Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TEST CASE COVERAGE ANALYSIS USING CLUSTERING METHODS FOR TEST CASE MINIMIZATION IN SOFTWARE TESTING

SANJAY SHARMA[1], JITENDRA CHOUDHARY[2], GOVINDA PATIL[3]

[1]Research Scholar, Department of Computer Science, Medicaps University, Indore
[2]Associate Professor, Department of Computer Science, Medicaps University, Indore
[3]Assistant Professor, Department of Computer Science, Medicaps University, Indore
[1]sanjaysharma1074@gmail.com, [2]jitendra.scsit@gmail.com, [3]govinda.patil@medicaps.ac.in

ABSTRACT

In test case minimization especially in software testing large-volume datasets are minimized the purpose is to separate unnecessary and duplicate datasets. The coverage of minimized datasets will be the same as original data sets. At present thousand techniques are available for dataset minimization and clustering is one of the main. The objective of the proposed paper is to justify different clustering methods, test case coverage and their role of in software testing, apart from the reason for choosing particular methods. As initial stats all clustering methods require data sets, data sets are generated by programmers or by automation tools. Duplicate data sets are the by-product of automation tools. This paper is based on an analysis of different test case coverage and test case minimization especially using clustering methods.
Keywords: *Test Case Minimization, Coverage Analysis, Data Mining, Clustering, Test Case Coverage*

1. INTRODUCTION

In testing phase of product development various test cases are generated for structural and functional checking of each unit and module. For result oriented and quality product it is necessary to optimize test cases and satisfy different coverage criteria like code coverage, statement coverage, condition coverage etc. At present test cases are optimized by different techniques, clustering is one of them. [24] [22]

In the field of computer science, clustering is a method of data mining that puts related data items in one group according to shared traits and attributes. It is the technique of organizing a collection of items into groups (called clusters) based on how similar the items are to one another. The patterns of clusters are based on collected data sets. The existence of core and outliers in the data has a significant impact on the clustering process. You must bring all of your characteristics to the same scale because the Euclidean distance is the distance metric utilized in the clustering process. [23] [27]

Clustering is a class of unsupervised machine learning techniques, groups together comparable data points according to a predetermined distance or similarity metric. Numerous fields, such as data mining, pattern recognition, image analysis, and bioinformatics, heavily rely on these techniques. Without labelled samples, clustering facilitates the discovery of underlying structures, relationships, and patterns within data. [1] [25]

The rest of this paper is organized as follows. Section II describes the literature review of papers related to the clustering method, used for minimization. Section III formulates the research analysis based on research questions and uses clustering methods, tools, and specifications. Section IV concludes our study and discusses possible future research directions.

1.1 Relation of Machine Learning and Data Mining

Data mining techniques [3] are part of machine learning where these techniques are further classified into supervised and unsupervised data mining techniques. Supervised contains classification and regression while unsupervised

contains clustering and association techniques. Clustering techniques are classified into partitioned, hierarchical, density, and other techniques. [21].

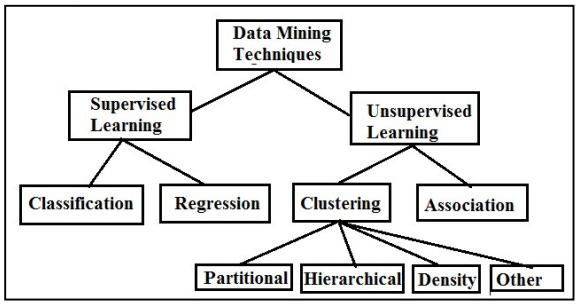

Figure 1: Data Mining Techniques

1.2 How Clustering Supports Data Minimization

One crucial component of data mining is data clustering [2]. An unsupervised machine learning technique called data clustering makes it possible to create logical groups of data from which patterns can be extracted. The idea is that items within a group should differ from (or be dissimilar from) those in other groups while still being similar to (or associated with) one another. When it comes to clustering, the objects within a cluster have low intra-class similarity and high inter-class similarity [3].

