CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

From Coarse Sampling to Insight: An Intelligent Testing Technique for Autonomous Haulage Systems

Master's Thesis in Computer science and engineering

Asiya Ismail Mohammed

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's Thesis 2025

From Coarse Sampling to Insight: An Intelligent Testing Technique for Autonomous Haulage Systems

Asiya Ismail Mohammed

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

From Coarse Sampling to Insight: An Intelligent Testing Technique for Autonomous Haulage Systems
Asiya Ismail Mohammed

© Asiya Ismail Mohammed, 2025.

Supervisor: Chih-Hong Cheng, Department of Computer Science and Engineering
Examiner: Christian Berger, Department of Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

From Coarse Sampling to Insight: An Intelligent Testing Technique for Autonomous Haulage Systems
Asiya Ismail Mohammed
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The mining industry is undergoing rapid transformation through the adoption of autonomous haulage systems (AHS), which promise improved safety, efficiency, and productivity. However, the testing of these systems remains a critical yet underdeveloped area. Based on a thematic analysis of interviews with domain expert engineers, this thesis investigates the challenges of testing AHS and examines the use of intelligent testing in virtual simulations to support their development. Coarse grid-search testing, combined with adaptive sampling strategies and tree-based models, enables feature importance analysis, allowing us to isolate scenario parameters that are largely independent of system performance. This dimensionality reduction also facilitates the construction of easy-to-understand, shallower decision trees and simplified random forests that can identify parameter regions likely to violate system specifications. Additionally, we identify input pairs with minimal differences that result in maximal output discrepancies, helping developers and testers alike to quickly localize error-inducing components. Our initial results demonstrate the value of intelligent testing as a proactive tool for uncovering edge cases, going beyond conventional scenario replay and widely used expert-driven methods. This thesis contributes to the development of a more robust and systematic testing framework for AHS, aiming to improve their reliability and operational safety.

Keywords: Autonomous Haulage System, Software Testing, Adaptive Sampling, Random Forest, Decision Tree, Combinatorial Testing, Clustering

Acknowledgements

I would like to express my sincere gratitude to my supervisors for their invaluable guidance and support throughout this project. Special thanks to my Chalmers University supervisor, Chih-Hong Cheng, for his insightful advice and encouragement. I am also deeply grateful to my Volvo supervisors, Maximilian Kneissl, Angelos Toytziaridis, and Stefan Kojchev, for their expert feedback and continuous support. This work would not have been possible without their contributions and trust. I would also like to thank Volvo Autonomous Solutions for sponsoring this thesis and providing the resources and environment that made this research possible. I also thank Volvo Autonomous Solutions for granting written permission to publish this thesis.

Asiya Ismail Mohammed, Gothenburg, June 2025

Declaration on the use of Generative AI

During this thesis work, the author used ChatGPT and Grammarly for grammatical corrections and to improve clarity of phrasing. These tools did not generate any substantial original content. All content, analysis, and conclusions are solely those of the author, who takes full responsibility for the publication.

Asiya Ismail Mohammed, Gothenburg, June 2025

Contents

List of Figures viii

List of Tables ix

1 Introduction 2
- 1.1 Background . . . 4
- 1.2 Problem Description . . . 4
- 1.3 Purpose of the Study . . . 5
- 1.4 Research Questions . . . 5
- 1.5 Scope . . . 6
- 1.6 Structure of the Thesis . . . 6
- 1.7 Significance of the Study . . . 7

2 Related Work 8
- 2.1 Autonomous Haulage Systems . . . 8
- 2.2 Challenges in Testing Autonomous Systems . . . 8
- 2.3 Testing Methods in Autonomous Systems . . . 9
- 2.4 Boundary Analysis for Failure Regions and Clustering Techniques . . . 10

3 Methods 12
- 3.1 Research Methodology . . . 12
- 3.2 Research Approach . . . 12
- 3.3 Experimental Setup . . . 13
  - 3.3.1 Independent Variables . . . 13
  - 3.3.2 Dependent Variables . . . 14
  - 3.3.3 Controlled Variables . . . 15
- 3.4 Testing Methods . . . 15
  - 3.4.1 Grid-search Testing . . . 15
  - 3.4.2 Random Testing: . . . 15
  - 3.4.3 Combinatorial Testing: . . . 16
  - 3.4.4 Adaptive Sampling and Decision Boundary Analysis: . . . 16
    - 3.4.4.1 Decision Tree and Random Forest Models . . . 16
    - 3.4.4.2 Handling Class Imbalance . . . 17
- 3.5 Nearest Neighbor Search: . . . 17
- 3.6 Data Collection and Analysis . . . 18
  - 3.6.1 Interviews . . . 18
  - 3.6.2 Input Space Coverage via Quantization . . . 19

3.6.2.1 Visualization Techniques 20
3.6.3 Evaluation and Stopping Criteria 20

4 Results **22**
4.1 Thematic Analysis and Expert Interviews 22
4.1.1 Environmental and Operational Challenges 22
4.1.2 Simulation Fidelity and Limitations 24
4.1.3 Reliance on Expert Judgment and Need for Systematic and Structured Testing 25
4.1.4 Summary 26
4.2 Simulation-Experiments 27
4.2.1 Obstacle Avoidance Test Scenario 27
4.2.2 Open Area Planner Test Scenario 28
4.3 Evaluation of grid-search testing and random testing 29
4.3.1 Note on Combinatorial Testing 30
4.4 Adaptive Sampling with Decision Tree and Random Forest Models . . 30
4.4.1 Feature Importance Analysis 30
4.4.1.1 Obstacle Avoidance Scenario 31
4.4.1.2 Open Area Planner Scenario 31
4.4.2 Decision Boundary Visualization 32
4.4.3 Evaluating the Adaptive Suite Against a Baseline 34
4.5 Transition Points and Output Drops 35

5 Discussion **38**
5.1 Addressing the Research Questions 38
5.1.1 RQ1 38
5.1.2 RQ2 39
5.1.3 RQ3 40
5.1.3.1 Adaptive Sampling and K-means Clustering 40
5.2 Limitations 41
5.2.1 Construct Validity 42
5.2.2 Internal Validity 42
5.2.3 External Validity 42
5.3 Future Work 43

6 Conclusion **44**

Bibliography **45**

A Appendix **II**

B appendix **V**
B.1 Model Performance Details V

List of Figures

1.1 AHS from Volvo Autonomous Solutions 3

3.1 Overview of methodology steps taken in this study 14

4.1 Obstacle Avoidance Test Scenario 28
4.2 Open Area Test Scenario
Note: The polygon has been modified for confidentiality, but the rest of the scenario reflects the actual test conditions. 29
4.3 Decision boundary visualization for Obstacle Avoidance scenario using adaptive sampling. 33
4.4 Decision boundary visualization for Open Area Planner scenario using adaptive sampling. 34
4.5 Feature importance of input parameters from Random Forest for the Obstacle Avoidance scenario 36
4.6 Feature importance of input parameters from Random Forest for the Open Area Planner scenario 36

List of Tables

4.1 Overview of Interview Participants 22
4.2 Summary of Themes from Thematic Analysis 23
4.3 Comparison Table of primary testing methods - Obstacle Avoidance . 30
4.4 Comparison Table of primary testing methods - Open Area Scenario . 30
4.5 Comparison of Adaptive Test Suite vs. Baseline (Volvo Autonomous Solutions) Test Suite 35
4.6 Pairs of nearly identical test inputs with different outcomes – Open Area Scenario. Colored values indicate changed inputs. Input Difference shows the specific change. 35
4.7 Pairs of nearly identical test inputs with different outcomes – Obstacle Avoidance scenario. Colored values indicate changed inputs. 37

5.1 Summary of how Research Questions were addressed 41
5.2 Comparison of Testing Methods applied in this thesis 41

B.1 Comparison of Random Forest and Decision Tree models for Open Area Planner Scenario: confusion matrices, classification reports, and feature importance. V
B.2 Feature importance for obstacle avoidance scenario using Random Forest and Decision Tree models. VI
B.3 Regression metrics for obstacle avoidance scenario using Random Forest and Decision Tree models. VI

1

Introduction

Mining provides raw materials essential to industries such as construction, energy, and manufacturing, thereby contributing to economic growth and development. Although mining is an essential industry that drives global economic development, it is also one of the most hazardous industries [20]. Mine workers face many dangerous health issues, such as skin conditions and respiratory diseases. The need to reduce these health risks and improve efficiency within this sector has therefore become a vital goal. These risks are amplified by traditional mining systems that have relied mainly on human operators for operations involving drilling rigs and haul trucks, which place workers in hazardous situations. Although semi-automated systems improve the safety of mining workers, they still require a high level of oversight to function effectively. Thus, these systems, while an improvement to the traditional systems, lack the advanced capability of a fully automated system that requires no human intervention and can perform complex tasks in even the harshest conditions. To address these challenges, Autonomous Haulage Systems (AHS) are increasingly being adopted in modern mining operations. As shown in Fig. 1.1, AHS typically consist of autonomous haulage trucks capable of navigating complex, semi-structured mining environments. Despite their potential to improve productivity, safety, and reduce operational costs, ensuring the reliability and safety of these systems remains a substantial challenge. Traditional testing methods, such as expert-driven scenario selection or replaying recorded scenarios, often fail to uncover critical edge cases and do not always provide engineers with actionable insights to isolate failure causes.

Building on the adoption of AHS, the automation of mining machinery has become a major focus in recent decades [29]. Autonomous systems have improved mining operations significantly, with Rio Tinto's Pilbara site achieving a 40 percent reduction in safety incidents and a 20 percent productivity boost. BHP's Jimblebar Mine also reported an 18 percent increase in ore throughput [20]. This shift is part of the broader Industry 4.0 movement. Industry 4.0 refers to the integration of smart technologies, such as automation, the Internet of Things (IoT), and artificial intelligence, into industrial processes. The mining industry leverages Industry 4.0 technologies, including cutting-edge autonomous haulage trucks capable of fully driverless operation. These systems carry up to 400 tons of ore and can operate entirely autonomously [15]. Furthermore, these systems are expected to enable continuous, round-the-clock operations, which can lead to increased productivity and efficiency in mining transport activities.

Figure 1.1: AHS from Volvo Autonomous Solutions

A notable gap exists in the current research landscape, as there is a lack of studies on developing and implementing effective testing strategies for AHS. Inspired by simulation-based testing techniques originally developed for urban autonomous driving, this thesis investigates how intelligent virtual simulation methods can better support the development and validation of AHS. Specifically, it aims to evaluate the effectiveness of grid-search testing, combinatorial testing, adaptive sampling, and random testing, as these methods collectively cover both exhaustive and exploratory testing strategies. Grid-search and combinatorial testing offer systematic coverage of the input space, while adaptive sampling and random testing support the discovery of failure-prone or underexplored regions through more dynamic or stochastic means. This combination allows for a comprehensive investigation of both common and edge-case behaviors in autonomous systems. A detailed explanation of each method, along with relevant literature support, is provided in Section 2.3. To improve test efficiency, this study proposes filtering out irrelevant input scenario parameters that have a negligible impact on AHS decision-making to reduce the dimensionality of the input space and focus testing efforts on the most influential factors. Additionally, it explores how clustering techniques and machine learning models, such as decision tree and random forest models, can enhance adaptive sampling, ultimately improving the efficiency and reliability of AHS testing.

1.1 Background

Significant research has been done to study the development and testing of other autonomous systems, such as urban autonomous driving systems. Testing methodologies such as grid-search testing, combinatorial testing, adaptive sampling, and random testing have been extensively studied in academia [34, 30, 45, 13]. These techniques have been analyzed and enhanced through research, with their practical applications increasingly seen in industry, such as collaborative projects with Volvo Cars on autonomous vehicle testing [34]. However, the application of these methodologies in the testing of AHS has not yet been explored or reported. The harsh conditions and rough terrains in mining fields introduce a new complexity that these methodologies have yet to address effectively. AHS are currently a new trending topic that the mining industry is focused on developing. These self-driving vehicles will work longer hours without requiring breaks or shift changes. Additionally, these systems are equipped to operate in rougher conditions than a human mine worker. The mining sector is said to have become 15% safer [20] with the rise of these autonomous transport systems. Improving testing techniques and the overall quality of AHS will improve the safety of human lives and the quality of life for mining workers while simultaneously increasing the efficiency of the work and delivering economic benefits.

Theories Guiding this Research: The study will build on the theories of autonomous vehicle testing and testing methods used in these systems. These methods include grid-search testing, adaptive sampling, combinatorial testing, and random testing. In addition, the research incorporates concepts from dimensionality reduction and machine learning to address the complexity of high-dimensional input spaces. Specifically, filtering techniques will be used to eliminate input parameters that have negligible influence on system behavior, while clustering methods and boundary analysis will support the identification of failure regions and decision boundaries. These theoretical foundations are adapted and extended to meet the unique challenges associated with testing AHS in mining environments.

1.2 Problem Description

The testing of AHS remains a largely unexplored area, with no universally accepted or standardized evaluation framework across the mining industry. While companies implement their own internal testing protocols and comply with regulatory standards, a common best-practice methodology has yet to be established. Effective testing is critical, not only to ensure the safety of mine workers, but also to protect mining companies from costly equipment failures and potential financial losses. Moreover, autonomous solution providers risk reputational damage and loss of client trust if their systems exhibit faults in operation. As discussed in the Introduction, traditional testing approaches, particularly expert-driven testing, frequently fail to uncover critical edge cases or provide engineers with actionable insights to isolate root causes of failure. These limitations have been highlighted by several studies that emphasize the subjectivity and incompleteness of manually designed tests, especially in complex

or unpredictable domains like autonomous systems [32, 44, 34]. This highlights an urgent need for intelligent testing methods capable of systematically identifying and focusing on the most relevant input parameters and challenging scenarios. This thesis addresses these challenges by proposing novel techniques designed to improve the safety, reliability, and efficiency of AHS testing.

