ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 3, Issue 1, pp: (98-106), Month: April 2016 – September 2016, Available at: www.paperpublications.org

# A Hybrid Density Based K Mean for Test Case Prioritization using Prism Algorithm

[1]Meena kumari, [2]Shivi Sharma

[1, 2]Deptt. Of CSE, L.R.I.E.T Solan (HPTU), India

*Abstract:* **Software testing is important phase of software development life cycle which ensures the developer that the developed software works according to specifications or not. Prioritization techniques that incorporate a clustering approach and utilize code coverage, code complexity and history faults as well to increase the effectiveness of the prioritization. To make testing efficient and effective a techniques of test case prioritization are used. An efficient Test case prioritization technique reduces the cost of testing and fault detection capabilities of testing. Results show that test case prioritization that utilizes a clustering approach can improve the effectiveness of test case prioritization techniques.**

*Keywords:* **Software Testing, Regression Testing, Test case prioritization, Test Case Prioritization Technique.**

## 1. INTRODUCTION

### 1.1 Software Testing:

Is done with an attention to find out the defect in existing software Software testing has been proven that testing, analysis and debugging costs usually consume over 50 % of the costs associates with the development of large software system.[1] Software testing is important phase of software development life cycle which ensure the developer that the developed software works according to specification or not. In other words software testing is verification and validation process of computer program or application/product to meet the requirements that guided its design and development, works as expected , can be implemented with the same characteristics , and satisfies the need of stakeholder . A defect is inconsistency among the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development phase.[2] Software testing is one element of a broader topic that is often referred to as verifying and validating that a software application or program. Software testing is useful for finding the defects, fundamental weakness in the application code that must be improved or checked.

**Software testing has three main purposes: verification, validation, and defect:**

- **Verification:** The process of verification confirms that software meets its specifications. It ensures that software correctly implemented for specific function
- **Validation:** Whereas the process of validation ensures that the software meets the business requirements. It provides the traceable activities to customers.
- **Defect:** A defect is inconsistency among the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development phases.[3]

### 1.2 Challenges of Software testing:

Testing may focus on the bug prevention as a part of quality assurance. To prevent our system from bug is testing first goal. There are many challenges in the software testing which is faced by software tester during the testing time. Both the manual testing and automation testing have many different challenges. These are the following challenges given below:

- Testing is always done on sampling basis. To test the whole system is never possible at one time i.e. complete testing. During test data selection methods, selection of test data needs good proficiency.
- Most of testers and their management are incognizant of training sources for testing which leads lack of testing knowledge to the tester. This is the biggest challenge for software tester.
- Due to lack of understanding of test tools.
- Due to management focuses on product not the process which leads to lack of management support and involvement of testing.
- Due to lack of customer involvement during testing.
- Due to lack of communication, requirements are not properly defined. This leads the problem to the software tester.
- Communication problem occurs when one test team work under multiple projects.
- Regression testing becomes challenge as project keep on growing. To overcome from this condition, both the testing and development teams need to work together.

### 1.3 Methods of Software Testing:

Software Testing consist different methods, which are white box testing, black box testing and gray box testing [4]

- **White box testing:** is also called glass testing, open box testing, structural testing and transparent box testing. In white box testing, the tester needs to have knowledge of the internal working of the code. Optimization of code and extra lines of code can be removed which can bring in hidden defects. But it is difficult to maintain white box testing as the use of specialized tools like code analyzers and debugging tools are required. Bugs can often be found before they cause trouble; so, this testing extremely effective in detecting the problems, resolving problems.
- **Black box testing:** is testing software based on output requirements and without any knowledge of the internal structure or coding in the program. Typically, when performing a black box testing, a software tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. [9]
- **Gray box testing:** is defined as testing software while already havingsome knowledge of its underlying code or logic. Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications. Testing is done from the point of view of the user and not the designer.

### 1.4 Techniques of Software Testing:

We performed two techniques of software testing in software development life cycle. These are:

- **Manual Testing:** In manual testing, tester takes over the role of an end user and test the software to identify any un-expected behaviour or bug. This kind of testing performed manually without using any automated tool or any script. Manual testing used by software developer to run test manually.
- **Automation testing:** Automation testing is also known as test automation. In this type of testing, tester writes scripts and uses software to test the software.This method takes automation tool support to execute the test caes is known as automation testing.Automation testing good for large project.
- In software testing there are many type of testing are used. The one most important type is regression testing.

