

## APPENDIX

The following paragraphs contain detailed information about the fault coverage and fault enhancement of FEST compared with ClassSRTS in category Dual\_F (Figure 15) and category FEST\_F (Figure 16) respectively, the size of selected tests for category Dual\_NF (Figure 17), the reasons for uncovering the faults detected by CI testing in category CI\_F, and the versions of each chosen project in the study (Figure 18).

- **Section IV-A1: Fault coverage and detection enhancement of FEST compared with ClassSRTS in category Dual\_F**

Figure 15 shows the fault coverage and detection enhancement of FEST compared with ClassSRTS in category Dual\_F. We can see that in this category, FEST can find all faults detected by ClassSRTS and find more faults in 19 (83%) versions.

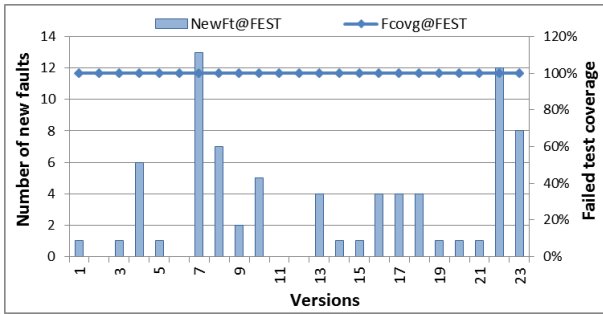


Fig. 15. Fault coverage and detection enhancement compared with ClassSRTS for category Dual\_F

- **Section IV-B1: Fault coverage and detection enhancement of FEST compared with ClassSRTS in category FEST\_F**

Figure 16 shows the fault coverage and detection enhancement of FEST compared with ClassSRTS in category FEST\_F. We can see that in this category, FEST can find all faults detected by ClassSRTS and find more faults in 29 (57%) versions.

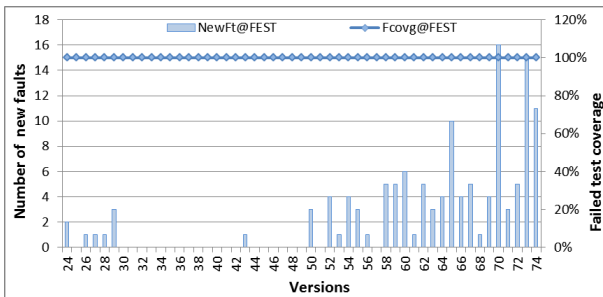


Fig. 16. Fault coverage and detection enhancement compared with ClassSRTS for category FEST\_F

- **Section IV-C: The size of selected tests in category Dual\_NF**

Figure 17 shows the number of tests in CI testing, FEST and ClassSRTS. The versions have been ordered by number of tests in CI testing and then assigned the ID sequentially.

From Figure 17, in 92% (168/184) versions, FEST can reduce the size of tests; on average, it can reduce the size of

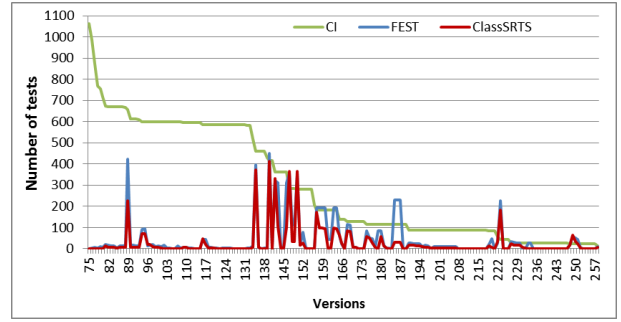


Fig. 17. Number of actual CI tests and number of selected tests for category Dual\_NF

tests by 251 (accounting for 86%) in all versions. Especially, in 59 out of 184 versions (e.g., v90-v130), the number of reduced tests is greater than 500, it is very economical. only in 16 versions, FEST selects more tests than CI testing. This is mainly because FEST searches for full dependencies of changed classes. Comparing with baseline approach, ClassSRTS can reduce the size of tests in 96% versions; on average, it can reduce the size of tests by 266 (accounting for 91%) in all versions. It means that FEST selects slightly more tests than ClassSRTS in few versions for necessity.

- **Section IV-D1: Reasons for uncovering the faults detected by CI testing in category CI\_F**

In v259, there is only one changed class. It is a test class without any affected classes existed. Both FEST and ClassSRTS select this test class, but did not find any faults. Regarding the 5 failed test classes in CI testing, we carefully analyzed the code and found they all have no dependency with the changed class. We also looked for their test history from all previous versions, and found that these failed test cases have no test history. It indicates that (1) these failed test cases were committed in the earlier versions, but first ran in v259; (2) these failed test class have no dependency relation with the current Version and the fault should be detected in the earlier versions when the test code was submitted.

In v260, there is one changed class, and FEST searches to get 19 affected classes of which 11 are test classes but none failed. CI testing reports one failed test. Similarly, we carefully analyzed the code to confirm that it did not have any dependency relations with the changed and affected classes. We also found that the failed test not only invoked the classes of program code, but also called the testing code from the third party libraries. We guess that the failed tests may be caused by the third party libraries. V261 has the same situation as v260.

- **Section V: Projects and the number of versions in different categories**

When we evaluate our approach by projects, we first analyze the categories and number of the versions of each chosen project (shown in Figure 18). The size of bubbles refers to the number of versions for each project in different categories.

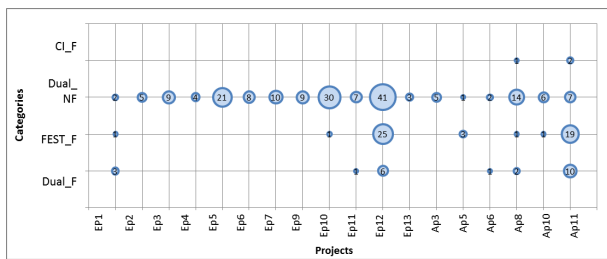


Fig. 18. The categories and number of the versions for each project