Among different minimization or optimization methods, clustering is also used for the test cases or data minimization. In clustering generation test cases are based on different conditions. Next test cases are converted into report format and then applied valid clustering method. The clustering method crates clusters and each cluster data may have some similarity but dissimilarity with other clusters. Similarity represents data redundancy and removing redundancy results in minimization. [20].

1.3 Importance of Test Coverage

Test coverage is important in software testing because it:

1. Ensures thorough testing: Test coverage metrics help ensure that the testing process is complete and covers all parts of the code.
2. Identifies gaps: By determining test coverage, teams can identify areas of the code that are not adequately tested and prioritize additional testing efforts.
3. Reduces defects: Higher test coverage can lead to fewer defects and issues in the software, resulting in higher quality and reliability.
4. Improves maintainability: Test coverage helps ensure that changes to the code do not introduce new defects or break existing functionality.
5. Provides confidence: High test coverage gives developers confidence that the software has been thoroughly tested and is ready for release.
6. Supports refactoring: With good test coverage, developers can refactor code with confidence, knowing that tests will catch any regressions.
7. Reduces risk: Test coverage helps mitigate the risk of defects and issues that could impact users, business operations, or reputation.

By prioritizing test coverage, teams can ensure that their software is robust, reliable, and meets user expectations.

1.4 Research Scope

This article explores test coverage's, data mining methods, and supporting tools in software testing, focusing specifically on these areas without covering all related methods exhaustively.

1.5 Research Questions (RQ's)

The purpose of this analysis is based on the following research questions (RQ's)

RQ1- Specify different coverage criteria used by authors

RQ2- Are clustering methods sufficient for test case minimization

RQ3- Why K-means clustering is mostly used by the researcher

RQ4 -How this analysis is supportive for researchers

We hypothesize that clustering methods, particularly K-means, are effective for test case minimization, and that a comprehensive analysis of coverage criteria and clustering techniques will provide valuable insights for researchers, ultimately enhancing the efficiency and effectiveness of software testing.

2. LITERATURE REVIEW

(4, 2011, KARTHEEK MUTHYALA), author integrated data mining techniques to decrease the number of test cases by using K-means and pickup cluster algorithm. The approach identifies duplication included in automatically created test cases thanks to data mining, which identifies comparable patterns in test cases.

The creation of test cases, cluster creation, pick-up cluster, and software testing using the updated test suite currently account for the typical testing duration. It was discovered that the running

time was far shorter than the time needed to complete every test case that was created. When the program to be tested is quite large, this optimization performs better than standard testing.

(5 ,2013, Yulei Pang) This paper suggests using k-mean clustering to categorize test situations into two groups. The goal is to remove the requirement to run non-effected test cases and concentrate on the program's affected sections and the associated affected test cases. Grouping a collection of objects so that those in the same group or cluster are more similar to one another than to those in other groups or clusters is the aim of cluster analysis [2]. For clustering various methods are available, popular technique is 'K-means' clustering, which divides a number of data sets or observations into 'K' clusters, each of which is assigned to the cluster with the closest mean [3]. According to study test case clustering can lower regression testing cost.

(6, Subashini and Jeyamala, 2014). Worked on created a collection of test cases in their method by applying the path coverage requirement. Then, using extremely basic programs that solely considered the path coverage condition, they presented a clustering strategy to cut down on the number of test instances.

.

In this article reduction is based on density-based clustering (7, Rashi et al., 2014). Author began by utilizing the Selenium tool to create a set of test cases. Then, they loaded the test cases into Weka and used the DBSCAN clustering algorithm. Finally, redundant test cases were eliminated using the proper filter. The present study employs knowledge mining techniques, particularly the K-means clustering algorithm, to choose the most efficacious test cases for fault detection and so minimize the total number of test cases.

(8, Fayaz Ahmad Khan, Anil Kumar Gupta) In this article, the K-Means approach was used for two test suites: one test suite was created using two well-known black box test case generation approaches for a single variable input scenario, while the second test suite was created for a two-variable input example. Along with test cases produced by utilizing boundary value analysis and equivalency partitioning, the approach included all potential values, including negative and out-of-range ones. There would be a waste of time, money, and effort because there are many redundant test cases that are generated. Therefore, the K-Means technique is performed on both test suites to automatically minimize the test cases in the test suite and obtain a representative collection of test cases from the entire test suite.

