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

In order to carry out crucial functions including trajectory prediction, lane maintenance, steering control, and vehicle motion planning, autonomous driving systems are depending more and more on machine learning based regression models. Even while these models have made great strides, it is still very difficult to guarantee their dependability and safety in actual driving situations due to the test oracle problem. A test oracle is defined as a mechanism capable of systematically verifying the correctness of a test result for any given input; however, a fundamental challenge arises when such an oracle either does not exist or is too expensive to be used. This problem presents a significant hurdle because it creates a verification difficulty, where it is impossible to clearly determine whether the continuous valued output of a program has passed or failed for a specific input. Furthermore, this leads to methodological restrictions, as the lack of an oracle limits the effectiveness of traditional test case selection strategies that rely on known outcomes. Consequently, while traditional testing is inadequate for ensuring reliable behavior in unexpected circumstances, alternative approaches like metamorphic testing seek to alleviate this problem by using metamorphic relations properties represented as relations among inputs and outputs of multiple executions to verify program correctness instead of relying on a single, predetermined outcome [**?**].

By validating software behavior through metamorphic relations that specify expected interactions between many system executions under deliberately altered inputs, Metamorphic Testing (MT) has emerged as a promising solution to the oracle problem. The lack of a universal, reusable, and scalable testing framework has prevented MT from being widely used in autonomous driving regression models, despite its successful application in other domains.

Regression models used in autonomous driving systems are the focus of this project's design and implementation of a generic metamorphic testing platform. The suggested method allows for the systematic creation and assessment of test cases without the need for explicit test oracles by introducing a model agnostic metamorphic testing template that abstracts common metamorphic features. Regression models for real world autonomous driving and proof of concept tests are used to validate the framework, which is constructed as a modular Python based system. A web based interface is also created to improve usability and accessibility for practitioners and researchers.
The suggested platform seeks to enhance the robustness, dependability, and real world preparedness of autonomous driving regression models by decreasing manual testing effort and expanding test coverage. This will help make the deployment of autonomous 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

List of Figures

List of Tables

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 [**?**]. 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 [**?**] [**?**].

1.1 Background

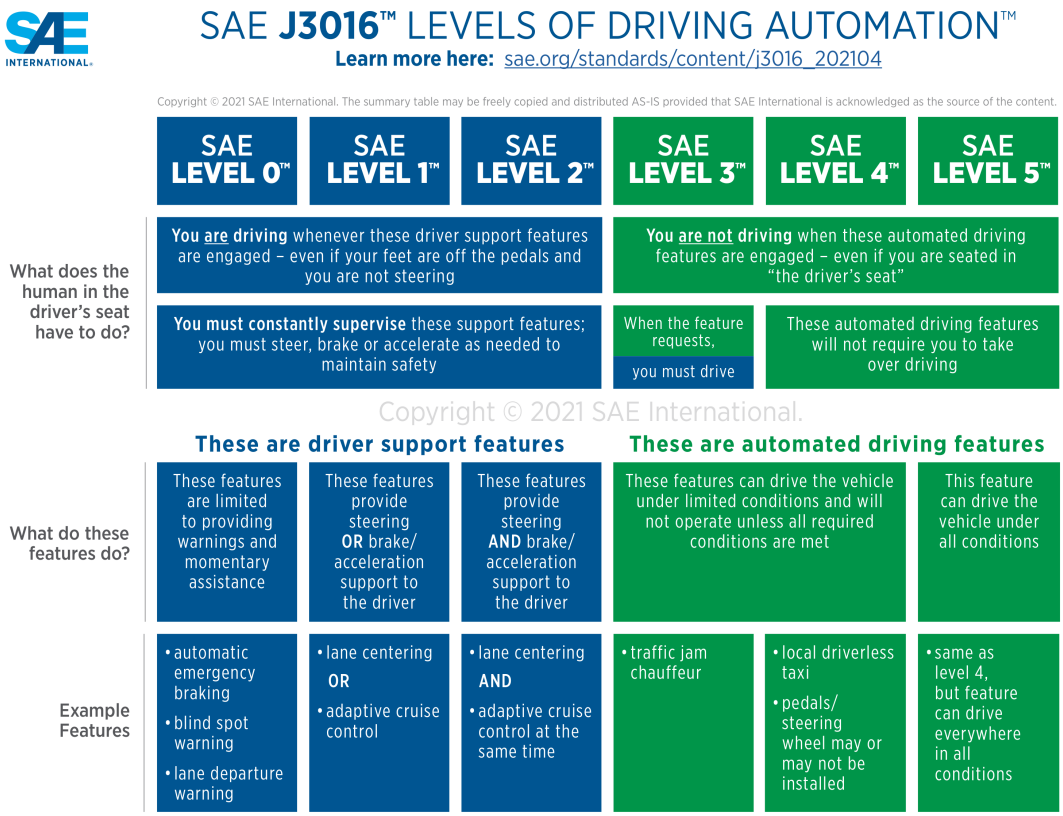

Figure 1.1: SAE International Levels of Driving Automation (J3016 standard), illustrating the progression from no automation (Level 0) to full autonomy (Level 5) [?].

