# A Generalized Metamorphic Testing Platform for Regression Models in Autonomous Driving Systems

An Undergraduate Project Proposal Report Submitted to the

Department of Electrical and Information Engineering
Faculty of Engineering
University of Ruhuna
Sri Lanka

in partial fulfillment of the requirements for the

**Degree of Bachelor of the Science of Engineering Honours**

by

Akurana B.N.T.M. – EG/2020/3812
Pabasara P.M.G.T – EG/2021/4701
Peiris P.R.S – EG/2021/4706
Perera G.C.M – EG/2021/4707

| | |
|---|---|
| Dr. Kaveen Liyanage | Mrs. Sithara Mahagama |
| (Supervisor) | (Co-Supervisor) |

# Abstract

Autonomous driving systems increasingly rely on machine-learning regression models to perform important tasks like predicting vehicle trajectories, maintaining lanes, controlling steering, and planning vehicle movements. Although these models have improved a lot, it is still challenging to ensure their reliability and safety in real-world driving situations because of the test oracle problem. A test oracle is a mechanism that can systematically verify whether a test result is correct for a given input. However, a major challenge occurs when such an oracle does not exist or is too costly to use. This problem creates a major challenge because it makes verification difficult: it becomes impossible to clearly determine whether the continuous-valued output of a program is correct or incorrect for a given input. Furthermore, this creates methodological limitations, because the absence of a test oracle reduces the effectiveness of traditional test case selection methods that depend on known expected results. As a result, traditional testing is not sufficient to ensure reliable behavior in unexpected situations. To address this issue, alternative approaches like metamorphic testing are used. Instead of relying on a single predefined expected result, metamorphic testing verifies program correctness by examining relationships between inputs and outputs across multiple executions [1].

Metamorphic Testing (MT) is a useful approach to solve the test oracle problem by checking how a system behaves when inputs are changed in a controlled way. However, even though MT has been successfully used in other fields, it is not widely used in autonomous driving regression models because there is no common, reusable, and scalable testing framework.

This project focuses on designing and implementing a generic metamorphic testing platform specifically for the regression models used in autonomous driving systems. This approach makes it easier to create and check test cases without needing a specific test oracle. It uses a general metamorphic testing template that works with any model and includes common testing features. The framework is a Python system made of separate modules and is tested using real autonomous driving models and example tests. A web-based interface was also built to make the system easier to use and access for researchers and practitioners.
The proposed platform aims to make autonomous driving regression models more reliable, robust, and ready for real-world use. By reducing the need for manual testing and increasing test coverage, it helps make the deployment of self-driving car technology safer.

# List of Acronyms

| | |
|---|---|
| ADAS | Advanced Driver Assistance Systems |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| Euro NCAP | European New Car Assessment Programme |
| GPR | Gaussian Process Regression |
| IEEE | Institute of Electrical and Electronics Engineers |
| LKAS | Lane Keeping Assist System |
| ML | Machine Learning |
| MPC | Model Predictive Control |
| MR | Metamorphic Relation |
| MT | Metamorphic Testing |
| SAE | Society of Automotive Engineers |
| SGPR | Sparse Gaussian Process Regression |
| WEF | World Economic Forum |

# Contents

**Contents** iii

**List of Figures** v

**List of Tables** vi

**1 Introduction** 1

1.1 Background . . . . . 2
1.2 Problem Statement . . . . . 4
1.3 Objectives and Scope . . . . . 4
1.3.1 Project Goal . . . . . 4
1.3.2 Objectives . . . . . 5
1.3.3 Scope of the Project . . . . . 6

**2 Literature Review** 7

2.1 Background of the Autonomous Driving Systems and Validation . . . . . 7
2.2 Regression Based Models in Autonomous Driving Systems . . . . . 8
2.3 Limitations of Conventional Testing Approaches in Autonomous Driving 9
2.4 Metamorphic Testing: Previous Work . . . . . 10
2.5 Metamorphic Testing in Machine Learning and Autonomous Driving Systems 11
2.6 Gap Identification: Limitations of Existing Metamorphic Testing Approaches 12
2.7 Summary of Existing Systems and Research Gaps and way forward . . . 12

**3 Methodology** 13

3.1 Comprehensive Analysis of Regression Models and Metamorphic Properties 14
3.2 Design of a Generalized Metamorphic Testing Template . . . . . 15
3.3 Proof of Concept: Demonstration on a Representative Regression Model 15
3.4 Validation Using Real World Autonomous Driving Regression Models . . 16
3.5 Implementation of a Python Based Metamorphic Testing Framework . . 16
3.6 System Architecture of the Proposed Framework . . . . . 16
3.7 Web Based Platform for Accessibility and Public Use . . . . . 18

**4 Timeline and Resource Required** 19

4.1 Timeline . . . . . 19

4.2 Resource Required . . . . . 22

**5 Conclusion** **23**

**References** **24**

# List of Figures

1.1 SAE International Levels of Driving Automation (J3016 standard), illustrating the progression from no automation (Level 0) to full autonomy (Level 5) [2]. . . . . 2
1.2 Illustration of the test oracle problem in software testing, highlighting the challenge of determining correct outputs without explicit expected results [3]. 3

2.1 Example of Gaussian Process based trajectory prediction in autonomous driving, showing predicted trajectories with uncertainty bounds for overtaking maneuvers [4]. . . . . 9
2.2 Conceptual overview of metamorphic testing, illustrating the generation of follow up test cases from source test cases using metamorphic relations to validate system behavior without an explicit test oracle [5]. . . . . 10
2.3 Diagram of a Lane Keeping Assist System (LKAS) in autonomous driving, illustrating lane detection, departure warning, and corrective steering [6]. 11

3.1 Overall workflow of the proposed metamorphic testing framework for autonomous driving regression models. . . . . 13
3.2 Structure of the generalized metamorphic testing template illustrating source inputs, transformations, and relational output constraints. . . . . 14
3.3 System architecture of the proposed Python based metamorphic testing framework for autonomous driving regression models. . . . . 17

# List of Tables

4.1 Project Timeline and Milestones . . . . . 21
4.2 Required Resources . . . . . 22

# Chapter 1

# Introduction

With growing use in early autonomous vehicle platforms and commercial driver assistance systems, autonomous driving has become one of the most significant uses of artificial intelligence. The World Economic Forum states that although widespread full autonomy is still limited by technical, legal, and safety issues, vehicle automation technologies are already present on public roads, especially at Levels 2 and 2+, and are anticipated to gradually expand toward higher levels of autonomy over the next ten years [7]. Current autonomous driving systems still have trouble operating dependably in a variety of real world scenarios, despite tremendous advancements in perception, prediction, and control models [8] [9].