(9, Fayaz Ahmad Khan, Dr. Anil Kumar Gupta) To divide the input domain or test suite of the example module into the required number (K) of partitions, the author used the K-Means Algorithm. The idea behind the method is to divide the test suite into a predetermined number of clusters to trim down on size and get rid of unnecessary test cases. In this scenario, the test cases are grouped based on the sample code's coverage criteria. When using any reduction technique, a coverage requirement is crucial since it provides an indication of how many structural components a test suite has executed. The code coverage criteria serve as a guide for the test suite reduction.

(10, Fayaz Ahmad Khan, et al 2015) The test suite that was initially created is examined and the hierarchical clustering approach is used to divide it into a predetermined number of clusters and minimize its size. To determine the number of clusters and the code coverage criterion, a branch coverage criterion is chosen. Creating clusters from the provided set of data or objects is the goal of cluster analysis. The two qualities listed below should be present in every cluster or division that is created: 1. Data that are homogeneous inside a cluster should have the greatest degree of similarity. 2. Data belonging to distinct clusters should have the greatest degree of heterogeneity feasible between them. It has been noted that certain test cases are unnecessary because they are

(11, Ahmad A. Saifan, Emad Alsukhin , 2016) The method begins with the source code and makes use of the integrated development environment (IDE) provided by Eclipse SDK 3.7.2 (Eclipse, 2016). It comes with a base workspace and an expandable plug-in system, primarily made in Java, allowing environment customization. Since they displayed the same behaviour and would

produce the same results, the test cases that are part of the same cluster are redundant. Thus, the author used a selective strategy to identify the most useful test cases to limit the number of test cases. Other than redundant test cases, which are those that are at the same distance from the cluster centre.

(12, Feng Liu et al., 2017) The streamlined test suite while ensuring the case coverage rate and the error detection rate ultimately gain the minimal test suite; this method uses the greedy algorithm to process the streamlined test suite. The paper first introduced the K-medoids clustering algorithm and then applied a method of parameters generated test suite characterized by cyclomatic complexity and code coverage rate. Using the test case as sample points for the K-medoids algorithm, the process first creates a test suite based on the testing requirements. This can be done using the Junit or CodeProAnalytix tools. Next, it computes code coverage and cyclomatic complexity for each sample point, using them as the algorithm's attributes. Finally, it uses these attributes to cluster the sample point to the center point for each classification. Give precedence to choosing the test required with the largest point to 'T' based on the distance between the near and the distant. Take equal sample points out.

(13, Mohammed Akour et al. 2018) By removing the unnecessary test cases, the study seeks to lower the test cost. The process starts with randomly creating the test scenarios. Test cases are created from the payroll system database functionalities using the Procedural Language/Structured Query Language (PL/SQL) tool. The K-means Clustering technique is used in the SPSS software package to minimize the number of test instances. Using the PL/SQL tool, we first created the test cases. The redundant test cases were then eliminated using the K-Means Clustering technique, which was applied, based on the test cases' distances from the cluster centers.

(14, A. Pandey et al. 2019) By removing unnecessary test cases, this paper reduces the amount of test cases and helps us save time when testing a large number of test cases. The author reduced the number of test cases that needed to be tested by using the elbow approach in conjunction with the K-means algorithm. The method significantly reduces the number of duplicated test instances and improves clustering accuracy, according to the experimental results. Our method mostly comprises of four steps. First, the source program is chosen, and test cases are created for it. The test cases created for the source software are then used to prepare the dataset in the following step. Using the elbow approach to anticipate the precise value of "K," the third step involved applying the clustering algorithm to the dataset that had been created in the preceding phase. After that, the clustered results are saved and the superfluous test cases are removed using the necessary filters.