1.3 Purpose of the Study

This study first identifies the key challenges in testing AHS, emphasizing the complexities involved in ensuring their reliability and safety. It then investigates the applicability and effectiveness of scenario sampling methodologies such as grid-search testing, random testing, combinatorial testing, and adaptive sampling to improve test scenario selection for AHS. Specifically, the study focuses on identifying critical failure scenarios and decision boundaries, defined here as regions in the input space where small changes in inputs cause significant changes in system behavior. Finally, it explores how clustering techniques and boundary analysis can be utilized to detect groups of similar failing scenarios. To improve testing efficiency, filtering techniques will be incorporated to eliminate irrelevant input parameters that have minimal impact on system behavior, thereby reducing input dimensionality and focusing test efforts on influential factors. By evaluating these techniques, this research aims to contribute to the development of a more robust, efficient, and reliable testing framework for AHS, ultimately improving safety and operational performance in mining environments.

1.4 Research Questions

This study focuses on identifying key challenges in testing AHS and evaluating the effectiveness of various testing methodologies. It also explores how clustering techniques and tree-based models can support adaptive sampling. The research is guided by the following questions:

RQ1: *What are the key challenges in testing AHS, and how can critical scenarios be identified?* This question aims to uncover the unique difficulties in testing AHS and to determine how critical failure scenarios can be effectively identified.

RQ2: *How can the testing methods grid-search testing, combinatorial testing, random testing, and adaptive sampling be applied to identify critical failure scenarios and decision boundaries in AHS?* This question investigates how different testing techniques can be utilized to assess the reliability and robustness of AHS by identifying critical failure scenarios.

RQ3: *How can tree-based models, clustering techniques, and nearest neighbor search methods be utilized to refine scenario selection and guide adaptive sampling strategies in AHS?* This question investigates how clustering and tree-based models can focus testing on high-risk areas of the input space, thereby reducing redundant tests and improving the likelihood of detecting failures.

1.5 Scope

This thesis investigates testing processes for AHS, aiming to enhance failure detection and improve coverage of the input space. The research is conducted using simulated environments designed to represent operational conditions of autonomous haulage trucks within mining contexts. The perspective adopted is that of AHS developers and testers, focusing on their practical needs for effective testing and validation in mining operations. The test scenarios used in this study are simulations modeled closely on real mining operations and challenges, specifically those provided by Volvo Autonomous Solutions for their system verification.

The conclusions of this thesis are limited to the specific testing scenarios, methods, and system configurations studied, including grid-search, combinatorial testing, adaptive sampling with tree-based models, and random testing. Although our results show promise within structured, industrial mining environments, their applicability to autonomous vehicles in highly dynamic or open-world settings such as urban areas remains uncertain. Additionally, more complex hybrid or AI-driven testing strategies beyond the adaptive sampling approaches explored here are outside the scope of this research.

1.6 Structure of the Thesis

The thesis is organized as follows:

- **Chapter 1 – Introduction:** Introduces foundational background and concepts relevant to autonomous system testing and AHS, outlining the motivation and goals of this research.
- **Chapter 2 – Related Work:** Reviews existing literature on autonomous system testing methods and challenges, with a focus on AHS, clustering techniques, and nearest neighbor methods for identifying input/output pairs.
- **Chapter 3 – Methods:** Describes the research methodology, experimental setup, and data collection procedures used to evaluate testing techniques, interviews and interview analysis, clustering, and boundary analysis.
- **Chapter 4 – Results:** Presents findings from applying grid-search, random, combinatorial, and adaptive sampling methods to identify critical failure scenarios, alongside figures and tables demonstrating the use of clustering and nearest neighbor search to find closely related input/output pairs with distinct outputs.
- **Chapter 5 – Discussion:** Interprets results, discusses limitations, and proposes directions for future research and improvements.
- **Chapter 6 – Conclusion:** Summarizes the main contributions and overall significance of the research outcomes.

1.7 Significance of the Study

Overall, the aim of this thesis is to first identify the key challenges of testing AHS and then study the applicability and effectiveness of different testing strategies on AHS. These findings are expected to improve the quality of AHS by allowing validation and verification engineers working on these systems to be more efficient. In particular, this study further contributes by identifying minimally different input scenarios that produce distinct outcomes, revealing critical decision boundaries. These contributions support safer and more reliable AHS deployment.

2

Related Work

2.1 Autonomous Haulage Systems

Automation of machinery is receiving increasing attention, especially in sectors such as mining, where there is often a high risk of serious or fatal accidents [38]. Miners face hazardous situations daily, such as health risks and the potential for vehicle collisions, highlighting the importance of autonomous systems. These autonomous systems have many advantages, including greater productivity, more efficient resource use, support for infrastructure maintenance, disaster management, and enhanced safety [27]. Beyond these advantages, these systems also offer potential economic benefits, such as reduced wear on components and streamlined operations [29].

While AHS have been extensively developed for productivity and safety improvements, there is still limited research focusing specifically on the testing and debugging of these systems. Emerging approaches, such as PMWorld [4], a virtual testing platform, and Parallel Mining [10], which leverages digital twins and diversified test scenarios, represent early steps toward this goal. Although there exist attempts to create technical specifications for autonomous mining [16], including proposals for maps [37, 42], testing methods are often not included in the specifications. Additionally, these solutions lack support for isolating parameter sensitivity or guiding engineers in debugging. Our work builds on these foundations and focuses on applying and evaluating advanced testing techniques to improve sample efficiency, test coverage, and insight generation for mining-specific AHS.

2.2 Challenges in Testing Autonomous Systems

The major challenge in autonomous systems is to ensure that they are sufficiently safe and they are developed in such a way that they can be approved for real-world use [25]. Since autonomous systems are, by design, not usually under human control or supervision, ensuring their behavior is safe is vital. This is especially important since "society holds robots to a higher standard and has a lower tolerance for their errors" [17]. Testing urban autonomous driving systems has been an active research field, and many autonomous driving safety standards, such as ISO 21448 SOTIF (Safety of Intended Functionality) [1], consider simulation-based testing to be crucial. Common practices include recording and replaying real-world scenarios from

databases like the NHTSA accident database [36], Pegasus [48], and SafetyPool [2]. Aside from scenario replay, programmatic scenario generation tools such as Scenic [14] and Paracosm [28] enable specifying and automatically creating diverse test scenarios.

Another challenge is that testing can become difficult when we cannot easily identify whether or not the system under test (SUT) delivers the correct result. This is particularly true for autonomous systems, where it is challenging to assess if the behavior of a safety-critical autonomous system aligns with its specifications. Autonomous systems are therefore difficult to test in a systematic manner because it is difficult to determine if these systems behave as expected [24]. A further critical challenge in testing autonomous systems is ensuring that the virtual test environment accurately and comprehensively represents the system's intended operational environment, often referred to as the operational design domain (ODD). The ODD can be vast and complex, encompassing diverse environmental conditions, dynamic agents, and rare edge cases [17]. Achieving sufficient coverage of the ODD is essential to build confidence in system safety prior to deployment.

Exhaustively testing across this immense state space is infeasible, so advanced scenario sampling, coverage metrics, and combinatorial testing techniques are employed to systematically explore meaningful regions of the ODD and uncover potential failures. Optimization techniques like genetic algorithms [19, 12], fuzzing [47], Gaussian Process-based adaptive sampling [30], Bayesian optimization [39], and multi-objective bandits [40] help efficiently explore the vast and complex scenario space. This line of research is strongly related to CPS falsification techniques [5, 7, 22, 11, 46, 6]. Additionally, tools such as ComOpT utilize constraint-enabled combinatorial testing to argue relative coverage of the ODD and guide local heuristic search to identify problematic scenarios [23]. However, even with these advanced methods, interpreting test results and isolating the causes of failures remain challenging for developers and testers. Addressing these challenges requires a range of testing methods, often implemented within simulation-based environments, to systematically generate and evaluate diverse scenarios that capture critical system behaviors.

2.3 Testing Methods in Autonomous Systems

Building on the challenges outlined above, testing autonomous systems is critical in ensuring their safety and reliability. There are many well-known testing methods and approaches used to validate autonomous systems and address testing challenges. These methods are often implemented within simulation environments and include scenario-based techniques that generate test cases to capture diverse system behaviors. The knowledge-based testing approach focuses on leveraging expert knowledge to generate, extract, and select scenarios for testing [34]. The data-driven testing approach, on the other hand, uses data from recent system performance records to define and create test scenarios [34].

The scenario-based testing approach aims to create test cases that simulate different system behaviors, either through random selections, specific algorithms, or systematic

coverage of input parameters. The goal is to explore a broad range of test cases, including edge cases, to ensure robust testing of the system's performance. Scenario-based testing is highly accepted and plays a key role in validating the safety of autonomous driving systems. It aims to reduce the test effort to a manageable number of scenarios by limiting the testing to meaningful scenarios based on the testing budget [34]. Some examples of testing methods that use this approach are random testing, grid-search testing, combinatorial testing, and adaptive sampling.

Random testing helps assess a broader range of potential conditions. Random testing is typically applied by selecting random parameters or values for the test cases used in the simulation [13]. Grid-search testing is a systematic method that explores all possible combinations of input parameter values within a predefined range. It ensures comprehensive coverage of the parameter space, but can become computationally expensive as the number of parameters increases. To mitigate the risk of combinatorial explosion, where the number of test cases increases exponentially, we limit the range of parameters and adjust the step size between input values. By selecting a subset of the grid, we reduce the number of test cases while maintaining broad coverage, allowing for efficient exploration of the input space. Previous studies have discussed the challenges of combinatorial explosion in exhaustive testing strategies [45], but with proper management of the parameter space, grid-search remains a powerful tool for thorough testing.

Combinatorial testing is very good at covering critical scenario combinations without concern for the combinatorial explosion in grid-search testing. Combinatorial testing is done by combining and adjusting multiple parameters to generate test cases for various scenarios [34]. Finally, adaptive sampling is a testing method that uses an iterative process in which, first, the SUT is tested systematically and exhaustively. The test results from the SUT are fed into a meta-model that will guide your search for the most insightful and critical scenarios [30]. This is a fast and efficient method guiding testers to find critical cases and to identify those cases where the system's performance dips suddenly (for example, from passing test to failing test). All these testing methods are important in addressing the complex challenges of autonomous systems. The environment can be unpredictable, and vehicles like urban autonomous driving systems must interact with pedestrians, other drivers, and various road conditions [18]. These methods can ensure the effectiveness of the test scenario generation by focusing on the critical cases [35]. In this thesis, we apply all these methods when developing a testing framework for AHS, combining efficient coverage from combinatorial, random, and grid-search testing, while also adapting to discover edge cases effectively using adaptive sampling techniques.

2.4 Boundary Analysis for Failure Regions and Clustering Techniques

Clustering techniques play a crucial role in identifying failure regions in autonomous systems. By grouping similar failure cases, these methods help in understanding

patterns in system failures and analyzing decision boundaries. Nearest neighbor search (NNS) has gained significant attention in various research areas, as it provides an efficient way to analyze and search large datasets. NNS is widely recognized as one of the most effective techniques for handling data search problems and has been applied to a wide range of fields, including pattern recognition, multimedia data searching, information retrieval, databases, data mining, and computational geometry [3]. By identifying the closest pairs of data points, NNS techniques help in detecting critical transition points and identifying failure regions, making it highly applicable in performance analysis and failure detection. In our work, we utilize NNS to compare pairs of test cases that are close in the input space but show divergent outputs. This enables the detection of regions where the system's behavior is highly sensitive or unstable. For example, two nearly identical inputs might lead to drastically different vehicle trajectories or safety scores, signaling a brittle or under-tested region of the model. Identifying such pairs allows us to highlight boundary cases and transition regions between safe and unsafe behavior, which are often hard to detect using traditional coverage metrics. This technique helps isolate root causes of failure and guides engineers in debugging by focusing their attention on high-risk transitions in the input space.

Decision tree models are also widely used for identifying failure regions, as they provide an interpretable way to classify test cases based on decision rules. The models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition [26]. These trees split the input space into different failure and non-failure regions, allowing for clear insight into what factors contribute to failures. Building on this, random forest models combine multiple tree predictors, each trained on a random subset of data and features. Each tree depends on a random vector sampled independently, and identically distributed for all trees, which enhances the diversity and strength of the ensemble [9]. In this study, random forest models are employed to classify failure regions, benefiting from their internal error estimates, variable importance measures, and strong performance in high-dimensional, noisy environments. K-means clustering is also used in this study and it is a widely used method for failure region analysis due to its simplicity and efficiency. It is one of the simplest unsupervised learning algorithms that solves the well-known clustering problem [21]. It partitions the input space into clusters and is useful for analyzing different modes of failure, though it may struggle with non-uniformly distributed failure cases. Despite its simplicity, K-means may require adjustments, as it is often affected by initializations and necessitates specifying the number of clusters a priori [33]. In this thesis, these techniques (NNS, decision trees, random forests, and K-means) are integrated into our testing framework for AHS. Together, they support the identification of boundary cases, provide interpretable failure classifications, and enable efficient clustering of failure modes for robust and scalable test analysis.

3

Methods

This chapter outlines the methodological approach taken in this thesis. It describes the research methodology, the experimental setup, and the procedures used for data collection and analysis. Figures, tables, and formulas are included where appropriate to support and clarify the methods applied.

3.1 Research Methodology

This research investigates how different testing techniques (random testing, grid-search testing, combinatorial testing, and adaptive sampling) can improve the verification process of AHS. The purpose of the study is to assess the effectiveness of these techniques in identifying failure test cases, failure regions in the input space, decision boundaries, and in improving input space coverage. We define the scope of the study using the template given by Wohlin et al. [43]. Figure 3.1 is a flowchart depicting the steps taken in the thesis to answer our research questions.