## 1.1 Background

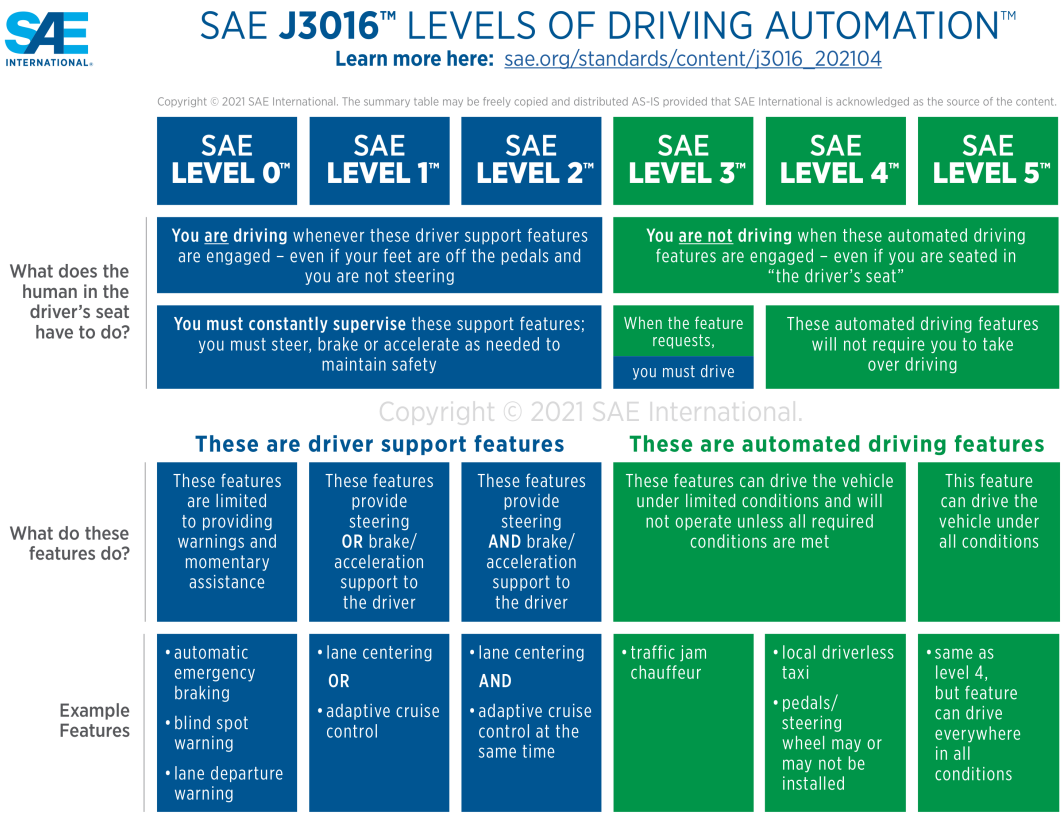


Figure 1.1: SAE International Levels of Driving Automation (J3016 standard), illustrating the progression from no automation (Level 0) to full autonomy (Level 5) [2].

In the SAE J3016 standard, there are six stages of automation of driving, starting with none. (Level 0) all the way to full automation (Level 5), as in Figure 1.1. Currently, the majority of commercially deployed systems act at Level 2 and 2+ and demand constant human supervision and just some automation. This is an illustration of the importance of sound testing. processes will be to ensure reliability and safety in the real implementation.

A major disadvantage that restricts the strength of au is lack of adequate and realistic testing. autonomous driving models. Autonomous systems operate in highly dynamic, com. complicated situations where listing and verification of all the situations in advance is impractical so de can be. ployment. In machine learning based systems which give continuous outputs such as tracks, turning angles or force, this has often been referred to as a problem. the oracle problem, in which it is difficult or even impossible to see the correct output. for a given test input. As has been stated above, the absence of credible test oracles is the bane of machine learning applications, e.g., autonomous driving and advanced driver assistant system (ADAS) [10], [8].

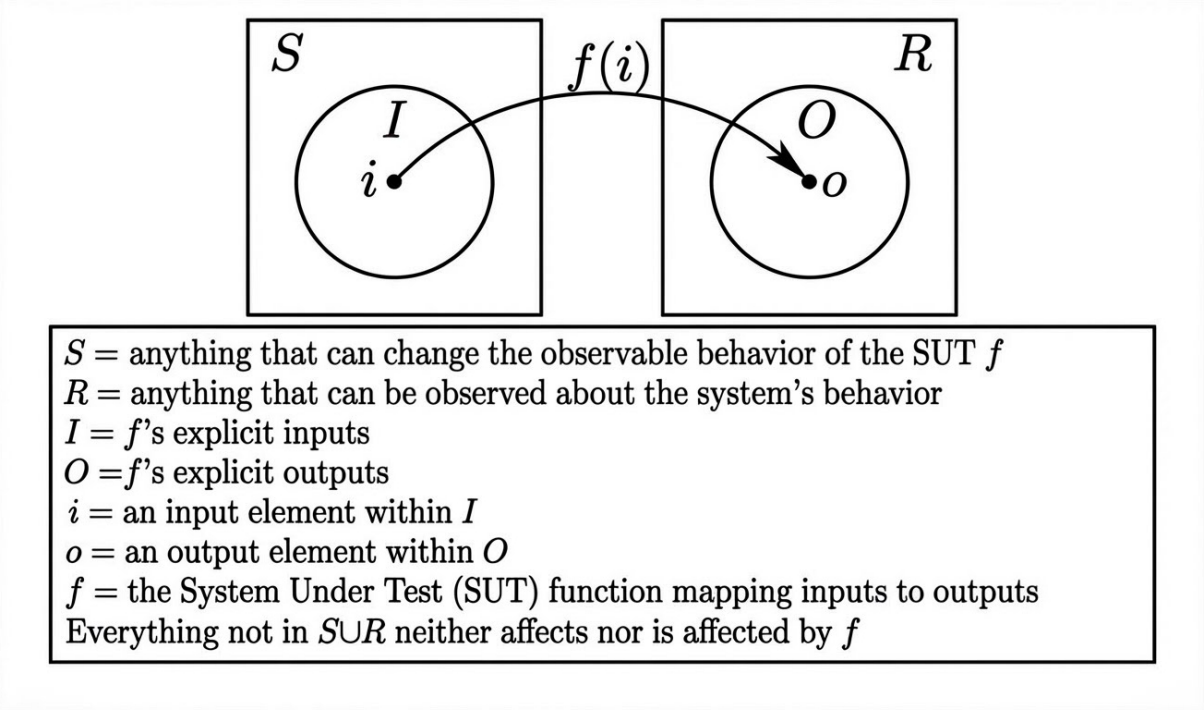


Figure 1.2: Illustration of the test oracle problem in software testing, highlighting the challenge of determining correct outputs without explicit expected results [3].

