TU Delft

LLM-Based Test Generation from Bytecode

An Empirical Comparison with EvoSuite and the Effect of Example Test Artifacts

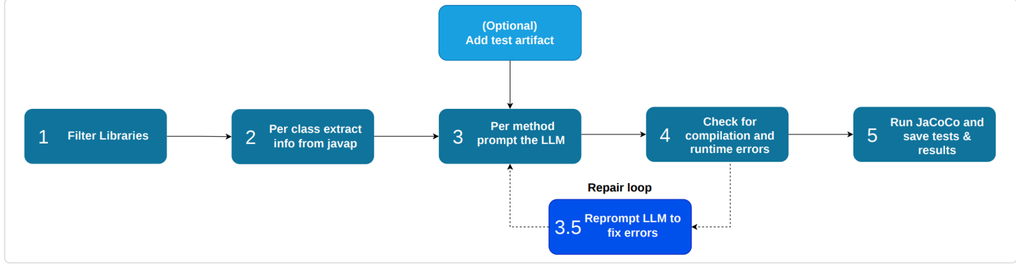
Julian Louis Overmars · Supervisors: Sebastian Proksch, Cathrine Paulsen · EEMCS, Delft University of Technology · June 2026

1 Motivation

- 70–90% of software depends on third-party libraries
- Library updates can break behavior — tests detect this
- Source code is **often unavailable**; bytecode always is
- EvoSuite achieved only **53% mutation score** (SBST 2021)
- Test Wars [Abdullin et al.] showed LLMs match EvoSuite on mutation score given *source code*
- LLMs hallucinate APIs → **low compilation rates** [Celik & Mahmoud]

Research question: Can LLMs generate effective tests from *bytecode alone*, and do example tests reduce hallucination?

2 Methodology



Setup

- 5 Apache Commons Java libraries
- Max 100 classes per library (seed 42)
- javap bytecode representation → LLM prompt per method
- deepseek-v4-flash via OpenRouter
- Up to 2 retries on compile / runtime errors

Three configurations

BC Bytecode only

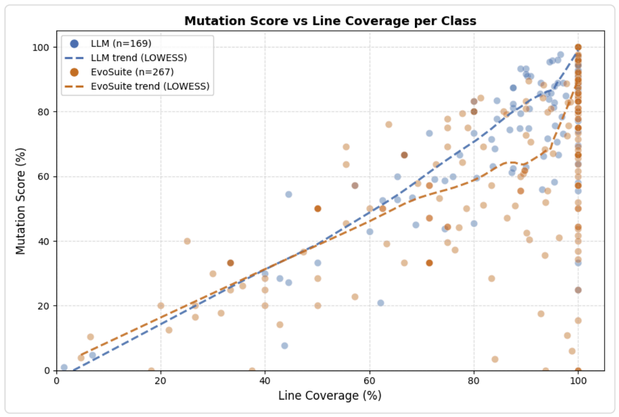
BTB + decompiled test bytecode

BTS + test source code

Metrics

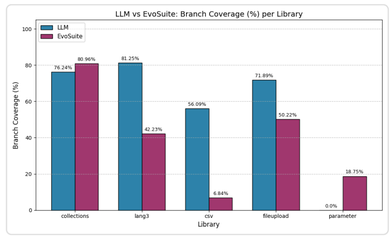
- Branch / line / method coverage (JaCoCo)
- Mutation score (PIT)

3 RQ1 LLM vs EvoSuite on Bytecode



LLM trend rises faster at high coverage → more fault-detecting tests

Branch coverage



82% LLM mean mut. score Median: 90.3%

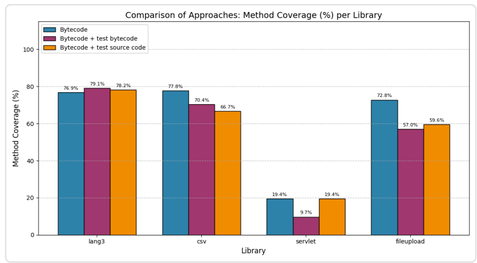
70% EvoSuite mean mut. score Median: 77.8%

Based on 103 classes with both LLM and EvoSuite tests

✔ LLM outperforms EvoSuite on branch coverage in **3/5 libraries** (up to +49 pp) and achieves higher per-class mutation scores — confirming Test Wars findings hold for bytecode

4 RQ2 Do Test Artifacts Help?

Method coverage (BC vs BTB vs BTS)



- Branch coverage improves on csv & fileupload with artifacts
- Method coverage **drops** in 3/4 libraries — hallucination not solved
- Mutation score: BC 77.1% · BTB 76.0% · BTS 77.0% — **no meaningful difference**
- BTB ≈ BTS — decompiled bytecode works as well as source code

✔ Artifacts improve branch coverage on some libraries
~ Do **not** consistently reduce hallucination · Source ≈ bytecode examples

5 Conclusions

RQ1 LLM vs EvoSuite

- LLM viable alternative to EvoSuite on bytecode
- Outperforms on coverage for 3/5 libraries
- Higher per-class mutation score (82% vs 70%)
- Slightly inconsistent on very large classes

RQ2 Test Artifacts

- Mixed results — library-dependent
- Branch coverage ↑ on small libraries
- Method coverage ↓ — hallucination persists
- Source code examples ≈ bytecode examples

Future Work

- Better bytecode representations (javap vs decompilers)
- Larger, more diverse library set
- Compare bytecode vs source code input directly
- Qualitative analysis of generated tests

6 Limitations

- Only 5 libraries — limits generalizability
- PIT ran on a non-random subset of classes (high coverage bias)
- LLM is nondeterministic despite seed
- Training data may include tested libraries (data leakage risk)
- RQ2 only tests classes with matching developer test class