Analyze AHS testing processes,
for the purpose of improving failure detection and input space coverage,
with respect to identifying critical failure cases,
from the point of view of AHS developers and testers,
in the context of autonomous haulage vehicles.

3.2 Research Approach

This study follows the Design Science Research (DSR) paradigm. DSR is a problem-solving paradigm that aims to increase human knowledge by the creation of new and innovative artifacts [41]. The artifact that is being created in this research is an improved AHS testing framework integrating systematic and adaptive sampling strategies. The DSR process in this study consists of:

- Problem Identification and Motivation (RQ1): Expert interviews were conducted to identify the key testing challenges in AHS. Insights from these interviews guided the identification of the core issues the testing methodology needs to address.

- Objectives of the Solution (RQ2, RQ3): The goal is to design an improved testing framework that effectively identifies collections of test cases and enhances AHS reliability via adaptive sampling test case generation.
- Design and Development: Implementing and evaluating different testing methods (grid-search testing, combinatorial testing, random testing, and adaptive sampling).
- Demonstration: The framework was implemented in an AB Volvo simulation environment to evaluate its applicability in real-world AHS testing.
- Evaluation: Assessing the testing methods' effectiveness using performance metrics such as failure detection, execution time, and input space coverage.
- Reflection and Refinement: Analyzing results with tree-based models and clustering techniques to refine the testing approach.

To establish problem relevance, we conducted expert interviews with industry professionals to gain insights into the unique challenges of testing AHS. These insights guided the design of our simulation-based experimental study, which incorporated grid-search testing and random testing as primary testing methods, with combinatorial testing also applied to explore its potential benefits. To identify and analyze failure-prone regions, decision tree and random forest models formed the core of the adaptive sampling approach, guiding the adaptive iteration to identify critical failure regions where a refined grid search enhanced test case selection. Additionally, clustering techniques—including k-means and manual clustering—were applied to provide supplementary insights and improve visualization of different success and failure regions within the input space.

3.3 Experimental Setup

This study focuses on AHS test scenarios, utilizing simulation-based evaluation primarily to detect system issues early, before proceeding to more costly testing frameworks. A digital simulation environment eliminates external influences from physical hardware while allowing for systematic and repeatable testing. The primary independent variables evaluated against the dependent variables are grid-search testing and random testing. Combinatorial testing and adaptive sampling were also applied; however, their evaluation focused on exploratory or supplementary analyses rather than on all performance metrics. The variables involved in the experiments are categorized as follows.

3.3.1 Independent Variables

Independent variables represent the factors that are manipulated or varied by the researcher to observe their effect on the system under test. In this experiment, they are as follows:

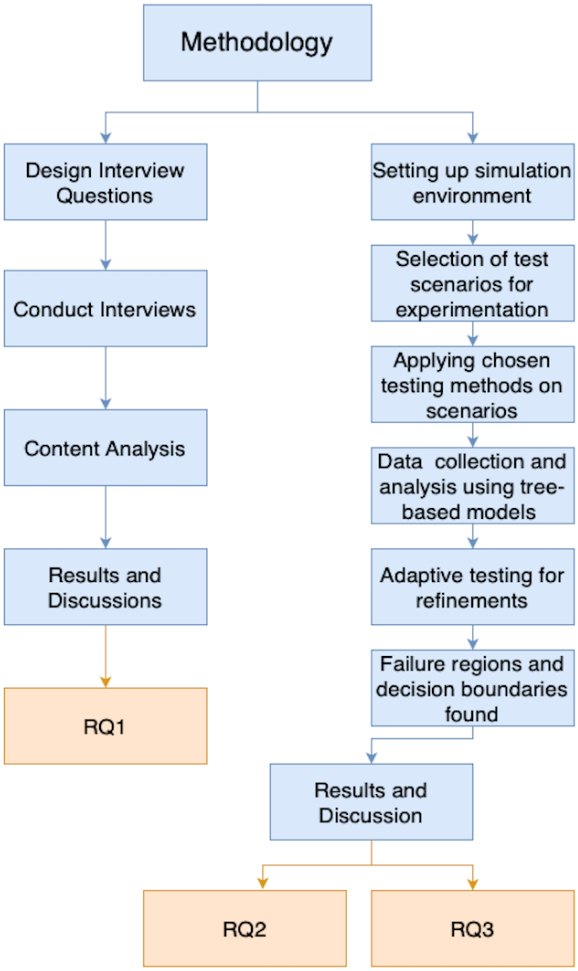

Figure 3.1: Overview of methodology steps taken in this study

1. Grid-search testing
2. Random testing

3.3.2 Dependent Variables

Dependent variables are the observed outcomes or responses that are measured to assess the effect of the independent variables. In this experiment, they are as follows:

1. Failure Cases Found: The number of detected failure cases in the test suite run for each method.
2. Input Space Coverage: How well each method explores the full range of input values. This was measured by dividing the space into grid cells and checking how many were covered.
3. Test Execution Time: The total time to complete all test cases.

3.3.3 Controlled Variables

Controlled variables are factors kept constant throughout the experiment to ensure a fair comparison and to eliminate confounding effects. In this experiment, they are as follows:

- Simulation Configuration: All methods were tested in the same simulated AHS environment to ensure a fair comparison.

3.4 Testing Methods

This thesis applied the following testing methods: grid-search testing, random testing, combinatorial testing, and adaptive sampling. These methods were selected to cover a range of test case generation strategies, from systematic to stochastic, and to explore adaptive approaches that focus testing efforts on failure-prone areas. Each method offers unique strengths in detecting failures and improving input space coverage in AHS testing.

3.4.1 Grid-search Testing

Grid-search testing systematically partitions the input space by dividing each input variable x, defined over the range $[l, u]$, into N equal intervals, resulting in $N + 1$ discrete test values. Formally, the test set T_x corresponding to variable x is defined as:

$$T_x \stackrel{\text{def}}{:=} \left\{ l,\ l + \frac{u-l}{N},\ l + 2\frac{u-l}{N},\ \ldots,\ l + (N-1)\frac{u-l}{N},\ u \right\}. \tag{3.1}$$

Assume we have n input variables, indexed as $x_1, x_2, \ldots, x_n$, where each x_i is discretized into a test set T_{x_i} using Equation (3.1). This partitioning ensures systematic coverage of the variable's range. The overall grid of test inputs is then constructed as the Cartesian product of such discrete partitions across all input variables.

3.4.2 Random Testing:

Random testing was used to sample input vectors from continuous parameter spaces. Unlike grid-search and combinatorial testing, which rely on discretized (countable) sets of input values, random testing samples from continuous intervals, which form an uncountably infinite space. For each input variable x_i, a value was sampled from a uniform distribution over its continuous range:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n], \quad \text{with each } x_i \sim \mathcal{U}(x_{i,\min}, x_{i,\max}). \tag{3.2}$$

All testing methods, except adaptive sampling, shared the same minimum and maximum limits for each input variable, ensuring that although the sampling strategies

differed, the overall input space explored remained consistent across methods. Multiple iterations were performed with different random seeds to ensure diverse and unbiased sampling across the entire continuous space. Both random and grid-search testing were executed using equivalent test sizes, and their results were analyzed to assess failure detection effectiveness, time taken, and input space coverage.

3.4.3 Combinatorial Testing:

Using the discrete values defined by the discretization in Equation (3.1), 2-way combinatorial testing ensures that for every pair of parameters (x_i, x_j), all combinations of their discrete values appear in at least one test case. Formally, the pairwise coverage criterion requires that for each $i < j$, the test set T satisfies:

$$\forall a \in D_i, \forall b \in D_j, \exists \mathbf{x} \in T \text{ such that } x_i = a \text{ and } x_j = b, \tag{3.3}$$

where $D_i = T_{x_i}$ and $D_j = T_{x_j}$ are discrete test sets constructed as per Equation (3.1).

3.4.4 Adaptive Sampling and Decision Boundary Analysis:

To identify failure-prone regions and begin adaptive sampling, we trained both Decision Tree and Random Forest models on test data generated via grid-search and random testing. The input to these models was the vector of input parameters $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, and the output was either a binary pass/fail (classification models) label or a continuous performance metric y (regression models). This analysis informed an adaptive sampling strategy focused on refining test coverage near critical decision boundaries.

3.4.4.1 Decision Tree and Random Forest Models

Decision trees learn a hierarchy of decision rules by recursively partitioning the input space based on feature splits that minimize variance (in regression) or impurity (in classification). Random Forest extends this approach by training an ensemble of decision trees on bootstrapped subsets of the data and aggregating their predictions. This helps to improve model robustness and capture nonlinear feature interactions. Key hyperparameters such as maximum tree depth, minimum samples per leaf, and the number of trees (for Random Forest) were left at their scikit-learn default values. Scikit-learn is a widely used open-source Python library that provides efficient implementations of many machine learning algorithms for both supervised and unsupervised learning tasks. Scikit-learn is designed to bring machine learning tools to non-specialists through a user-friendly Python API, and is commonly used in both academic and industrial applications due to its balance of performance and usability [31]. Models were trained using an 80/20 split of the dataset to assess generalization. Feature importance measures how much each input parameter contributes to the predictive power of the model. Feature importance metrics extracted from these models revealed which input parameters most strongly influenced failure predictions.

Model performance was evaluated using the following metrics:

- For regression tasks, we used Mean Absolute Error (MAE), which measures the average absolute difference between predicted and true values, indicating prediction accuracy, and R-squared (R^2), which represents the proportion of variance in the target variable explained by the model.

- For classification tasks, we assessed performance with accuracy, measuring the overall proportion of correct predictions; precision, indicating the proportion of true positive predictions among all positive predictions; recall, representing the proportion of true positives detected among all actual positives; and the F1-score, which is the harmonic mean of precision and recall, balancing the two. Additionally, we analyzed the confusion matrix, which summarizes true positives, true negatives, false positives, and false negatives, providing detailed insight into classification errors.

3.4.4.2 Handling Class Imbalance

In the Open Area scenario, Synthetic Minority Over-sampling Technique (SMOTE), a method that generates synthetic samples of the minority class to balance datasets, was applied to mitigate class imbalance and enhance the models' ability to learn from both passing and failing cases. Conversely, in the Obstacle Avoidance scenario, where failures comprised roughly 90% of the data, SMOTE was not used, as the imbalance reflected the goal of thoroughly exploring failure-prone regions. This contrast underscores the importance of aligning data balancing techniques with the specific objectives and characteristics of each testing scenario.

3.5 Nearest Neighbor Search:

To detect sensitive transition zones in the input space, where small changes in input cause large performance differences, we applied the NNS method in each adaptive iteration. The input space was split into two sets: - D_{success}: inputs that produced successful (passing) outcomes, - D_{fail}: inputs that produced failure outcomes. We computed the Euclidean distance between all cross-pairs of these sets:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \tag{3.4}$$

The nearest neighbor pair $(\mathbf{x}_s, \mathbf{x}_f)$ is defined as the pair across D_{success} and D_{fail} with the minimum distance:

$$(\mathbf{x}_s, \mathbf{x}_f) = \arg \min_{\mathbf{x}_i \in D_{\text{success}}, \mathbf{x}_j \in D_{\text{fail}}} d(\mathbf{x}_i, \mathbf{x}_j) \tag{3.5}$$

This pair represents the tightest observed boundary between success and failure, a point where the system is most sensitive. We computed the output change:

$$\Delta y = |y_s - y_f| \tag{3.6}$$

This quantifies how drastically the system reacts to a minimal input change. Importantly, we selected the closest success–failure pair, not the pair with the maximum Δy, because our goal is to localize the region where even small variations in input can flip the system's behavior. If multiple pairs are equally close, we rank them by their corresponding Δy values, i.e., the output change, prioritizing those with larger shifts. This reflects boundary sensitivity more effectively than just large output changes, which could arise from far-apart inputs.

Illustrative Example:

Suppose a system takes two parameters x_1 and x_2, and produces a binary outcome (pass/fail). Consider three test points:

- $\mathbf{x}_A = (1.0, 1.0) \rightarrow$ pass
- $\mathbf{x}_B = (1.1, 1.0) \rightarrow$ fail
- $\mathbf{x}_C = (3.0, 3.0) \rightarrow$ fail

Here,
- $d(\mathbf{x}_A, \mathbf{x}_B) = 0.1$ — smallest distance
- $d(\mathbf{x}_A, \mathbf{x}_C) = \sqrt{8} \approx 2.83$

Though both $\mathbf{x}_B$ and $\mathbf{x}_C$ are failures, only $\mathbf{x}_B$ shows that a tiny change from $\mathbf{x}_A$ results in failure. That's the kind of critical edge we seek with NNS. By integrating NNS into adaptive sampling, we can prioritize test regions near these sensitive boundaries and generate new test cases that probe the reliability limits of the system more effectively.

3.6 Data Collection and Analysis

This section outlines the methods used to collect and analyze data to support the evaluation of our testing framework for AHS. We begin by describing the qualitative insights gathered through interviews with domain experts, followed by the quantitative approaches used to measure input space coverage and assess testing effectiveness. Visualization and evaluation techniques are also detailed to provide a comprehensive overview of our analysis strategy.