The SAE J3016 standard specifies six levels of driving automation, from no automation (Level 0) to complete automation (Level 5), as seen in Figure 1.1. Currently, the majority of commercially deployed systems function at Levels 2 and 2+, requiring constant human supervision and only limited automation. This highlights how crucial sound testing procedures are to guaranteeing dependability and safety in practical implementation.

Insufficient and unrealistic testing is a significant barrier limiting the robustness of autonomous driving models. Autonomous systems function in extremely dynamic, complicated contexts where it is not feasible to list and verify every scenario before to deployment. For machine learning based systems that provide continuous outputs like trajectories, steering angles, or acceleration values, this problem is commonly known as the oracle problem, where it is challenging or impossible to discern the proper output for a given test input. As previously mentioned, the lack of trustworthy test oracles is a common problem for machine learning applications, such as autonomous driving and advanced driver assistance systems (ADAS) [?], [?].

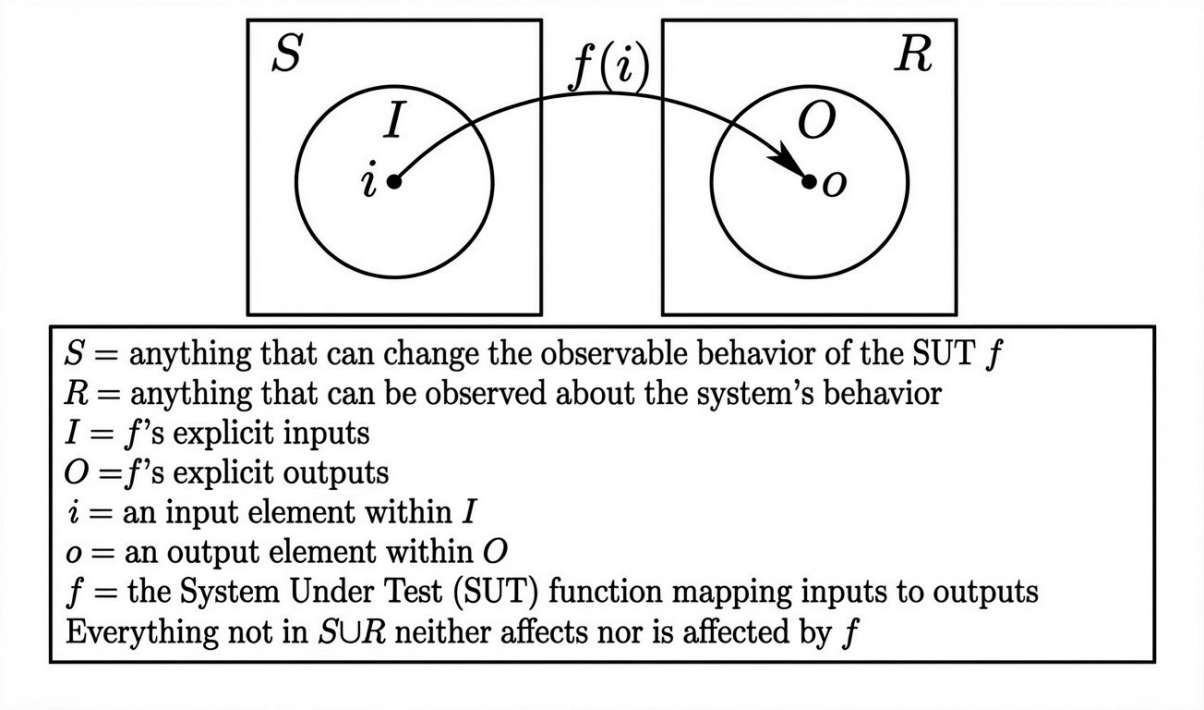

Figure 1.2: Illustration of the test oracle problem in software testing, highlighting the challenge of determining correct outputs without explicit expected results [**?**].

Figure **??** illustrates the relationship between stimuli and observations in the context of software testing. Stimuli (S) represent all factors that can influence the behavior of the system under test (SUT), including explicit test inputs provided by the tester, environmental conditions, configuration parameters, and sensor inputs. A subset of these stimuli, denoted as $I \subset S$, corresponds to the explicit inputs directly applied to the system. Observations (R) represent all aspects of the system's behavior that can be measured after stimulation, including explicit outputs ($O \subset R$) as well as non functional properties such as 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 [**?**].

Testing fundamentally consists of applying stimuli to the system and observing the resulting behavior. However, as depicted in the figure, this process alone does not guarantee the availability of a clear mechanism for determining whether the observed behavior 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 intended specification. Formally, a test oracle can be expressed as a function

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

where E denotes the set of all possible executions of the system under test [**?**].

The oracle problem arises when such a function cannot be reliably defined, implemented, or evaluated for all executions. Barr *et al.* emphasize that this limitation is particularly severe in modern software systems characterized by non determinism, high complexity,

and data driven behavior, where determining correct outputs in advance is impractical or impossible [**?**]. This issue is especially prominent in machine learning based autonomous driving systems, which produce continuous outputs that depend on complex and dynamic environmental stimuli. Consequently, the absence of explicit expected results motivates the adoption of alternative testing strategies, such as metamorphic testing, that reduce or eliminate dependence on traditional test oracles [**?**, **?**, **?**].