### 1.5 Types of Software Testing:

- **Unit Testing:** Unit testing is testing of individual unit or group of related units.It falls under the class of white box testing. This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases.
- **Integration Testing:** Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. In this group of component are combined to produce output.Also interaction between hardware and software is tested in integration testing .If software and hardware have any relation .This type of testing falls under black box testing.

- **System Testing:** System testing tests the system as a whole. Once all the components are integrated the application as a whole is tested rigorously to see that it meets the specified quality standards.This type of testing is performed by a specialized testing team.System testing is the first step in the SDLC where the application is tested as a whole. In this by putting the software in different environments it still works.System testing is done with full implementation and environment.

- **Stress Testing:** Stress testing mean how system behave under unfavourable condition.Stress Testing includes the behavior of a software under abnormal conditions.

- **Performance Testing:** is the testing to assess the speed and effectiveness of the system and to make sure it generating result within a specified time as the performance requirement.

- **Regression Testing:** Regression testing is testing after modification of system component or group of related units to ensure that modification is working correctly and is not demaging or imposing other modules to produce unexpected results.

- **Acceptance Testing:** This is the most imp type of testing as it is conducted by the Quality Assurance Team Who will gauge whether the application meets the intended specification and satisfies the clients requirement.Acceptence testing is often done by the customer to ensure that the delivered product meets the requirement and works as the customer expected.

- **Usability Testing:** Usability is a black box technique and is used to identify any error and improvements in the software by observing the uses through their usage the operation.

- **Alpha testing**:This test is the first stage of testing and will be performed amongst the teams .Unit testing , integration testing and system testing when combined together is known as alpha testing.

- **Beta Testing:** Which is done in end user. A team outside developments or publically releasing full perversion of the product Which is known as beta version .The aim of beta testing to cover unexpected errors.

### 1.6 Regression Testing:

Regression testing is testing after modification of system component or group of related units to ensure that modification is working correctly and is not demaging or imposing other modules to produce unexpected results.

Regression testing is process of retesting the modified software and ensures that new error does not introduce into the previously tested source code due to these modifications. Regression testing is very expensive testing process. In order to decrease the cost of regression testing the software tester may prioritize the test case so that the test case which are more important as run earlier during regression testing process

**Regression Testing Goal:** Testing modified software to ensure that changes are correct and do not adversely affect other parts of the software.

- Make use of existing test cases developed for previous versions of the software

- May have to create new test cases as well

**Why is Regression Testing Important?**

- Software is buggy

- Software is modified over time

**Need of regression testing:-**

- When there are changes in requirements and developer may modify the code according to the requirement.

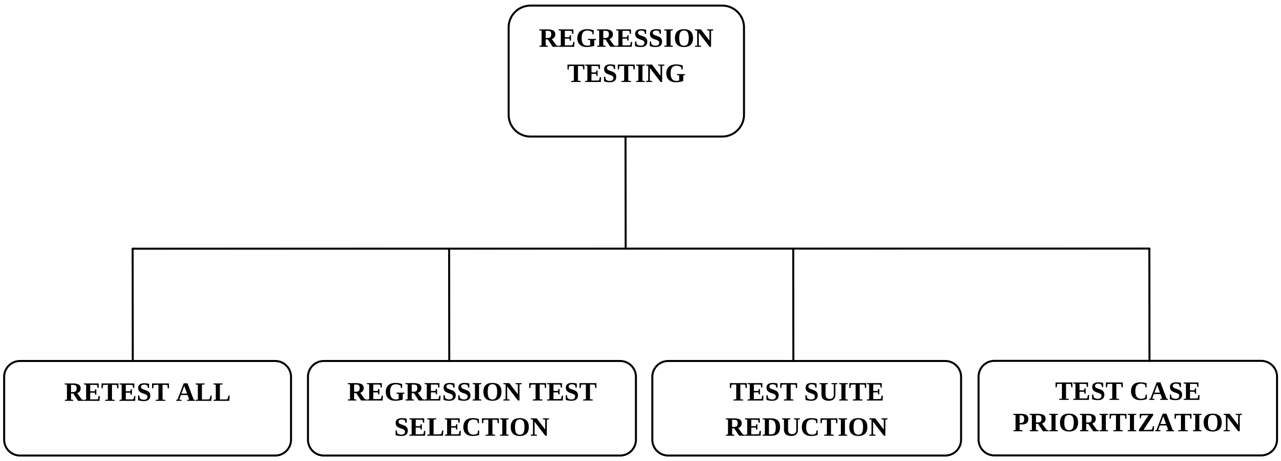
- For defect fixing.