A machine learning technique for test suite reduction that combines binary search and k-means clustering was presented in this research (15, Nour Chetouane et al. 2020). The algorithm's concept is to group test cases that are similar to one another and then choose a representative test case to be included in the new, smaller test suite from each cluster. The binary search method is used to find the right number of clusters, which permits the test suite to be reduced as long as the coverage or mutation score from the original test suite is maintained.

The test reduction method, according to approach (16, C. Xia, et al. 2021), consists of two components: the evolution algorithm and cluster analysis. During the grouping phase, a basic K-means algorithm is utilized. The developed individuals are then sorted using the non-dominated genetic algorithm, and the individuals that are at the same level determine the level of congestion. To assess new individuals, crossover, mutation, and selection processes based on genetic algorithms are then carried out. Ultimately, a smaller collection of test suites is produced by decoding each test case and eliminating superfluous ones using the overriding all coding principle.

(17, SARAH M. NAGY, et al. 2023) employs a clustering-based strategy to significantly reduce the number of tests. Using K-mean++, test cases are categorized into groups based on how similar they are to one another. The test suite in

each cluster is then reduced using a multi-objective genetic algorithm based on code coverage. The elbow and silhouette analysis method defines the ideal "K." The improved method fared better in terms of code coverage rate and test suite reduction than earlier published methods.

(18, Dhanashree Kedare, et al. 2025) In this author worked on k-means and DBSCAN for test case optimization. The research focus on regression testing optimization, reducing the number of test cases, improve the efficiency and effectiveness of regression testing. The statistical significance of the differences in performance between the algorithms will be assessed using appropriate statistical tests, such as ANOVA or t-tests

3. PROPOSED ANALYSIS

The proposed analysis is based on previous work, which focused on the systematic analysis of available articles related to test case minimization using various clustering methods. As per the analysis most of papers worked on K-means, Hybrid and some based on other clustering methods.

3.1 Analysis of Papers Based on Methods, Tools & Coverage Analysis

This analysis is based on existing papers and clustering methods used by different researchers, the summary is based on selected field's like- Author name with year, Method used, tools, and coverage.

Table 1: Analysis as per, clustering methods, coverage and tools used by author

Ref. No.	Author name with year	Method used	Tool used	Coverage Based on
4	2011,Kartheek Muthyala	K-means and Pickup cluster	Weka Etc.	Hybrid method for distribution and coverage based
5	2013, Yulei Pang	K-means	Java, Junit	Code Coverage Based on Hamming Distances
6	2014, .B.Subashini	K-means	Weka, java	Based on path coverage
7	2014, Rashi	DBSCAN	Weka, Selenium	Coverage based on input and output values
8	2014, Fayaz Ahmad	K-means	Weka	Statement coverage
9	2015, Fayaz Ahmad	K-means	CodeProAnalytiX, CodeProJUnit	Based on code & decision coverage
10	2015, Fayaz Ahmad	K-means	Weka	Based on branch coverage criteria
11	2016, Saifan, Alsukhni	K-means	CodeProAnalytiX, CodeProJUnit	Based on complexity & code coverage
12	2017, F. Liu, J. Zhang	K-medoids	CodePro AnalytiX, Eclemma	Based on code coverage and Cyclomatics complexity
13	2018, M. Akour, Iman	K-means	PL/SQL,SPSS	Based on code coverage
14	2018, A. Pandey, A. K. Malviya	K-means & Elbow	Weka ,Eclipse SDK	Coverage based on input and output values
15	2020, N. Chetouane, F. Wotawa	K-means and binary search	Weka	Based on code and statement coverage
16	2021, C. Xia, Y. Zhang	K-means with Genetic Algorithm	Random Selection	Based on code coverage
26	2022, Chen-Hua-Lee	K-means, K-modes, and Hierarchical Agglomerative Clustering (HAC)	Defects4j with software framework	Function and Statement Coverage
17	2023, SARAH M. NAGY	K-means++ with Genetic Algorithm	NSGA and MATLAB	Based on Code Coverage
18	2024, D. Kedare	K-means with DBSCAN	ANOVA, T-test	Based on Code Coverage

3.2 Analysis Based on Methods, Tools & Coverage Analysis

Table 2: Detailed Analysis as Per Reduction Approach, Analysis and Final Minimization