3.6.1 Interviews

Prior to the simulation-based experiments, we performed semi-structured interviews with professionals in the field of AHS to gain insights into the unique challenges of testing AHS. The interviews were geared towards answering our first research question (RQ1), identifying key challenges in AHS testing, and guiding the development of

the testing framework in this thesis to focus on the most relevant and addressable challenges within this scope. Participants were selected through purposive sampling, guided by discussions with industry supervisors at Volvo Autonomous Solutions. The inclusion criteria required that participants worked closely in the development or testing of AHS. This resulted in a group of seven interviewees with diverse roles, including developers, verification and validation engineers, requirements managers, and product owners. No formal pilot testing of the interview guide was conducted. The questions were developed based on insights gathered from initial discussions with supervisors about AHS challenges and a review of relevant literature. Given these inputs and project time constraints, pilot testing was not deemed necessary. At the beginning of each interview, participants were informed about the purpose and focus of the study. Verbal consent was obtained to record the interviews and to use and review their responses for analysis. All responses were treated confidentially and reported anonymously to protect participant privacy. All participants were asked the same questions, but follow-up questions varied depending on their responses.

The interviews were divided into four main topics: Challenges in Testing AHS, Identifying Critical Performance Scenarios, Performance Validation and KPIs, and Desired Improvements in Testing. Interview data were analyzed using thematic analysis [8], following the six-phase framework: familiarization with the data, generating initial codes, searching for themes, reviewing themes, defining and naming themes, and producing the final report. This systematic approach enabled the identification, analysis, and reporting of patterns and meaningful themes across participants' responses. To support the credibility of the analysis, anonymized participant quotes illustrating key themes are presented in Section 4.1. The full interview guide used in this study is provided in Appendix A.

3.6.2 Input Space Coverage via Quantization

To quantify input space coverage, we employ a quantization-based approach that partitions the input space into uniform grid cells (or bins) defined by the resolution vector $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_n]$ used in the grid search. Each test point, regardless of the testing method, is mapped to a corresponding grid cell in the discretized input space. The input space coverage, C, is computed as the percentage of unique grid cells visited by at least one test sample:

$$C = \frac{N_{\text{visited}}}{N_{\text{total}}} \times 100\%, \tag{3.7}$$

where

- N_{visited} is the number of unique grid cells (bins) containing at least one test case.
- N_{total} is the total number of grid cells in the discretized input space.

This quantization approach allows a direct and unbiased comparison between different

testing methods, such as structured grid-search testing and stochastic random testing, since both are evaluated against the same discretized input space. It is important to note that achieving 100% coverage becomes increasingly difficult as the number of input parameters grows due to the combinatorial explosion of possible grid cells. This exponential growth in the size of the input space makes exhaustive testing infeasible for high-dimensional input spaces. Consequently, methods such as adaptive sampling become essential. Adaptive sampling strategically focuses testing efforts by sampling sparsely in regions with few detected errors and intensively in regions where errors are more frequent. This targeted approach improves testing efficiency by prioritizing critical regions of the input space, enhancing fault detection while mitigating the combinatorial explosion problem.

3.6.2.1 Visualization Techniques

To support interpretation of model behavior and failure regions, several visualization techniques were employed:

- **Feature importance bar charts:** these indicate which variables contributed the most to predictions.
- **2D and 3D input projections:** for example, velocity vs. obstacle distance vs. reverse, color-coded by pass and different failure categories.
- **Clustering techniques:** Unsupervised clustering methods such as k-means (which partitions data into k clusters based on feature similarity) and manual labeling were used purely for visualization purposes, helping to highlight patterns within failure cases in projected feature spaces.

These visualizations provided intuitive insights into input-output relationships and guided the refinement of adaptive sampling strategies near critical regions.

3.6.3 Evaluation and Stopping Criteria

Test results from the two primary testing methods, grid-search testing, and random testing, were analyzed to evaluate their effectiveness at detecting failures. Combinatorial (pairwise) testing was excluded from direct comparison on these metrics, as it was introduced later in the study and involved a smaller test suite, which would make the comparison unfair and inconsistent. The following key evaluation metrics were used:

- The number of failure cases detected per method.
- The breadth of input space coverage by each method.
- Time taken for each of these methods.

The iterative nature of adaptive sampling involves stopping when the testing process has converged. We define convergence as the point when the same failure-prone

regions are repeatedly identified in subsequent iterations, signaling that no new insights are gained from further testing.

4

Results

This chapter presents the results obtained from the conducted experiments and expert interviews. It outlines key findings, observations, and any patterns or insights that emerged during the course of this thesis.

4.1 Thematic Analysis and Expert Interviews

To support our simulation-based experiments, we conducted interviews with seven AHS professionals at Volvo Autonomous Solutions to gain a better understanding of the practical challenges in system testing (RQ1). Table 4.1 provides an overview of the participants, highlighting their roles, domains, and years of experience. Thematic analysis revealed key pain points such as environmental and operational challenges, limited test coverage, uncertainty around scenario criticality, and infrastructure limitations. Table 4.2 summarizes these themes, illustrating how practitioners' concerns informed the design of our testing methodology.

#	Position	Domain	Experience (years)
1	Product Owner	Automotive	9
2	Requirements Manager	Automotive	6.5
3	Software Developer	Automotive	6
4	Senior Software Engineer	Automotive	8.5
5	System Verification Engineer	Automotive	2
6	System Verification Engineer	Automotive	4
7	Scrum Master	Automotive	9

Table 4.1: Overview of Interview Participants

4.1.1 Environmental and Operational Challenges

The operational environment in which AHS function presents significant challenges for effective testing. Harsh conditions such as uneven terrain, extreme weather, and steep slopes create unique obstacles for autonomous systems deployed in mining contexts. These are not only difficult to simulate accurately but are also the conditions under which AHS must demonstrate robust and reliable behavior. Several interviewees emphasized that such environmental constraints remain a primary hurdle.

Theme	Description
Environmental and Operational Challenges	Harsh conditions like dust, terrain variability, and slopes make real-world testing unpredictable and difficult to replicate in simulation.
Simulation Fidelity and Limitations	Inaccurate or overly simplistic simulation models hinder reliable validation, especially for component interaction and edge-case reproduction.
Reliance on Expert Judgment	Identifying critical scenarios and validating performance still depends heavily on expert intuition, rather than formal or automated methods.
Need for Systematic and Structured Testing	Practitioners desire more structured processes and systematic methods (e.g., coverage-based, data-driven, or smarter automation) to handle complex input spaces.

Table 4.2: Summary of Themes from Thematic Analysis

For instance, one participant highlighted the unique challenges posed by the demanding terrain in mining operations:

> *I think one of the key challenges is the environment itself, which we are facing in mining applications. Compared to other autonomous driving use cases, there is a strong focus on ensuring the machine can operate reliably in its environment. We are dealing with steep uphills and downhills, whether the vehicle is loaded or unloaded, while also needing to account for loads.* – P1

Beyond terrain gradients, other physical factors such as dust accumulation and slippery surfaces further complicate testing. As another participant noted, these environmental elements directly impact system performance in ways that are often difficult to anticipate:

> *Yes, terrain and dust affect the system, especially in mining operations. You will have trucks that get dirty much quicker, and there's mud, and it's slippery.* – P7

Furthermore, the interaction of weather phenomena with sensor technology introduces additional complexities. One participant pointed out the challenges posed by water accumulation after rain, which can interfere with visual sensors by creating misleading reflections:

> *I would say introducing water—what's often not thought of is after rain, not during. These visual sensors view puddles as just... reflecting light, which is hard for the sensors to grab, and so that's a potential big challenge.* – P6

These reflections illustrate how the physical and environmental variability inherent in mining operations introduce both mechanical and perceptual challenges for AHS.

Conditions such as dust, mud, and low traction not only affect the vehicle's drivability but also degrade the performance of onboard sensors responsible for perception and navigation. Visual and lidar sensors may be obstructed or confused by dust clouds, water accumulation, or surface reflectivity, thereby increasing the likelihood of perception errors.

Moreover, compounding weather conditions such as heavy rain, fog, and rapid temperature fluctuations further complicate the validation process. Unlike urban or highway environments, mining sites lack the predictability of well-maintained roads, consistent lighting, and standardized infrastructure. These environmental factors evolve rapidly and unevenly across the site, making it challenging to create a consistent and representative test environment, whether in real-world trials or simulated settings. These insights underscore the importance of developing testing strategies and simulation tools that can adequately capture and replicate these environmental complexities to ensure AHS reliability in their true operating contexts.

4.1.2 Simulation Fidelity and Limitations

Closely tied to the environmental complexities faced by AHS is the critical issue of simulation fidelity. Many participants expressed concerns that current simulation tools may fall short in accurately representing the real-world conditions under which AHS operate. This gap in realism not only undermines the reliability of testing outcomes but also risks engendering a false sense of confidence in system performance, potentially leading to undetected failures in deployment. One participant stressed the importance of understanding the scope and limitations of the simulation environments used in testing:

> *We need to know the test environments we are using. How well can we trust the models, or what do the models cover, and what are the limitations of the model? If the models are not validated, that will limit how much we can trust them.* – P7

This concern about trustworthiness highlights a fundamental challenge: without thorough validation, simulation results may not accurately predict real-world behavior. Another interviewee pointed out that many control system issues arise from complex interactions with the environment and perception systems, which are difficult to capture fully in simulation:

> *One of the problems with testing the control systems is that some of these issues are caused in conjunction with perception or environment, and we do not have a sufficiently accurate representation of reality.* – P3

This gap between simulation and reality underscores the need for models that can faithfully reproduce not only the vehicle dynamics but also the nuanced environmental factors that impact system performance. As another participant noted, ensuring the accuracy of both vehicle and environmental models is essential for meaningful

simulation results:

> *I work with simulations, and you need to know that the simulated model of the vehicle and the environment it drives on is accurate.* – P5

However, even when high-fidelity models exist, their computational demands often pose practical challenges. Longer simulation times can slow feedback loops, delaying test iterations and reducing overall efficiency:

> *Another limitation or trouble when working with high fidelity is that it takes quite some time to simulate, so then the feedback loops are longer.* – P5

Balancing model fidelity with computational efficiency remains a key challenge, especially in industrial contexts where rapid development cycles and extensive scenario coverage are necessary. Participants expressed a clear desire for advancements in simulation technologies to better capture real-world physics and operational nuances:

> *Better physical models—we want to have a more accurate representation of reality.* – P3

Overall, these reflections reveal that simulation fidelity is not merely a technical detail but a central factor influencing the credibility and utility of AHS testing. Improving simulation accuracy while managing computational costs is essential to enable more reliable validation, accelerate development, and reduce reliance on costly and potentially hazardous real-world testing.

4.1.3 Reliance on Expert Judgment and Need for Systematic and Structured Testing

Another key challenge identified was the heavy reliance on expert intuition and manual effort in scenario selection and test design. While expert knowledge is invaluable, participants acknowledged that this approach often lacks structure and systematic rigor.

> *I am speaking for myself, but we try to think of edge cases. So it comes from your knowledge, and then also during code reviews, your peers can suggest additional tests. But mainly, it comes from expert knowledge about the function and what it's not supposed to do.* – P4

This informal approach to scenario generation depends heavily on individual experience and collaboration among peers. While it helps identify many important cases, it may miss less obvious or rare failure modes.

> *Tests can take a lot of time, and even when simplified, they still take time to run. So we need to be cautious about adding too many of them and instead prioritize. Ideally, we would test all the cases across all vehicle types, but that would take too long. Selecting critical edge cases can be a cheap way to achieve high coverage.* – P3

Given these practical constraints, testers must carefully select scenarios that balance thoroughness with feasibility. This necessity often results in focusing on perceived high-impact cases at the expense of systematic coverage. Recognizing these limitations, participants expressed enthusiasm for more intelligent and automated approaches that could expand testing efficiency without increasing manual effort.

> *More use of simulation and finding a more optimized way of identifying difficult cases would be quite nice.* – P4

Such insights reflect a broader consensus on the need to complement expert judgment with intelligent methods that can systematically explore the input space and highlight potential failure regions.

> *On our driver component, it's mostly expert-driven testing and edge case detection based on expert knowledge. We need smarter testing methods.* – P1

> *More automation is always good. So we have the potential of testing more and that could save us some time.* – P6

Overall, these reflections highlight a clear gap in current testing practices, the absence of systematic, repeatable methods capable of supplementing expert insight. This study aims to fill that gap by proposing a structured testing framework. By combining grid-search, random, and combinatorial scenario generation, which ensures broad coverage, with adaptive sampling, which targets high-risk regions efficiently, this framework seeks to enhance testing effectiveness while maintaining the practical benefits of expert knowledge. The goal is to move toward a more systematic, intelligent testing method that supports the complexity and variability inherent in autonomous heavy systems testing.

4.1.4 Summary

The following summary synthesizes key insights from the interviews on challenges and practices in testing AHS:

- Participants highlighted the complexity of the operational environment, noting that harsh conditions such as uneven terrain, dust, mud, steep slopes, and variable weather significantly impact both vehicle performance and sensor reliability.

- Simulation fidelity emerged as a critical concern, with many interviewees expressing doubts about the accuracy and representativeness of current simulation models. Limitations in model validation and the trade-off between high fidelity and computational efficiency were repeatedly mentioned.

- Testing efforts are heavily reliant on expert judgment, with scenario selection often based on individual experience and intuition rather than systematic or intelligent approaches. This limits consistency and makes it difficult to ensure comprehensive coverage.

- Interviewees emphasized the difficulty of scaling testing across diverse operational conditions, vehicle configurations, and environmental variations, highlighting a need for more structured, optimized, and repeatable testing frameworks.

- Participants expressed a desire for improvements in both physical modeling and testing methodologies to better capture real-world complexities and increase confidence in AHS validation results.

4.2 Simulation-Experiments

In this section, we introduce the scenarios used for our simulation experiments. These scenarios serve as test environments to compare different testing strategies, including grid search and random testing. Additionally, we incorporate combinatorial testing as a method to improve input space coverage while keeping the number of test cases manageable. We further use these scenarios to evaluate the applicability and efficiency of adaptive sampling techniques, leveraging tree-based models and clustering as supporting methods. For each scenario, we describe the setup parameters, define the success criteria and failure conditions, and specify which parameters are varied and which are held constant. Finally, we include illustrations to provide a visual understanding of each test scenario.

