



| FuSeBMC - AI

Efficient Hybrid Fuzzing for Detecting Vulnerabilities and Achieving High Coverage in Software

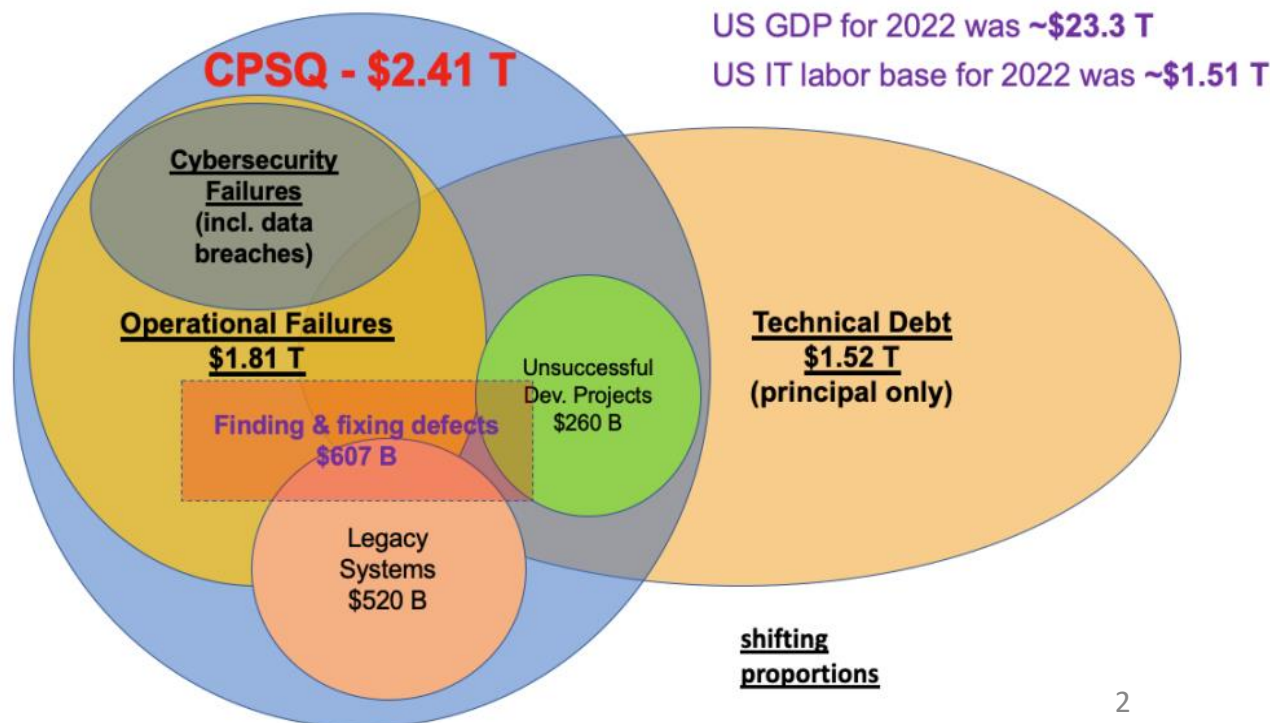
Dr. Kaled Alshmrany

The University of Manchester

kaled.alshmrany@manchester.ac.uk

How much could software errors cost your business?