Ref. No	Reduction Approach	Analysis	Final minimization
4	Total no of 'K' Based on total no condition	From clusters, Remove similar patterns	Select single test case from each cluster
5	Using K-means method, data sets are classified into two classes effective and non-effective test cases	Identification of test cases as per high ratio of recall and as per accuracy	Remove non-effective test cases, result minimization
6	Using k-means data mining techniques, data object are grouped into required set of clusters, same data sets are grouped into single cluster, other cluster may have dissimilar data	Test data measurement used as a coverage criteria as the basis of white box testing. As in method the source code is converted into a control flow graph. Output test data considered as the coverage criteria	The algorithm minimized objective function using squared error function with supporting method
7	For to identify redundant test cases, similar data objects are club into a single cluster after applying density based clustering method	Focus on density based clusters, very low density based clusters subject to low priority test cases	DBSCAN data mining method applied on source test cases and removed duplicate test cases using filter as per the condition
8	Using K-means and supporting method, obtain effective and non-effective test cases after generating valid test set of test cases, and remove non-effective test cases	After measurement of "Euclidean" distance, test data are portioned into required number of cluster (K) as cluster (0), cluster (1) and cluster (n)	In each cluster, calculate seed point, known as centroids of a cluster after selecting random point. Cluster mean point works as like cluster centroids
9	Remove duplicate data set from 'K; clusters	As per the information on data, meaningful data are sorted and stored into groups, so similar data are identified and removed	Select one test case from each cluster
10	Clusters 'K' equals to total number of output condition, select one from each cluster	Data sets are splits into clusters as per coverage criteria. After generating test cases as per the condition data sets are divides into fix size clusters.	Method closely observe each data group/ cluster, reduced test cases after observing same result multiple times
11	If in the group of different classes, some data points have the same distance from the cluster center point, consider as redundant test cases	In Eclipse developer toolkit, CodeProAnalytix is an automated software testing and quality tool that can support to test data generation, code analysis and error detection	Based on code and decision coverage analysis test cases are measured, test cases that have same behavior treat as redundant test cases
12	Test suite generated by CodeproAnalytix or Junit, apply k-mediods algorithm, calculate code coverage and cyclomatic complexity	Use the greedy algorithm to find the test suite with the largest coverage testing requirements in the final screening of the test suite	After k-mediods, remove equal data points
13	Avoid and remove test cases based on the dame distance from the cluster centre	Started by generating the test cases from the PL/SQL tool. Then, we applied the clustering algorithm	After generating test cases, they applied rule based on Fuzzy logic, they chose only one data member from each cluster
14	Remove common test cases based on output	Calculate correct 'K' and eliminated duplicated data	Open the obtained clustered result in WEKA and apply appropriate filter to eliminate redundant test cases
15	To remove duplicate data, combine K-means with binary	From each cluster centroid, select the closest test cases based on distance	To remove reduced test cases, data seta are divides

	search algorithm, after each iteration test data are sorted and contain less redundant data sets	metric method. The representative test cases have reduced test cases.	by 2 as like a binary data search, until a specific set of data not further divided
16	Based on clustering result, duplicate test case are removed	Reduced object are shown by cost related, fault related and coverage related criteria	With clustering method genetic algorithm is applied to all the clusters, test cases in the same cluster contains more similarity
26	Model working on three methods, K-means, K-modes, and Hierarchical Agglomerative Clustering (HAC)	Experiments based on real subject programs were evaluated using fault detection effectiveness (FDE) loss and other comparison criteria.	Reduce test suite size with the same or higher fault detection capability by applying cluster-based test suite reduction (CBTSR) methods with two testing criteria
17	Calculating the silhouette value for different values of k, K-Medoids utilizes the Medoid, which represents the data point most centrally positioned within the cluster	Multiple test cases to be executed simultaneously on different machines or devices, reducing the overall testing time.	The Medoid is the data point that minimizes the overall dissimilarity to all other points within the cluster.
18	K-means, FSK-means (Fractional Sigmoid K-means) and DBSCAN. Optimized test cases that are redundant by keeping the high fault detection ratios.	The chosen clustering algorithms (K-means, FSK-means, DBSCAN) will be evaluated with respect to precision, recall, F-measure and fault detection rate	Robust statistical methods are used for the remediation of outliers in the procedure

3.3 Widely used Clustering Methods (result as per graph)

As per the study of selected papers K-means and Hybrid (K-means with other) methods are mostly used by researchers. As per analysis, most researchers used the K-means clustering technique. The second widely used method is hybrid, where a combination of K-means and other methods are used.