Chen first proposed Metamorphic Testing (MT) in 1998, and it offers a practical solution to the oracle problem. MT determines whether a system satisfies required qualities, called metamorphic relations (MRs), spanning several executions with related inputs rather than requiring explicit expected outcomes. Even when the correct output is unknown, errors can be identified if these relationships are broken. Numerous fields, such as navigation software, search engines, cybersecurity, machine learning systems, and autonomous driving applications, have effectively used MT [**?**], [**?**]. Geometric transformation based metamorphic relations have proven particularly successful in exposing discrepancies under perspective, spatial, or environmental alterations in the context of autonomous systems.

1.2 Problem Statement

However, using metamorphic testing in practice is still difficult despite its benefits. For each model and situation, current methods usually require developers to manually create metamorphic relations and apply unique test scripts. Due to irregular test design or inadequate domain coverage, this method is laborious, prone to errors, and may still overlook important edge situations. Because of this, metamorphic testing is currently neither readily available or methodically incorporated into the process of developing autonomous driving regression models.

1.3 Objectives and Scope

1.3.1 Project Goal

The primary goal of this project is to design and implement a generalized metamorphic testing environment for autonomous driving regression models. The proposed platform aims to simplify the definition, execution, and evaluation of metamorphic relations, enabling effective testing of both newly developed and existing autonomous driving models.

1.3.2 Objectives

To achieve the above project goal, the following objectives are defined. Each objective is accompanied by clear verification metrics to assess successful completion.

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

This objective focuses on defining meaningful metamorphic relations (e.g., rotation, translation, symmetry, and invariance) that reflect real world driving properties.

Verification metrics:

- Definition of at least six metamorphic relations applicable to autonomous driving.
- Formal specification of input output constraints for 100% of the defined relations.

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

This objective ensures that the testing approach can be reused across different regression models without requiring model specific test oracles.

Verification metrics:

- Successful instantiation of the template for at least three different regression models.
- No structural modification required when applying the template across models.

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

This objective demonstrates the practical applicability and effectiveness of the framework 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 ensures that the framework can be used by practitioners without deep 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 interaction.

1.3.3 Scope of the Project

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

- The project focuses exclusively on regression based models used in autonomous driving tasks such as lane keeping, trajectory prediction, and steering angle estimation.
- Testing is conducted using metamorphic testing techniques to address the oracle problem in machine learning based systems.
- The proposed framework is model agnostic and operates at the software testing level; it does not replace existing simulation platforms or vehicle control systems.
- Validation is performed using simulated driving scenarios and software level testing rather than real world vehicle deployment.
- Classification models, sensor hardware validation, perception pipelines, and low level vehicle actuation control are explicitly outside the scope of this study.

Chapter 2

Literature Review

2.1 Background of the Autonomous Driving Systems and Validation

Autonomous driving systems are safety critical cyber physical systems that integrate perception, decision making, and control components to enable vehicles to operate with limited or no human intervention [?] [?]. These systems continuously process data from multiple onboard sensors, including cameras, LiDAR, radar, and Global Positioning System (GPS) units, to perceive the surrounding environment and understand road geometry, traffic participants, and dynamic obstacles. Based on this perception, control actions such as steering, acceleration, braking, and lane keeping are generated to ensure safe vehicle operation.

The operational environment of autonomous vehicles is inherently dynamic and uncertain. Factors such as varying road conditions, complex traffic interactions, weather changes, sensor noise, and unpredictable human behavior significantly increase system complexity [?] [?]. As a result, ensuring the reliability, robustness, and safety of autonomous driving software is a critical requirement throughout the development lifecycle. Failures or unexpected behaviors in these systems may lead to severe safety consequences, highlighting the importance of rigorous testing and validation methodologies [?].

Testing and validation play a central role in the development of autonomous driving systems. Conventional validation approaches typically rely on predefined datasets, simulation based testing, and scenario based evaluations using known traffic situations and standardized benchmarks [?] [?]. While these methods enable controlled experimentation and functional verification, their effectiveness becomes limited when applied to machine learning based systems, particularly those producing continuous outputs and exhibiting non deterministic behavior.

Many core components of autonomous driving systems, especially regression based machine learning models, generate continuous outputs such as steering angles, vehicle trajectories, velocity profiles, and control commands [?]. [?]. In real world driving scenarios, determining the correct output for a given input in advance is often infeasible. This leads to the well known test oracle problem, where the absence of explicit expected outputs prevents reliable automated validation [?] [?]. Additionally, predefined datasets and scenario based testing fail to comprehensively capture rare edge cases and unseen driving conditions, making it difficult to guarantee robust real world performance.

As autonomous driving systems increasingly rely on machine learning for perception and control, alternative validation techniques that do not depend on explicit expected outputs have gained increasing attention. This has motivated the exploration of metamorphic testing, which evaluates system correctness by verifying expected relationships between multiple executions rather than individual outputs [?] [?].

