# Faculty Development Research Project – Final Report

**Project Title:** *A Comparative Survey of Mutation Testing Across Large Language Models, REST APIs, and Database Engines*
**Investigator:** Dr. Bouchaib Falah
**Department:** Computer Science, Lincoln University
**Project Period:** June–July 2025
**Conference Presentation:** IEEE IEMCON 2025, University of California, Berkeley
**Manuscript Status:** Accepted for publication (IEEE Xplore)

## 1. Introduction

This project aimed to conduct a systematic comparative survey of mutation testing across three major software system types: Large Language Models (LLMs), REST APIs, and Database Engines. As software systems continue to grow in complexity, mutation testing has become a powerful method for assessing the quality of test suites by injecting artificial faults called mutants, and evaluating whether existing tests can detect them.

Although mutation testing is well studied in traditional software systems, its application across heterogeneous architectures, probabilistic neural networks, distributed APIs, and database engines, remain fragmented. This project fills this gap by offering a structured analysis of 30 peer-reviewed studies (10 per domain), identifying cross-domain patterns, challenges, and future opportunities.

The work was completed in Summer 2025, and the resulting article was submitted, presented (**See the appendix**), and accepted for publication at the IEEE International Conference on Computing, Networking and Communications (IEMCON 2025), held at the University of California, Berkeley.

## 2. Project Objectives

The approved proposal outlined the following goals:

1. Conduct a comparative survey of mutation testing as applied to LLMs, REST APIs, and Database Engines.
2. Analyze strengths, weaknesses, and error-handling behavior in each system category.

3. Identify prevailing mutation operators, tools, and evaluation metrics used in the literature.
4. Summarize trends, challenges, and future research directions in mutation testing across domains.
5. Produce a publishable research article and disseminate results to Lincoln University faculty and students.

All goals were completed successfully.

## 3. Methodology

The project followed a structured research methodology consistent with the FDP application and industry-academic standards.

### 3.1 Article Selection

Thirty peer-reviewed studies were collected from IEEE, ACM, Springer, and top-tier software engineering venues. Selection criteria required that each article:

- Focus explicitly on mutation testing (or mutation-based fuzzing/testing),
- Target one of the three system types,
- Provide sufficient technical detail on mutation operators, evaluation, or tooling.

### 3.2 Data Extraction

Each study was summarized according to:

- System type (LLM, REST API, Database Engine)
- Mutation operators and test strategies
- Tooling and frameworks (e.g., EvoMaster, SQLancer, PRIMG)
- Evaluation metrics
- Key findings and limitations

### 3.3 Comparative Analysis

Both quantitative and qualitative comparison techniques were applied, supported with:

- Tables showcasing mutation operators, tool maturity, and metrics

- Bar and pie charts illustrating operator diversity and tool distribution
- Cross-domain evaluation of challenges, strengths, and fault-detection behavior

### 3.4 Validation

Consistency checks were performed during coding and classification. Limitations such as sample size and heterogeneous evaluation environments were identified and discussed in the report and publication.

## 4. Summary of Findings

The completed project highlights several important insights.

### 4.1 Mutation Operators

- **LLMs**: Semantic, prompt-level, fairness-aware, and equivalence-based mutations.
- **REST APIs**: Endpoint mutations, schema-guided mutations, malformed requests, authentication mutations.
- **Database Engines**: SQL clause rewrites, grammar mutations, semantic mutations, and query plan-guided mutations.

### 4.2 Tool Maturity

- **LLM mutation tools** are in early developmental stages; mostly prototypes.
- **REST API tools** such as EvoMaster, RESTler, and DeepREST are highly mature and widely used.
- **Database mutation tools** (SQLancer, GDsmith, Squirrel) demonstrate strong performance and robust automated guidance.

### 4.3 Evaluation Metrics

- Mutation score
- Test-suite robustness
- Code and query-plan coverage
- Fault-detection rate
- Fairness violation detection (LLMs)

### 4.4 Fault Detection and System Behavior

- REST APIs and Database Engines exhibit higher determinism and achieve higher mutation scores.
- LLMs demonstrate vulnerability to minor prompt changes, fairness shifts, and semantic inconsistencies.
- Database engines are especially sensitive to logical and structural query mutations, revealing deep optimization bugs.

### 4.5 Key Challenges

- **LLMs**: lack of oracle standards, nondeterminism, reproducibility issues.
- **REST APIs**: complexity of stateful/multi-step interactions, limited semantic mutation operators.
- **Databases**: difficulty identifying equivalent mutants, limited NoSQL mutation studies, and performance-guided mutation gaps.

## 5. Project Impact on Teaching and Scholarship

### 5.1 Impact on Teaching at Lincoln University

The project will directly enhanced course content in:

- **CSC390 – Software Testing**:
    - Integration of real-world mutation testing case studies
    - Exposure to industry tools such as EvoMaster and SQLancer
    - Demonstrations of mutation operators across modern systems
- **CSC4303 – Applied Research**:
    - Students will be introduced to research methodology, data extraction, and systematic surveys
    - This project will become a model of a faculty-research pipeline

### 5.2 Impact on Research and Faculty Development

- Strengthened Lincoln University's research footprint in software testing and reliability

- Supported interdisciplinary collaboration in AI, Databases, and Software Engineering
- Led to an accepted IEEE conference paper and future journal extension plans
- Prepared the groundwork for future grant proposals, research assistantships, and joint projects

**6. Dissemination of Results**

The project outcomes were shared through:

1. **Conference Presentation –** Delivered at IEEE IEMCON 2025 (UC Berkeley)
2. **Accepted Publication –** To appear in IEEE Xplore (2026)
3. **Departmental Dissemination –** Shared with Dr. Olamide Tawose.
4. **Planned Future Talks –** Potential CS seminar presentation in Spring 2026

**7. Project Completion Statement**

All phases of the project were completed within the approved Summer 2025 timeline:

- Literature review
- Data extraction and analysis
- Comparative tables, figures, and charts
- Full research article writing
- Conference submission, presentation, and acceptance
- Preparation of this final report

This Faculty Development Research Project successfully achieved every proposed objective, resulting in both scholarly dissemination and meaningful impact on teaching and student engagement.

**Appendix:**

**Certificate of Presentation at IEEE IEMCON 2025, University of California, Berkely**


# CERTIFICATE OF PRESENTATION

THIS IS HEREBY PRESENTED TO

**Bouchaib Falah**
**(Lincoln University, USA)**

FOR PRESENTING THE PAPER TITLED

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

IN

**IEEE IEMCON 2025**
29TH - 31ST OCTOBER 2025
UNIVERSITY OF CALIFORNIA, BERKELEY, USA

**SON VUONG**
General Co-chair, IEEE IEMCON 2025

**SATYAJIT CHAKRABARTI**
President, SMART Society