evoLLve'M: Improving Test Assertions and Mutation Score using ChatGPT-4o and EvoSuite

Arda Turhan*, Annibale Panichella**, Mitchell Olsthoorn**

*d.a.turhan@student.tudelft.nl, **Professor and supervisor from the Software Engineering Research Group (SERG)

1 INTRODUCTION

Software testing is a vital yet time consuming process during the development lifecycle, often causing engineers to limit its use in practice. In this research, we leverage the strengths of the following two prominent test generation techniques, with the aim of creating improved test assertions that encourage active testing.

Search-Based Software Testing:

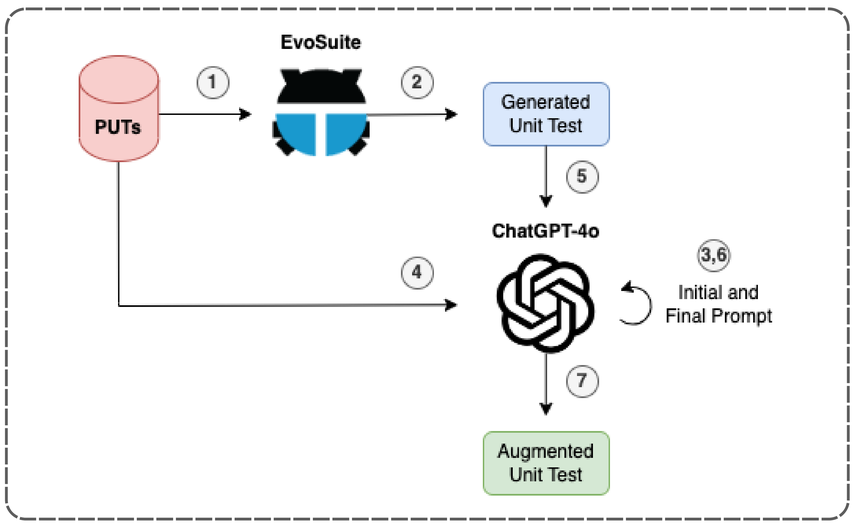
+ Efficient and effective test case generation
+ Proven SOTA approach (EvoSuite)
- Exploring edge cases for the entire input space

Large Language Models:

+ Unique inference and pattern matching capabilities
+ Language comprehension
- Hallucinations during code synthesis

2 APPROACH

Our approach, called evoLLve'M, aims to generate unit tests with meaningful test assertions by utilizing the strengths of both search-based test generation and that of LLMs. These strengths are effective test structure generation and comprehension and inference capabilities, respectively. Shown in the figure below, the approach uses EvoSuite to create an initial JUnit test case for each Program Under Test (PUT), of which the assertions will be further improved by ChatGPT-4o using few-shot prompting.



3 RESEARCH QUESTIONS

RQ 1: How effective are tests produced by evoLLve'M compared to only using EvoSuite, measured by **mutation score**?

RQ 2: What is the impact of evoLLve'M on **types of mutations killed**, compared to only using EvoSuite?

4 EVALUATION

Benchmark:
20 classes gathered from the SF110 corpus.

Configurations:
EvoSuite V1.2.0 (default), ChatGPT-4o-2024-05-13, and PIT 16.1.1

Analysis:
6 runs for each approach. Unpaired Wilcoxon test, and Vargha-Delaney $\hat{A}_{12}$ on mutation score.

5 RESULTS

ID	Project Name	Program Under Test	CC	EvoSuite	evoLLve'M	p-value	$\hat{A}_{12}$
1	Tullibee	client.Util	22	1.00	1.00	-	- (0.5)
2	Tulibee	client.Contract	26	1.00	0.99	0.26	S (0.36)
3	templateit	OpMatcher	27	0.72	0.94	0.04	L (0.92)
4	sfms	crypt.Base64	22	0.78	0.78	0.85	- (0.5)
5	imSMART Migration	servlet.HTMLFilter	8	1.00	1.00	-	- (0.5)
6	BeanBin	search.WildcardSearch	11	0.96	0.96	-	- (0.5)
7	saxpath	Axis	28	1.00	1.00	-	- (0.5)
8	Java View Controller	tools.Base64Coder	35	0.95	0.96	0.26	M (0.72)
9	Java View Controller	tools.HtmlEncoder	14	0.73	0.67	0.08	L (0.25)
10	Corina	util.NaturalSort	47	0.67	0.66	0.84	- (0.44)
11	Corina	util.Sort	29	0.07	0.19	0.10	L (0.75)
12	Corina	util.StringComparator	7	1.00	1.00	-	- (0.5)
13	Corina	util.StringUtils	26	0.88	0.89	0.71	S (0.63)
14	SchemaSpy	util.Version	14	0.84	0.95	0.03	L (0.92)
15	Java Interactive Profiler	ByteVector	33	0.33	0.46	0.03	L (1.00)
16	Lagoon	util.Utils	25	0.96	0.96	0.16	M (0.67)
17	openjms	util.CommandLine	24	0.88	0.84	0.41	M (0.71)
18	biblestudy	util.Queue	28	0.81	0.88	0.04	L (0.94)
19	Battlecry	bcWord	27	0.78	0.87	0.03	L (0.83)
20	fim1	utils.StringEncoder64	38	0.77	0.71	0.06	M (0.28)
Mean over all projects				80.6%	83.5%		

Type	EvoSuite killed	EvoSuite total	evoLLve'M killed	evoLLve'M total
CB	359 (67.2%)	534	370 (69.3%)	534
ER	249 (90.2%)	276	260 (93.9%)	277
FR	90 (100%)	90	88 (100%)	88
TR	199 (97.1%)	205	202 (97.1%)	208
INC	290 (65.3%)	444	368 (82.9%)	444
MTH	1107 (76.0%)	1456	1082 (75.1%)	1440
NC	1529 (90.69%)	1686	1565 (91.4%)	1712
NR	133 (88.7%)	150	146 (97.3%)	150
PR	241 (84.9%)	284	249 (85.3%)	292
VMC	189 (79.4%)	238	188 (79.7%)	236
Total	**4386 (81.8%)**	5363	**4518 (84.0%)**	5381

6 CONCLUSION

31.3% out of all classes are significantly improved, without having a negative impact on the other classes. For improved classes, **the average increase in mutation score is 20.6%.**

Killed mutations of type **INC** and **NR** are increased by **26.9%** and **8.9%**, respectively. Other types show similar results.

TU Delft Delft University of Technology