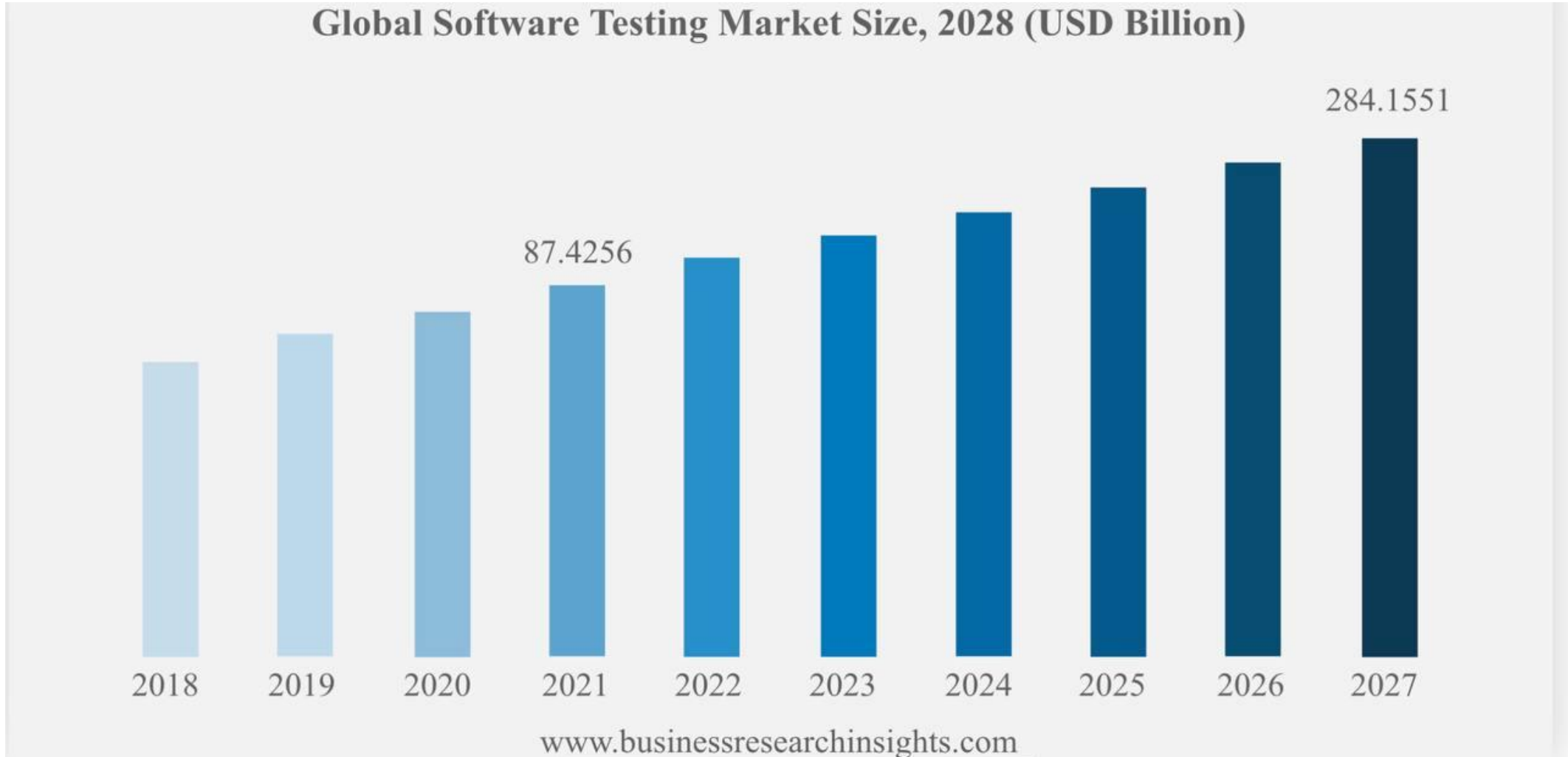
Poor software quality cost US companies \$2.41 trillion in 2022, while the accumulated software Technical Debt (TD) has grown to ~\$1.52 trillion



TD relies on temporary easy-to-implement solutions to achieve short-term results at the expense of efficiency in the long run