- New feature is added to the software.

- Performance issue fix.

**1.7 Techniques of Regression Testing:**

- Retest All
- Regression Test Selection
- Test Suite Reduction
- Test Case Prioritization



- **Retest-all**: It is very expensive because it requires huge time and resource. In this method, the entire test suite in existing test bucket or suite should be re-executed. Those test cases that no longer apply to modified version of program are discarded and all the remaining set of test cases is used to test the modified program. It is the most straightforward approach of regression testing. In this simply execute all the existing test cases in the test suite.[5]
- **Regression Test Selection-** Retest all technique takes time and effort as all test cases are used to test the program again, so may be quite expensive. This technique much better as it uses information about program, modified program, test cases to select subset of test cases for testing.[6]
- **Test Suite Reduction** – This technique uses information about program and test suite to remove the test cases, which have become redundant with time, as new functionality is added. It is different from Regression test selection as former does not permanently remove test cases but selects those that are required. Advantage of this technique is that it reduces cost of validating, executing, managing test suites over future releases of software, but the downside of this is that it might reduce the fault detection capability with the reduction of test suite size[6]
- **Test cases Prioritization:** In this method, selection of test cases based on priority will greatly reduced the regression test suite. In this technique each test case is assigned a priority. Priority is set according to some criterion and test cases with highest priority are scheduled first For example criterion may be that the test case which has faster code coverage gets the highest priority. Advantage to previous techniques is that it doesn’t discard or permanently remove the test cases from test suite. Another criterion may be rate at which fault is detected[1]

**1.8 Introduction to Test Case Prioritization:**

Test case prioritization techniques schedule test cases in order to increase their effectiveness according to some criterion. Test case prioritization concerns with the identification of the perfect ordering of test cases. The purpose of this technique is to meet some performance goals like rate of fault detection, increase the effectiveness etc. Rate of fault detection is used to measure how rapidly faults are detected within the testing process, which provide feedback on the system under test and let software engineers begin correcting faults earlier than might otherwise be possible. The main purpose of prioritization will be minimizing the test suits.[7]

**There exist different types of test case prioritization methods:**

- **Code-based test case prioritization:** In code-based test case prioritization, source code of the system is used to prioritize the test cases. Most of the test case prioritization methods are code based.[8]
- **In model-based test case:** Prioritization a system's model is used to prioritize the test cases. System models are used to capture some aspects of the system behavior. The model based test case prioritization may improve the early fault detection as compared to the code-based test case prioritization. Model-based test prioritization may be an inexpensive alternative to the existing code-based test prioritization methods. However, model-based test case prioritization may be sensitive to the correct/incorrect information provided by the testers/developers. Hence model based test case prioritization is the best one compared to code based test case prioritization.[8]

**1.9 Classification based on the characteristics of the prioritization algorithms-**

- Based on customer requirements
- Based on coverage
- Based on cost effective
- Based on chronographic history

1. **Based on customer requirements**-In these techniques various customer requirement factor are considered. Assign some weight to these factors. Test cases having high weight value are executed first and test cases with low weight value are executed later.
2. **Based on coverage-** For detecting faults earlier in testing, we have to achieve more coverage. These techniques test internal structure of data and may be consider as white box testing.
3. **Based on cost effective**-There are much kind of cost related to test cases like cost of analysis and cost of prioritization. In cost effective based techniques test cases are ordered for execution based on cost.

Leung and white propose a cost modal that compare the various regression strategies. They divide the total cost into two parts

- Direct cost
- Indirect cost

**Direct cost includes**

System analysis cost Ca

Test selection cost Cs

Test execution cost Ce

Result analysis cost Cr

**Indirect cost includes**

Overhead cost

Tool development cost

One disadvantage of this technique is that they did not include the cost of undetected faults. They used the cost factors like

- $Ca\ (T)$ cost of analysis
- $Ce(T)$ cost of execution
- $Cc(T)$ cost of result checking
- $Cs(T)$ cost of selection
- $Cm\ (T)$ cost of maintenance of the test suite

In experiment for test case prioritization they consider two factors cost of analysis and cost of prioritization $Cp\ (T)$. In their work they divide the testing process in two phases one preliminary and second critical phase. These two phases have different costs. The result shows that the optimal ordering, total function coverage and additional function coverage have maximum saving.

