Chapter 56
An Effective Approach to Test Suite Reduction and Fault Detection Using Data Mining Techniques

B. Subashini
K.L.N College of Engineering, Madurai, India

D. Jeya Mala
Department of Computer Applications, Thiagarajar College of Engineering, Madurai, India

ABSTRACT

Software testing is used to find bugs in the software to provide a quality product to the end users. Test suites are used to detect failures in software but it may be redundant and it takes a lot of time for the execution of software. In this article, an enormous number of test cases are created using combinatorial test design algorithms. Attribute reduction is an important preprocessing task in data mining. Attributes are selected by removing all weak and irrelevant attributes to reduce complexity in data mining. After preprocessing, it is not necessary to test the software with every combination of test cases, since the test cases are large and redundant, the healthier test cases are identified using a data mining techniques algorithm. This is healthier and the final test suite will identify the defects in the software, it will provide better coverage analysis and reduces execution time on the software.

1. INTRODUCTION

Software testing is a process of executing a program or application with the intent of finding errors in the software. A test case is an identity which is associated with a program. The primary purpose of a test case is to find bugs. A potential drawback in testing is the creation of a large number of test cases, a test suite. Test cases should be created with high probability to uncover bugs. Testing the software or application with test suite takes an enormous amount of time in execution and it also increases the

DOI: 10.4018/978-1-6684-3702-5.ch056

computational effort of running the entire test suite. Complete software testing means every statement in the program and every possible path combination with every possible combination of data must be executed. In this paper, the combinations of test cases are created using IPOG_D algorithm with the help of parameters and constraints for the specific system (Lei, Kacker, Kuhn et al., 2008). Lot of test cases are generated using this IPOG_D algorithm and it may be redundant. It is necessary to identify irrelevant and redundant test cases and reduce it. For the reduction of test case, the feature selection is the first step for eliminating irrelevant attributes in the data set. Then classification technique is applied to find the accuracy of the testing application. Secondly, the test data is checked with the training data for the fault detection. The resultant and reduced test data is used to test the program and check the program for coverage analysis and execution time. By this usage of reduced test suite, the time, cost and effort for execution of the program may be reduced as because of the removal of redundant test cases by the mining technique. It may improve the effectiveness of software testing by the selection of effective test cases.

2. MATERIALS AND METHODS

2.1. Background

Software testing is an action to confirm the actual outcomes with the expected outcomes and assure that the product framework is without defect. Test Case is a cluster of activities executed to check a specific aspect or effectiveness of programming application. The objective of any product venture is to formulate test cases which meet client prerequisite. In this article, huge records of the test case are automatically generated by using the combinatorial testing method, it may be redundant and it is required to eliminate repeated test cases. Test suites are minimized and the faults are forecasted by using the classification technique. Reduction in test suite will minimize the time of execution, effort and it will provide better coverage analysis.

2.2. Combinatorial Testing

Pairwise testing is a combinatorial method of test cases in software testing, the input parameters for the system is tested with all possible discrete combinations of the values of parameters. The combinatorial testing focuses on t-way test data generation, where each test t contains a set of values for parameters. When the combinations of input parameters increase the testing will be effective with all possible combinations of values (Lei et al., 2008). Since, the effort of testing is based upon the input parameters, each and every parameter much contribute to a fault and the faults are caused by the interaction between the inputs, these possible combinations of test cases are produced with high-quality testing of coverage in the system.

Each and every program or a system contains a distinct series of behaviors; first, the input parameters for the system must be fully identified. The input values pass for the parameters is identified as second. Thus, the possible input values for each and every parameter for the system are identified and the values are passed. The values passed to the input parameter may be passed upon boundary value analysis, equivalence partitioning method or random value testing method. Each and every value for the parameter contains both the valid as well as invalid data.