2.2 Regression Based Models in Autonomous Driving Systems

For autonomous driving systems to operate safely and effectively, key functionalities such as trajectory prediction, lane centerline tracking, steering control, and motion planning must be performed reliably. Although deep neural network based methods have achieved strong performance in perception tasks, regression based models remain essential for control oriented and prediction based applications due to their interpretability, uncertainty estimation capabilities, and robustness when handling structured continuous data [?] [?].

Among regression approaches, Gaussian Process Regression (GPR) has been widely adopted in autonomous driving for trajectory prediction and vehicle behavior modeling. GPR provides probabilistic predictions with uncertainty estimates, which are crucial for safety aware decision making in dynamic driving environments. However, traditional GPR suffers from high computational complexity, limiting its scalability in real time systems [?].

To address this limitation, Sparse Gaussian Process Regression (SGPR) has been proposed as a scalable alternative that maintains prediction accuracy while enabling real time inference [?]. SGPR based models have been successfully integrated into Model Predictive Control (MPC) frameworks for autonomous driving tasks such as lane keeping and overtaking maneuvers. Experimental studies demonstrate improved trajectory prediction accuracy and control stability across diverse traffic conditions [?] [?].

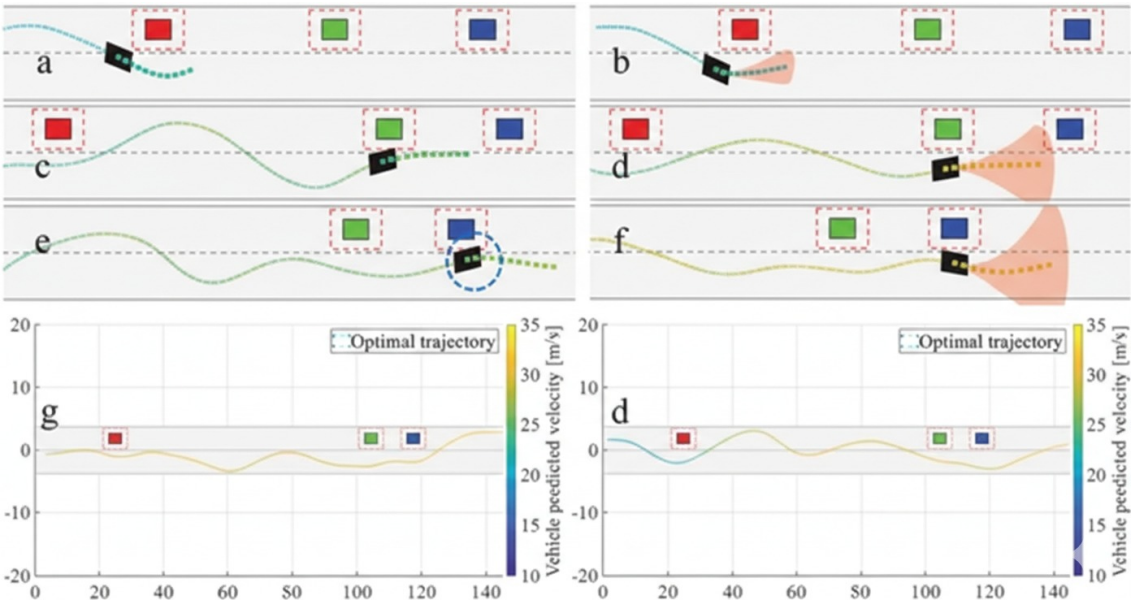

Figure 2.1: Example of Gaussian Process based trajectory prediction in autonomous driving, showing predicted trajectories with uncertainty bounds for overtaking maneuvers [?].

An example of Gaussian Process based trajectory prediction in an autonomous driving setting is shown in Figure 2.1, which shows expected vehicle trajectories and related uncertainty bounds. Because they can give uncertainty estimates, which are essential for making safety aware decisions in dynamic driving conditions, these regression based models are very useful.

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

2.3 Limitations of Conventional Testing Approaches in Autonomous Driving

Despite advances in modeling techniques, the testing and validation of autonomous driving systems remain a major challenge. Conventional testing approaches are often limited to predefined datasets, simulation based environments, and manually designed scenarios [?] [?]. While these methods provide controlled and repeatable evaluations, they struggle to represent the full complexity of real world driving conditions.
Autonomous vehicles operate in highly dynamic environments involving unpredictable agents, rare corner cases, and continuously evolving traffic situations [?] [?]. As a result, models that perform well in controlled testing environments may exhibit unexpected or unsafe behavior when deployed in novel conditions. This limitation is particularly severe for regression based models with continuous outputs, where defining correct expected

results for all possible inputs is impractical.

Consequently, there is a growing need for systematic testing approaches that can evaluate model robustness beyond predefined datasets and scenarios, without requiring explicit output oracles [?] [?].

2.4 Metamorphic Testing: Previous Work

Metamorphic testing (MT) was first introduced by Chen in 1998 as a software testing technique designed to address the oracle problem [?]. Instead of relying on predefined expected outputs, MT verifies program correctness by examining relationships between multiple executions under systematically transformed inputs.

These expected relationships, known as metamorphic relations (MRs), represent fundamental properties that the system under test should satisfy. If a metamorphic relation is violated, a fault is detected even when 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 [?].