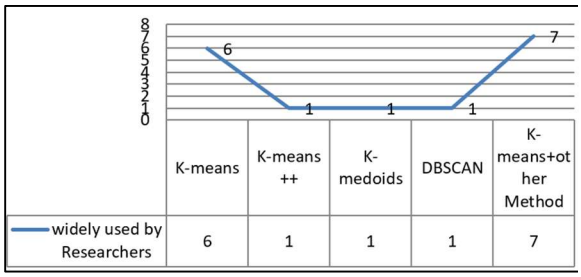

Figure 2: Usage of Clustering Methods

3.4 Supporting Tools used for Test Case Minimization using Clustering

The approach, test case minimization is not possible without the use of supporting programs such as Java, Junit, Weka, Selenium, etc. where Java is used to edit and create program code, Weka and Selenium are for test case generation and implementation of the clustering algorithm. As per the graph, Weka is a widely used computer program support to test case minimization.

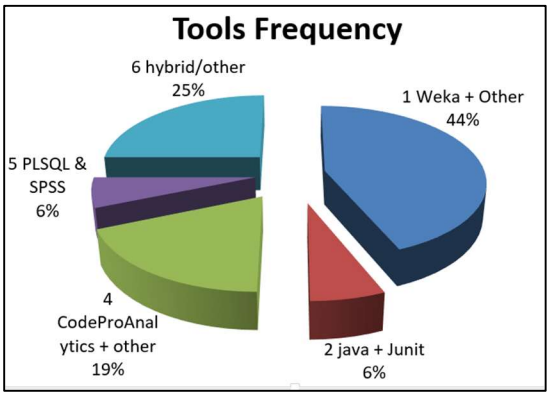

Figure 3: Usage of Major Tools

3.5 Selection of Coverage for Test Cases and Minimization

As an analysis, most researchers worked on hybrid coverage while others worked on code coverage, path coverage, condition coverage, and branch coverage.

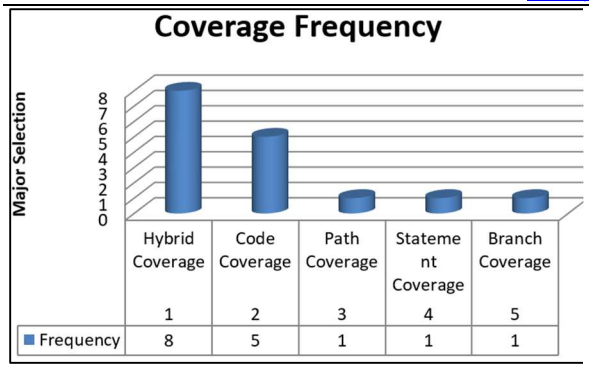

Figure 4: Usage of Different Test Case Coverage

3.5 Result Analysis

As per the analysis of clustering methods for test case minimization, the researcher worked on the clustering analysis plot in chart number -1. The researcher frequently used K-means and hybrid methods. Data mining clustering methods are one of the methods that support test case minimization. Minimization constraints on different parameters, not a single method is available for complete minimization. K-means is a very easy result-oriented simple method, used in major frequency. This analysis specifies what available methods, tools, and coverage used by researchers.