The cost of poor software quality in the US: A 2022 Report

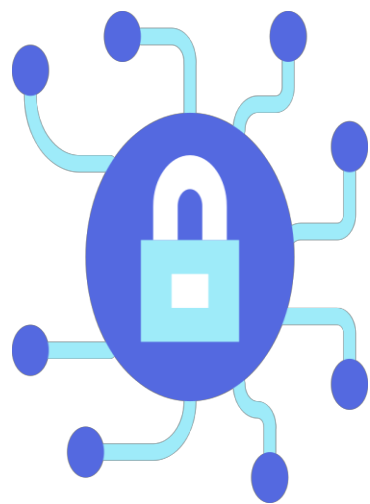
Market Size



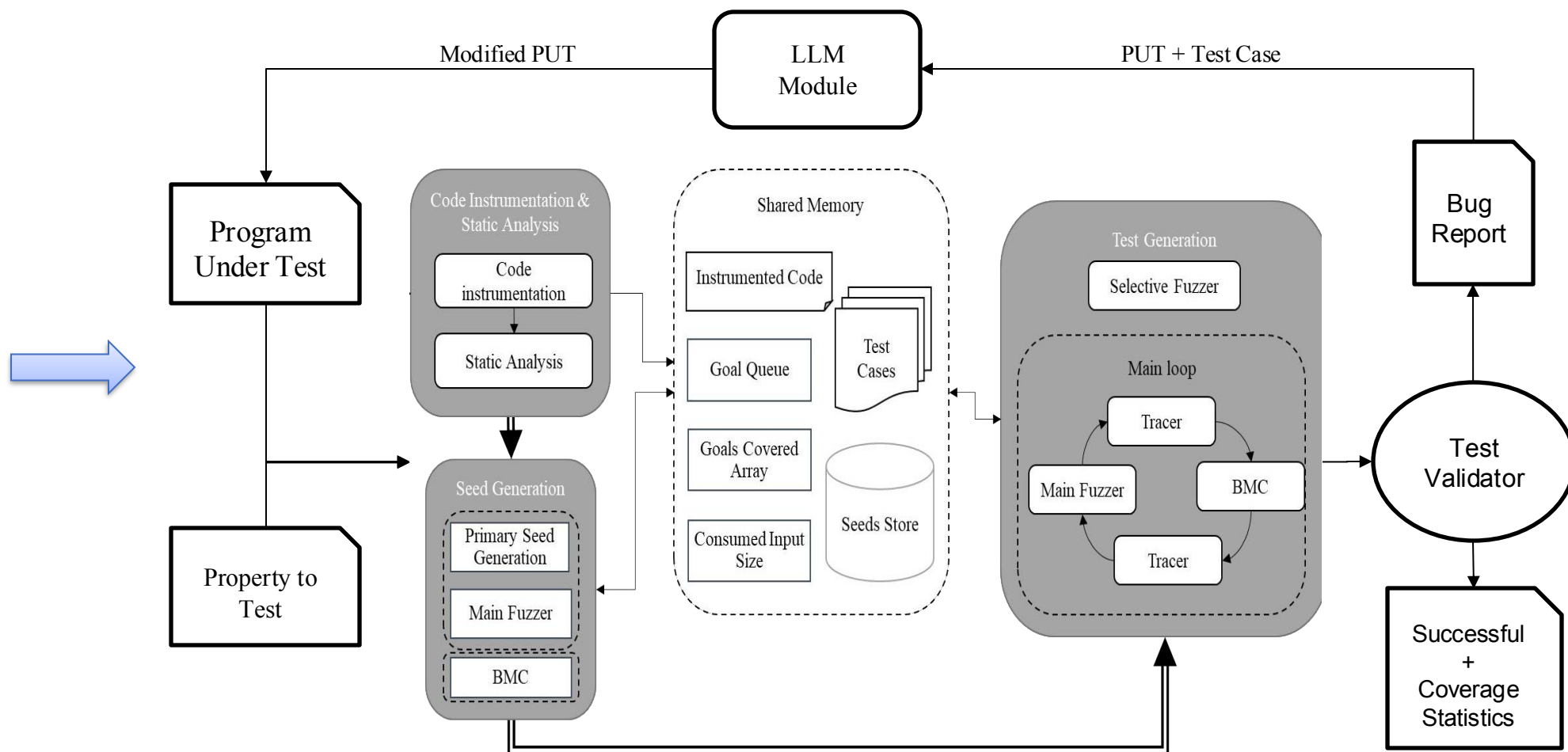
Proposed Solution

FuSeBMC-AI

- Use **Clang** tooling infrastructure
- Employ three engines in its **reachability analysis: one BMC and two fuzzing engines**
- Use a **tracer** to coordinate the various engines



FuSeBMC AI



FuSeBMC-AI Software Project

FuSeBMC-AI source code is written in C++ and Python; it is available for download on GitHub. Also, the instructions for using the tool FuSeBMC-AI are given in the file README.