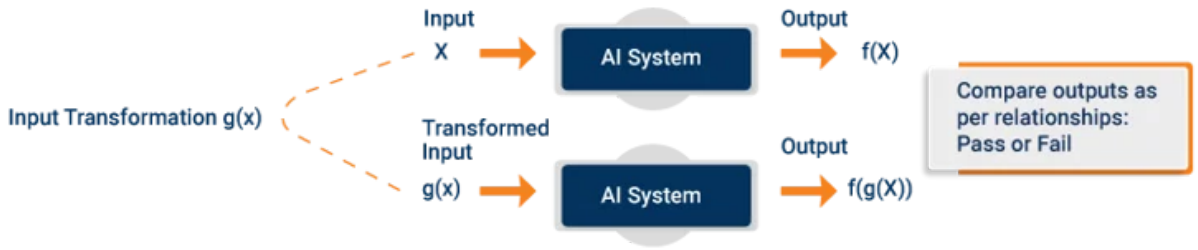

Figure 2.2: Conceptual overview of metamorphic testing, illustrating the generation of follow up test cases from source test cases using metamorphic relations to validate system behavior without an explicit test oracle [?].

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

2.5 Metamorphic Testing in Machine Learning and Autonomous Driving Systems

In recent years, metamorphic testing has gained increasing attention in the validation of machine learning based systems, including autonomous driving applications [?] [?]. Due to 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 [?] [?].

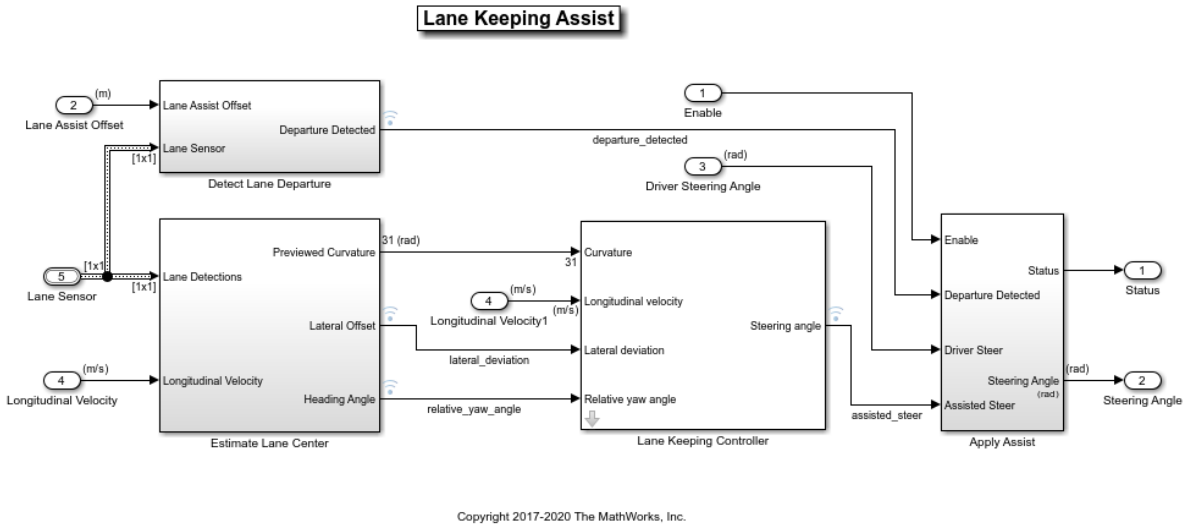

Figure 2.3: Diagram of a Lane Keeping Assist System (LKAS) in autonomous driving, illustrating lane detection, departure warning, and corrective steering [?].

A Lane Keeping Assist System (LKAS) usually includes of lane detection, vehicle position assessment, and corrective steering control components, as shown in Figure 2.3. Because 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 choice for metamorphic testing.

Researchers have demonstrated that symmetry and rotation based MRs can reveal critical system faults that are often missed by conventional testing. By generating large numbers of follow up test cases from existing scenarios, MT enables systematic exploration of model behavior without relying on explicit expected outputs [?] [?].

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 perception and control models [?] [?].

2.6 Gap Identification: Limitations of Existing Metamorphic Testing Approaches

Despite its effectiveness, the practical adoption of metamorphic testing in autonomous driving remains limited. Most existing MT implementations rely on ad hoc test scripts, where metamorphic relations are manually defined and tightly coupled to specific models, simulators, or experimental setups [**?**] [**?**].

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

Furthermore, current MT approaches typically require extensive domain knowledge and familiarity with simulation platforms, increasing development and maintenance overhead. These challenges are particularly pronounced for regression based autonomous driving models, where continuous outputs and complex interactions complicate MR design [**?**].

2.7 Summary of Existing Systems and Research Gaps and way forward

Existing autonomous driving systems widely employ regression based models such as GPR, SGPR, and learning based control models for trajectory prediction, lane keeping, and steering control [**?**] [**?**] [**?**]. While these models offer strong predictive capabilities and uncertainty estimation, their validation remains challenging due to the oracle problem and the limitations of conventional testing approaches [**?**] [**?**].

