robotic and Autonomous Systems: A Systematic Review. *ACM Transactions on Software Engineering and Methodology*, 32(2): 1–61.

[7] Wang J, Huang Y, Chen C, et al., 2024, Software Testing with Large Language Models: Survey, Landscape, and Vision. *IEEE Transactions on Software Engineering,* 50(4): 911–936.

[8] Chen L, Teng S, Li B, et al., 2023, Milestones in Autonomous Driving and Intelligent Vehicles—Part II: Perception and Planning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(10): 6401–6415.

[9] Betz J, Betz T, Fent F, et al., 2023, TUM Autonomous Motorsport: An Autonomous Racing Software for the Indy Autonomous Challenge. *Journal of Field Robotics*, 40(4): 783–809.

[10] Rejali S, Aghabayk K, Esmaeli S, et al., 2023, Comparison of Technology Acceptance Model, Theory of Planned Behavior, and Unified Theory of Acceptance and Use of Technology to Assess a Priori Acceptance of Fully Automated Vehicles. *Transportation Research Part A: Policy and Practice*, 2023(168): 103565.

[11] Chen Y, Chen ST, Zhang T, et al., 2018, Autonomous Vehicle Testing and Validation Platform: Integrated Simulation System with Hardware in the Loop. In *2018 IEEE Intelligent Vehicles Symposium*, 949–956.

[12] Deng WW, Lee YH, Zhao AN, 2008, Hardware-in-the-loop Simulation for Autonomous Driving. *Proceedings of the 34th Annual Conference of IEEE Industrial Electronics*, 1742–1747.

[13] Martinet P, Roux OH, 2018, A Formal Approach for the Design of a Dependable Perception System for Autonomous Vehicles. *Proceedings of the 21st International Conference on Intelligent Transportation Systems*, 2452–2459.

[14] Li L, Huang WL, Liu YH, et al., 2016, Intelligence Testing for Autonomous Vehicles: A New Approach. *IEEE Transactions on Intelligent Vehicles*, 1(2): 158–166.

[15] Geiger A, Lenz P, Stiller C, et al., 2013, Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

[16] Li W, Pan CW, Zhang R, et al., 2019, AADS: Augmented Autonomous Driving Simulation Using Data-driven Algorithms. *Science Robotics*, 4(28): eaaw0863. https://doi.org/10.1126/scirobotics.aaw0863

[17] Gietelink OJ, Verburg DJ, Labibes K, et al., 2004, Precrash System Validation with PRESCAN and VEHIL. *2004 IEEE Intelligent Vehicles Symposium*, 913–918.

[18] Helle P, Schamai W, Strobel C, 2016, Testing of Autonomous Systems—Challenges and Current State-of-the-Art. *INCOSE International Symposium*, 26(1): 571–584.

[19] Isele D, Rahimi R, Cosgun A, et al., 2018, Navigating Occluded Intersections with Autonomous Vehicles Using Deep Reinforcement Learning. *Proceedings of 2018 IEEE International Conference on Robotics and Automation*, 2034–2039.

[20] Jha S, Raman V, 2016, Automated Synthesis of Safe Autonomous Vehicle Control Under Perception Uncertainty. *The 8th International Symposium on NASA Formal Methods*, 117–132.

[21] Jo K, Kim J, Kim D, et al., 2014, Development of Autonomous Car—Part I: Distributed System Architecture and Development Process. *IEEE Transactions on Industrial Electronics*, 61(12): 7131–7140.

[22] Jo K, Kim J, Kim D, et al., 2015, Development of Autonomous Car—Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture. *IEEE Transactions on Industrial Electronics*, 62(8): 5119–5132.

[23] Joshi A, 2017, Powertrain and Chassis Hardware-in-the-loop (HIL) Simulation of Autonomous Vehicle Platform. SAE Technical Paper No. 2017-01-1991.

[24] Jöckel L, Kläs M, Martínez-Fernández S, 2019, Safe Traffic Sign Recognition Through Data Augmentation for Autonomous Vehicles Software. *Proceedings of the 19th IEEE International Conference on Software Quality, Reliability and Security Companion*, 540–541.

[25] Ko MK, 2015, *Autonomous Vehicle Driving Support System and Autonomous Driving Method Performed by the Same*. USA Patent No. 14/132,249.

[26] Lin SC, Zhang YQ, Hsu CH, et al., 2018, The Architectural Implications of Autonomous Driving: Constraints and Acceleration. *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems*, 751–766.

[27] Koopman P, Wagner M, 2017, Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1): 90–96.