Table 3: Analysis Summary of Major used Factors

S.No.	Remarks	Description (as per usage)	Rank (in between 1 to 10)
1	Method	K-means + other Methods	7
2	Coverage	Hybrid Coverage	8
3	Tools	Weka + Other	7

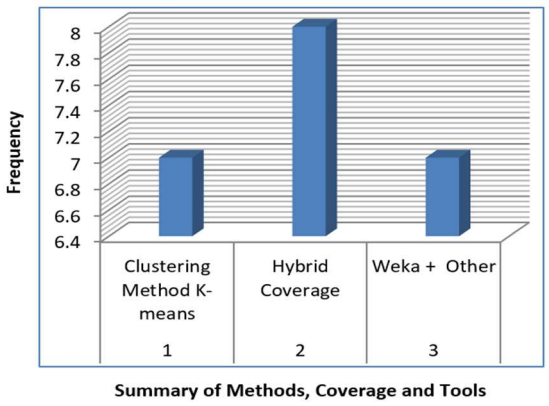

Figure 5: Graphical Representation of Analysis

4. CONCLUSION AND FUTURE WORK

The approach of test case minimization may be implemented by many methods like Genetic, Fuzzy, data mining, and hybrid. In the field of data mining widely used approaches are classification, association, and clustering, and this paper is specially based on Test Case Minimization based on different clustering methods. Sometimes the selection of clustering methods depends on primary data and the form of primary data is spherical, non-spherical, or hybrid form. Specific clustering methods work well on spherical data while some preferable only on non-spherical data. As per first research question, selection of coverage's are based on different types of testing's conducted by development team specially based on program code or functionality testing. As per second, only clustering method is not sufficient for test case minimization, the result may vary method by method. As per third, clustering method or clustering with other methods are mostly used for test case minimization and the last this analysis support to researcher for selection of specific clustering method, tool and specific coverage.

REFERENCES:

[1] N. Gupta, A. Sharma and M. K. Pachariya, "An Insight into Test Case Optimization: Ideas and Trends with Future Perspectives," IEEE Access, vol. 7, pp. 22310-22327, 2019, doi: 10.1109/ACCESS.2019.2899471.

[2] C. Coviello, S. Romano, G. Scanniello, A. Marchetto, G. Antoniol and A. Corazza, "Clustering support for inadequate test suite reduction," 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), Campobasso, Italy, 2018, pp. 95-105, doi: 10.1109/SANER.2018.8330200.

[3] N. Mottaghi and M. R. Keyvanpour, "Test suite reduction using data mining techniques: A review article," 2017 International Symposium on Computer Science and Software Engineering Conference (CSSE), Shiraz, Iran, 2017, pp. 61-66, doi: 10.1109/CSICSSE.2017.8320118.

[4] Kartheek Muthyala, Rajshekhar Naidu,"A Novel Approach to Test Suite Reduction

using Data Mining", IJCSE, ISSN : 0976-5166 Vol. 2 No. 3 Jun-Jul 2011.

[5] Y. Pang, X. Xue and A. S. Namin, "Identifying Effective Test Cases through K-Means Clustering for Enhancing Regression Testing," 2013 12th International Conference on Machine Learning and Applications, Miami, FL, USA, 2013, pp. 78-83, doi: 10.1109/ICMLA.2013.109.

[6] Subashini, B. & and JeyaMala, D. (2014). Reduction of Test Cases Using Clustering Technique. In Proceedings of International Conference on Innovations in Engineering and Technology, ICIET'14.

[7] Rashi Chauhan, Pooja Batra, Sarika Chaudhary, "An Efficient Approach for Test Suite Reduction using Density based Clustering Technique", International Journal of Computer Applications. 97, July 2014, 1-4. doi:10.5120/17048-6276.

[8] Fayaz Ahmad Khan,Anil Kumar Gupta "Profiling of Test Cases with Clustering Methodology " ,International Journal of Computer Applications (0975 8887) Volume 106 – No. 14, November 2014.

[9] Fayaz Ahmad Khan, Anil Kumar Gupta, " An Efficient Approach to Test Suite Minimization for 100% Decision Coverage Criteria using K-Means Clustering Approach", IJAPRR International Peer Reviewed Refereed Journal, Vol. II, Issue VII, p.n. 18-26, 2015, ISSN: 2350-1294.

[10]Fayaz Ahmad Khan, Anil Kumar Gupta, " An Efficient Technique to Test Suite Minimization using Hierarchical Clustering Approach", International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-3 Issue-11, September 2015.

