A Comparative Survey of Mutation Testing Across Large Language Models, REST APIs, and Database Engines

1. A concise description of the project

Software testing is a crucial phase in the software lifecycle process, ensuring a system functions correctly and satisfies requirements. It involves the generation and execution of test cases designed to detect potential bugs and verify the software's reliability. These test cases must be evaluated to determine how well they detect faults. One powerful technique for assessing test quality is mutation testing, which introduces small modifications to the software code to form so-called mutants. The goal is to determine whether the test cases can successfully identify these seeded faults, thereby measuring their strength and effectiveness.

This research aims to use mutation testing to conduct a comparative survey of three distinct software components: Large Language Models (LLMs), REST APIs, and Database Engines, assessing the effectiveness, challenges, and future prospects of this technique for each application. The study will explore how mutation testing is applied in these systems, by reviewing and comparing peer-reviewed studies that have used mutation testing in these contexts. The research will provide insights into how these systems handle faults, detect errors, and ensure reliability. Thus, identify key trends, strengths, and weaknesses in the application of mutation testing across the three domains. The results from this survey will contribute to understanding the comparative performance of these technologies in terms of their reliability, robustness, and ability to adapt to real-world error scenarios.

2. Measurable goals and objectives for the project

The primary objective of this research is to conduct a comparative survey of mutation testing technique applied to Large Language Models (LLMs), REST APIs, and Database Engines. This survey will systematically review how mutation testing is used in each system, analyzing the effectiveness, challenges, and future prospects of mutation testing. The purpose of this comparative analysis is to find important trends, patterns, and weaknesses in the way mutation testing is used in various systems.

The study will focus on evaluating how each system handles and responds to mutants by analyzing LLMs' ability to process faulty data, assessing REST APIs' behavior under mutated requests, and investigating how database engines execute and manage mutated queries. Through peer-reviewed studies, this research seeks to analyze the error-detection and fault-tolerance capabilities of each system. Thus, identify strengths, weaknesses, and common challenges faced by each component when subjected to mutation testing. Ultimately, the findings from this survey will contribute to advancements in software reliability and fault tolerance, offering valuable insights into the effectiveness of mutation testing as a tool for evaluating the robustness of different software components.

3. The timeframe for the project

This research project will be conducted over two months, June and July 2025. The first phase, in early June, will focus on gathering and reviewing peer-reviewed studies related to mutation testing in Large Language Models (LLMs), REST APIs, and Database Engines. During this phase, the research will also involve establishing a framework for categorizing and comparing the effectiveness, challenges, and future prospects of mutation testing in these domains.

By mid-June, Data collection and preliminary analysis, which includes a systematic review and analysis of previous studies will start. With an emphasis on the advantages, disadvantages, and error-handling techniques covered in the literature, I will shed light on how each system manages and identifies errors.

In July, the project will enter its comparative analysis phase, where the results from the reviewed studies will be analyzed. In this stage, the results will be combined to assess how well mutation testing methods work for each system and identify major issues. The results will be organized using a structured comparison of the systems' fault detection and robustness.
The final phase, set for late July, will focus on synthesizing findings, preparing a comprehensive report, and drafting a research paper for potential publication.

4. How the project will enhance teaching and research at Lincoln University

This project will significantly enhance both teaching and research at Lincoln University by incorporating state-of-art- software testing methodologies into the curriculum and encouraging a research-learning environment. The results of this study will provide valuable insights into the reliability and robustness of modern computing systems, which can be incorporated into advanced courses on software engineering, database management, and artificial intelligence, ensuring that students are exposed to innovative testing methodologies and real-world case studies.
The research will enhance Lincoln University's in software reliability and testing expertise, promoting faculty scholarship and interdisciplinary collaboration, and potentially laying the groundwork for future grant applications, faculty-student research initiatives, and industry partnerships.
Additionally, this research will create opportunities for independent studies, research projects, and student involvement in software testing beyond the classroom, potentially leading to internships, conference presentations, or research collaborations. By integrating the results into future courses and fostering a stronger research culture, this project will strengthen Lincoln University's role in advancing software reliability and quality assurance research.

5. How the success of the project will be measured

The success of this project will be measured through a quantitative and qualitative analysis from the comparative survey of the mutation testing results which will be presented in the form of tables, charts, and figures, comparing the fault detection capabilities, robustness, and error-handling mechanisms of each system, and highlighting strengths and weaknesses identified in the literature.
Furthermore, the impact of the project will be confirmed by sharing the results orally at a nationally or internationally recognized conference in September, ensuring scholarly participation and a contribution to the software dependability and fault tolerance field.

6. How, when, where, and with whom the project's outcome will be shared

I plan to share the outcomes of this project with faculty members in my department, particularly Dr. Olamide Tawose and Dr. Tiffanie Smith, who have expressed interest in collaborating on future research in software testing. Additionally, I will present the findings to students in my current Software Testing class, who have studied mutation testing in detail and demonstrated enthusiasm for the subject.
By engaging both faculty and students, this project will foster further research opportunities and encourage deeper exploration of software testing methodologies.