4.2.1 Obstacle Avoidance Test Scenario

The first simulation scenario, illustrated in Figure 4.1, involves two connected road segments, where the lengths of the two segments are fixed. Given a concrete scenario θ, the goal is for the mining vehicle to stop at the goal position gp. In addition, the system considers a safety buffer of K meters, meaning that at any time t, the vehicle reference point, rp_θ^t, should be distant from any reference point of an obstacle rp_{obs} by at least K meters, i.e., $\forall t, dist(rp_\theta^t, rp_{obs}) \geq K$.

Within a scenario θ, the following six input parameters can be varied to derive new test cases.

1. Velocity v_0, characterizing the starting velocity of the vehicle.

2. Obstacle lateral offset Δ_{obs}, determining the lateral position of the obstacle.

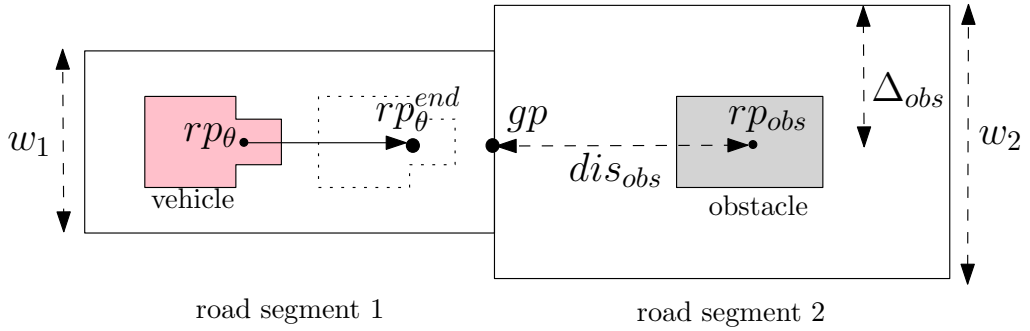

Figure 4.1: Obstacle Avoidance Test Scenario

3. Obstacle distance dis_{obs} determining the distance between the obstacle and the goal point gp at the end of the first segment.[1]

4. The width of the first segment w_1.

5. The width of the second segment w_2.

6. In this scenario, r indicates whether the vehicle is in reverse driving mode (true/1 for reverse). This affects the vehicle's heading direction.

4.2.2 Open Area Planner Test Scenario

For the second scenario, we illustrate the Open Area Planner scenario depicted in Fig. 4.2, where a transport vehicle navigates within a polygonal open area. The scenario is defined by an initial vehicle position and heading (yaw angle), and a target terminal position and heading. The initial reference position rp_θ corresponds to the vehicle's starting coordinates (x_{init}, y_{init}) and initial yaw angle ψ_0. The goal is for the vehicle to reach the terminal reference position rp_θ^{end}, corresponding to (x_{term}, y_{term}), and terminal yaw angle ψ_{end}. A test case is considered successful if the vehicle travels from rp_θ to rp_θ^{end}, achieves the terminal yaw ψ_{end}, and comes to a complete stop.

Within this scenario, the following input parameters are varied to generate test cases:

1. Terminal x, y position $(rp_\theta = x_{term}, y_{term})$, defining the target coordinates the vehicle must reach within the open area.

2. Terminal yaw angle (ψ_{end}), specifying the desired final vehicle heading at the destination.

[1] Note that if dis_{obs} is smaller than K meters, then the goal location gp can never be reached without violating the constraint regulated by the safety buffer.

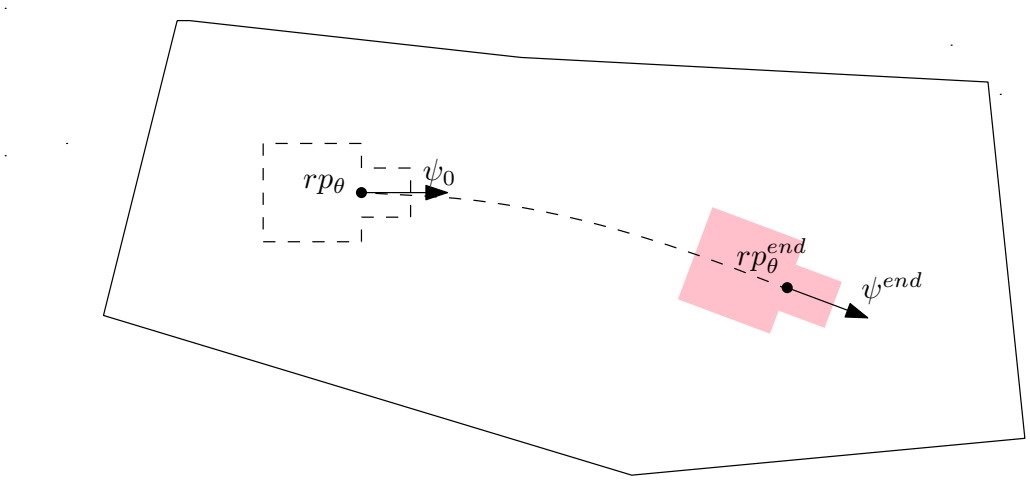

Figure 4.2: Open Area Test Scenario
Note: The polygon has been modified for confidentiality, but the rest of the scenario reflects the actual test conditions.

4.3 Evaluation of grid-search testing and random testing

To compare the effectiveness of grid-search testing and random testing, the same number of test cases were generated for each method. We compared both methods across two scenarios and evaluated their performance based on key performance metrics, which are summarized in the Tables 4.4 and 4.3. The results across different scenarios show that both grid-search and random testing uncovered a comparable number of failure cases. However, grid-search consistently achieved significantly higher input space coverage, especially in the Open Area scenario (100% vs. 59.27%). This higher coverage suggests that grid-search provides a more structured and comprehensive sampling of the input space, making it more suitable for downstream analysis and model training in adaptive testing.

Consequently, these qualities made the grid-search test data well-suited for training the initial models used in the first adaptive testing iteration. In subsequent iterations, new models (e.g., decision trees and random forests) were trained using the data collected from the previous adaptive tests. The structured sampling of grid search provided a consistent and comprehensive foundation for identifying critical patterns and behaviors. Furthermore, we observed a disparity in the failure percentages between the two scenarios. While the exact cause is not yet known, this difference likely stems from the varying complexities and constraints of the scenarios, with the Obstacle Avoidance test having stricter spatial requirements that may lead to more failures. Further analysis is needed to determine the precise reasons.

4.3.1 Note on Combinatorial Testing

Combinatorial testing was also applied during the study and demonstrated promising results, particularly in terms of input space coverage and reduced execution time. In the Open Area scenario, combinatorial testing even outperformed random testing in coverage. However, as this method was introduced later in the project timeline, it was not included in the main comparative evaluation with grid-search and random testing. Consequently, to maintain consistency and fairness in the primary analysis, only grid-search and random testing results are directly compared. Nonetheless, the preliminary findings suggest that combinatorial testing is a valuable approach worth further exploration in future work.

Testing Method	**Number of Test Samples**	**Time Taken**	**Total Failure cases**	**Input Space Coverage**
Grid Search	5120	10 h 18 min	90%	100%
Random	5120	10 h 48 min	91%	100%

Table 4.3: Comparison Table of primary testing methods - Obstacle Avoidance

Testing Method	**Number of Test Samples**	**Time Taken**	**Total Failure cases**	**Input Space Coverage**
Grid Search	2367	9 h 22 min	0.4%	100%
Random	2367	9 h 23 min	0.34%	59.27%

Table 4.4: Comparison Table of primary testing methods - Open Area Scenario

4.4 Adaptive Sampling with Decision Tree and Random Forest Models

To refine the exploration of the input space and focus testing efforts more effectively, we employed adaptive sampling based on decision tree and random forest models. These models were selected due to their interpretability, robustness, and ability to highlight key input features that influence output behavior. Initial comparisons between grid-search and random testing (Table 4.3 and 4.4) revealed that grid-search testing was better suited for training decision tree and random forest models compared to random testing. As such, we used grid-sampled data as the foundation for adaptive iterations.

4.4.1 Feature Importance Analysis

We performed feature importance analysis using both decision tree and random forest models to assess which input parameters had the greatest impact on system behavior across scenarios. Although both model types were explored, the random forest models consistently achieved slightly better predictive performance across scenarios, such as lower mean absolute error (MAE) and higher R^2 values in regression tasks,

as well as more balanced classification metrics. As a result, the results of the random forest models are presented in the main text for brevity. The results of the decision tree models, which follow similar trends, are omitted but available upon request.

4.4.1.1 Obstacle Avoidance Scenario

In this scenario, the feature importance analysis with random forest consistently identified three out of six parameters as dominant influencers of system performance: *Obstacle Distance*, *Reverse Parameter*, and *Velocity*. Notably, the decision tree model produced a nearly identical ranking of feature importance in this scenario, reinforcing the conclusion that these three parameters predominantly govern the system's obstacle avoidance behavior. To quantitatively support this, Appendix Table B.2 summarizes the feature importance rankings from both the random forest and decision tree models trained on the obstacle avoidance data. The random forest model yielded a MAE of 0.0297 and R^2 of 0.9803, outperforming the decision tree model, which achieved an MAE of 0.0311 and R^2 of 0.9766. These metrics, shown in Appendix Table B.3, reinforce the random forest's slightly better predictive performance while confirming the critical role of *Obstacle Distance*, *Reverse Parameter*, and *Velocity*.

Our feature importance analysis shows that the widths of the two road segments (w_1 and w_2) have little to no influence on the system's internal decision variables or control logic. This is confirmed by the development team, who explained that the AHS prototype is designed to follow the center of the lane and does not initiate complex maneuvers such as turning. Similarly, under the current test configuration, where the road width is moderate, the sensor reliably detects obstacles regardless of their lateral offset. Our observations suggest that even when an obstacle is positioned toward the side, it effectively forms a virtual barrier perpendicular to the road. Therefore, the lateral offset parameter does not significantly affect the system's obstacle avoidance decisions.

Guided by these insights, we focused our adaptive sampling exclusively on the three identified key parameters, *Obstacle Distance*, *Reverse Parameter*, and *Velocity*, while holding the other three parameters fixed. This targeted refinement allowed us to perform a finer-grained exploration within the relevant parameter subspace, such as reducing the sampling step size from 1.0 meter to 0.1 meter. Consequently, this strategy enabled more precise delineation of the decision boundaries, enhancing the efficiency and effectiveness of the testing process by concentrating efforts where the system's behavior is most sensitive.

4.4.1.2 Open Area Planner Scenario

In this scenario, all three input parameters, *Terminal x, y position* (x_{term},y_{term}), and *Terminal yaw angle* (ψ_{end}), showed significant but varying influence on system behavior. Feature importance analysis (see Fig. 4.6) revealed that the terminal positions, particularly the y_{term} coordinate, had the most dominant effect. This is likely because the planner's path heavily depends on reaching a precise spatial goal. Notably, the top and bottom parts of the actual polygon used in the simulation represented the

narrowest regions, creating tighter constraints for the system; however, the exact shape of the polygon cannot be shown for confidentiality reasons. An illustrative scenario is provided in Fig. 4.2. This example includes an open area polygon with straight line segments forming an irregular shape, with some edges angled inward and others outward. While different from the actual polygon, it demonstrates how narrow or constricted areas can pose greater challenges for the system compared to wider sections. Meanwhile, the terminal yaw angle ψ_{end} contributed meaningfully by influencing the vehicle's final orientation and alignment.

Appendix Table B.1 summarizes the classification performance of the random forest and decision tree models for the open area scenario, along with their confusion matrices and classification reports. The random forest achieved an overall accuracy of 99%, demonstrating balanced precision and recall across all classes. The feature importance rankings reveal that both models identified y_{term} as the most influential feature, with the random forest assigning it an importance of 0.464 and the decision tree assigning it a substantially higher value of 0.805. However, a notable difference lies in the relative importance of the remaining features. The random forest ranked x_{term} (0.388) above ψ_{end}(0.148), while the decision tree reversed this order, placing more weight on ψ_{end} (0.160) than on x_{term} (0.035).

This switch in importance between the x-position and yaw indicates that the models use different decision criteria: the random forest, leveraging its ensemble structure, finds horizontal position more informative, whereas the decision tree relies more on orientation when forming splits. Despite these differences, both models consistently prioritize spatial and orientation features critical to behavior classification in open area navigation. Given their combined impact, all three parameters were included in the adaptive search space for this scenario. This comprehensive approach allowed finer-resolution sampling, enhancing the precision in identifying critical decision boundaries relevant to the planner's performance.

4.4.2 Decision Boundary Visualization

Figures 4.3 and 4.4 illustrate the decision boundaries derived from the random forest models trained on the last iteration and final point of convergence of adaptively sampled data. The colored regions represent areas classified as pass or failure, based on system performance under varying input parameters. In Figure 4.3, the model focuses on the three dominant parameters: obstacle distance, velocity, and reverse mode, as introduced in Section 4.2.1. This enables the model to accurately capture the boundary between safe and failure states in the Obstacle Avoidance scenario. The vertical separation in the figure represents the reverse mode parameter, which takes only two values (true or false). These correspond to forward or reverse driving but are not explicitly labeled to maintain confidentiality. For the same reason, numerical axis ticks are hidden and random jitter has been added to obscure exact test values while retaining the overall structure and interpretability. The figure presents a manual clustering, where test cases within similar failure groups are colored alike. Failure groups were defined based on severity thresholds (e.g., within 20% below expected, indicating minor failure; 40% below expected, moderate failure; etc.). This refined

boundary demonstrates the model's effectiveness in guiding adaptive sampling toward critical transition zones; regions where small input variations can cause the system to shift from safe operation to failure.