Although metamorphic testing has shown promise in addressing these challenges, current solutions lack abstraction, automation, and scalability. This clearly motivates the need for a generalized, reusable, and model agnostic metamorphic testing framework for autonomous driving regression models, which forms the 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.

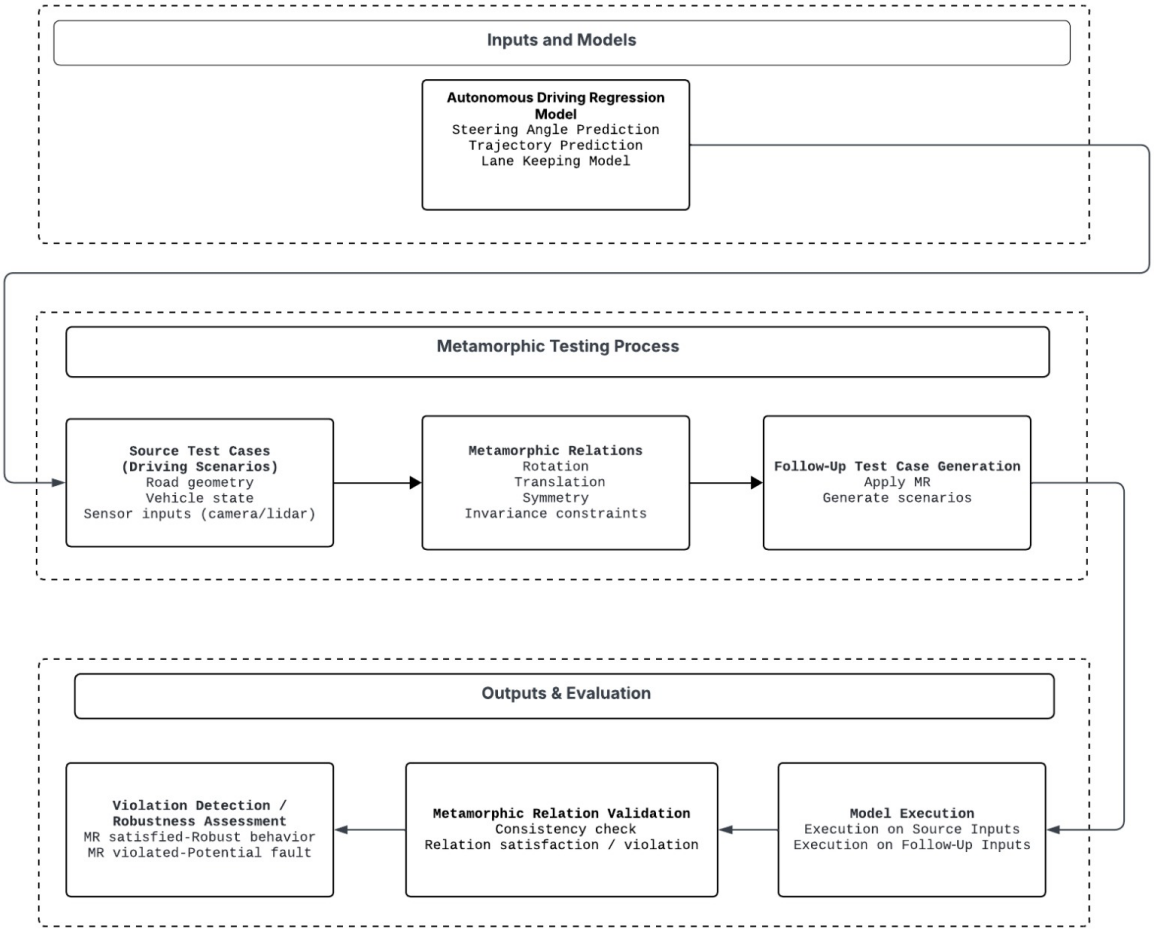

Figure 3.1: Overall workflow of the proposed metamorphic testing framework for autonomous driving regression models.

An overview of the suggested metamorphic testing framework workflow is given in Figure 3.1, which shows the steps from regression model analysis to the 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 include, but are not limited to, 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 that the proposed framework remains model agnostic and is capable of supporting diverse regression behaviors and output characteristics typical of autonomous driving systems.