Figure 1.2 show the relationship between stimuli and observations with regard to software testing. Stimuli ($S$) represent all factors that can regulate the behaviour of the system under test (SUT), including explicit test inputs provided by the tester, environ mental conditions, configuration parameters, and sensor inputs. A subset of these stimuli, denoted as $I \subset S$, correlate with the explicit inputs directly applied to the system. Observations ($R$) stand in for all aspects of the system's behaviour that can be measured after stimulation, including explicit outputs ($O \subset R$) as well as non functional properties like execution time, resource utilization, and internal state changes. Elements that do not belong to either the stimulus set or the observation set neither affect nor are affected by the execution of the SUT [3].
Testing fundamentally contains applying stimuli to the system and observing the resulting behaviour. However, as demonstrated in the figure, this process alone does not guarantee the availability of a clear mechanism for determining whether the observed behaviour is correct. This challenge is formally captured by the concept of a test oracle, which is commonly defined as a predicate that determines whether a given program execution satisfies its considered specification. Formally, a test oracle can be expressed as a function

$$O : E \to \{\text{true}, \text{false}\}, \tag{1.1}$$

where $E$ dsymbolize the set of all possible executions of the system under test [3].
The oracle problem occurs when such a function cannot be reliably defined, implemented, or evaluated for all executions. Barr *et al.* stress the fact that this restriction is especially acute in the present day software systems, which tend to be non deterministic,

highly complex, 3 and data driven and in which specifying the correct outputs is either impractical or impossible [3]. This is particularly acute when it comes to machine learning autonomous driving systems, which generate continuous values which are reliant on intricate and changing environmental inputs. This therefore leads to the need to adopt alternative testing approaches like metamorphic testing, which minimize or do away with reliance on conventional test oracles [1, 3, 11].

Chen first came up with Metamorphic Testing in 1998. Metamorphic Testing is an useful way to deal with the oracle problem. Metamorphic Testing checks if a system has the qualities it needs which are called Metamorphic Relations. These Metamorphic Relations are like rules that apply to runs of the system with related inputs rather than just looking at what the system should output. Even if we do not know what the correct output is we can still find errors if these rules are not followed. Many areas have used Metamorphic Testing. It has worked well. These areas include navigation software, search engines, cyber security, machine learning systems and autonomous driving applications. Metamorphic Testing has been very effective, in all these fields [11], [9]. Geometric transformation based metamorphic relations have proven particularly successful in exposing discrepancies under perspective, spatial, or environmental alterations in the context of autonomous systems problem statement.

## 1.2 Problem Statement

Though metamorphic testing has its benefits, it remains difficult to apply the testing in actual practice. The existing methods often involve actions by the developers to construct metamorphic relations manually and execute certain test scripts with each model and scenario. Due to the inconsistent test design, This is an inadequate domain coverage method, which is time-consuming, and may contain error-prone, and even fail to capture important edge instances. Consequently, the metamorphic testing is not a readily available and a systematically integrated part of the autonomous driving regression model creation process yet.

## 1.3 Objectives and Scope

### 1.3.1 Project Goal

Creating and implementing a generalized metamorphic testing environment for autonomous driving regression models is the main objective of this project. In order to facilitate efficient testing of both recently created and current autonomous driving models, the suggested platform seeks to streamline the definition, implementation, and assessment of metamorphic relations.

### 1.3.2 Objectives

In accomplishing the above project goal the following objectives are established. Each objective is with transparent verification measures to measure completion success.

#### 1.3.2.1 Objective 1: Identify and formalize applicable metamorphic relations for autonomous driving regression models

This goal aims at establishing significant metamorphic relationships (e.g. rotation, trans). real world driving properties are represented by relation, symmetry, and invariance.

Verification metrics:

- • Definition of at least six metamorphic relations that can be applied to autonomous driving.
- • Specification of the input output constraints of 100% of defined relations.

#### 1.3.2.2 Objective 2: Design a generalized and model agnostic metamorphic testing template

This goal would guarantee that the methodology of testing can be replicated in other regression models that do not need model specific test oracles.

Verification metrics:

- Effective initiation of the template of at least three regression models.
- No structural adjustment is needed in implementing the template on models.

#### 1.3.2.3 Objective 3: Validate the proposed framework using representative autonomous driving regression models

This objective shows that the framework is really useful and works well in life situations. The framework is something that can be applied in a way and it is effective. This means the framework is good, for use and it gets the job done under realistic testing scenarios.

Verification metrics:

- Application of the framework to at least two autonomous driving regression models.
- Execution of at least five metamorphic relations per model.
- Identification of violations or robustness confirmation for each evaluated model.

#### 1.3.2.4 Objective 4: Develop a web based interface for improved accessibility and usability

This objective makes sure that the framework can be used by people who work with it every day without needing to know a lot, about the framework. The framework is something that people should be able to use knowledge of metamorphic testing implementation.

Verification metrics:

- A functional web interface supporting model selection, metamorphic relation configuration, and result visualization.
- Successful end to end test execution through the interface with minimal user inter action.

### 1.3.3 Scope of the Project

The scope of this project is defined to ensure clarity and feasibility and is outlined as follows:

- This project is limited to the regression based models applicable in autonomous driving activities such as lane keeping, trajectory prediction and estimation of steering angle.
- Metamorphic testing is done to manage the oracle problem in machine learning based systems.
- The suggested framework is model-neutral and runs on the software testing level, it does not substitute the current simulation platforms or vehicle control systems.
- Validation is done based on simulated driving conditions and software level testing as opposed to actual vehicle deployment.
- Classification models, sensor hardware validation, perception pipelines and low level vehicle actuation control are also explicitly beyond the framework of this work.

# Chapter 2

# Literature Review

## 2.1 Background of the Autonomous Driving Systems and Validation

With limited or no human dependence in vehicle controlling and operating autonomous driving systems use safety critical cyber physical systems which are doing decision making and controlling components [2] [7]. These systems continuously process data from multiple sensors in the vehicle, including cameras, LiDAR sensors, radar, and Global Positioning System (GPS) units, to understand road geometry, traffic participants, and dynamic obstacles overall the surrounding environment. Based on this idea, control actions like steering, acceleration, braking, and lane keeping are generated to make sure the safety of the vehicle operation.

The surrounding environment of autonomous driving vehicles is dynamic and uncertain. Factors like sudden variations of road conditions, complex traffic interactions and intersections, sudden weather changes, sensor noise and unpredictable human behavior largely increase system complexity and unpredictability [12] [13]. Hence, ensuring the reliability, robustness, and safety of autonomous driving software is an important requirement while developing the lifecycle. The proper and harder testing and validation of systems is required due to the high unpredictability and sudden variations of inputs [14].

Testing systems play a vital role in the development of ADS. Currently widely used validation techniques normally dependant on predefined datasets, simulation based testing, and scenario based evaluations using known traffic situations and standardized benchmarks [15] [16]. While these techniques make it controlled for experimentation and functional verification, their efficiency becomes limited when applied to machine learning based systems, specially those generating continuous outputs and showing non deterministic behavior.

Most of the core components of ADS, especially regression based machine learning models, generate continuous outputs like steering angles, vehicle trajectories, velocity profiles, and control commands [4]. [8]. In real world driving scenarios, planning the correct output decision for a given input in advance is often impossible. This causes the well known test oracle problem, where the absence of exact expected outputs prevents reliable validation [1] [3]. Furthermore, predefined datasets and scenario based testing fails to properly capture rare edge cases and unseen driving conditions, making it more difficult to validate sophisticated real world performance.

As ADS heavily rely on machine learning for perception and control, alternative validation techniques that do not depend on exact expected outputs have attracted increasing attention. This has caused the exploration of metamorphic testing, which evaluates system correctness by verifying expected relationships between multiple executions rather than individual outputs [9] [11].

