International Journal of Computer Applications (0975 – 8887)
Volume 134 – No.5, January 2016

Amalgamation of Automated Test Case Generation Techniques with Data Mining Techniques: A Survey

Yogita Dubey
Department of CSE
Student M.Tech VI Semester
BUIT ,Bhopal ,India

Divakar Singh
Department of CSE
Head of Department
UIT,BU,Bhopal , India

Anju Singh
Department of CSE and IT
Assistant Professor
UTD, BU, Bhopal ,India

ABSTRACT
A testing module in the life cycle of a software development plays a crucial role for its development and its successful deployment using the defined cases. For this previous practitioner incorporated data mining techniques to reduce the number of test cases. During the software development process appropriate selection of unit tests is vital when many unit tests exist. Poor test selection may lead to Faults. This is true when the application is large and many developers are involved with some tests in which the developer can be unfamiliar and non-obvious relationships between application code and test code may be extant By the application of association rule mining and the unit test selection process and with extant selection techniques by comparison, Researchers facilitates a quantitative analysis of the advantages of heuristic and its disadvantages for the development where process patterns are stable. In test case generation technique data mining knowledge engineering area plays a vital role where the algorithm are capable of analyzing based on the pattern description such that the effective and accurate test cases can be generated, in this paper author presents a method to reduction the number of test suites by using mining methods thereby facilitating the mining from test cases. Here we are discussing about the Automated Test case Generation Techniques with Data Mining Techniques.

Keywords
Software Engineering, Software Testing, data mining, test cases, data item, test suite, Weka, item set, classifier, Cluster, Automated, and Redundancy.

1. INTRODUCTION
1.1 Software Testing
Testing is essential parameter of a software development where the usability and failure of the software due to minute disturbance can be observed and cure using this step, thus an important part of the phase to deploy and make success to a project is testing phase of a project.

A test case [1, 2] is a collection of multiple available parameters of the development project. Multiple inputs implemented and generated are to the development project. Such that a test can be covered the problems and parameters such as database attribute and input attributes entered by the user. Hence the output of test case is a pass/fail. A test suite [1, 2] is an automatic testing and generated the cases using which proficiency and dynamicity of the software can be computed. Redundancy is the recapitulation of data between test cases and the other. So process of test suite is indispensable to achieve on which lot of time will be able to save from executing redundant or not needed test cases. By the test suite the behavioral patterns exhibited helps us in this process of automation.

1.2 Definition of Automation
"The task executions which take part self, which reduces user burden to monitor, to detect and cure".

Automated Testing
1. Need to dart a set of tests repeatedly
2. Helps performing "compatibility testing"(on different configurations and platforms)
3. Long term costs are reduced
4. Possible to run regressions on a code that is continuously changing and in shorter time
5. It's more expensive to automate (bigger initial investments)
6. You will not be able to automate everything, still some tests have to be done manually

1.2.1 Automated Software Testing
Software testing [11, 12], automatic testing make avoidance of the technique which made by the user often in case. Some case by mistake also missed by the technique. The results may be automatically fed into a database that can facilitate useful statistics on how skillfully the software development procedure is going. How test cases [12] may be automatically generated from state charts. The application of the states produces the different test cases as solutions to a planning problem. The test cases can be used by automated or physical software testing on system level. Also the paper dispenses a method for the reduction of test suites by using mining methods facilitating the mining from test cases.

Automation [12] performs repeated and robust testing and case generation automatically as the test platform change it optimize accordingly.

A process which covers**:**
1. Software usage to control over the test cases and access.
2. Expected and actual output can be compared.
3. Test arrangement requirement setup phase.
4. Similar test control and testing task. "Software Testing is usually the portion of the project which makes grown men weep"
5. Ever-shrinking schedule and slightest resources .It involves automating a physical process of testing.

