Research Proposal On The Role of Artificial Intelligence in Enhancing Software Testing

Student Name	Student ID	Contribution percentages
CHIN ZHEN HO	1221102540	25%
ERIC TEOH WEI XIANG	1221102007	25%
BERNARD RYAN SIM KANG XUAN	1221101777	25%
GAN SHAO YANG	1221103201	25%

Introduction

Background:
As software grows more complex, traditional manual testing is becoming inefficient. AI, especially LLMs like GPT-3.5 and GPT-4, offers potential for automation but faces challenges in reliability, integration testing, and testing AI/ML models.

Research Topic:
This proposal aims to optimize AI-driven tools to improve reliability, scalability, and adaptability in diverse testing environments.

Importance:
Solving these challenges is essential for faster, more accurate, and reliable software testing, critical for high-stakes industries.

Problem Statement with Justification/Motivation

1. Inconsistent LLM Performance:

- LLM-generated unit tests are mostly explored for Java, limiting their use across languages like Python and C++.

Justification:

- Ensuring LLM reliability across languages enables broader AI-based testing adoption.

2. Limited AI in Integration Testing:

- AI tools have focused on unit testing, with less attention on complex integration testing.

Justification:

- Automating integration testing reduces manual effort and enhances software reliability.

3. Challenges in Testing AI/ML Models:

- AI/ML models face overfitting and unpredictable behavior, making them hard to test with traditional methods.

Justification:

- Specialized frameworks are needed to ensure AI/ML reliability, especially in critical sectors like healthcare and autonomous systems.

Research Questions

1. How reliable are LLMs, such as GPT-4, in generating unit tests for different programming languages?

2. Can AI-driven tools effectively automate integration testing, and how do they compare with traditional manual methods in terms of accuracy and efficiency?

3. What specialized frameworks can be developed to improve the testing of AI/ML models, particularly in mitigating issues related to overfitting and algorithmic uncertainty?

Hypotheses

1. LLM-generated unit tests, when appropriately fine-tuned, will perform reliably across a variety of programming languages.
2. AI-driven tools can automate integration testing more efficiently and accurately than traditional manual approaches.
3. New testing frameworks designed for AI/ML systems will improve the models' generalizability and robustness, reducing overfitting and enhancing performance in real-world scenarios.

Research Objectives

1. To evaluate LLM-generated unit tests' performance across different programming languages.

2. To design AI-driven tools for automating integration testing and measure their effectiveness against manual methods.

3. To develop and validate testing frameworks for AI/ML systems that address overfitting and uncertainty.

Literature Review

AI-driven testing with Large Language Models (LLMs)

- highlight the use of GPT-3.5
- Performance depends heavily on fine-tuning factors

Automation of testing processes

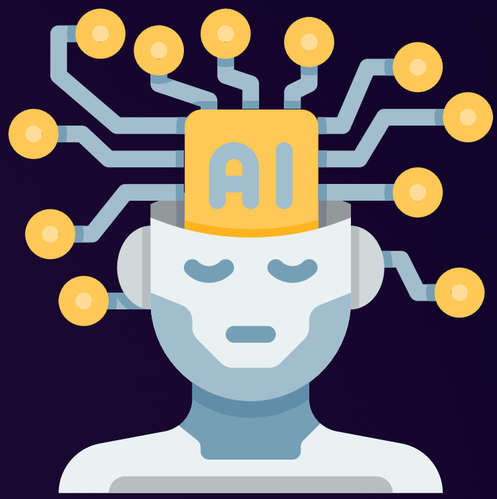
- significantly reduce manual effort
- Integration testing

Challenges in testing AI/ML models

- identify key issues like overfitting and algorithmic uncertainty
- Traditional testing methods aren't suited

Gaps in research

- lack of research into the cross-language applicability
- Limited exploration of AI-driven integration testing
- Absence of specialized frameworks



Research Methodology

Step 1: Empirical Evaluation of LLM-Generated Unit Tests Across Multiple Programming Languages