## 2.2 Regression Based Models in Autonomous Driving Systems

Key functionalities like trajectory prediction, lane centerline tracking, steering control, and motion planning must be performed correctly to make sure the safety and efficiency and operational safety of ADS. Even though deep neural network based methods have accomplished strong evidence of efficiency in perception tasks, regression based models stays vital for control focused and prediction based applications because of their interpretability, uncertainty estimation capabilities, and robustness when handling continuous data [8] [17].

Among regression approaches, Gaussian Process Regression (GPR) has been widely used in autonomous driving for trajectory prediction and vehicle behavior modeling. GPR provides probabilistic predictions with uncertainty estimates, which are essential for safety aware decision making in dynamic driving environments. However, conventional GPR is more vulnerable for its computational complexity, limiting its scalability in real time systems [18].

To address this issue, Sparse Gaussian Process Regression (SGPR) has been introduced as a reliable solution that maintains accuracy while ensuring the acceptance of real time inference [4]. SGPR based models have been successfully applied into Model Predictive Control (MPC) frameworks for autonomous driving tasks like lane keeping and overtaking maneuvers. Studies show improved trajectory prediction accuracy and control stability across various traffic conditions [9] [19].

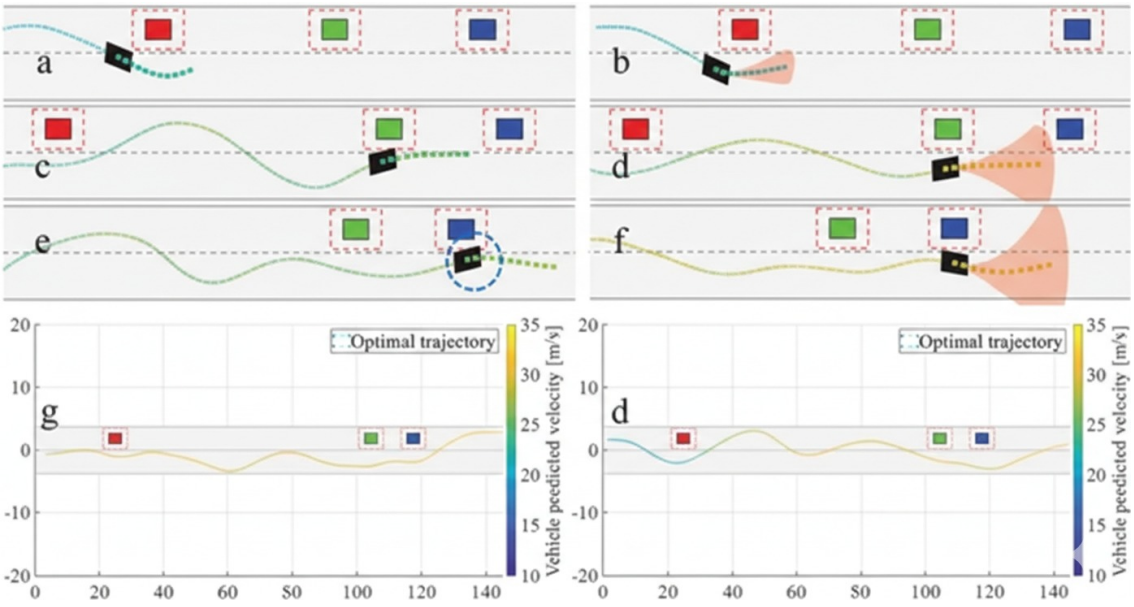


Figure 2.1: Example of Gaussian Process based trajectory prediction in autonomous driving, showing predicted trajectories with uncertainty bounds for overtaking maneuvers [4].

As an example for Gaussian Process based trajectory prediction in an autonomous driving setting can be shown in Figure 2.1, which indicates the expected vehicle trajectories and related uncertainty bounds. These regression based models are very useful since they can give uncertainty estimates, which are essential for making safety aware decisions in dynamic driving conditions.

Recent research presented at the IEEE Intelligent Vehicles Symposium further show that SGPR based trajectory prediction models can outperform deep learning approaches in terms of inference efficiency, interpretability, and adaptability, specially when combined with geometric transformations such as translation and rotation equivariance [4] [20].

## 2.3 Limitations of Conventional Testing Approaches in Autonomous Driving

In spite of the advances in modeling technologies, the testing and validation of ADS remain a major challenge. Traditional testing techniques are normally limited to predefined datasets, simulation based environments, and manually designed driving scenarios [15] [16]. While these methods give proper evaluations, they struggle to weigh the full complexity of real world driving scenarios.

Autonomous vehicles operate in highly dynamic environmental conditions involving unpredictable agents, rare edge cases, and continuously complicating traffic situations [12] [21]. Therefore, models which perform well in controlled testing environments may show unexpected or unsafe behavior when they are exposed to untested novel conditions. This limitation is especially worse for regression based models with continuous outputs, where

defining correct expected outputs for all possible inputs is impossible.

Henceforth, there is an increasing need for systematic testing techniques that can evaluate model robustness and efficiency beyond predefined datasets and scenarios, without needing exact output oracles [1] [3].

## 2.4 Metamorphic Testing: Previous Work

Metamorphic testing (MT) was first introduced by Chen in 1998 as a software testing technique designated to address the oracle problem [11]. Instead of relying on predefined expected results, MT verifies program accuracy by checking relationships between multiple executions under carefully transformed inputs.

These expected relationships, known to be metamorphic relations (MRs), represent basic properties that the system under test should achieve. If a metamorphic relation is violated, a fault is detected even if the correct output is unknown. This paradigm significantly reduces dependence on explicit test oracles and has proven effective for complex, data driven systems where traditional testing methods fail [1].

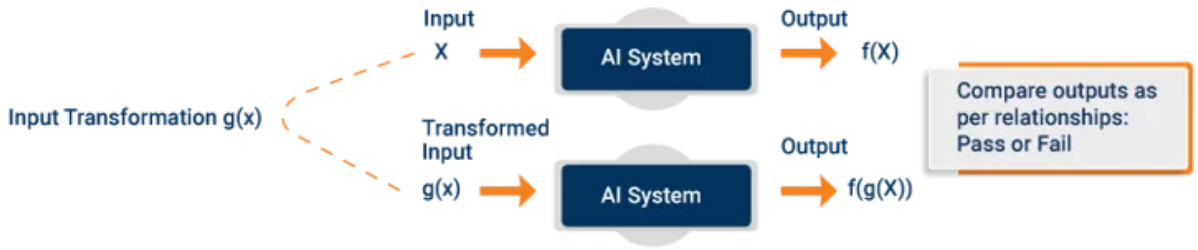


Figure 2.2: Conceptual overview of metamorphic testing, illustrating the generation of follow up test cases from source test cases using metamorphic relations to validate system behavior without an explicit test oracle [5].

Instead of depending on explicit expected outputs, metamorphic testing uses prefound metamorphic relations to generate follow up test cases from an initial source test case, as shown in Figure 2.2. The diagram in the figure indicates how the oracle problem in complex systems, where correct outputs are hard to predict ahead of time, is addressed by using connections between many executions to check system behavior.
MT has since been successfully applied to numerical computation, search engines, cybersecurity, and machine learning systems, showing its general applicability and reliability in oracle challenged environments [8] [22].