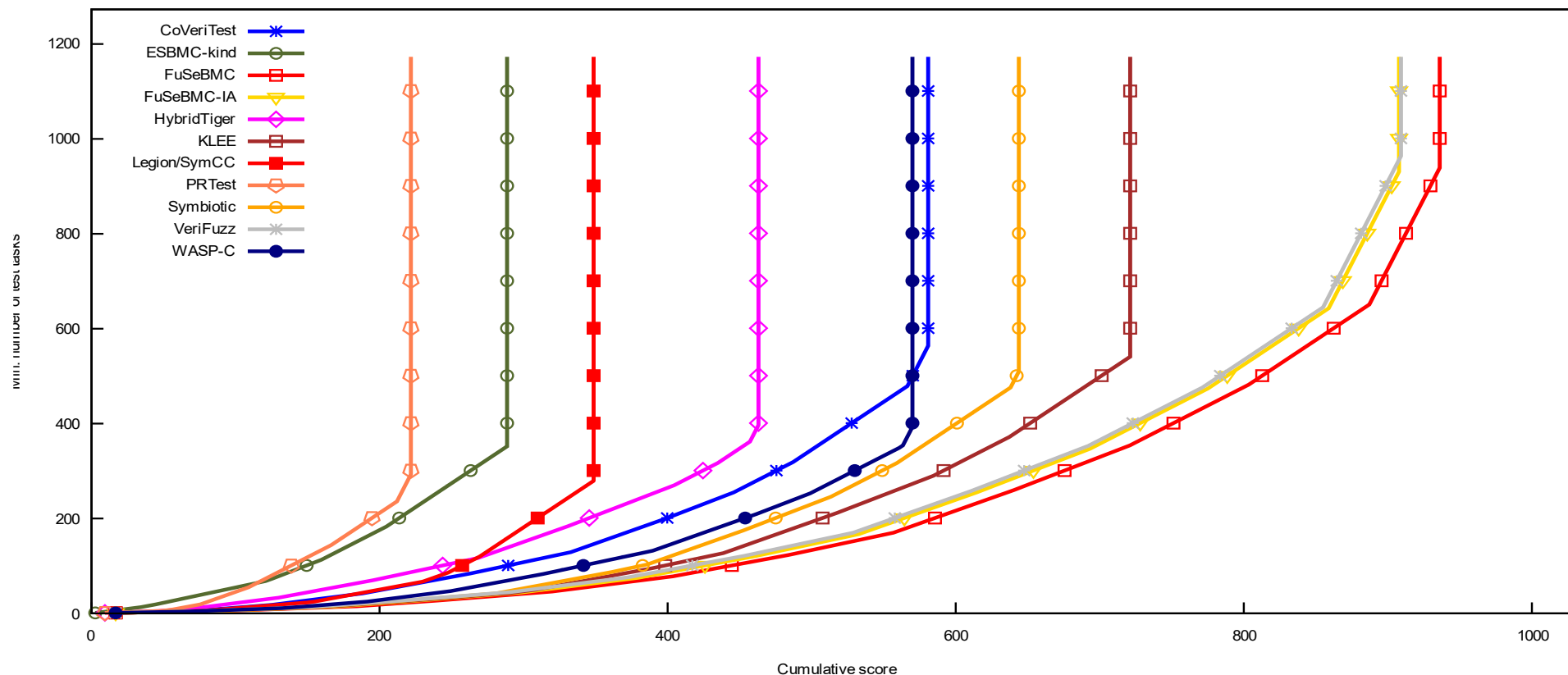
The screenshot displays the FuSeBMC-AI web interface with the following configuration:

- Benchmark:** sv-benchmarks/c/reducercommutativity/rangesum.i
- Property:** ./properties/coverage-branches.prp
- Strategy:** kinduction
- Arch:** 32
- Timeout:** 300 second(s)
- Machine Learning:** Predicate FuSeBMC Paramerters
- Model:** Decision Tree Classifier (Classification: 4.0)
- Cover-Branches:**
 - unlimited-k-steps: max-k-step: 10 k-step: 1 unwind: 1 context-bound: 2
 - max-inductive-step: 12
 - GoalTracer:
 - Fuzzer 1: 20 second(s)
 - Fuzzer 2: 287 second(s)
 - Min Num of TCs to Run AFL: 1
 - Handle Infinite While Loop: 20 second(s)
 - Handle Selective Inputs: 20 second(s)
 - GoalSorting: DEPTH_THEN_TYPE
 - Global Depth of Goals:
- Run TestCov:**
- Result Dir:** [empty field]
- Buttons:** Stop, Generate Cmd, Start
- Command:** XML Parameters
./fusebmc.py -p ./properties/coverage-branches.prp --arch 32 --run-testcov --timeout 300 --ml 2 --ml-model 0 sv-benchmarks/c/reducercommutativity/rangesum.i
- Run Output Dir:** /home/hosam/sdb1/FuSeBMC/fusebmc_output/rangesum.i_jbvPALwOeHHsOINHjwmEXlebT
- Test Results Table:**

	Test	Individual	Accumulated	Part of reduced suite
1	Testcase_24_Fu.xml	13.33	13.33	True
2	testcase_16_ES.xml	6.67	20.0	True

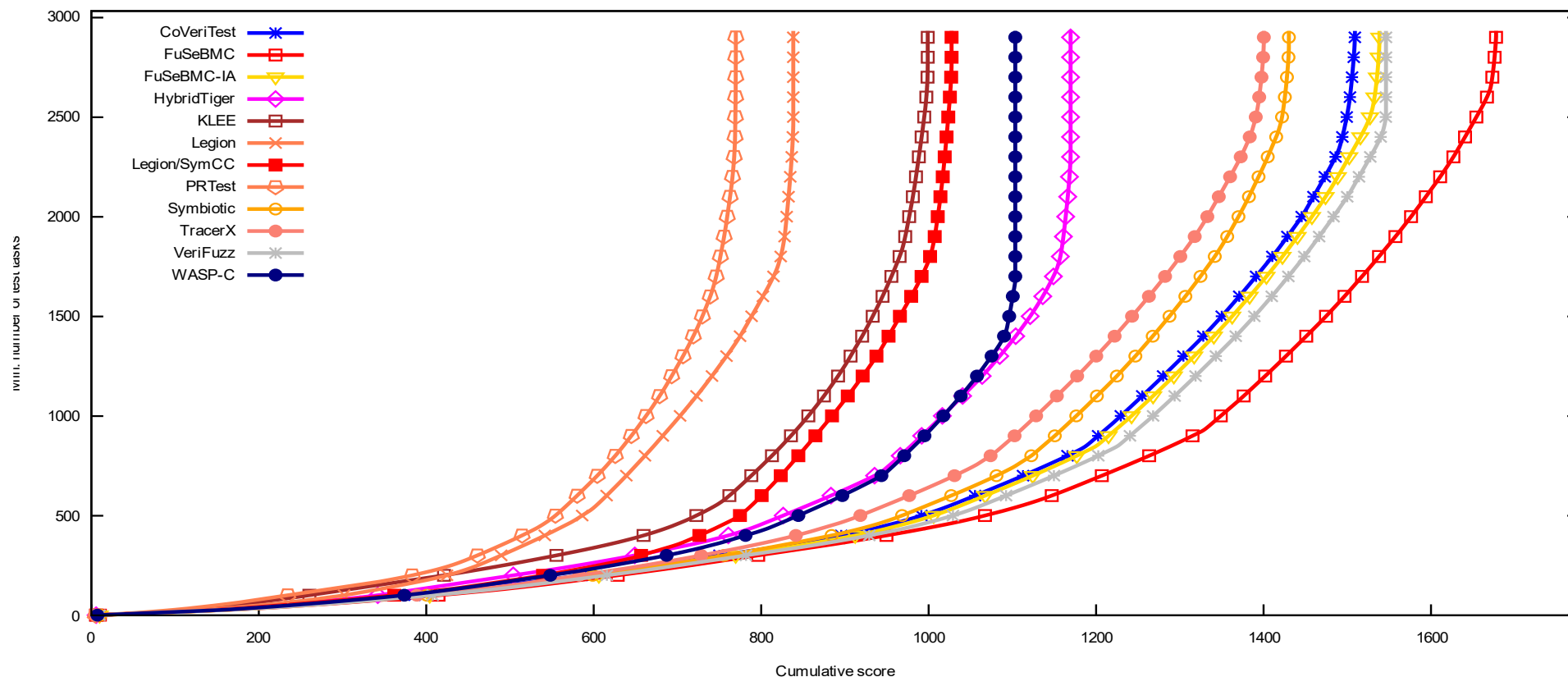
```
kaled@kaled-VirtualBox:~/Desktop/FuSeBMC_v3.6.6$ ./fusebmc.py -s incr -p properties/coverage-branches.prp sv-benchmarks/c/array-tiling/skippedu.c
```