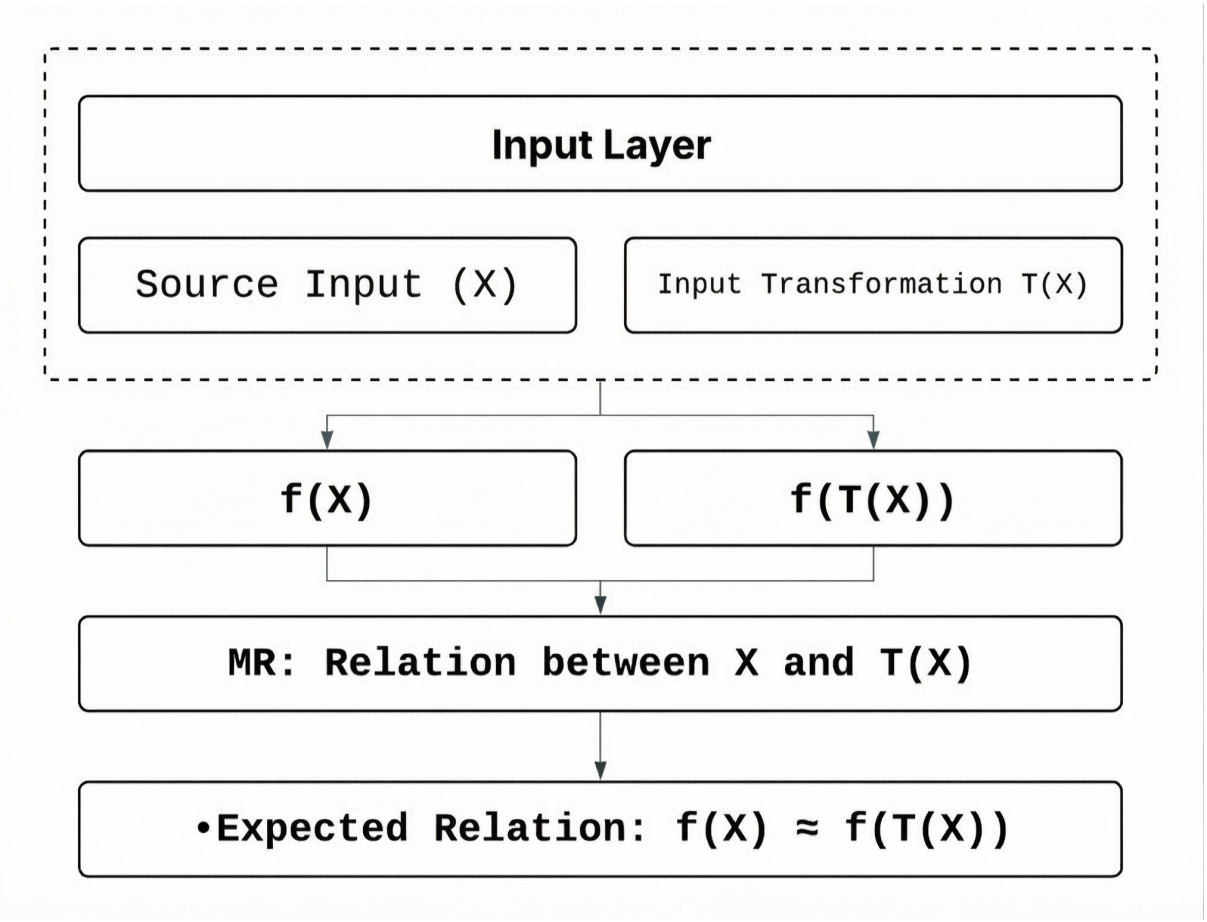

Figure 3.2: Structure of the generalized metamorphic testing template illustrating source inputs, transformations, and relational output constraints.

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 while 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 the viability and accuracy of the method, a rudimentary regression based model such as a 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, while 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 like lane keeping, trajectory prediction, steering angle estimation, and vehicle control are used to further validate the suggested metamorphic testing template. Based on the 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