1.2.2 *Why automated testing?*

1. A time savings translates directly into cost savings
2. Improves testing productivity
3. Improves accuracy
4. Increases test coverage
5. Verification and Validation of the requirements of an organisation
6. Defects and defeats caused by misunderstood requirements or coding errors
7. Compatibility of the Software

The test case generation is the requirement of verifying the robustness of the program. The need for effective test automation adds to this problem. Data mining algorithms can be applied at different extent of abstraction and it helps the user locate more meaningful patterns. Data mining will design the patterns from the existing database. Using well-established data mining technicians, practitioners and researchers can cover the potential of this valuable data in order to organize their project and to produce better quality software systems that are delivered on time and within budget. Various available software case and testing [3] is a case where one of the essential areas of the SDLC. It is the scenario which is generally performed on large code to find out the defects in software and also to test the software for correct results. It aims at providing assurance to the client that our software works accurately at any circumstances. This requires building test cases that exploits each and every possible path of the software; it is very tedious to explore each and every possible test case manually. Automated test case generation [1] has already been started where test cases are built automatically. This generates thousands of test cases at a faster rate through a simple program. The problem with the above approach is if the software is built on thousands of lines of code, it takes a lot of time to know the output for execution of each test case and if the test suite is more the execution of the test suite [2, 4, and 9] may take even days to complete. Test suite contains machine generated test cases. It contains redundant test cases too. It deals with the above Issue.

1.2.3 *The Automated Test Lifecycle Methodology*

Fig- 1 below describes the life cycle method for the life cycle such that how the steps process with the input given such that the proper required steps and test cases can be generated. First of all the program is loaded and the decision making rules can be applied first to pre-process the program structure to recognize its content. Furthermore we load the pre-processed data into the aquatint and further move the data for the transaction. Thus upon applying the defined rules a test cases for particular program input can be get generated as per the steps defined in fig-1.

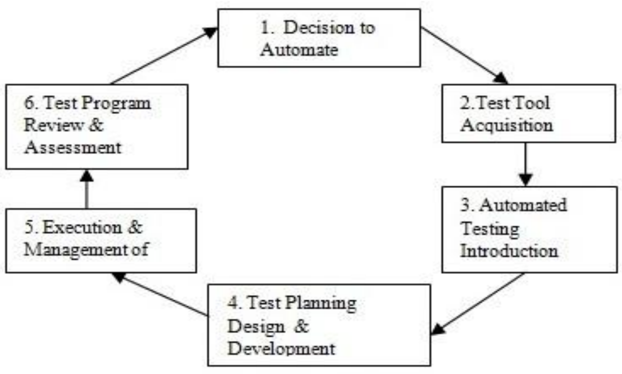
1. Decision to Automate
2.Test Tool Acquisition
3. Automated Testing Introduction
4. Test Planning Design & Development
5. Execution & Management of
6. Test Program Review & Assessment

Figure 1: Life cycle of Automated Testing methodologies

1.2.4 *Advantages and Disadvantage of Automated Testing*

Table 1: Advantage and disadvantages in automated testing technique scenario

Advantage	**Disadvantage**
Reliable: eliminating human error	High investment is needed in the tools and testing.
Reusable, Fast	High man power required for test preparations
Better quality software, cost reduction	A lot of test areas left uncovered

1.2.5 *Future Enhancements of Automated Testing*

- Tool needs to be "smarter"
- Self-healing – probes that ship with the software
- Determines an area that's buggy and help develop additional test cases
- Can detect sounds/flashes
- Image detection less sensitive

Know which areas of code changed only run those test cases applicable to that changed area at the system level.

2. LITERATURE REVIEW ON AUTOMATED TEST CASE GENERATION TECHNIQUES WITH DATA MINING TECHNIQUES

2.1 Automated testing with Association Rule Mining

In this paper[14] author described about the technique which is genetic algorithm for the test case generation technique where the technique use a concept of spanning tree, the concepts of spanning sets is used in this concept to set a bound to the number of test cases. Also, this technique overcomes the problem of the redundant test cases and guides the test case selection by concentrating only on the elements of the spanning set. The fitness function of the algorithm is use to find and fix the error outperform in the programming input which got an advantage of the algorithm. The genetic approach generates unique test sequences or scenarios until the errors are found. Thus the genetic algorithm performs and gives the best output from the input program and execution.

In this paper [15] author described the genetic algorithm to optimize the test cases that are generated using the category-partition and test harness patterns. They have performed the unit testing using category partition. In order to load the program and execute it from the drive they perform the usage of harness pattern. GA is used to cover all the paths of a graph-equivalent of Unit under Test (UUT). Optimal test suites are derived by using the method of sampling statistics. It is efficient approach of optimization using both genetic algorithm and sampling strategy.

