

Stacking High-Level Fuzz Mutations in Big Data Applications

How does stacking high-level fuzz mutations affect the test performance for big data applications?

Background



Fuzz testing is an automated testing technique where input is mutated to find new paths in code.



BigFuzz is a newly proposed method which applies fuzz testing to Big Data Applications.



High-level mutations are error type guided mutations based on real faults.

Stacking Methods



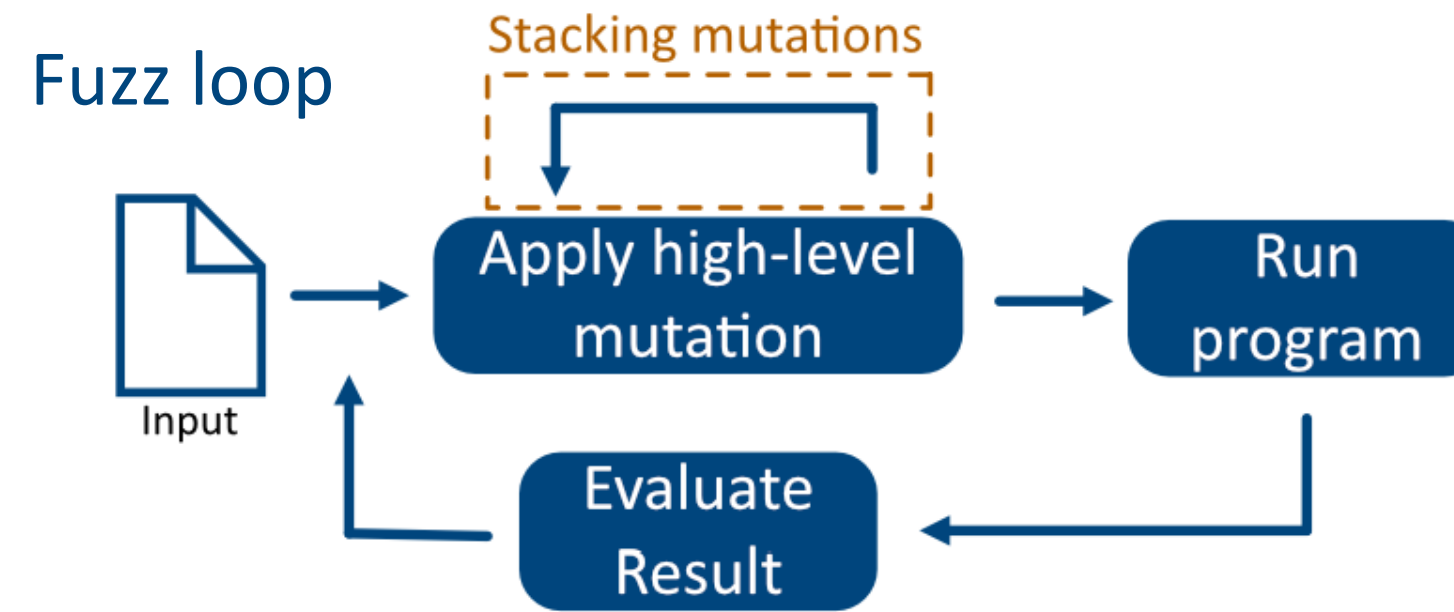
Random Stack: Stack mutations randomly

1

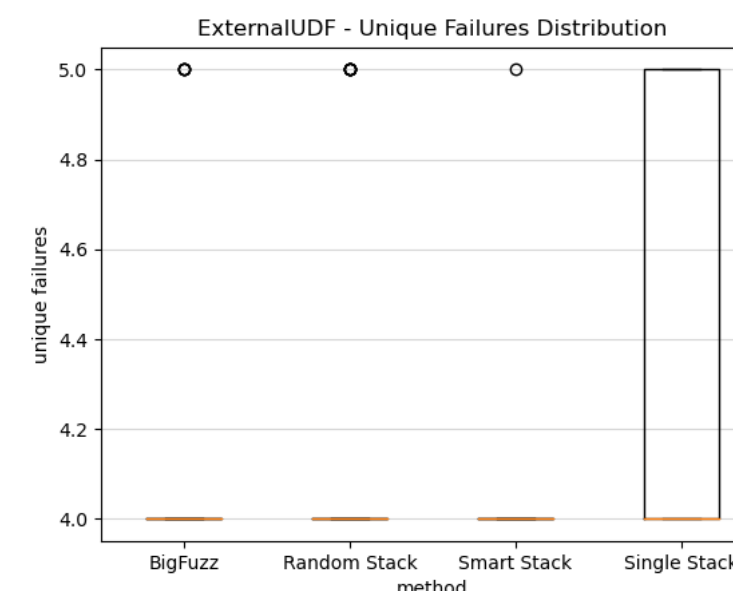
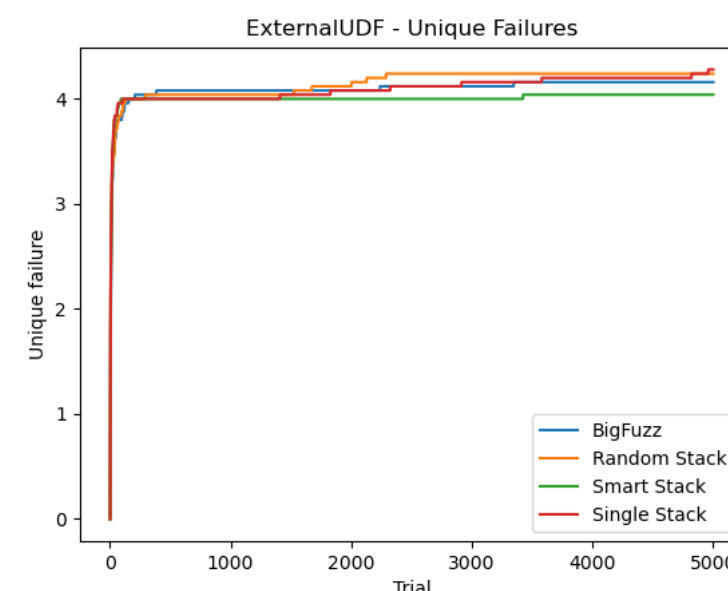
Single Stack: Apply at most one mutation per datapoint



Smart Stack: Stack mutations following the stack rules



3. Results



Stacking rules

Seco	First	M1	M2	M3	M4	M5	M6	M7
d	M1	Green	Green	Green	Green	Orange	Orange	Green
	M2	Green	White	Green	Green	Orange	Orange	Green
	M3	Green	Green	White	Green	Green	Green	Green
	M4	Orange	Orange	Green	Green	Orange	Orange	Green
	M5	Orange	Green	Green	Orange	Orange	Orange	Orange
	M6	Orange	Orange	Green	Orange	Orange	White	Green
	M7	Green	Green	Green	Green	Orange	Green	White
		Mutations do not interfere						
		Mutations interfere						
		No benefit of again mutating						

Conclusion

- More unique failures are found in less tests
- Amount of unique failures found is more reliable
- Best performing stacking algorithm differs per benchmark

Future Work

- Apply the stacking of mutations on a more diverse benchmark suite
- Improve the stacking of mutations by extending the rules set or use biased high-level mutations