## 2.5 Metamorphic Testing in Machine Learning and Autonomous Driving Systems

In recent past few years, metamorphic testing has attracted a huge attention in the validation of machine learning based systems, including autonomous driving applications [9] [23]. As a result of continuous outputs and complex system behaviors, MT is particularly suitable for testing Advanced Driver Assistance Systems (ADAS).

Previous studies have applied geometric transformation based metamorphic relations, such as rotation, translation, and symmetry, to verify Lane Keeping Assist Systems (LKAS) under Euro NCAP driving scenarios [6] [9].

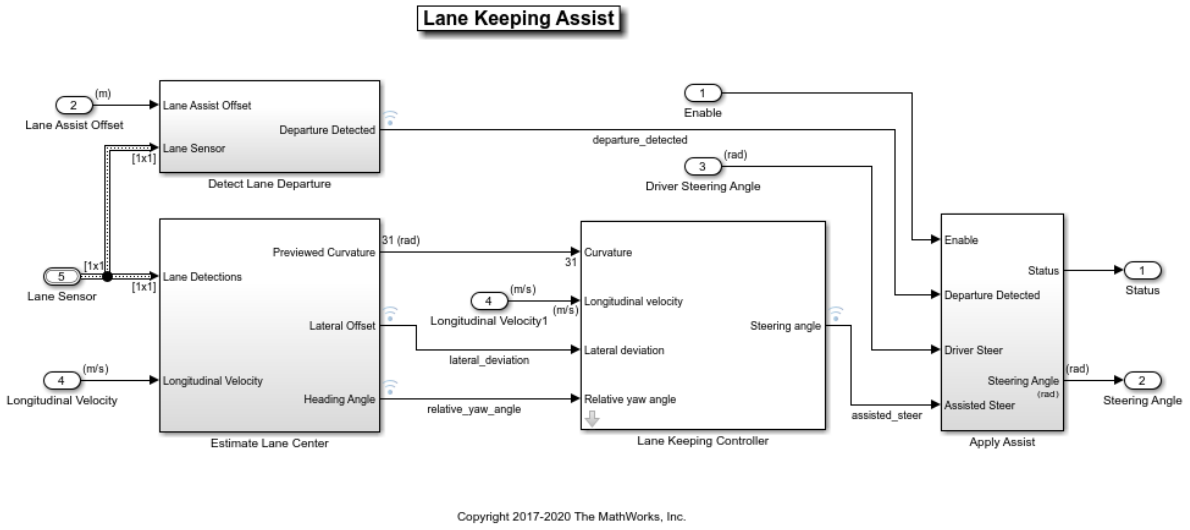


Figure 2.3: Diagram of a Lane Keeping Assist System (LKAS) in autonomous driving, illustrating lane detection, departure warning, and corrective steering [6].

A Lane Keeping Assist System (LKAS) usually includes lane detection, vehicle position assessment, and accurate steering control components, as shown in Figure 2.3. Since geometric transformations and symmetry based metamorphic relations can be used to verify system consistency under various driving conditions, LKAS's modular structure makes it a good case for metamorphic testing.

Researchers have shown that symmetry and rotation based MRs can reveal critical system faults that are often missed by current traditional testing. From generating huge numbers of follow up test cases from existing driving scenarios, MT enables systematic exploitation of the respected model behavior without depending on explicit or exact expected outputs [11] [24].

Additional metamorphic relations explored in the literature include translation invariance, trajectory preserving transformations, environmental condition variations, and sensor symmetry relations, which are particularly effective for testing regression based per-

ception and control models [9] [25].

## 2.6 Gap Identification: Limitations of Existing Metamorphic Testing Approaches

In spite of its performance, the practical applications of metamorphic testing in autonomous driving remains limited. Most currently used MT applications rely on ad hoc test scripts, where metamorphic relations are manually defined and tightly coupled to specific models, simulators, or experimental setups [9] [11].

Although geometric transformation based MRs have proven successful, they are frequently implemented in a case specific and non reusable manner. This makes it high manual effort, limited scalability, and poor integration into continuous development pipelines [23].

Furthermore, current MT approaches usually require extensive domain knowledge and familiarity with simulation platforms, increasing development and maintenance complexity for the users. These challenges are especially pronounced for regression based autonomous driving models, where continuous outputs and complex interactions complicate MR design [8].

## 2.7 Summary of Existing Systems and Research Gaps and way forward

Existing currently used autonomous driving systems widely use regression based models such as GPR, SGPR, and learning based control models for trajectory prediction, lane keeping, and steering contro [4] [8] [9]. While these models give strong prediction abilities and unpredictable estimation, their validation remains challenging because of the oracle problem and the limitations of traditional testing approaches [1] [3].

Although metamorphic testing has shown positivity in overcoming these challenges, current solutions lack abstraction, automation, and scalability. This surely motivates the need for a generalized, reusable, and model agnostic metamorphic testing framework for autonomous driving regression models, which makes the fundamental basis of the proposed research.

# Chapter 3

# Methodology

To develop a generalized and reusable metamorphic testing framework applicable to a wide range of regression models in autonomous driving, this work follows a structured, multi stage methodology as outlined below.