Thus the strategy driven by them was efficient in mean of using two different individual techniques to load the dataset and to perform the genetic based test case generation. The fitness function has given a multiple roles in order to find best fitness of different node and test cases used in this paper. Furthermore author [16] proposed another parallel processing related concept TGen that generates test data based on the genetic approach but in the efficient way of using thread based parallel processing technique which further improves the execution time policy using the algorithm provided.

In this paper author [17] proposed an algorithm hill climbing based which is used in phases of test case generation. The usage of fitness function they performed average usage of three factors i.e. likelihood, close to boundary and branch coverage and analyze the results. Also they have worked on a hybrid approach which is called GA-NN i.e. combination of genetic algorithm and neural network. They used neural network as an estimator for the fitness of test suites. In paper [18] author described a new approach which is effective in its field to generate the test cases. They take two different methods local search and global search method. Local search uses the fitness function to evaluate possible moves within the search space from a single current solution point until a local optimal is reached. Further they performed branching operation which leads for the test case generation process.

In this paper author [18] represent a new algorithm which is CONCOLIC algorithm. Which is having two module of implementation which are Concrete and symbolic techniques. They perform the execution based on symbol and further more a concrete execution at a similar time to perform the execution of finding test case generation. Furthermore author given a new algorithm which performs an analysis on the basis of dynamic and static symbolic analysis which execute and produce exact test suite to explore execution paths of a target program. In this algorithm they also perform a stopping criteria where the different and multiple un-necessary iteration can be avoided using the technique.

2.2 Automated Testing with Classification

Generation of Classification Rules

Researchers provide the Software Requirements Specification to the classifier system. For classifying we use Weka. The Weka classifier is trained at the first with a training set. Later it is provided with the SRS. It classifies user requirements in functional and nonfunctional requirements by generating a classification rules. The classification rules are applied to the user requirements to get FR and NFR. We derive the state machine From NFR. State machines specify the behavior of a system/subsystem.

Generation of Test Cases:

Transformation describes from State diagrams in to Test cases. In computer Science State machines and state diagrams have a long history. Recent versions of UML include an expressive state diagrams concept. In the UML state machine formalism especially the abstraction mechanisms i.e. nesting of states and stubs, allow Researcher's to map all the important elements of use case documents to State machines.

From State Machines to Test Cases

For some time to derive test cases using state models has been common practice in the software testing world. The final goal of model-based testing is to automate the test case generation as much as possible. Researcher's approach from test models to generates a set of valid test sequences, where all preconditions transitions are established either by previous actions or by properties of the test data. The scope of Researcher's method on the test data is the generation of test sequences supplemented by constraints, as far as it can be derived from the state machine which presenting information. The method specified in [11] can be used to produce test cases from state machine diagrams which are as follows:

There are 3 main steps in test case generation, in the first step on a transition a predicate is selected from a UML state machine diagram. In the next step, the selected predicate will transform into a predicate function. In the third step, corresponding to the transformed predicate function test data is generated. The generated test data are stored for use with an automatic tester. Particular predicate are determined once the test data corresponding to a particular predicate, the steps are repeated until the next predicate is selected on the state machine diagram. The process is repeated on the state machine diagram until all Predicates have been considered.

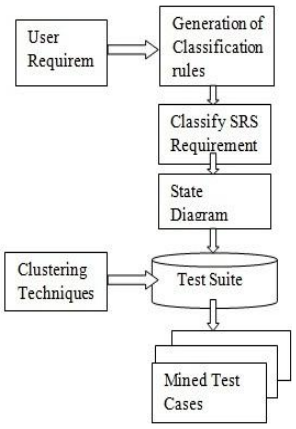

Figure 2: Automatically generated test cases from software requirement specification mining system.

2.3 Automated Testing with Clustering:

Cluster analysis [19] an important domain of data mining, groups the data into meaningful groups/ clusters based on the information found in the data. The aim is that the objects in a cluster are related to one other and separate or different from the objects in other groups or clusters. Based on these distance measures, over the years different number of clustering techniques has been proposed. The most commonly used techniques are Hierarchical clustering (or nested) and Partitioned clustering (or Unnested) techniques. Hierarchical techniques produce a nested sequence of partitions, with a single inclusive cluster on the top and at the bottom singleton clusters of individual points. In Partition technique create a one level (Un-nested) partition of the data points into a desired number of clusters at once. Here, in this study, K-Means Partitioned clustering technique is implemented to group the objects or test cases in the desired number of clusters.