Competition on Software Testing 2023: Results of the Cover-Error Category



FuSeBMC achieved 1st place in Cover-Error

Competition on Software Testing 2023: Results of the Cover-Branches Category

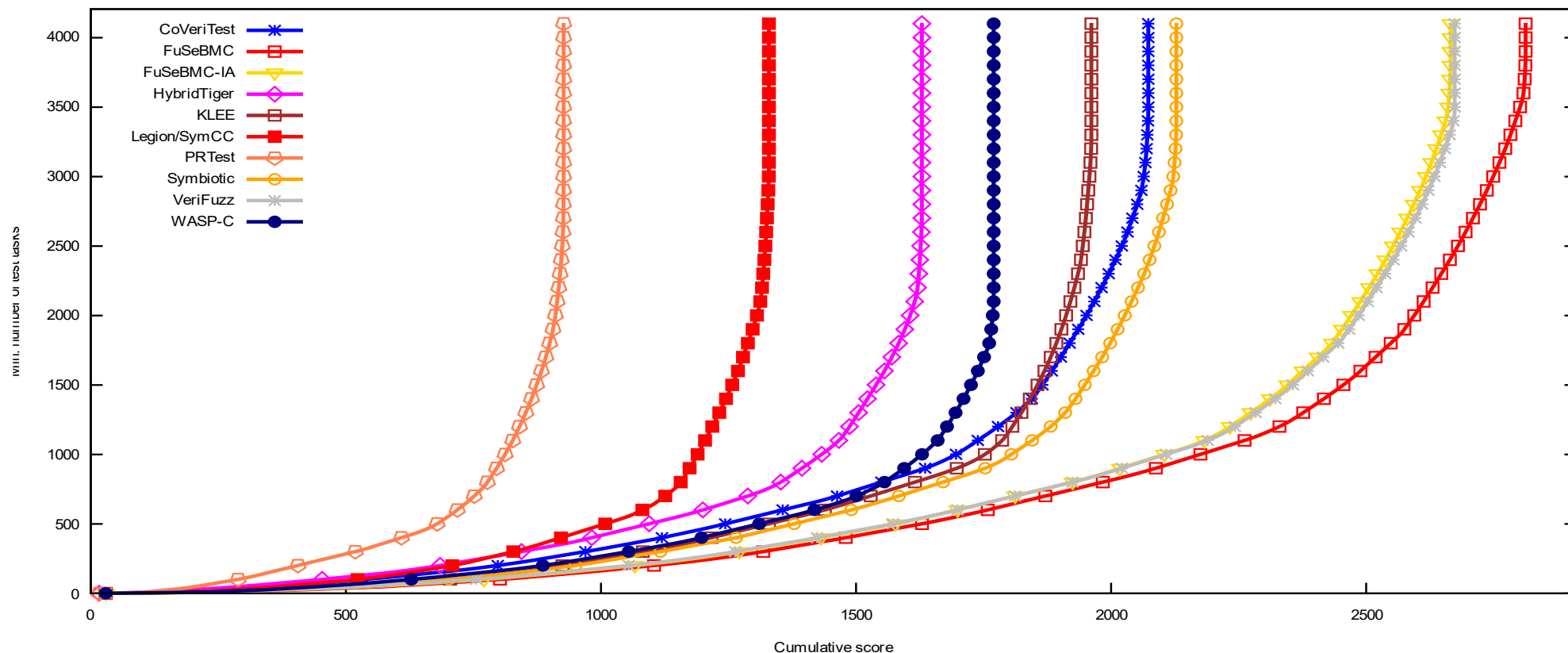


FuSeBMC achieved 1st place in Cover-Branches

Alshmrany, K., Aldughaim, M., Bhayat, A., Cordeiro, L.: FuSeBMC v4: Smart Seed Generation for Hybrid Fuzzing - (Competition Contribution). FASE 2022: 336-340

<https://test-comp.sosy-lab.org/2023/>

Competition on Software Testing 2023: Results of the Overall Category



FuSeBMC achieved 3 awards: 1st place in Cover-Error, 1st place in Cover-Branches, and 1st place in Overall

Awards

FuSeBMC-AI received 18 significant awards from the International Competition on Software Testing (Test-Comp 2021 - 2024) organised by the European Joint Conferences on Theory and Practice of Software (ETAPS).

 ETAPS EUROPEAN JOINT CONFERENCES ON THEORY & PRACTICE OF SOFTWARE	FASE 2021	 ETAPS EUROPEAN JOINT CONFERENCES ON THEORY & PRACTICE OF SOFTWARE	FASE 2022	 ETAPS EUROPEAN JOINT CONFERENCES ON THEORY & PRACTICE OF SOFTWARE	FASE 2023	 ETAPS EUROPEAN JOINT CONFERENCES ON THEORY & PRACTICE OF SOFTWARE	FASE 2024
	Test-Comp 2021: Cover-Error (find a test that covers a bug).		Test-Comp 2022: Cover-Error (find a test that covers a bug).		Test-Comp 2023: Cover-Error (find a test that covers a bug).		Test-Comp 2024: Cover-Error (find a test that covers a bug).