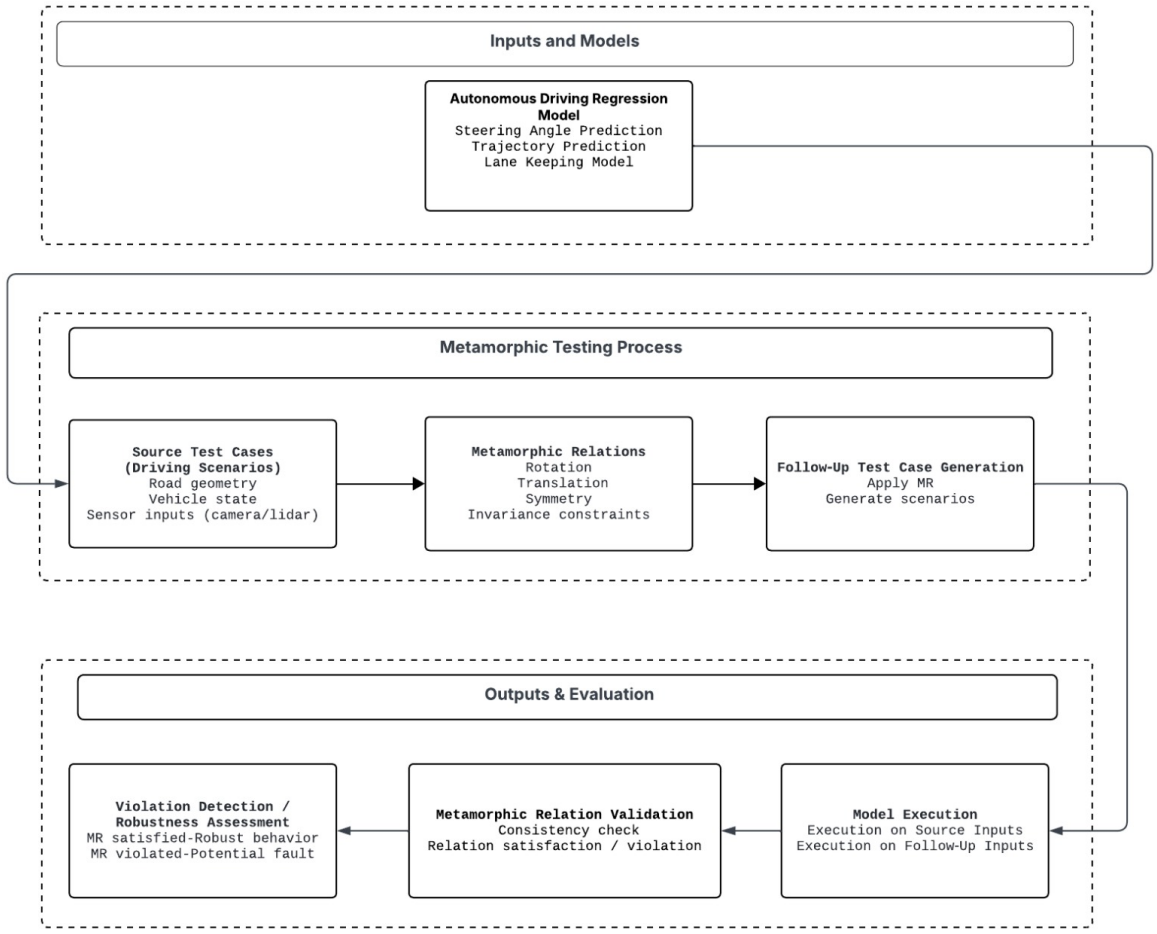


Figure 3.1: Overall workflow of the proposed metamorphic testing framework for autonomous driving regression models.

An overview of suggested metamorphic testing framework workflow is given in Figure 3.1. It shows the steps from regression model analysis to execution and evaluation of metamorphic tests. The workflow demonstrates how metamorphic relations are used to

change source test cases, which are then assessed to find violations without the need for conventional test oracles.

## 3.1 Comprehensive Analysis of Regression Models and Metamorphic Properties

The first stage involves a comprehensive study of regression models commonly employed in autonomous driving and related cyber physical systems. These includes linear and nonlinear regression models, Gaussian Process Regression (GPR), Sparse Gaussian Process Regression (SGPR), neural network based regression models, and hybrid learning based regression approaches. For each category, existing metamorphic testing strategies and applicable metamorphic relations (MRs) such as geometric transformations, symmetry relations, monotonicity, invariance, and consistency relations are systematically analyzed. This analysis ensures the proposed framework remains model agnostic and it is capable of supporting diverse regression behaviors and output characteristics typical of autonomous driving systems.

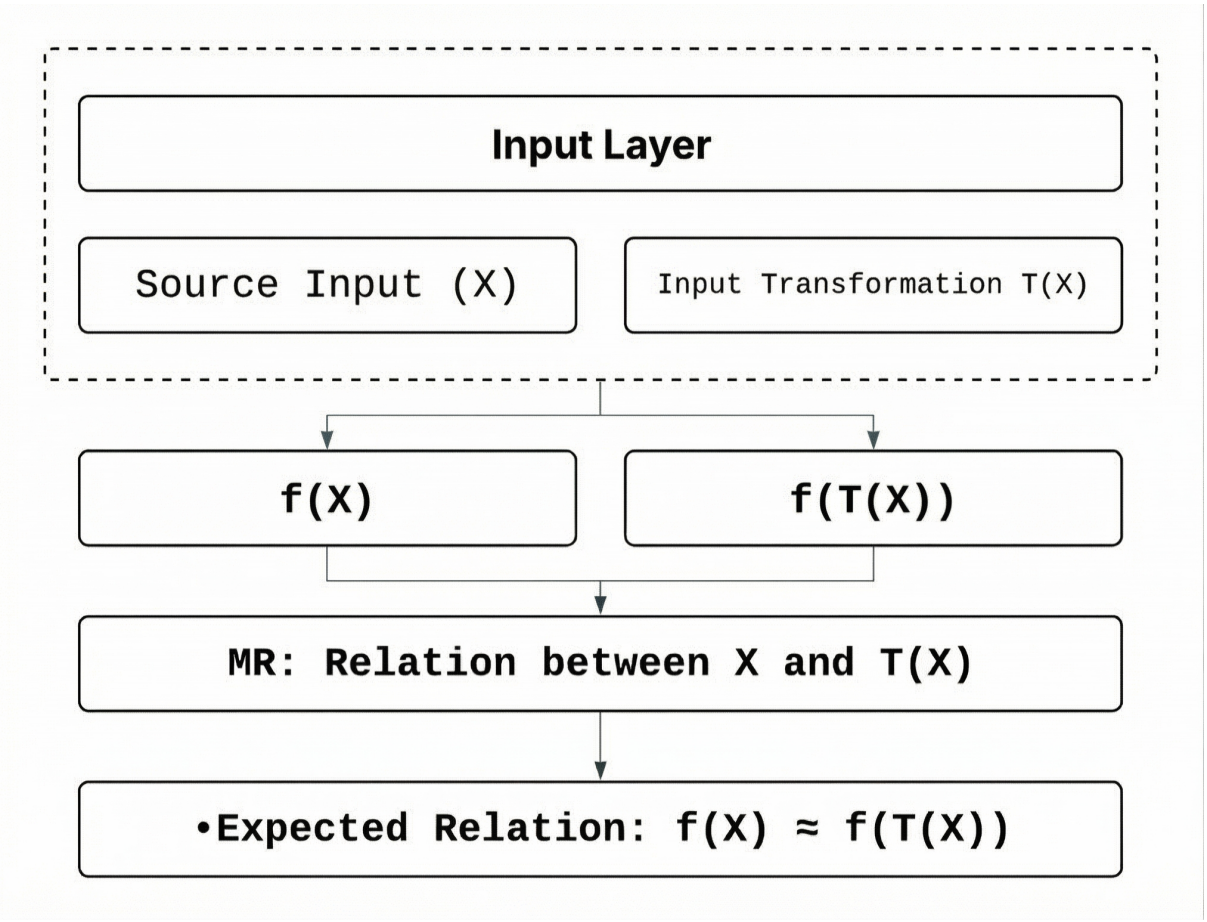


Figure 3.2: Structure of the generalized metamorphic testing template illustrating source inputs, transformations, and relational output constraints.

As shown in Figure 3.2, the generalized metamorphic testing template defines the relationship between source inputs, transformed follow up inputs, and expected relational

constraints on regression outputs. This abstraction enables the framework to remain independent of specific regression algorithms. Supporting diverse model types and autonomous driving applications.

## 3.2 Design of a Generalized Metamorphic Testing Template

Based on the insights obtained from the regression model analysis, a generalized mathematical and conceptual metamorphic testing template is designed. The template formally defines:

- The structure of source and follow up test inputs
- A standardized representation of metamorphic relations
- Expected relational constraints between regression outputs

The proposed template is independent of any specific regression algorithm. Instead, it allows instantiation for different regression models by specifying appropriate metamorphic relations and input transformation rules, thereby enabling reuse across multiple model types and applications.