The *k*-Means method is considered one of the simplest and most classical methods for data clustering and is also perhaps one of the most widely used methods in practical implementations because of its simplicity. The decision of K is very important in K-means Algorithm because they yield different results based on the location of K initial centroids The K-Means approach initially treats a data set as a single cluster and proceeds by partitioning it into the K‟ specified number of clusters. The algorithm seeks to minimize the sum of the squared distance of a point from the center of its cluster. Finally the algorithm stops when the centroids do not change.

Researchers [19] have implemented the K-Means Algorithm to partition the input domain or test suite of sample module into a desired number (K) of partitions. The intuition behind the approach is that, the given test suite in Figure shown in comparative analysis Table is to be partitioned into a specific number of clusters in order to reduce the size and eliminate the redundant test cases. Here, we have specified the value of K=4 in order to group the test cases according to a coverage criteria of the sample code. A coverage criterion is an important factor while applying any reduction techniques because it gives the measure of structural components executed by a test suite. The test suite reduction is guided by the code coverage criteria defined in order to select a test suite that will give 100% code coverage. There are many ways to measure code coverage; like statement coverage, branch coverage, decision coverage, multiple decisions / condition coverage, loop coverage and path coverage. So it very important to consider any of the above code coverage criteria while reducing the test cases. A selected test suite after reduction should be able to give 100% code coverage; otherwise the faults will remain hidden in other parts or components of software under test.

In below table 1, the comparative study is shown between different data mining technique for test case generation , where the sample taken dataset, their real time application possibility and complexity is been mentioned.

3. COMPARATIVE ANALYSIS OF AUTOMATED TEST CASE GENERATION TECHNIQUES WITH DATA MINING TECHNIQUES:

Table-2: Comparison Analysis table

TECHNIQE-S	DATASET	REALTIME APPLICATION	COMPLEXITY / GENERATION OF TEST CASES.
Genetic Algorithm	ATM withdrawal transaction, < t0, t1, t3, t8, t9, t10, t11, t12 >	ATM transaction application.	High/ GOOD
Genetic algorithm with fitness and harness value.	Triangle classifier, A remainder calculation program, Linear search program, Binary search program	Real world program testing.	Intermediate/ MEDIUM
GA-NN(Hill-climbing based algorithm)	They have taken example of finding gcd of two numbers program	Solving Small program test case generation problem.	Low/GOOD
CONCOLIC	They have used the programs that use yacc and lex to describe their inputs, the programs are bc,logictree,cuetools,lua,wuftpd	Real time application.	Intermediate/ Best
K-Means Algorithm	Segment of code, any programming language code to determine the test cases.	Performing testing with huge program, it outperforms normal testing.	Low/GOOD
Association rules mining	An artificial dataset, which consist of 8 item set.	Large multi-layered software application.	Less/GOOD

4. CONCLUSION

Previous Researchers discussed different problems in relation of software testing and the test suite size. They discussed about the automated test case generators and the redundancies they incorporate. They took help of data mining and proposed a new technique in software testing which reduced the size of the test suite significantly. They tested their reduced test suite for coverage.

They proposed future works of these lines of research. So deployment of testing techniques with this methodology significantly reduced time and effort spent in executing thousands of test cases. A new approach to automatically generate test cases from user requirements and mining of test cases has been discussed by Researcher’s. Firstly a formal transformation of a detailed user requirements to a UML state model, secondly the generation of test cases from the state model and lastly Mining of Test cases. In future we will be focusing on Automated Test Case Generation with Data Mining Techniques. Previous Researchers have discussed the above procedures and further enhancements regarding can be made which are discussed below. In order to further perform our continuation with the work we are going to take efficient optimize technique which may perform better with some parameter such as computation time, number of test case generated, percentage of branch coverage, range of the input variables. Further technique, which may apply with the classification such as Ant colony, recursive ant colony optimization technique to obtain further automated test case generation.

5. REFERENCES

[1] Ajitha Ranjan.”Automated Requirements-Based test case Generation”. Communications of ACM, 2006

[2] T. Y. Chen and M. F. Lau. A new heuristic for test suite reduction. Information and Software Technology, 40(5):347-354, 1998.