	Test-Comp 2021: Consumption of CPU and Memory.		Test-Comp 2022: Cover-Branches (find a test that covers a branch).		Test-Comp 2023: Cover-Branches (find a test that covers a branch).		Test-Comp 2024: Cover-Branches (find a test that covers a branch).
	Test-Comp 2021 's Overall category.		Test-Comp 2022 's Overall category.		Test-Comp 2023 's Overall category.		Test-Comp 2024 's Overall category.

<https://test-comp.sosy-lab.org/2024/>

Publications



- Published paper in Fundamental Approaches to Software Engineering – 24th International Conference, FASE 2021



- Published paper in The International Conference on Tests and Proofs, TAP 2021



- Published paper in Fundamental Approaches to Software Engineering – 25th International Conference, FASE 2022



- Published paper in The Formal Aspects of Computing Journal FAC 2024



- Published paper in IEEE Secure Development Conference, SecDev 2022



- Published paper in Fundamental Approaches to Software Engineering – 26th International Conference, FASE 2023

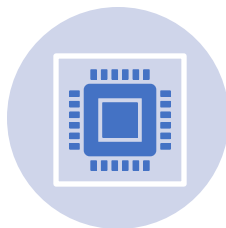
FuSeBMC-AI 's Impact: Awards and Industrial Deployment



18 awards from the international competitions on software testing (Test-Comp) 2012-2024 at **FASE**.



Most Influential tool at international Competitions 2021-2024.