## 3.3 Proof of Concept: Demonstration on a Representative Regression Model

The suggested metamorphic testing template is created and used to a typical regression model used in autonomous driving as a proof of concept. To illustrate viability and accuracy of method, a rudimentary regression based model such as steering angle prediction or trajectory estimate model is chosen.

In this proof of concept study, simulated driving scenarios are used to build a collection of source test inputs. Well defined metamorphic relations, such as geometric transformations (e.g., translation or rotation of trajectories), symmetry relations, and invariance properties, are then used to construct subsequent test cases. Both source and follow up inputs are used to run the regression model, and the results are compared against the expected metamorphic constraints.

Observed violations of metamorphic relations indicate potential robustness or correctness issues in the regression model. satisfied relations provide increased confidence in model behavior. This proof of concept implementation demonstrates that the generalized metamorphic testing template can be practically applied to real regression models without

requiring model specific test oracles, thereby validating the core idea of the proposed framework.

## 3.4 Validation Using Real World Autonomous Driving Regression Models

Following the proof of concept demonstration, real world regression models used in autonomous driving tasks. Lane keeping, trajectory prediction, steering angle estimation, and vehicle control are used to further validate the suggested metamorphic testing template. Based on specified relations, metamorphic test cases are automatically created. The outputs are then examined to find any deviations from the expected metamorphic qualities. This validation stage shows how the methodology may evaluate regression models' accuracy and resilience under realistic and methodically altered driving conditions.

## 3.5 Implementation of a Python Based Metamorphic Testing Framework

The generalized metamorphic testing template is developed as a modular and expandable Python based framework following validation. The implementation offers:

- Application Programming Interfaces (APIs) for integrating arbitrary regression models
- Configurable definitions of metamorphic relations
- Automated test case generation and execution
- Output comparison and metamorphic violation reporting

This framework greatly reduces manual labor and increases repeatability by allowing researchers and practitioners to apply metamorphic testing to regression models without creating ad hoc testing scripts.

## 3.6 System Architecture of the Proposed Framework

The overall system architecture is designed in a modular manner and consists of the following core components:

- Regression Model Interface
- Metamorphic Relation Engine
- Test Case Generator

- Execution and Evaluation Module
- Reporting and Visualization Module

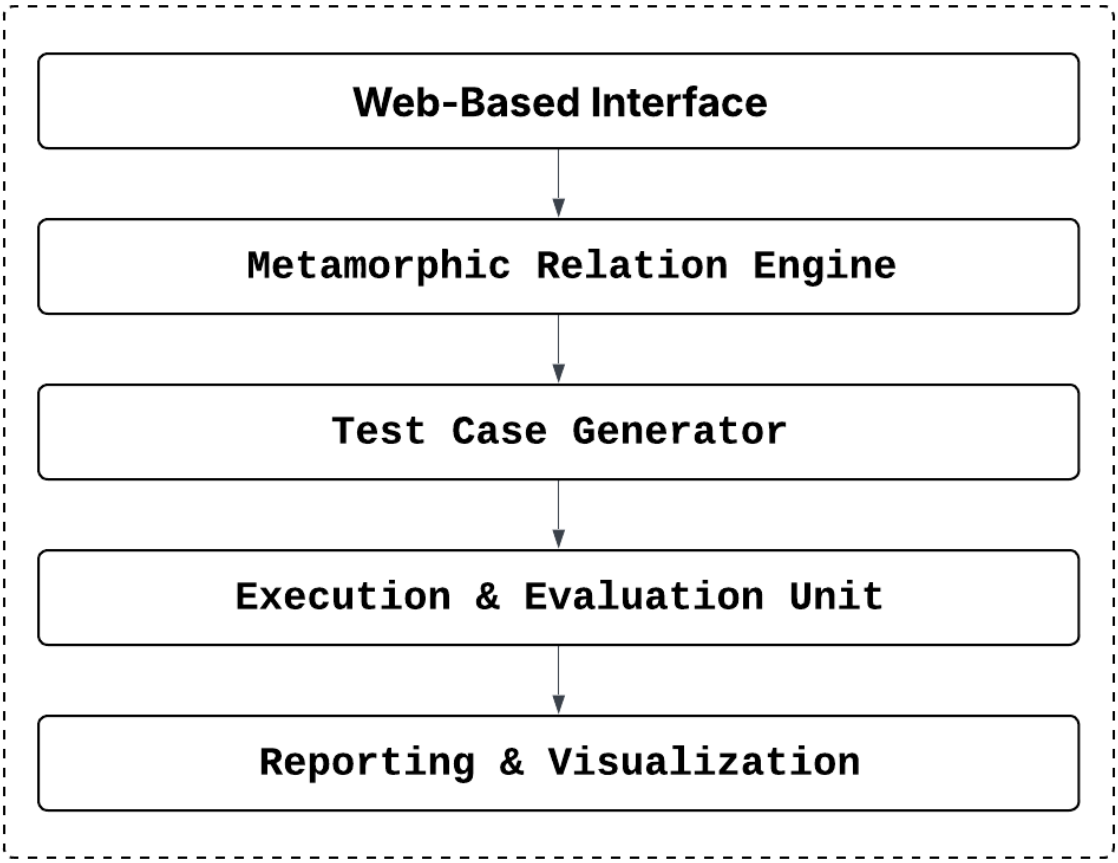


Figure 3.3: System architecture of the proposed Python based metamorphic testing framework for autonomous driving regression models.

The system architecture of the suggested Python based metamorphic testing framework is shown in Figure 3.3. Scalability, reusability, and ease of expansion are ensured by, the architecture's modular components. Which include the regression model interface, metamorphic relation engine, test case generator, execution and evaluation module, and reporting and visualization module.

This architecture ensures scalability, reusability, and ease of extension to new regression models and autonomous driving scenarios.

## 3.7 Web Based Platform for Accessibility and Public Use

To promote usability and broader adoption, on top of the Python frameworka web based interface is developed. The platform enables users to:

- Upload or connect regression models
- Select predefined or custom metamorphic relations
- Execute metamorphic tests through a graphical interface
- Visualize testing results and detected violations

# Chapter 4

# Timeline and Resource Required

## 4.1 Timeline

The project is split into 18 milestones, each showing a planned task, to make sure it completed effectively within 28 weeks (two semesters).

1. Identification of Metamorphic Properties and Relations

   - Identify suitable metamorphic properties for regression models.
   - Define metamorphic relations based on the expected outputs and how the inputs are changed.
   - Check the identified relations using theoretical analysis.

2. Design of Generalized Metamorphic Testing Template

   - Create a universal, reusable template for metamorphic testing.
   - Explain how to create, run, and check tests.
   - Make sure the template can be easily adapted for other regression models.

3. Proof of Concept (PoC) Development

   - Build a proof of concept to show that the proposed approach works.
   - Test important metamorphic relations using sample regression models.
   - Review the results and make any needed improvements to the design.

