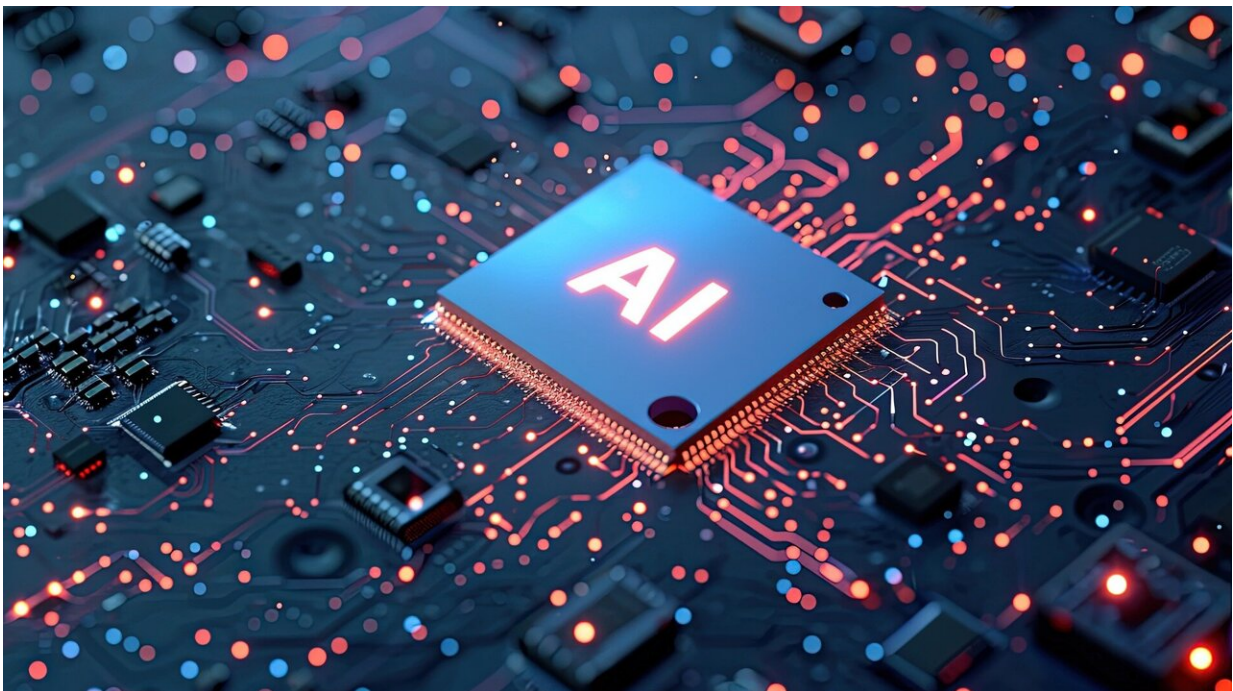


Novel approach improves automatic software repair by generating test cases

July 23 2024



Credit: Pixabay/CC0 Public Domain

IMDEA Software researchers Facundo Molina, Juan Manuel Copia and Alessandra Gorla present FIXCHECK, a novel approach to improve patch fix analysis that combines static analysis, randomized testing and large language models.

Their innovations, embodied in the paper: "Improving Patch Correctness

Analysis via Random Testing and Large Language Models" were presented at the International Conference on Software Testing, Verification and Validation ([ICST 2024](#)), and additional details are [available](#) on the *Zenodo* server.

Generating [patches](#) that fix software defects is a crucial task in the maintenance of software systems. Typically, software defects are reported via [test cases](#), which unveil undesirable behaviors in the software.

In response to these defects, developers create patches that must undergo validation before being committed to the codebase, ensuring that the test provided no longer exposes the defect. However, patches may still fail to effectively address the underlying bug or introduce new bugs, resulting in what is known as bad fixes or incorrect patches.

The detection of these incorrect patches can significantly impact the time and effort spent on bug fixes by developers and the overall maintenance of software systems.

Automatic program repair (APR) provides [software developers](#) with tools capable of automatically generating patches for buggy programs. However, their use has uncovered numerous incorrect patches that fail to address the bug.

To tackle this problem, researchers at IMDEA Software have created FIXCHECK, a novel approach for improving the output of patch correctness analyses that combines static analysis, random testing and [large language models](#) (LLMs) to automatically generate tests to detect bugs in potentially incorrect patches.

FIXCHECK employs a [two-step process](#). The first step consists of generating random tests, obtaining a large set of test cases. The second

step is based on the use of large language models, from which meaningful assertions are derived for each test case.

In addition, FIXCHECK includes a selection and prioritization mechanism that executes new test cases on the patched program and then discards or ranks these tests based on their probability of revealing bugs in the patch.

"The effectiveness of FIXCHECK in generating test cases that reveal bugs in incorrect patches was evaluated on 160 patches, including both developer-created patches and patches generated by RPA tools," states Facundo Molina, postdoctoral researcher at Institute IMDEA Software.

The results show that FIXCHECK can effectively generate bug detection tests for 62% of incorrect developer-written patches, with a high degree of confidence. In addition, it complements existing patch fix evaluation techniques by providing test cases that reveal bugs for up to 50% of incorrect patches identified by state-of-the-art techniques.

FIXCHECK represents a significant advance in the field of software repair and maintenance by providing a robust solution for automating test generation and detecting faults during software maintenance. This approach not only improves the effectiveness of patch validation, but also promotes wider adoption of automated program repair methods.

More information: Facundo Molina et al, Improving Patch Correctness Analysis via Random Testing and Large Language Models (Replication Package), *Zenodo* (2024). [DOI: 10.5281/zenodo.10498173](https://doi.org/10.5281/zenodo.10498173)

Provided by IMDEA Software Institute

Citation: Novel approach improves automatic software repair by generating test cases (2024, July 23) retrieved 23 July 2024 from

<https://techxplore.com/news/2024-07-approach-automatic-software-generating-cases.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.