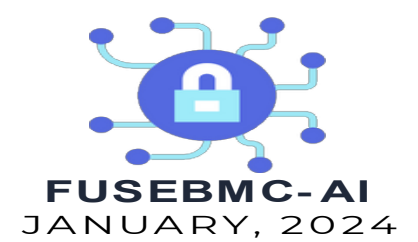
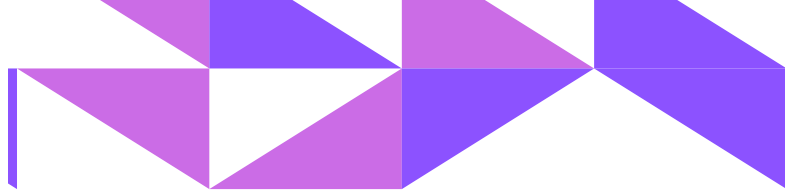
It is classified as **Green testing tool (low Consumption of CPU and Memory)**.



The ability to **detect vulnerabilities** effectively and quickly compared with **state-of-the-art tools**.



Practical and academic contributions illustrated in **6 Published papers** in the field.



SOURCE CODE SECURITY WITH FUSeBMC-AI

Overview

FuSeBMC-AI aims to revolutionize software testing. Originating from collaborations with ARM and Intel, it addresses the need for robust automated testing tools, targeting software developers.

Industry Context

The rising complexity of software has made manual testing impractical. The market for automated testing tools exceeds \$51.8 Billion annually. FuSeBMC-AI offers a powerful solution to detect and repair security vulnerabilities whilst keeping the number of false alarms minimal, meeting the growing demand for secure coding tools.

The Need

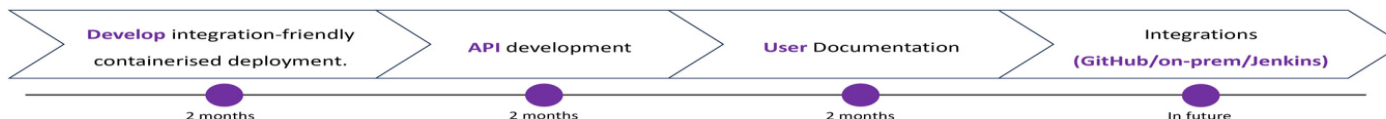
Manual testing is impractical due to high costs, complexity, and a shortage of skilled testers. Existing automated tools often fail to identify or fix security issues and can generate many false alarms, creating additional challenges for developers. There is a pressing need for powerful and reliable testing tools to prevent cyber-attacks.

Innovative Solution

FuSeBMC-AI combines advanced verification and AI methods to detect and fix over 40 types of security vulnerabilities in C language source code (can be extended to other languages). It generates detailed bug reports with locations, types, and suggested fixes, enhancing security and reliability, and enabling continuous learning of software developers. This has earned FuSeBMC-AI 18 Intl. awards in competitions compared to tools from Amazon, Tata, Intel, and others.

Development Stage and Roadmap

We have validated our core technologies in the industry via partnerships with Arm, Ethereum, Intel, and Nokia. From market validation interviews, we have identified a tangible need for this technology across several high-security sectors. Next, we need to develop the core technology in an integration-friendly manner.



Our Team



Prof. Richard Allmendinger
AI Specialist,
Advisor, Professor



Dr. Kaled Alshmrany
Consultant, Researcher,
Cybersecurity Specialist

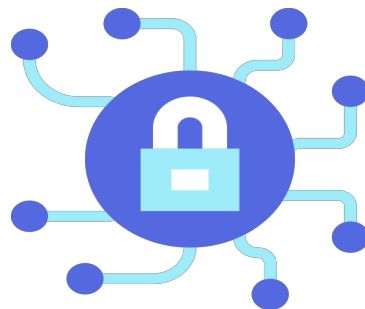


Prof. Lucas Cordeiro
Cybersecurity Specialist,
Advisor, Professor



Dr. Rachel Pooley
Innovation Discovery
Manager

Thank you ...



FuSeBMC AI

Find out more about FuSeBMC-AI at :

<https://github.com/FuSeBMC/>



kaled.alshmrany@manchester.ac.uk

LinkedIn