**4. Based on chronographic history-**

In these type of prioritization techniques test execution history considered to be the main factor for prioritization of test cases. Jung-Min-Kim and Adam Porter proposed a history based test case prioritization technique. It is for regression testing in resource constrained environment. Their main motive behind this is to show that historical information can be useful for decreasing the cost and it may be useful increasing the effectiveness of testing process. [10]

**Goals:** Following are possible goals of prioritization:

- To increase the rate of fault detection of a test suite.
- To increase the coverage of coverable code in the system under test at a faster rate, so proving a code coverage criterion to be met earlier in the test process.
- To increase their confidence in the reliability of the system under test at a faster rate.
- To increase the rate at which high risk faults are detected by a test suite, thus positioning such faults earlier in the testing process.

## 2. METHODOLOGY AND PROPOSED WORK

**2.1 Proposed Methodology:** It is proposed to use a novel density based K-means algorithm for test case prioritization for regression testing. The naive k-means algorithm partitions the dataset into ‘k’ subsets such that all records, from now on referred to as points, in a given subset "belong" to the same center. Also the points in a given subset are closer to that center than to any other center. The algorithm keeps track of the centroids of the subsets, and proceeds in simple iterations. The initial partitioning is randomly generated, that is, we randomly initialize the centroids to some points in the region of the space.

Clustering error measures the same criterion and is sometimes used instead of distortion. Infect the points minimizes the distortion for the points in the cluster. Also when another cluster center is closer to a point than its current cluster center, moving the cluster from its current cluster to the other can reduce the distortion further. The above two steps are precisely the steps done by the k-means cluster. Thus k-means reduces distortion in every step locally. The k-Means algorithm terminates at a solution that is locally optimal for the distortion function. Hence, a natural choice as a convergence criterion is distortion. Among other measures of convergence used by other researchers, we can measure the sum of Euclidean distance of the new centroids from the old centroids. In this thesis we always use clustering error/distortion as the convergence criterion for all variants of k-means algorithm.

**DBKMEANS Clustering:**

DBKmeans clustering algorithm is a combination of DBSCAN and K-mean clustering. This algorithm performs better then DBSCAN when handling clusters of circularly distributed data points and slightly overlapped clusters. The criteria for splitting or joining a cluster can be decided based on the number of expected points in a cluster or the expected density of the cluster (derived by using the number of points in a cluster and the area of the cluster). There is lot of scope for the DBKmeans clustering algorithm in different application areas such as medical image segmentation and medical data mining. Basically DBKmeans clustering algorithm overcome the drawbacks of DBSCAN and K-means clustering algorithms.

## 3. RESULT AND DISCUSSION

**3.1 EXPERIMENTAL SETUP:**

Our whole experiments are performed in MATLAB framework, which is used for prioritizing the test cases. Our main aim is to increase the rate of fault detection within less time by prioritizing the test cases. For this, we used novel Density

Based K-means clustering algorithm for test case prioritization. Firstly, we implement K-means algorithm in MATLAB framework. Then, improve the density in this algorithm by applying DBSCAN algorithm. After that Density Based K-means used the standard data set to make the clusters on the bases of their density.