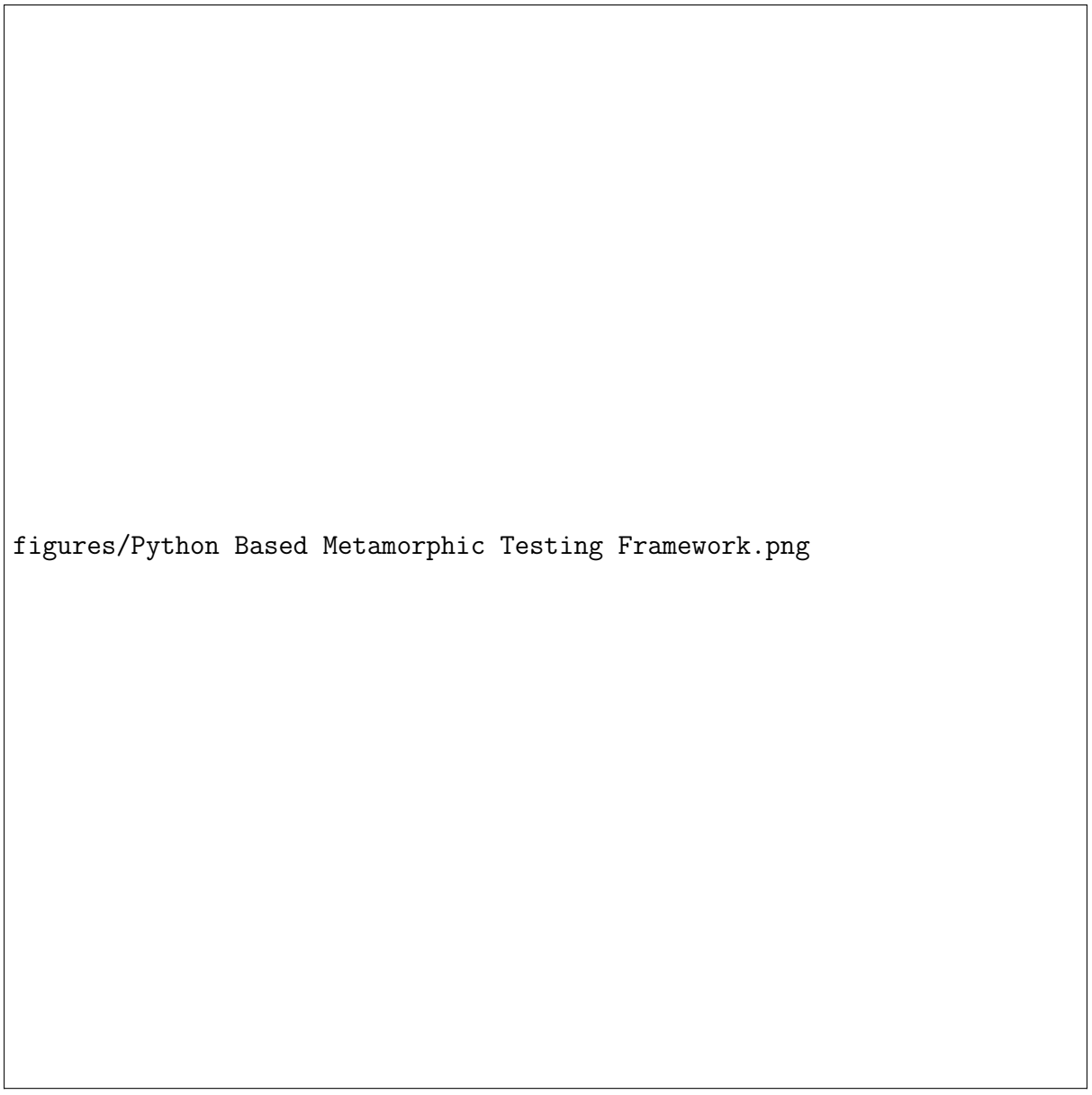

Figure 3.3: System architecture of the proposed Python based metamorphic testing framework for autonomous driving regression models.

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, a web based interface is developed on top of the Python framework. 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 has been divided into 18 sequential milestones, each of which corresponds to the planned project activities, in order to guarantee its effective completion within the allotted time period of 28 weeks (two semesters).

1. Identification of Metamorphic Properties and Relations
   - Identify suitable metamorphic properties for regression models.
   - Based on expected outputs and input transformations, define metamorphic relations.
   - Validate identified relations through theoretical analysis.

2. Design of Generalized Metamorphic Testing Template
   - Create a template for universal and reusable metamorphic testing.
   - Describe the processes for creating, executing, and validating tests.
   - Make sure it can be expanded to accommodate other regression models.

3. Proof of Concept (PoC) Development
   - Conduct a proof of concept to show that the suggested strategy is feasible.
   - Use sample regression models to validate important metamorphic linkages.
   - Examine PoC results and make any necessary design adjustments.

4. Development of Python Based Metamorphic Testing Framework
   - Develop the core Python based metamorphic testing framework.

- Implement modules for the creation and execution of automated test cases.
- Integrate metamorphic relations into the framework.

5. Experimental Evaluation
   - Create test configurations to assess the framework's efficacy.
   - Execute experiments using regression models.
   - Evaluate the performance and capability of fault detection by analyzing the data.

6. Web Based Platform Development
   - Design and implement a web based platform for the testing framework.
   - Enable user interaction for test execution and result visualization.
   - Integrate the web platform with the backend testing framework.

7. Documentation and Final Writing
   - Prepare comprehensive documentation of system design and implementation.
   - Write chapters of the final project report.
   - Verify adherence to the formatting requirements of the university.

8. Final Demonstration
   - Demonstrate the complete metamorphic testing system.
   - Display the characteristics of the web based platform and experimental findings.
   - Conclude the project with 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

The growing popularity of autonomous driving systems has brought attention to the urgent need for dependable and methodical testing techniques that can guarantee robust and safe model behavior in a variety of real world scenarios. Due to their continuous outputs and lack of trustworthy test oracles, regression based machine learning models which are frequently employed in autonomous driving for tasks like trajectory prediction, steering control, and lane keeping present particular testing issues. The complexity and diversity of real world driving situations are frequently not well captured by conventional testing methods, which reduces confidence in model deployment.

A generalized metamorphic testing platform created especially for regression models in autonomous driving systems is proposed in this study. Instead of depending on explicit expected outcomes, the suggested method uses metamorphic testing concepts to validate relational coherence across successive executions, so addressing the Oracle problem. Systematic, reusable, and scalable testing across a variety of regression models and autonomous driving applications is made possible by the development of a model agnostic metamorphic testing framework.

The manual effort needed to define and run metamorphic tests is greatly decreased by the use of a modular Python based framework and an intuitive web based interface. The suggested platform shows its efficacy in detecting any robustness and consistency problems that might be missed by traditional testing techniques through proof of concept demonstrations and validation using real world autonomous driving regression models.

All things considered, this work advances the robustness, dependability, and practicality of autonomous driving regression models. The suggested metamorphic testing platform facilitates the safer and more reliable deployment of autonomous driving technologies by bridging the gap between model development and systematic validation. It also offers a basis for further research and expansion in software testing for machine learning based cyber physical 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.