[3] David Alex Lamb, “Software Engineering, planning for change,” Prentice Hall, Englewood Cliffs, NJ 07632, pp. 109–112, 1988.

[4] M. J. Harrold, R. Gupta, and M. L. Soffa. A methodology for controlling the size of a test suit. ACM Trans. on Soft.Eng. And Meth, 2(3):270-285, 1993.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn. A Data clustering: review. ACM Computing Surveys, 31(3):264–323, 1999.

[6] Lilly Ramesh, “Knowledge Mining of Test Case System,” International Journal on Computer Science and Engineering Vol.2 (1), 2009, 69-73.

[7] Mark Last and Menahem Friedman.”The Data Mining approach to automated software testing.”.Communications of ACM, 2003.

[8] Martina marre and Antonia Bertolino, “using spanning sets for coverage testing”. IEEE transactions on software Engineering, vol.29.

[9] Remco R. Bouckaert.’weka Manual 3-6-1”. Software manual,June 4,2009,pp-11-14.

[10] Zhenyu Chen and Baowen Xu.”A novel approach for test suite reduction based on requirement relation contraction”.Communications of ACM, 2006.

[11] P. Samuel R. Mall A.K.Bothra “Automatic test case generation using unified Modeling language (UML) state diagrams” Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, West Bengal, India E-mail: philips@cusat.ac.in

[12] Vasilache S., and Tanaka J., "Synthesis of State Machines from Multiple Interrelated Scenarios Using Dependency Diagrams," Journal of Systemics, Cybernetics and Informatics, Vol.3, No.3, 2006.

[13] Lilly Raamesh and G.V. Uma “Reliable Mining of Automatically Generated Test Cases From Software Requirement Specification (SRS)” IJCSI Vol. 7, Issue 1, No. 3, January 2010.

[14] Abdelaziz M Khamis, Moheb R Girgis, and Ahmed S Ghiduk. Automatic software test data generation for spanning sets coverage using genetic algorithms. Computing and Informatics, 26(4):383–401, 2012.

[15] [MR Keyvanpour, H Homayouni, and Hasein Shirazee. Automatic software test case generation. Journal of Software Engineering, 5(3):91–101, 2011.

[16] Roshni Rajkumari and BG Geetha. Automated test data generation and optimization scheme using genetic algorithm. In Proceedings of International Conference on Software and Computer Applications (ICSCA 2011), 2011.

[17] Dharmalingam Jeya Mala, Elizabeth Ruby, and Vasudev Mohan. A hybrid test optimization framework-coupling genetic algorithm with local search technique. Computing and Informatics, 29(1):133–164, 2012.

[18] R. Pandita, T. Xie, N. Tillmann, and J. de Halleux, “Guided test generation for coverage criteria,” in Software Maintenance (ICSM), 2010 IEEE International Conference on, pp. 1–10, IEEE, 2010.

[19] Fayaz Ahmad Khan, Dr. Anil Kumar Gupta, Dibya Jyoti Bora Department of Computer Science and Applications, Barkatullah University Bhopal, (M.P) “An Efficient Approach to Test Suite Minimization for 100% Decision Coverage Criteria using K-Means Clustering Approach” IJAPRR ISSN (2350-1294) Vol. II, Issue VII, 2015

[20] Parag Deoskar , Dr. Divakar Singh and Dr. Anju Singh,“An Efficient Support Based Ant Colony Optimization Technique for Lung Cancer Data ,” International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 9, September 2013.

[21] Mrs. Nidhi Sing ,Dr. Divakar Singh , “The Improved K-Means with Particle Swarm Optimization” Journal of Information Engineering and Applications ISSN 2224-5782 (print) ISSN 2225-0506 (online) Vol.3, No.11, 2013.

[22] Surendra Kumar Chadokar , Dr. Divakar Singh and Ashutosh Singh , “Optimizing Network Traffic by Generating the Association rules using hybrid apriori genetic algorithm, “ 10th IEEE International Conference on Wireless and optical Communication Netwaork 2013

[23] Nidhi Singh , Dr. Divakar Singh, “The Improved K-Means with particle Swarm Optimization , Journal of Information Engineering and Application.