

An Evaluation of the Statement Deletion Mutation Operator in Real-World Python Projects

Shan Cao, Dongyuan Liu

Introduction

Many open-source libraries use test coverage as a measurement of test suite quality. However, test coverage only guarantee the code is being executed when running the tests, it does not check that the tests are able to detect real faults. To get deeper knowledge of the flaws, running test set against slightly modified versions of programs is one approach. This technique is called *mutation testing*. In this paper, we specifically evaluate the statement deletion mutation operator (SDL). We evaluate on real-world Python projects and their test suites.

MutPy

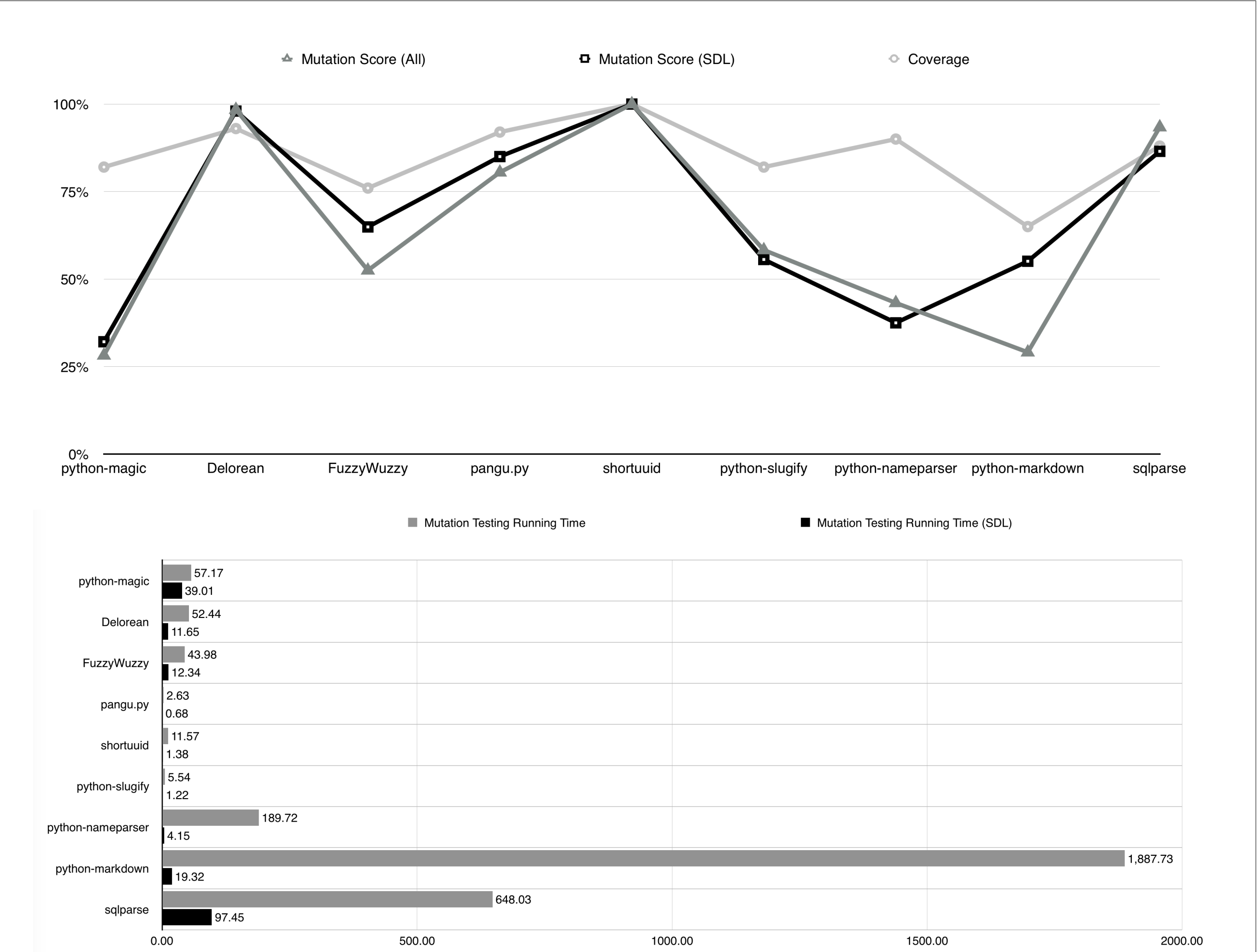
MutPy is a mutation testing tool for Python projects. We use it to compute and compare the result of SDL and experimental mutation. This tool has not received any updates since 2014. To make it compatible to recent libraries, we updated it to support Python 3.5 and fixed some issues in loading unit tests before starting the experiment. The version we are using is in this forked repository.

Python projects for evaluation

Project	LOC	Description
python-magic	391	A Python wrapper for libmagic
Delorean	1991	A library for datetimes in Python
FuzzyWuzzy	1352	Fuzzy string matching in Python
pangu.py	446	Paranoid text spacing in Python
shortuuid	355	A generator library for UUIDs
python-slugify	425	Returns unicode slugs
python-nameparser	4027	Parse human names into components
python-markdown	8199	A Python implementation of Markdown
sqlparse	19434	A non-validating SQL parser for Python

Mutation Operators Supported by MutPy	
Operation	Description
AOD	arithmetic operator deletion
AOR	arithmetic operator replacement
ASR	assignment operator replacement
BCR	break continue replacement
COD	conditional operator deletion
COI	conditional operator insertion
CRP	constant replacement
DDL	decorator deletion
EHD	exception handler deletion
EXS	exception swallowing
IHD	hiding variable deletion
IOD	overriding method deletion
IOP	overridden method calling position change
LCR	logical connector replacement
LOD	logical operator deletion
LOR	logical operator replacement
ROR	relational operator replacement
SCD	super calling deletion
SCI	super calling insert
SIR	slice index remove

Experimental Mutation Operators	
Operation	Description
CDI	classmethod decorator insertion
OIL	one iteration loop
RIL	reverse iteration loop
SDI	staticmethod decorator insertion
SDL	statement deletion
SVD	self variable deletion
ZIL	zero iteration loop



SDL-mutation compared to full mutation

Further Work

- Currently MutPy only loads unit tests based on the Python standard unittest module. However, there are many popular Python projects that do not use the unittest module.
- A big issue for mutation testing with SDL operator is that after removing a statement, it could form an infinite loop. We can record the running time for all the tests before running mutation testing. For each test, we set the timeout separately based on its running time.
- Mutation testing is slow because it runs the whole test suite for every mutant. We can run a subset of tests that covers the mutant.
- Investigate the relevancy between survived mutants and bugs.
- Explore appropriate use cases for mutation testing.