Step 2: Development and Evaluation of AI-Driven Integration Testing Tools

Step 3: Designing Specialized Frameworks for AI/ML System Testing

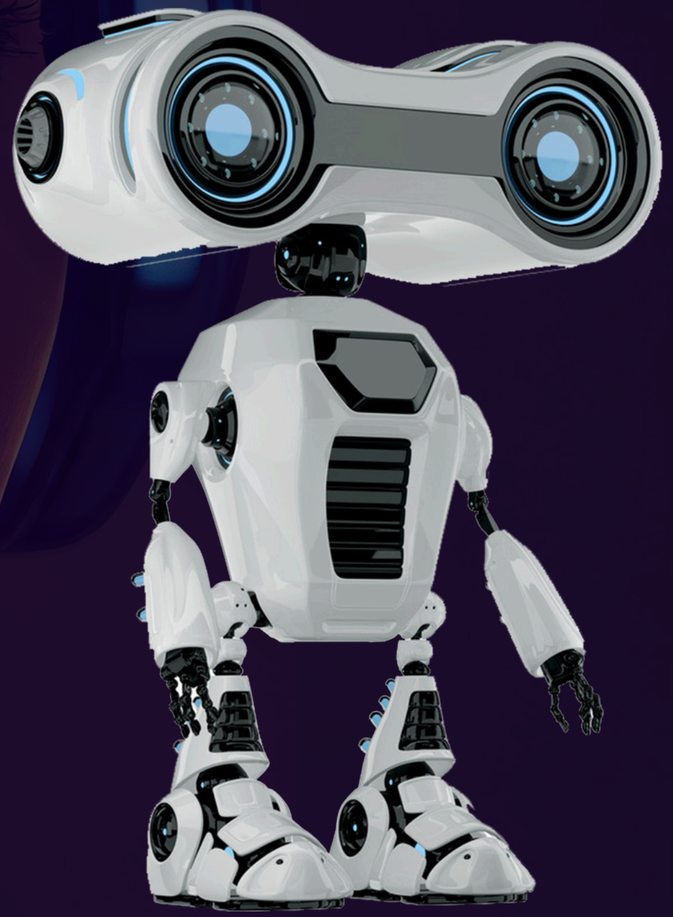
Metrics:

Key performance indicators (KPIs) include:

- **Code Coverage**
- **Test Accuracy**
- **Fault Detection Rate**
- **Reduction of Overfitting**
- **Robustness**

Research Activities and Milestones

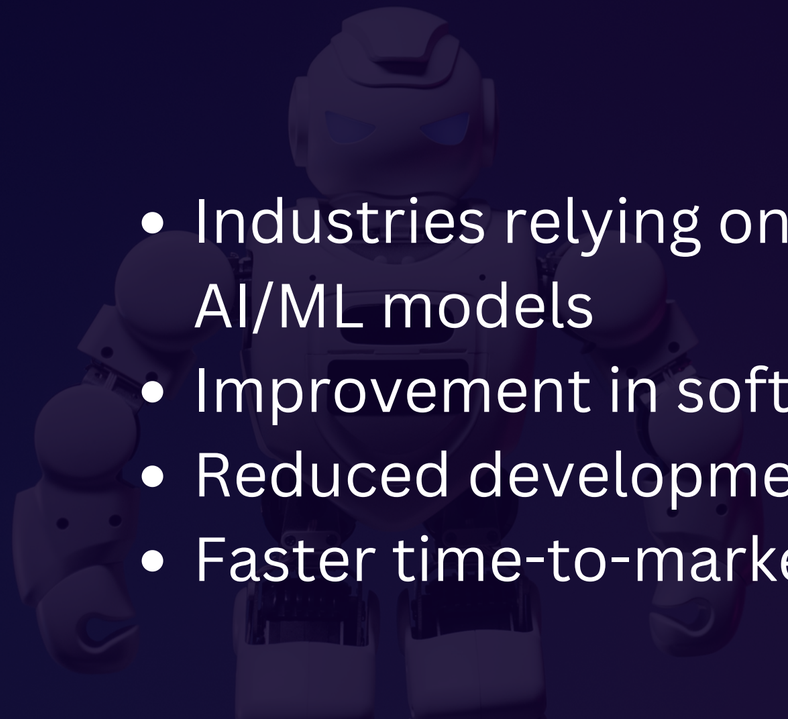
- Month 1-2:
  Activity: Set up the experimental framework.
  Milestone: Complete the experimental design and preparation of datasets for testing.
- Month 3-4:
  Activity: Conduct comparative analysis.
  Milestone: Completion of initial LLM-generated test evaluation and data collection.
- Month 5-6:
  Activity: Begin implementation.
  Milestone: Initial implementation of AI-driven tools for integration testing.
- Month 7-8:
  Activity: Conduct experiments.
  Milestone: Completion of integration testing tool evaluation and analysis of results.
- Month 9-10:
  Activity: Develop specialized frameworks.
  Milestone: Framework development completed and initial testing environment prepared.
- Month 11:
  Activity: Conduct preliminary testing.
  Milestone: Completion of preliminary testing and initial results analysis.
- Month 12:
  Activity: Finalize the specialized testing frameworks.
  Milestone: Submission of results for publication and finalization of research report.



Expected Result & Impact

Expected Result:

- Produce reliable, standardized methods for generating unit tests across multiple programming languages
- Creation of AI-driven tools
- Development of specialized frameworks for testing AI/ML models



IMPACT:

- Industries relying on complex software systems and AI/ML models
- Improvement in software reliability and quality
- Reduced development costs
- Faster time-to-market for software solutions


Thank You!