4. Development of Python Based Metamorphic Testing Framework

   - Build the main Python framework for metamorphic testing.

- Create modules to automatically generate and run test cases.
- Add metamorphic relations into the framework.

5. Experimental Evaluation
   - Set up tests to check the framework's efficacy.
   - Execute experiments using regression models.
   - Analyze the results to see how effectively faults are detected.

6. Web Based Platform Development
   - Design and implement a web based platform for the testing framework.
   - Allow users to run tests and view results.
   - Connect the web platform with the backend framework.

7. Documentation and Final Writing
   - Prepare detailed documentation of the system design and implementation.
   - Write the chapters of the final project report.
   - Make sure the report meets the university's formatting requirements.

8. Final Demonstration
   - Demonstrate the complete metamorphic testing system.
   - Highlight the features of the web platform and key experimental results.
   - Wrap up the project with a final evaluation and submission.

Table 4.1: Project Timeline and Milestones

| PROJECT ACTIVITY | DURATION | PROJECT WEEK | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| Identification of metamorphic properties & relations | 3 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Design of generalized metamorphic testing template | 3 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proof of Concept (PoC) | 4 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python-based MT framework | 6 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Experimental evaluation | 4 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Web-based platform | 4 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Documentation & final writing | 3 weeks | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Final Demonstration | 1 week | | | | | | | | | | | | | | | | | | | | | | | | | | | |

## 4.2 Resource Required

Table 4.2: Required Resources

| Category | Requirement Item | Purpose / Description | Availability / Cost |
|---|---|---|---|
| Computing Resources | High-Performance Computing System | Used for implementing regression models, processing datasets, executing metamorphic tests, and analyzing results. Required performance includes a multi-core CPU with >200 GFLOPS compute capability, minimum 16 GB RAM, and a dedicated CUDA-enabled (or equivalent) GPU for accelerated computation. | Available through university laboratory systems or existing institutional resources |
| Software Platform | Operating System | Required to run machine learning, data processing, and simulation tools. | Linux or Windows (available in university labs) |
| Programming Environment | Python Programming Language | Core language used for model implementation, data preprocessing, metamorphic testing, and evaluation. | Free and open-source |
| Machine Learning Libraries | NumPy, SciPy, Scikit-learn | Used for numerical computation, regression model development (GPR/SGPR), and mathematical transformations. | Free and open-source |
| Visualization Software | Matplotlib, Seaborn | Used for software-level visualization of trajectories, lane positions, and comparison of original versus transformed outputs. | Free and open-source |
| Simulation / Demonstration Environment | Data-Level Simulation Environment | Used to demonstrate results by applying controlled transformations (e.g., rotation, translation, symmetry) to autonomous driving datasets instead of full physics-based simulation, enabling validation of metamorphic relations. | Implemented using open-source Python libraries (Free) |
| Real-World Demonstration Facility | Faculty Autonomous Robot Car Test Bed | Used for limited real-world visualization and demonstration of selected results in a controlled faculty environment. | Available within the faculty (No additional cost) |
| Datasets | Public Autonomous Driving Datasets | Used for training and testing regression-based models such as trajectory prediction and steering control. | Publicly available datasets (Free) |

# Chapter 5

# Conclusion

As autonomous driving systems become more popular, there is an increasing need for reliable and systematic testing methods to ensure that these models behave safely and consistently in real-world situations. Machine learning models used in self-driving cars, like those for predicting paths, steering, and staying in lanes, are hard to test because their outputs keep changing and there is no reliable way to check them. Traditional testing methods often fail to capture the complexity and variety of real-world driving situations, which makes it harder to trust the models when they are deployed.

This study proposes a general metamorphic testing platform designed specifically for regression models used in autonomous driving systems. Instead of checking for exact expected results, this method uses metamorphic testing to see if the outputs of several runs make sense together, helping to solve the oracle problem. This testing framework works with any model and allows different autonomous driving systems to be tested in an organized, reusable, and scalable way.

The Python framework and web interface make metamorphic testing easier and less manual. Tests with real autonomous driving models show it can catch issues that traditional methods might miss.

Overall, this work makes autonomous driving models more reliable and safer. The proposed testing platform connects model development with systematic validation and supports future research in testing AI-driven systems.

# References


[1] H. Liu, F.-C. Kuo, D. Towey, and T. Y. Chen, "How effectively does metamorphic testing alleviate the oracle problem?" *IEEE Transactions on Software Engineering*, vol. 40, no. 1, pp. 4–22, 2014.

[2] SAE International. (2021) Sae j3016 levels of driving automation visual chart. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[3] E. T. Barr *et al.*, "The oracle problem in software testing: A survey," *IEEE Transactions on Software Engineering*, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/6963470

[4] H. Liu, K. Chen, and J. Ma, "Incremental learning-based real-time trajectory prediction for autonomous driving via sparse gaussian process regression," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[5] QA Source. (2020) Metamorphic testing concept diagram. [Online]. Available: https://blog.qasource.com/what-is-metamorphic-testing

[6] MathWorks. (2024) Lane keeping assist with lane detection system diagram. [Online]. Available: https://www.mathworks.com/help/driving/ug/lane-keeping-assist-with-lane-detection.html

[7] World Economic Forum, "Autonomous vehicles: Timeline and roadmap ahead," 2025.

[8] X. Zhang *et al.*, "Machine learning testing: Survey, landscapes, and horizons," *IEEE Transactions on Software Engineering*, 2020.

[9] Z. Zhou *et al.*, "Metamorphic testing for autonomous driving systems," *IEEE Transactions on Software Engineering*, 2019 - 2021.

[10] E. T. Barr *et al.*, "The oracle problem in software testing," *IEEE Transactions on Software Engineering*, 2015.

[11] T. Y. Chen, "Metamorphic testing: A new approach for generating next test cases," in *Technical Report*, 1998.

[12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the DARPA grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[13] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[14] *ISO 26262: Road Vehicles – Functional Safety*, International Organization for Standardization Std., 2018.

[15] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2017.

[16] NVIDIA, "Simulation-based autonomous vehicle testing," NVIDIA Corporation, White Paper, 2022.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[18] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2006.

[19] D. Nguyen-Tien, L. L. Pape, and R. Siegwart, "Gaussian process-based model predictive control for autonomous driving," *IEEE Control Systems Letters*, 2020.

[20] M. Liu, Y. Zeng, Y. Wang, and R. Urtasun, "Equivariant trajectory prediction for autonomous vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[21] K. Pei, Y. Cao, J. Yang, and S. Jana, "DeepXplore: Automated white-box testing of deep learning systems," in *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, 2017.

[22] Y. Zhou, X. Zhang, and T. Y. Chen, "Metamorphic testing of machine learning programs," in *Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE)*, 2018.

[23] X. Murphy *et al.*, "Testing and validation of AI-based systems: A survey," *IEEE Access*, 2021.

[24] M. Tian *et al.*, "Metamorphic testing of autonomous systems: Principles and applications," *IEEE Access*, 2022.

[25] *Taxonomy and Definitions for Terms Related to Driving Automation Systems*, SAE International Std. SAE J3016, 2021.