[11] Saifan, A. A., Alsukhni, E., Alawneh, H., & Sbaih, A., "Test Case Reduction using Data Mining Technique", International Journal of Software Innovation (IJSI), 4(4), 56-70. doi:10.4018/IJSI.2016100104

[12] F. Liu, J. Zhang and E. -Z. Zhu, "Test-Suite Reduction Based on K-Medoids Clustering Algorithm," 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Nanjing, China, 2017, pp. 186-192, doi: 10.1109/CyberC.2017.38.

[13]Mohammed Akour, Iman Al Jarrah,"An Efficient Approach For Test Suite Reduction Using K-Means Clustering", Journal of Theoretical and Applied Information Technology,15th September 2018. Vol.96. No 17, ISSN: 1992-8645.

[14] A. Pandey1, A. K. Malviya ,"Enhancing test case reduction by k-means algorithm and elbow method ", 2018, IJCSE , Vol.6(6), Jun 2018, E-ISSN: 2347-2693.

[15] N. Chetouane, F. Wotawa, H. Felbinger and M. Nica, "On Using k-means Clustering for Test Suite Reduction", 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2020, pp. 380-385, doi: 10.1109/ICSTW50294.2020.00068.

[16] C. Xia, Y. Zhang and Z. Hui, "Test Suite Reduction via Evolutionary Clustering," IEEE , vol. 9, pp. 28111-28121, 2021, doi: 10.1109/ACCESS.2021.3058301.

[17]Sarah H M. Nagy1, Huda A. Maghawry, "An Enhanced Approach for Test Suite Reduction Using Clustering And Genetic Algorithms" JATIT, 15th June 2023. Vol.101. No 11, ISSN: 1992-8645."

[18] Dhanashree Kedare, Mukesh Kumar, Dhara Vyas, "Clustering Driven Machine Learning Framework for Scalable Test Case Minimization and Optimization, JISEM, 2025, 10(39s), ISSN: 2468-4376.

[19] Neha Sharma, Dr. Shilpi Singh, "Software Testing Techniques: A Literature Review", 2020| IJIRT , Volume 7 Issue 5, ISSN: 2349-6002.

[20] Gupta, A. Sharma and M. K. Pachariya, "An Insight into Test Case Optimization: Ideas and Trends with Future Perspectives," IEEE Access, vol. 7, pp. 22310-22327, 2019, doi: 10.1109/ACCESS.2019.2899471.

[21] Neha D., B.M. Vidyavathi,"A Survey on Applications of Data Mining using Clustering Techniques. International Journal of Computer Applications", 126/2, 2015, 7-12. doi:10.5120/ijca2015905986.

[22] Meenu, Navita," Study and Analysis of Software Testing", ISSN: 2321-8169, 6674 – 6678, Volume: 3 Issue: 12, IJRITCC , 2015.

[23] S. Parsa, A. Khalilian and Y. Fazlalizadeh, "A new algorithm to Test Suite Reduction based on cluster analysis," 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 2009, pp. 189-193, doi: 10.1109/ICCSIT.2009.5234742.

[24] Sakshi Rastogi, "A Comparative Study on Testing Techniques of Software", Volume 6, Issue 7, July – 2021, ISSN No:-2456-2165.

[25] A. Mehmood, Q. M. Ilyas, M. Ahmad and Z. Shi, "Test Suite Optimization Using Machine Learning Techniques: A Comprehensive Study," in IEEE Access, vol. 12, pp. 168645-168671, 2024, doi: 10.1109/ACCESS.2024.3490453.

[26] Chen-Hua Lee and Chin-Yu Huang. Applying Cluster-based Approach to Improve the Effectiveness of Test Suite Reduction [J]. Int J Performability Eng, 2022, Vol-18(1), doi:10.23940/ijpe.22.01.p1.110.

[27] P. Kreutzer, T. Kunze and M. Philippsen, "Test Case Reduction: A Framework, Benchmark, and Comparative Study," 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), Luxembourg, 2021, pp. 58-69, doi: 10.1109/ICSME52107.2021.00012.