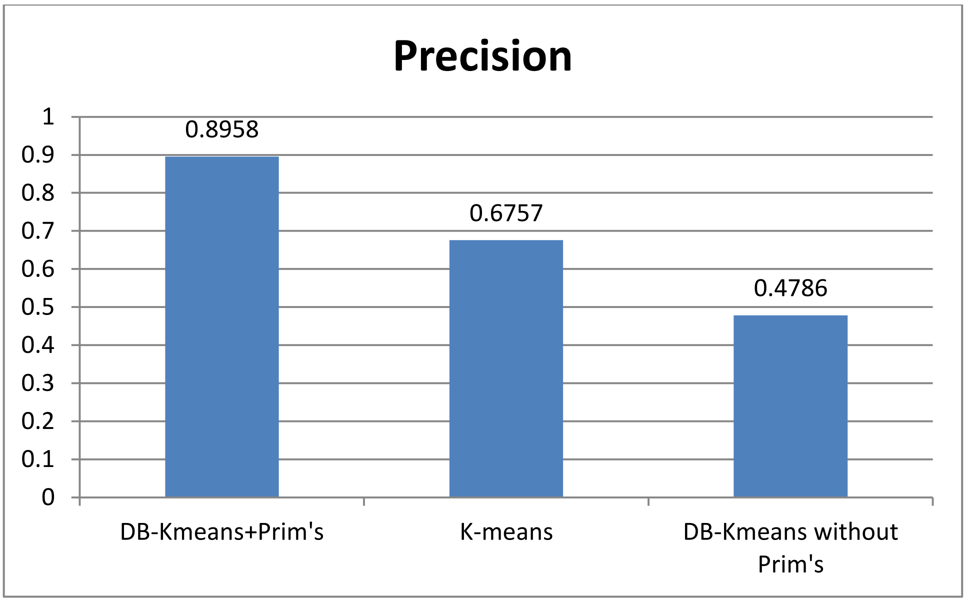
**Parameter metrics used:**

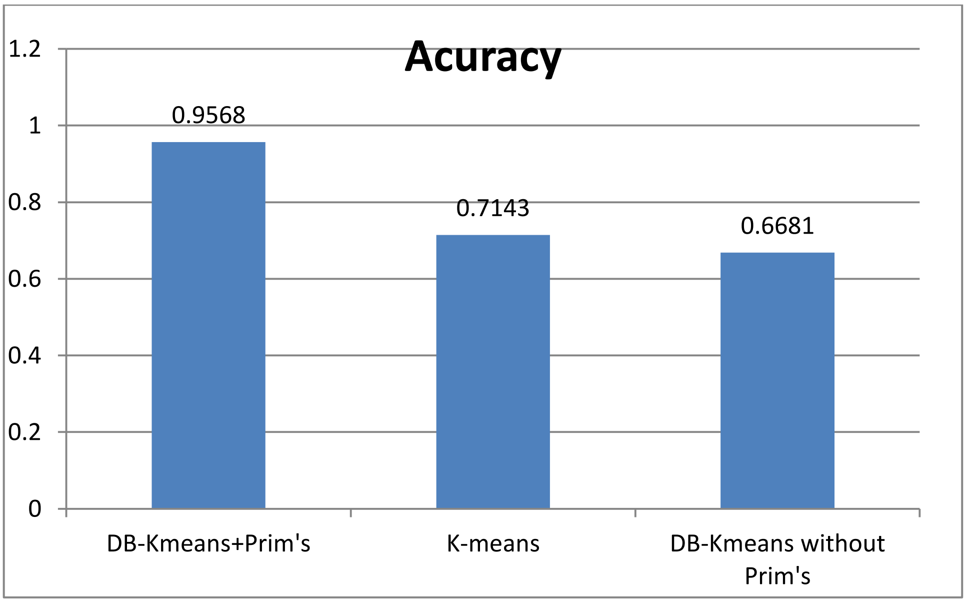
**Prim's algorithm:**

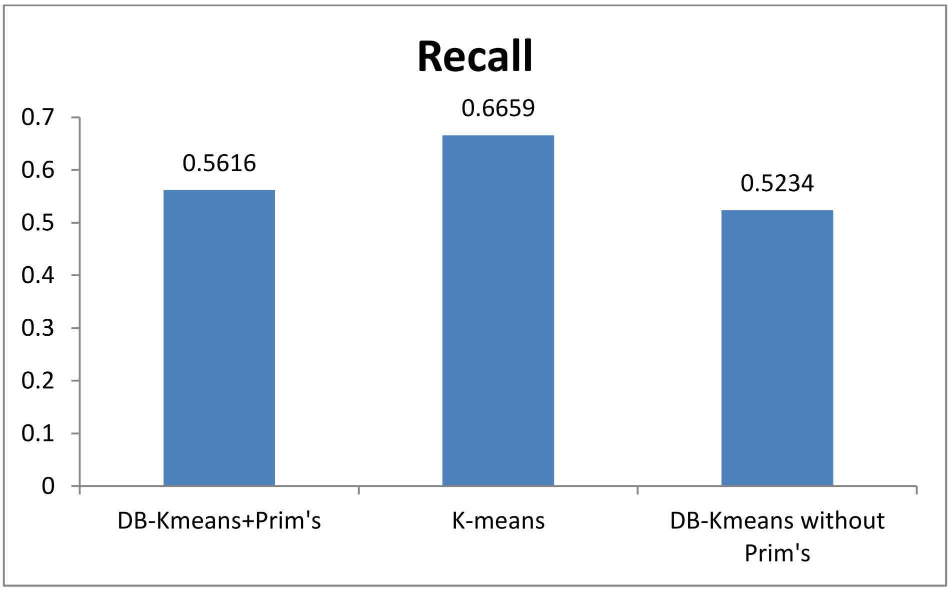
It is a Greedy algorithm which used the concept of minimum spanning tree for connected weighted undirected graph. Firstly, initialize a tree with single vertex which is randomly selected from the cluster. Then, develop the tree by one edge that connect tree to vertices not yet in the tree by calculating the minimum-weight edge and add it to the tree. This process is continuously repeated until all the vertices in the tree are connected with minimum weighted graph. The given formula is: Number of edge= Numbers of Nodes – 1