Similarly, Figure 4.4 illustrates how adaptive sampling across the three key parameters in the Open Area scenario captures complex interactions that influence success or failure. The figure shows all terminal points the SUT attempted to reach, with a clear red boundary outlining locations the SUT could not reach and green indicating successful reaches. While the exact cause of the failure region (red area) has not been determined, its close proximity to the SUT's starting location and the top narrower region of the open area polygon suggest possible reasons, although some nearby locations did result in success. Clustering of sample points near the red boundary highlights the efficiency of adaptive sampling by focusing test efforts on the most informative, failure-prone regions of the input space. These visualizations demonstrate that model-guided sampling improves fault detection by prioritizing exploration of high-risk zones and minimizing effort in already-safe areas.

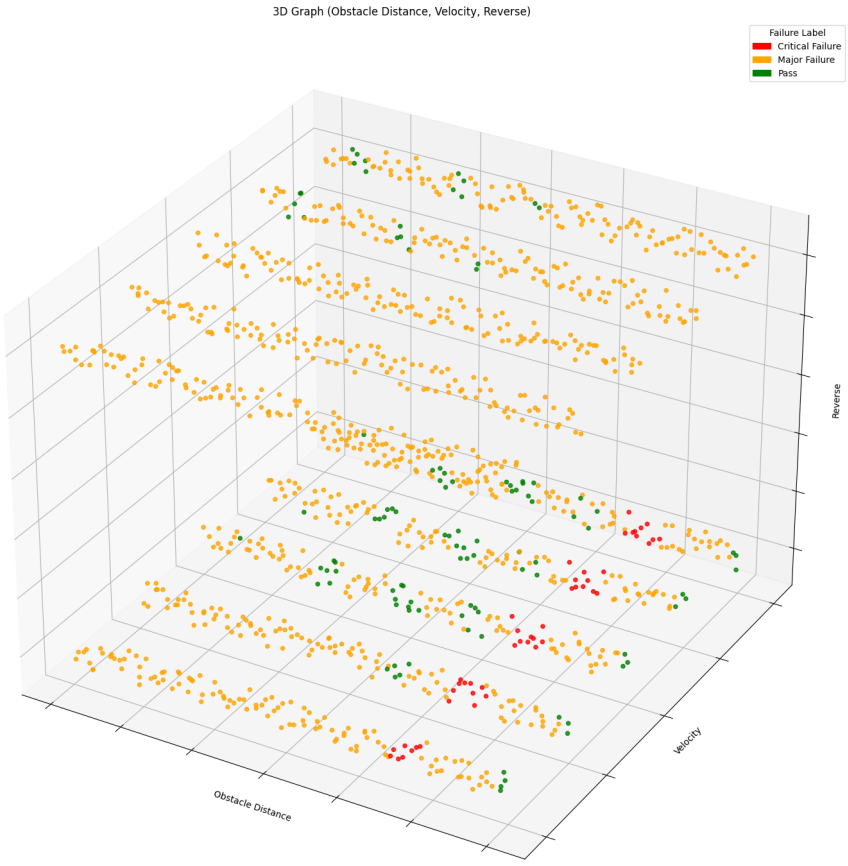

Figure 4.3: Decision boundary visualization for Obstacle Avoidance scenario using adaptive sampling.

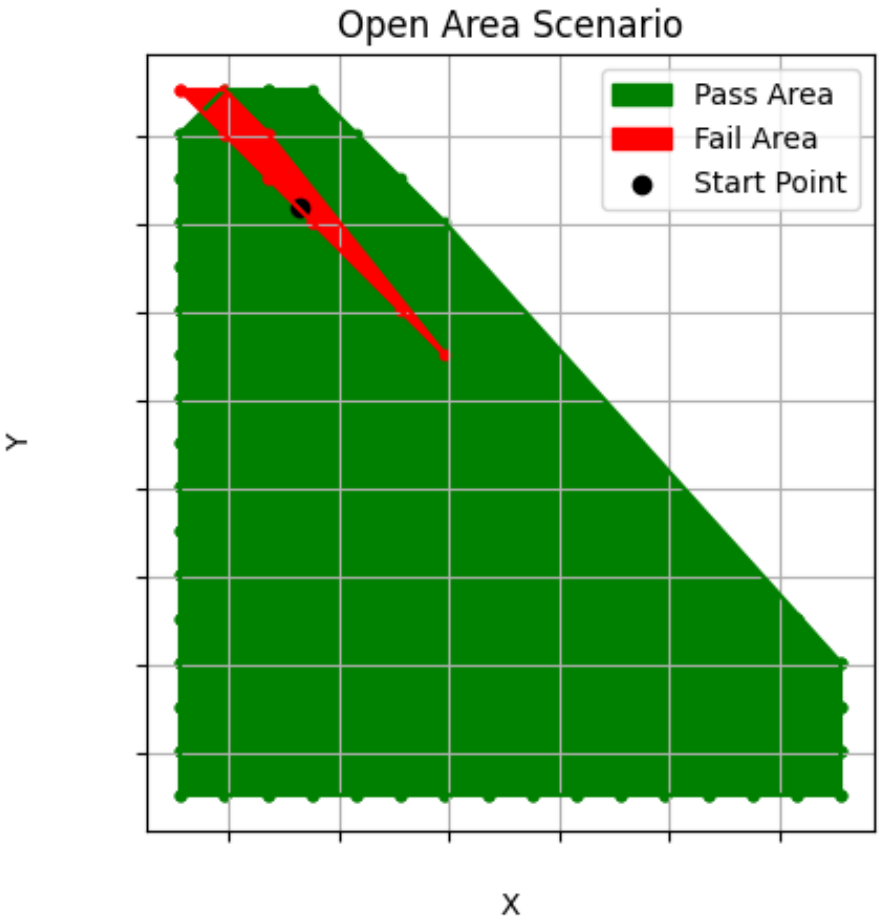

Figure 4.4: Decision boundary visualization for Open Area Planner scenario using adaptive sampling.

4.4.3 Evaluating the Adaptive Suite Against a Baseline

Table 4.5 reveals a comparison of the final adaptive test suite from the fourth iteration with the baseline test suite used by Volvo Autonomous Solutions for the Open Area scenario. Despite being significantly smaller in size (34 vs. 143 tests), the adaptive suite detected more failures and achieved better failure density. Although the adaptive suite ran three minutes longer in total time, the improved fault detection rate highlights the effectiveness of targeting critical regions with fewer, more informative tests. The three failures found in the baseline test suite were not part of the adaptive suite. These baseline failure cases were located very close to the boundary of the polygonal driveable area, near its lower left edge. It was unclear whether these failures were system issues or due to the vehicle slightly touching or exceeding the boundary constraints. In contrast, the failures detected by the adaptive approach occurred farther from the boundary edges, indicating more robust violations not attributable to proximity artifacts. This suggests that adaptive sampling was effective at identifying diverse and unambiguous failure cases. This comparison was primarily conducted to evaluate the adaptive sampling techniques against the expert-driven test suite used by Volvo Autonomous Solutions, as one of the key challenges identified

in the interview portion of the study (see Subsection 4.1.3) was the need to move toward more automated and smarter testing methods to efficiently focus on edge cases. These results indicate that adaptive sampling can outperform traditional expert-driven approaches in terms of efficiency and effectiveness, supporting the case for incorporating more automated testing strategies in AHS development.

Test Suite	**Number of Tests**	**Failures Detected**	**Execution Time (h:mm)**
Adaptive (Iteration 4)	34	14	0:38
Volvo Baseline	143	3	0:35

Table 4.5: Comparison of Adaptive Test Suite vs. Baseline (Volvo Autonomous Solutions) Test Suite

4.5 Transition Points and Output Drops

To further investigate the sensitivity of the system to small input changes, we applied the NNS method (see Subsection 3.5) across both test scenarios: Obstacle Avoidance and Open Area. This method identifies pairs of inputs (one resulting in success and the other in failure) that are closest in Euclidean distance, effectively highlighting the transition boundaries in the input space. By analyzing these pairs, we can pinpoint where minor variations in terminal conditions or obstacle parameters result in significant changes in system performance. Tables 4.6 and 4.7 present representative examples of such critical transition points in the Open Area and Obstacle Avoidance scenarios, revealing pass and failure test inputs and outputs. In Table 4.7, the Output Difference indicates how much closer the vehicle came to the obstacle in the failure scenario compared to the passing scenario, which maintained a safe distance.

Success Scenario Input *terminal yaw angle*, x_{term}, y_{term}	**Failure Scenario Input** *terminal yaw angle*, x_{term}, y_{term}	**Input Difference**	**Success Output**	**Failure Output**
[2.24076 -210.111 -84.962]	[2.26076 -210.111 -84.962]	0.02	[0]	[1]
[2.24076 -210.611 -84.962]	[2.22076 -210.611 -84.962]	0.02	[0]	[1]
[2.26076 -210.111 -84.462]	[2.26076 -210.111 -84.962]	0.5	[0]	[1]
[2.22076 -209.611 -86.962]	[2.22076 -209.111 -86.962]	0.5	[0]	[1]
[2.26076 -211.111 -84.462]	[2.22076 -211.111 -84.462]	0.04	[0]	[1]

Table 4.6: Pairs of nearly identical test inputs with different outcomes – Open Area Scenario. Colored values indicate changed inputs. Input Difference shows the specific change.

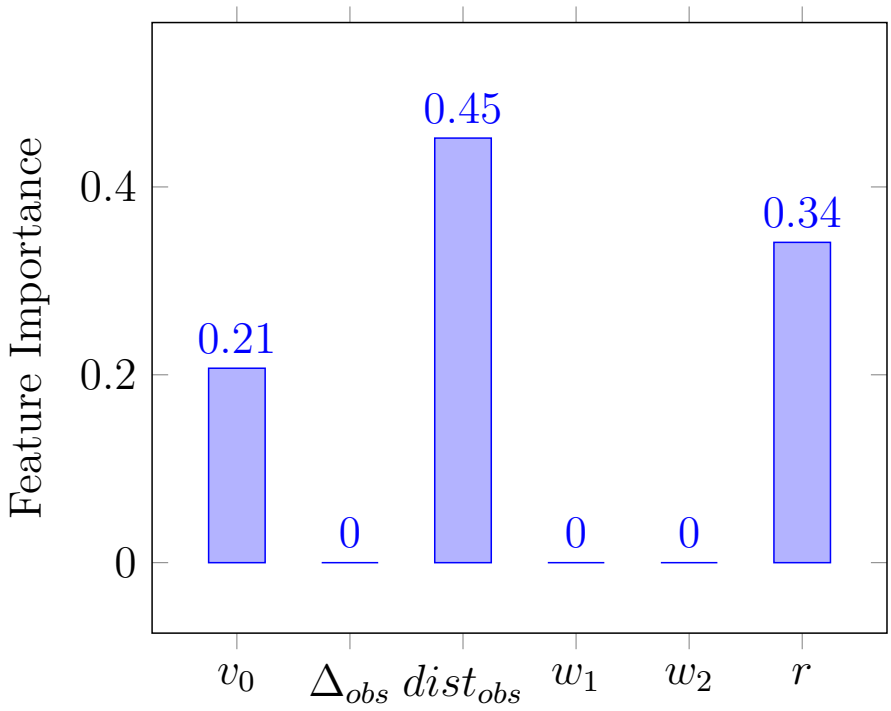

Figure 4.5: Feature importance of input parameters from Random Forest for the Obstacle Avoidance scenario

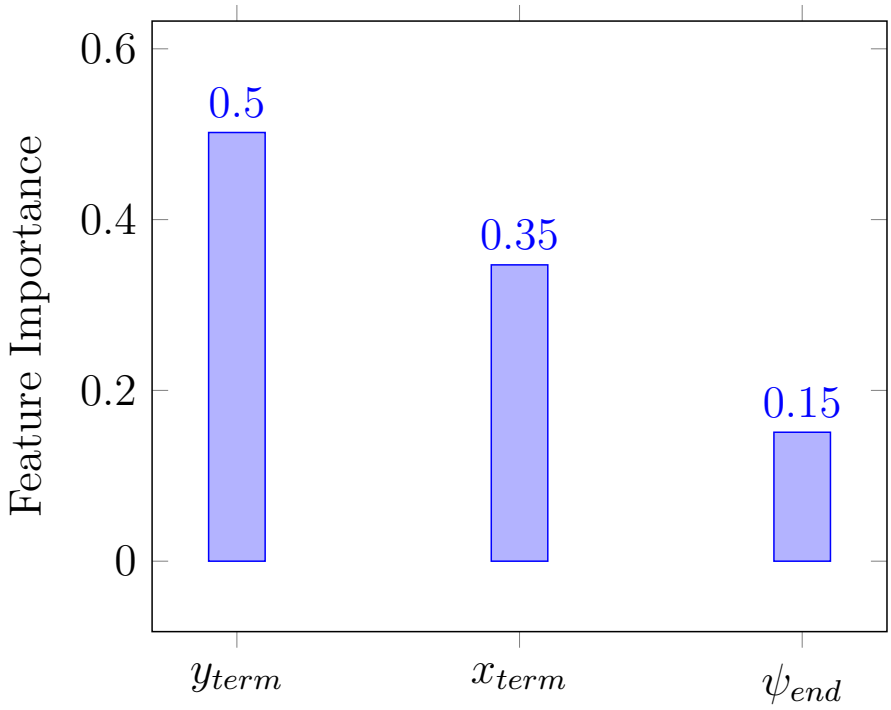

Figure 4.6: Feature importance of input parameters from Random Forest for the Open Area Planner scenario

Success Scenario Input dis_{obs}, r (reverse mode), v_0	**Failure Scenario Input** dis_{obs}, r (reverse mode), v_0	**Output Difference**
6.2 m, true, 5.0 m/s	6.3 m, true, 5.0 m/s	0.37 m
6.2 m, false, 3.0 m/s	6.3 m, false, 3.0 m/s	0.32 m
7.4 m, false, 2.0 m/s	7.5 m, false, 2.0 m/s	0.10 m

Table 4.7: Pairs of nearly identical test inputs with different outcomes – Obstacle Avoidance scenario. Colored values indicate changed inputs.

