An empirical analysis of test input generation tools for android apps through a sequence of events

ABSTRACT

Graphical User Interface (GUI) testing of Android apps has gained considerable interest from the industries and research community due to its excellent capability to verify the operational requirements of GUI components. To date, most of the existing GUI testing tools for Android apps are capable of generating test inputs by using different approaches and improve the Android apps' code coverage and fault detection performance. Many previous studies have evaluated the code coverage and crash detection performances of GUI testing tools in the literature. However, very few studies have investigated the effectiveness of the test input generation tools, especially in the events sequence length of the overall test coverage and crash detection. The event sequence length generally shows the number of steps required by the test input generation tools to detect a crash. It is critical to highlight its effectiveness due to its significant effects on time, testing effort, and computational cost. Thus, this study evaluated the effectiveness of six test input generation tools for Android apps that support the system events generation on 50 Android apps. The generation tools were evaluated and compared based on the activity coverage, method coverage, and capability in detecting crashes. Through a critical analysis of the results, this study identifies the diversity and similarity of test input generation tools for Android apps to provide a clear picture of the current state of the art. The results revealed that a long events sequence performed better than a shorter events sequence. However, a long events sequence led to a minor positive effect on the coverage and crash detection. Moreover, the study showed that the tools achieved less than 40% of the method coverage and 67% of the activity coverage.