The idea behind selected the Prim's algorithm is to connect all the vertices to make spanning tree. This helps us to make select the test cases which cover the maximum coverage with minimum time.

**3.2 RESULT AND COMPARISION:**







## 4. CONCLUSION AND FUTURE WORK

The thesis aims at test case prioritization of test cases such that the testing effort reduces significantly while the code coverage remains almost same. This is achieved by using clustering approach such that the test cases are selected from each cluster thereby ensuring uniform distribution of code coverage. A novel density based K-means Clustering is utilised in this thesis and the sublist of test cases are found by using a minimum spanning tree using Prim’s algorithm.

The results are shown to be effective and it is found that although there is a significant reduction in the total number of test cases, the code coverage is almost same. This is desired as the total test cases represent the testing effort. The testing effort is reduced with minimum effect on performance.

There are numerous applications of the test case prioritization and can be utilised in all software industry. There is a large scope of improvement in the proposed technique. Other clustering algorithms can be applied and tested for efficacy. A hybrid of various other TCP techniques can be designed and tested for efficacy. Other test case parameters can be used for evaluation of results. To reduce this cost by prioritizing test cases and running the tests for the selective test cases as per the available time and manpower. There are a number of test cases available which can consume a lot of time and effort. A selective number of test cases needs to be selected which would be otherwise used for the same purpose. The priorities of the test cases need to be decided on the basis of several parameters. The parameters for the test case prioritization need to be chosen and a model needs to be developed which would set priority among the test cases. First of all a data set needs to be generated which would be utilized for our proposed algorithm testing. Then the dataset needs to be pre-processed for outlier removal and redundancy removal. Then a technique for clustering of the test cases needs to be developed which would be utilized for the above mentioned problem. The parameters has to be selected such that the code coverage remains almost the same while there must be significant reduction in the number of test cases which implies a reduction in the total testing effort.

## REFERENCES

[1] Siripong Roongruangsuwan, Jirapun Daengdej, **“**Test Case Prioritization Techniques”, *Journal of Theoretical and Applied Information Technology, 2010.*

[2] Medhun Hashini D.R, “Clustering Approach to Test Case Prioritization Using Code Coverage Metric”, *International Journal of Engineering And Computer Science*, Volume 3 Issue 4, 2014.

[3] Ashima Singh, “ Prioritizing Test Cases in Regression testing using Fault Based Analysis”, *International Journal of Computer Science Issues,* Vol.9, Issue 6, No 1, 2012**.**

[4] Kumar, Mukesh. "An optimized farthest first clustering algorithm." InEngineering (NUiCONE), 2013 Nirma University International Conference on, pp. 1-5. IEEE, 2013.

[5] K.Sandya "Test Case Prioritization Using Clustering Approach And Multiple Modular Based Approach" *International Conference* 2012.

[6] Praveen Ranjan Srivastava, "Test Case Prioritization" *Journal of Theoretical and Applied Information Technology, 2008.*

[7] Di Nardo, Daniel, Nadia Alshahwan, Lionel Briand, and YvanLabiche. "Coverage-based test case prioritisation: An industrial case study." In Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on, pp. 302-311. IEEE, 2013.

[8] Sanjukta Mohanty, Arup Abhinna Acharya, Durga Prasad Mohapatra, Sanjukta Mohanty, "A Survey On Model Based Test Case Prioritization", *International Journal of Computer Science and Information Technologies*, Vol. 2 (3), 2011.

[9] Liu, Wenhong, Xin Wu, WeiXiang Zhang, and Yang Xu. "The research of the test case prioritization algorithm for black box testing." In Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on, pp. 37-40. IEEE, 2014.

[10] Monika, Ajmer singh "Test Case Prioritization: A Review*", International Journal of Engineering Research & Technology* Vol. 3 Issue 5, 2014.