5

Discussion

This chapter interprets and contextualizes the findings presented in Chapter 4, focusing on how they address the research questions posed by this study. It reflects on the key challenges and insights revealed through interviews and simulation experiments, examining their implications for the testing and validation of AHS. Drawing on qualitative and quantitative evidence, the discussion highlights how current practices can be enhanced through structured, systematic, and adaptive testing frameworks. It also considers the practical constraints faced by industry practitioners and the potential for future advancements.

5.1 Addressing the Research Questions

This section revisits the research questions introduced in Section 1.4, synthesizing the key findings from the empirical investigation and their contributions to understanding AHS testing challenges and solutions.

5.1.1 RQ1

What are the key challenges in testing AHS, and how can critical scenarios be identified?

The thematic analysis of expert interviews highlighted significant challenges faced in AHS testing, such as environmental and operational difficulties, simulation fidelity limitations, reliance on expert judgment, and the need for systematic, structured testing approaches. Similar challenges have been reported in the literature for the testing of autonomous systems and show some overlap between challenges faced in AHS and other autonomous systems. Helle et al. [17] identify complex environments as a major cause of difficulty in testing autonomous systems. They argue that the large input space and partially unknown design context make exhaustive testing infeasible.

Another challenge noted by Helle et al. [17] is the inherent complexity of autonomous systems' software. This is because the software in these systems can adapt its plans, learn from experience, and reconfigure itself automatically. Accurately replicating such dynamic and intricate behavior in simulation is difficult, thereby limiting the

reliability of virtual testing environments. The reliance on expert judgment and the need for more structured and systematic testing approaches are also echoed by other researchers [32, 44, 34], who highlight that scenario generation and selection may be subjective, where simple yet critical scenarios are sometimes overlooked. This can lead to insufficient evidence for safety validation in real-world traffic conditions [32]. Furthermore, manually designing scenarios is time-consuming and often lacks the diversity and complexity of real accident scenarios [44]. Many dangerous situations exceed human imagination and are difficult to capture through expert design alone.

While simulation fidelity and environmental or operational challenges were outside the scope of this thesis, the difficulties stemming from reliance on expert judgment and the need for more structured, systematic testing approaches motivated the development of the presented testing framework. This framework combines systematic and adaptive methods to better address complex failure modes and enhance input space coverage. The findings from the interviews directly address RQ1 by identifying core testing obstacles, emphasizing the critical need for broader and deeper coverage of the test space, and highlighting the importance of uncovering critical performance scenarios. The proposed framework responds to these challenges by integrating model-guided adaptivity with structured test generation, enabling more thorough and efficient exploration of failure-prone regions.

5.1.2 RQ2

How can the testing methods grid-search testing, combinatorial testing, random testing, and adaptive sampling be applied to identify critical failure scenarios and decision boundaries in AHS?

Simulation experiments in this study were guided not only by challenges expressed by experts during interviews but also by insights from prior research on testing strategies for autonomous systems. For example, Fogelberg and Hult Pappas [13] emphasized the ability of random testing to expose a wide range of scenarios but noted its inconsistency in discovering rare or edge-case failures. Grid-search testing, while offering more thorough coverage of the input space, has been repeatedly flagged as computationally expensive and prone to combinatorial explosion unless tightly managed [45]. Combinatorial testing, as highlighted by Song [34], provides a structured yet efficient way to uncover critical input interactions without requiring full input space traversal. Finally, Mullins et al. [30] demonstrated that adaptive sampling, when guided by meta-models, effectively targets the most failure-prone regions by iteratively focusing on high-risk areas.

These insights align with this study's findings: grid search offered the most comprehensive input coverage but was computationally intensive, while adaptive sampling was the most efficient at detecting critical failures. Combinatorial testing struck a middle ground—faster than grid-search testing but less targeted than adaptive sampling—while random testing was not necessarily faster but occasionally uncovered more failures than grid-search testing. Table 5.2 summarizes these trade-offs. These results address RQ2 by showing that each method has distinct strengths: grid search

provides thoroughness, adaptive sampling excels in precision and efficiency, and the others offer practical compromises. The choice of method ultimately depends on whether the priority is broad coverage, targeted failure detection, or runtime efficiency.

5.1.3 RQ3

How can tree-based models, clustering techniques, and nearest neighbor search methods be utilized to refine scenario selection and guide adaptive sampling strategies in AHS?

The integration of qualitative and quantitative results suggests that addressing the challenges raised by practitioners requires flexible and targeted testing methods. Clustering and tree-based models, including decision trees and random forests, enhanced adaptive sampling by steering test generation toward failure-prone regions. These tools refined scenario selection by focusing on inputs where the system was most sensitive, effectively balancing broad exploration with targeted testing in high-risk zones. The nearest neighbor search method successfully identified sensitive transition points where small input changes led to system failures. These represent areas of particular interest for developers and testers, as they highlight the edge of the system's operational boundaries and serve as entry points for debugging and system refinement.

5.1.3.1 Adaptive Sampling and K-means Clustering

For adaptive sampling, we selected two models: Decision Tree and Random Forest. While a single decision tree is more interpretable and can provide insights by manually examining decision paths, it lacks built-in mechanisms for uncertainty estimation. In contrast, random forests, due to their ensemble structure, naturally support uncertainty-based adaptive sampling, allowing for automated identification of regions in the input space where the model's predictions are less certain, indicating where additional testing is most needed. Both models performed well overall, but random forest models proved slightly more effective, especially when test cases were limited or failure instances were sparse. The decision tree's interpretability remains valuable for understanding failure patterns, but random forests enable more scalable and automated adaptive testing.

Additionally, k-means clustering was used to group test cases; however, it did not produce visually informative results. This appeared to be because input points were often too close together, making it difficult for the algorithm to separate failure cases from successful ones. Interestingly, k-means clusters containing a mix of failure and success cases proved especially valuable, as they represented regions near critical decision boundaries. Although a clear k-means clustering figure could not be generated, Figure 4.3 illustrates a manually grouped clustering of failure cases from the final adaptive iteration, with similar failures grouped by color. Together, these findings underscore the value of combining model-guided testing with coverage-focused testing methods to create more efficient and effective AHS testing protocols.

Summary of Research Questions and Findings

This section provides a concise summary of each research question along with the key results and conclusions drawn from this thesis.

Research Question	Summary of Findings
RQ1: What are the key challenges in testing AHS, and how can critical performance scenarios be identified?	Expert interviews identified simulation fidelity, reliance on expert judgment, and a need for systematic and structured testing approaches as key challenges. These findings informed the development of the testing framework.
RQ2: How can the testing methods grid-search testing, combinatorial testing, random testing, and adaptive sampling be applied to identify critical failure scenarios and decision boundaries in AHS?	Grid-search testing offered the highest coverage but was costly. Combinatorial testing was more efficient but less targeted than adaptive sampling. Random testing occasionally found more failures than grid-search testing, and adaptive sampling was the most efficient at detecting failures.
RQ3: How can tree-based models, clustering techniques, and nearest neighbor search methods be utilized to refine scenario selection and guide adaptive sampling strategies in AHS?	Tree-based models helped estimate feature importance and locate decision boundaries, clustering highlighted distinct behavioral regions, and NNS identified similar scenarios with different outcomes (pass/fail). Together, these techniques guided adaptive sampling toward failure-prone and under-tested areas.

Table 5.1: Summary of how Research Questions were addressed

Testing Method	Description
Grid-search Testing	Exhaustive, structured sampling across the parameter grid. Provides high coverage but is computationally expensive and time-consuming.
Random Testing	Offers broad, unconstrained sampling but may miss critical edge cases due to its randomness.
Combinatorial Testing	Pairwise combinations of parameters to efficiently cover interactions between inputs. Balances efficiency and coverage well.
Adaptive Sampling	Focused sampling in failure-prone regions using adaptive, model-guided approaches (e.g., Decision Tree and Random Forest models). Offers high efficiency and excels at detecting critical failures with fewer samples.

Table 5.2: Comparison of Testing Methods applied in this thesis

5.2 Limitations

This section discusses limitations to the study's validity, going through construct validity, internal validity and external validity.

5.2.1 Construct Validity

A limitation in construct validity was the use of feature importance derived from the selected models to guide the testing strategy. The interpretation of this feature importance is bounded by the limitations of grid-based sampling and should not be treated as definitive proof of feature irrelevance. To mitigate this, uniform grid-based sampling within the parameter space can be applied repeatedly to generate diverse training datasets. By training models multiple times and observing if a feature consistently shows zero importance, one can apply hypothesis testing. In other words, if a feature repeatedly fails to impact the model, this provides statistical evidence that the feature is likely irrelevant with high confidence.

5.2.2 Internal Validity

This study used simulations to conceptually evaluate testing methods rather than fully replicating real-world environments. To address this limitation, we involved AHS experts from Volvo Autonomous Solutions and academic professionals. Their feedback helped validate that the chosen scenarios represent real-world challenges faced by AHS. The simulations were conducted under controlled AHS-specific conditions but include inherent limitations such as approximations in physics modeling and sensor simulation, which may affect test accuracy. Care was taken to select parameters and scenarios that closely reflect realistic operational conditions. The AHS models include simplifying assumptions about system behavior and environmental interactions, which could impact generalizability. The scenarios and models used are those routinely employed by AHS experts; independent verification of their accuracy was beyond this study's scope, so findings should be interpreted accordingly. To mitigate bias, scenario selection and data analysis were guided by expert input and iterative reviews, reducing subjective influence and enhancing result robustness. Although expert input was critical for scenario selection and validation, there remains a potential risk of subjective bias or variability in expert opinions, which could influence the scope and focus of the testing scenarios.

5.2.3 External Validity

The data used in this study comes from specific systems (Volvo Autonomous Solutions), which may pose a threat to generalizability. By using diverse test scenarios and a range of mining-specific conditions, the study aims to increase the relevance and applicability of the findings to real-world mining environments. For this study, we have selected a small subgroup of testing methods, which are grid-search, combinatorial, random, and adaptive sampling, and may exclude newer or less explored methodologies that could be important for the broader field. This is a deliberate choice on our part as these methodologies are among the most well-researched and widely used, making our findings relevant to academics and industry professionals. It is possible that newer, hybrid, or less-explored testing methodologies could offer additional benefits or insights, but these were beyond the scope of this study. Furthermore, the confidentiality of certain real-world data may limit the replicability of the study, and some abstractions were necessary in how results are presented.

5.3 Future Work

Future work should address these limitations by validating findings through hardware-in-the-loop testing or real-world field trials, and by incorporating additional machine-learning techniques. For example, large language models (LLMs) could be leveraged to analyze system logs and automatically isolate bugs, significantly accelerating the debugging process. They can assist in identifying the root causes of critical failures and prioritizing issues for faster resolution. Additionally, LLMs could support the identification of irrelevant input features, complementing the feature relevance analysis proposed in this thesis.

6

Conclusion

This thesis presents a DSR study in which we interviewed experts in AHS on the challenges they face when testing these systems. We further developed and evaluated a testing framework that combines systematic approaches with adaptive sampling techniques. The thematic analysis of the expert interviews revealed that experts' main challenges during testing revolved around the complex mining environment in which AHS operate, the fidelity of simulations, and the ability to accurately represent this challenging real-world environment in simulation to validate functionality correctly before HIL testing. Furthermore, experts expressed their need for smarter, more automated methods to test systems that maintain good coverage, are time-efficient, and effectively catch critical or edge cases.

Grid-search testing, random testing, and combinatorial testing were the methods selected for the experimental phase, followed by adaptive sampling using tree-based models (decision trees and random forests) and clustering techniques. Our findings showed that grid-search testing can be computationally expensive, and that methods like combinatorial and random testing can be used in combination or as substitutes to achieve good coverage while reducing the number of test cases needed. This aligns with feedback from several experts who highlighted the need to balance efficiency with coverage of more critical scenarios. Furthermore, adaptive sampling has shown promising results, helping experts identify critical edge cases in a time-efficient manner. However, adaptive sampling techniques rely on model-driven approaches and therefore require training data. This makes it vital that early-stage primary or exploratory testing, conducted prior to model training, uses methods that collect sufficient systematic data for the models to fit effectively. We found that grid-search testing performed best for this purpose, with combinatorial and random testing following closely behind.

The choice of models for adaptive sampling is also vital. The decision tree and random forest models, for instance, were used in this thesis because they are highly interpretable, widely used, and well-documented. These models provide experts in AHS with valuable tools to understand system behaviors and identify important decision boundaries where behavior may change. We used feature importance techniques from decision tree and random forest models to reduce input space dimensionality and filter out irrelevant parameters, enabling adaptive iterations to focus on the most influential factors. To further support this, the NNS method was used to identify

clusters of similar input scenarios that produced different outcomes. This helped isolate decision boundaries and highlight which input parameters triggered behavioral changes, thereby aiding in debugging and system refinement. Ultimately, the results show that this testing framework enhances failure detection and coverage of the input space more efficiently than traditional exhaustive or random methods alone. Insights from expert interviews ensure the framework is grounded in real-world challenges, increasing its relevance and applicability to practical testing scenarios.

Bibliography

[1] ISO/PAS 21448:2019 road vehicles — safety of the intended functionality. `https://www.iso.org/standard/70939.html`, 2019.

[2] The Safety Pool Initiative. `https://www.safetypool.ai/`, 2021.

[3] Mohammad Reza Abbasifard, Bijan Ghahremani, and Hassan Naderi. A survey on nearest neighbor search methods. *International Journal of Computer Applications*, 95(25), 2014.

[4] Yunfeng Ai, Yuhang Liu, Yu Gao, Chen Zhao, Xiang Cheng, Jinpeng Han, Bin Tian, Long Chen, and Fei-Yue Wang. Pmworld: A parallel testing platform for autonomous driving in mines. *IEEE Transactions on Intelligent Vehicles*, 9(1):1402–1411, 2023.

[5] Yashwanth Annpureddy, Che Liu, Georgios Fainekos, and Sriram Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 254–257. Springer, 2011.

[6] Benoît Barbot, Nicolas Basset, Thao Dang, Alexandre Donzé, James Kapinski, and Tomoya Yamaguchi. Falsification of cyber-physical systems with constrained signal spaces. In *NASA Formal Methods Symposium (NFM)*, pages 420–439. Springer, 2020.

[7] Raja Ben Abdessalem, Shiva Nejati, Lionel C Briand, and Thomas Stifter. Testing advanced driver assistance systems using multi-objective search and neural networks. In *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 63–74. ACM, 2016.

[8] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[9] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[10] Long Chen, Yuting Xie, Yuhang He, Yunfeng Ai, Bin Tian, Lingxi Li, Shirong Ge, and Fei-Yue Wang. Autonomous mining through cooperative driving and operations enabled by parallel intelligence. *Communications Engineering*, 3(1):75,

2024.

[11] Tommaso Dreossi, Daniel J Fremont, Shromona Ghosh, Edward Kim, Hadi Ravanbakhsh, Marcell Vazquez-Chanlatte, and Sanjit A Seshia. VerifAI: A toolkit for the design and analysis of artificial intelligence-based systems. In *International Conference on Computer Aided Verification (CAV)*, pages 432–442. Springer, 2019.

[12] Hamid Ebadi, Mahshid Helali Moghadam, Markus Borg, Gregory Gay, Afonso Fontes, and Kasper Socha. Efficient and effective generation of test cases for pedestrian detection-search-based software testing of baidu apollo in svl. In *International Conference on Artificial Intelligence Testing (AITest)*, pages 103–110. IEEE, 2021.

[13] D. Fogelberg and E. Hult Pappas. Automated robustness simulation testing of an autonomous vehicle. *Department of Computer Science*, 2018.

[14] Daniel J Fremont, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. Scenic: a language for scenario specification and scene generation. In *ACM SIGPLAN Conference on Programming Language Ddesign and Implementation (PLDI)*, pages 63–78. ACM, 2019.

[15] Tarek Gaber, Yassine El Jazouli, Esraa Eldesouky, and Ahmed Ali. Autonomous haulage systems in the mining industry: cybersecurity, communication and safety issues and challenges. *Electronics*, 10(11):1357, 2021.

[16] Shirong Ge, Fei-Yue Wang, Jianjian Yang, Zhen Ding, Xiao Wang, Yuchen Li, Siyu Teng, Zhongmin Liu, Yunfeng Ai, and Long Chen. Making standards for smart mining operations: Intelligent vehicles for autonomous mining transportation. *IEEE Transactions on Intelligent Vehicles*, 7(3):413–416, 2022.

[17] Philipp Helle, Wladimir Schamai, and Carsten Strobel. Testing of autonomous systems–challenges and current state-of-the-art. In *INCOSE international symposium*, volume 26, pages 571–584. Wiley Online Library, 2016.

[18] WuLing Huang, Kunfeng Wang, Yisheng Lv, and FengHua Zhu. Autonomous vehicles testing methods review. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 163–168. IEEE, 2016.

[19] David Kaufmann, Lorenz Klampfl, Florian Klück, Martin Zimmermann, and Jianbo Tao. Critical and challenging scenario generation based on automatic action behavior sequence optimization: 2021 ieee autonomous driving ai test challenge group 108. In *International Conference on Artificial Intelligence Testing (AITest)*, pages 118–127. IEEE, 2021.

[20] Dilnawaz Kaur. The impact of autonomous vehicles on mining operations: Enhancing safety and productivity through technological advancements | Published

in Scholarly Review Journal — scholarlyreview.org. [Accessed 12-12-2024].

[21] Trupti M Kodinariya, Prashant R Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[22] Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J Kochenderfer. Adaptive stress testing for autonomous vehicles. In *Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2018.

[23] Changwen Li, Chih-Hong Cheng, Tiantian Sun, Yuhang Chen, and Rongjie Yan. Comopt: Combination and optimization for testing autonomous driving systems. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7738–7744. IEEE, 2022.

[24] Mikael Lindvall, Adam Porter, Gudjon Magnusson, and Christoph Schulze. Metamorphic model-based testing of autonomous systems. In *2017 IEEE/ACM 2nd International Workshop on Metamorphic Testing (MET)*, pages 35–41. IEEE, 2017.

[25] Tilo Linz. Testing autonomous systems. *The Future of Software Quality Assurance*, pages 61–75, 2020.

[26] Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

[27] Danellie Lynas and Tim Horberry. Exploring the human factors challenges of automated mining equipment. In *46th Annual Human Factors and Ergonomics Society of Australia Conf. 2010, HFESA 2010*, pages 115–22, 2010.

[28] Rupak Majumdar, Aman Mathur, Marcus Pirron, Laura Stegner, and Damien Zufferey. Paracosm: A test framework for autonomous driving simulations. In *International Conference on Fundamental Approaches to Software Engineering (FASE)*, pages 172–195. Springer, 2021.

[29] Mauricio Mascaró, Isao Parra, Carlos Tampier, Mauricio Correa, and J Ruiz del Solar. Development and testing of autonomous mining equipment in underground mines. In *7th International Congress on Automation in Mining–Automining 2020*, 2020.

[30] Galen E Mullins, Paul G Stankiewicz, and Satyandra K Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1443–1450. IEEE, 2017.

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[32] Thomas Ponn, Christian Gnandt, and Frank Diermeyer. An optimization-based method to identify relevant scenarios for type approval of automated vehicles. In *Proceedings of the ESV—International Technical Conference on the Enhanced Safety of Vehicles, Eindhoven, The Netherlands*, pages 10–13, 2019.

[33] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020.

[34] Qunying Song. Critical scenario identification for testing of realistic autonomous driving systems. *Department of Computer Science*, 2022.

[35] Qunying Song, Per Runeson, and Stefan Persson. A scenario distribution model for effective and efficient testing of autonomous driving systems. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–8, 2022.

[36] Eric Thorn, Shawn C Kimmel, Michelle Chaka, Booz Allen Hamilton, et al. A framework for automated driving system testable cases and scenarios. Technical report, NHTSA, 2018.

[37] Bin Tian, Junhui Wang, Yuting Xie, Yuchen Li, Yachen Zhu, and Long Chen. High-definition mapping for autonomous driving in surface mines. *IEEE Transactions on Intelligent Vehicles*, pages 1–27, 2024.

[38] Paweł Trybała. Lidar-based simultaneous localization and mapping in an underground mine in złoty stok, poland. In *IOP Conference Series. Earth and Environmental Science*, volume 942. IOP Publishing, 2021.

[39] Kesav Viswanadha, Francis Indaheng, Justin Wong, Edward Kim, Ellen Kalvan, Yash Pant, Daniel J Fremont, and Sanjit A Seshia. Addressing the ieee av test challenge with scenic and verifai. In *International Conference on Artificial Intelligence Testing (AITest)*, pages 136–142. IEEE, 2021.

[40] Kesav Viswanadha, Edward Kim, Francis Indaheng, Daniel J Fremont, and Sanjit A Seshia. Parallel and multi-objective falsification with scenic and verifai. In *International Conference on Runtime Verification (RV)*, pages 265–276. Springer, 2021.

[41] Jan Vom Brocke, Alan Hevner, and Alexander Maedche. Introduction to design science research. *Design science research. Cases*, pages 1–13, 2020.

[42] Philip Wickberg, Anas Fattouh, Sara Afshar, and Markus Bohlin. Exploring dynamic map validation at construction sites: A case study and feasibility analysis. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 3865–3871. IEEE, 2024.

[43] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, et al. *Experimentation in software engineering*, volume 236.

Springer, 2012.

[44] Zhang Xinxin, Li Fei, and Wu Xiangbin. Csg: Critical scenario generation from real traffic accidents. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1330–1336. IEEE, 2020.

[45] Mohammed I Younis, Kamal Z Zamli, and Nor Ashidi Mat Isa. A strategy for grid based t-way test data generation. In *2008 First International Conference on Distributed Framework and Applications*, pages 73–78. IEEE, 2008.

[46] Zhenya Zhang, Paolo Arcaini, and Ichiro Hasuo. Constraining counterexamples in hybrid system falsification: Penalty-based approaches. In *NASA Formal Methods Symposium (NFM)*, pages 401–419. Springer, 2020.

[47] Yuan Zhou, Yang Sun, Yun Tang, Yuqi Chen, Jun Sun, Christopher M Poskitt, Yang Liu, and Zijiang Yang. Specification-based autonomous driving system testing. *IEEE Transactions on Software Engineering*, 49(6):3391–3410, 2023.

[48] Adrian Zlocki, Julian Bock, and Lutz Eckstein. Database of relevant traffic scenarios for highly automated vehicles. In *Autonomous Vehicle Test and Development Symposium*, pages 20–22, 2017.

A

Appendix

Interview Guide for Challenges in Testing AHS

This appendix presents the interview protocol used to collect insights from professionals involved in the development, testing, and requirements management of AHS. The purpose was to understand key challenges, scenario identification strategies, and performance validation practices in AHS testing.

1. Introduction

1. Please introduce yourself and describe your role related to autonomous haulage systems.

2. How does your role involve testing or ensuring the correct operation of AHS?

2. Key Challenges in Testing AHS

1. What are the biggest challenges you encounter when testing or working with autonomous haulage systems?

2. How does testing AHS differ from testing other types of systems you have worked with?

3. From your perspective, what are the primary challenges in verifying that autonomous decisions are correct and reliable?

For Requirements Managers only:

1. What challenges do you face when defining requirements for AHS?

2. How do testing requirements for AHS differ from those for other systems?

3. What difficulties exist in ensuring that autonomous decision-making is correctly specified and validated through requirements?

3. Identifying Critical Performance Scenarios

1. How do you or your team determine which performance scenarios are most important to test or validate in AHS?

2. Are there specific environmental or operational factors (e.g., dust, terrain, weather) that complicate testing?

3. How do you ensure rare or high-risk scenarios are considered in testing or system design?

For Requirements Managers only:

1. How do you define and prioritize critical performance scenarios in system requirements?

2. What environmental or operational conditions are crucial to capture for reliable system performance?

3. How are rare or high-risk scenarios incorporated into requirements to support testing and validation?

4. Performance Validation and Key Performance Indicators (KPIs)

1. What KPIs do you use to assess whether AHS has passed or failed testing, particularly regarding safety and reliability?

2. How do you ensure the KPIs reflect real-world system performance and readiness for deployment?

For Requirements Managers only:

1. What KPIs are defined in requirements to assess AHS safety and reliability during testing?

2. How do you verify that KPIs align with actual performance and deployment conditions?

5. Closing: Suggestions for Improvement

1. If you could improve one aspect of AHS testing, what would it be?

2. What advice would you offer to enhance scenario identification and validation processes?

For Requirements Managers only:

1. What improvements would you suggest in defining or structuring testing requirements?

2. Any advice on refining scenario identification and validation from a requirements

and system design perspective?

Note: Prior to the interviews, participants were asked for consent to record the sessions for transcription and analysis purposes.

B

appendix

B.1 Model Performance Details

Metric / Model	Random Forest	Decision Tree
Confusion Matrix		
Pass (0) predicted as Pass (0)	477	477
Pass (0) predicted as Failure (1)	1	1
Failure (1) predicted as Pass (0)	8	8
Failure (1) predicted as Failure (1)	127	127
Classification Report		
Pass (0) Precision	0.98	1.00
Pass (0) Recall	1.00	0.99
Pass (0) F1-score	0.99	0.99
Support	478	478
Failure (1) Precision	0.99	0.97
Failure (1) Recall	0.94	0.99
Failure (1) F1-score	0.97	0.98
Support	135	135
Accuracy	0.99	0.99
Macro Avg Precision	0.99	0.98
Macro Avg Recall	0.97	0.99
Macro Avg F1-score	0.98	0.99
Weighted Avg Precision	0.99	0.99
Weighted Avg Recall	0.99	0.99
Weighted Avg F1-score	0.99	0.99
Feature Importance		
terminal_position_y	0.464	0.805
terminal_position_x	0.388	0.035
terminal_yaw	0.148	0.160

Table B.1: Comparison of Random Forest and Decision Tree models for Open Area Planner Scenario: confusion matrices, classification reports, and feature importance.

Model	Feature	Importance
Random Forest	ob_Dist	0.4520
	is_Rev	0.3407
	velocity	0.2073
	seg_W_2	0.0
	seg_W_1	0.0
	ob_Lat	0.0
Decision Tree	ob_Dist	0.4249
	is_Rev	0.3696
	velocity	0.2055
	seg_W_2	0.0
	ob_Lat	0.0
	seg_W_1	0.0

Table B.2: Feature importance for obstacle avoidance scenario using Random Forest and Decision Tree models.

Model	MAE	R^2
Random Forest	0.0297	0.9803
Decision Tree	0.0311	0.9766

Table B.3: Regression metrics for obstacle avoidance scenario using Random Forest and Decision Tree models.