REST Journal on Emerging trends in Modelling and Manufacturing
Vol: 5(4), 2019
REST Publisher
ISSN: 2455-4537

Website: www.restpublisher.com/journals/jemm

Using a ELECTRE MCDM method for Software Testing Techniques

Krishna Kumar TP[1]*, M.Ramachandran[2]

[1]Nehru School of Management @ NCERC, Pamapady, Thiruvilwamala, Thrissur, Keral, India.

[2]MPSTME SVKM'S Narsee Monjee Institute of Management Studies, Shirpur, Maharashtra 425405, India

*Corresponding Author: tpk683@gmail.com

Abstract

Software testing techniques are gleaned from business to test the application under test against functional or non-functional requirements The methods used are Each testing technique is specific Helps to detect type deficiency. Software Development Life Cycle (SDLC) is a structured process, which is low productivity To develop high-quality, low-cost software over time helps. The goal of SDLC is, all customers meet expectations and demands and It's about building great software that breaks. ELECTRE (Elimination and Choix Traduisant La Realite - Elimination and Choice expresses reality) methods from recruitment to many real-world decision-making There are problems widely used. Transportation and more. Theoretical research on the fundamentals of electre methods is also active at this time. Alternative: Equal distribution, Boundary Value Analysis, Chaos, Cause- Effect graph, orthogonal array Test, all pair test. Evaluation Preference: Control Flow Testing, Branch Testing, Basic path testing, and data flow Test. From the result it is seen that data flow testing is got the first rank whereas is the control flow testing has the lowest rank. The value of the dataset for Range of Software Testing Techniques in ELECTRE Method shows that it results in data flow testing and top ranking.

I. Introduction

Internal structures or functions of an application examine its operation without looking closely at Software testing methodology. This test is valid at each stage of software testing and can be used for: unit, coordination, structure and Acceptance. Basically, the process of QA can be divided into 5 main stages: Identify, plan, design, implement and optimize. QA team functional and non-functional software requirements Helps define and how they relate to business objectives checking. Sanity testing is a subset of regression testing. After receiving the software build, Code changes introduced are as expected sanity checks to make sure it works. This test is a checkpoint that decides whether or not to continue testing for the build. Smoke testing is a preliminary verification of software. This type of testing identifies fundamental and critical issues in an application before performing critical testing. Electre TRI is a sorting multicriteria method, that is, the method assigns alternatives to predetermined categories. Alternative assignment to a class is the result of comparing these alternative and category profiles ELECTRE III Comparison of Feasibility and Criteria To measure importance Used when desirable, while ELECTRE IV is used in cases where this measurement is not possible. In most cases, ELECTRE methods are traditional Independent for Multicriteria Decision Making (MCDM) problems that fit the criteria. However, the real more or between criteria in situations are more or less interdependent. A special case is MCDM, Priority is called MCDM. The latest over the years, MCDM in the context of priority criteria How to deal with problems is increasing. Test Planning – Test Team Strategy and planning the approach. Test Case Design – Testing based on objectives and criteria Making cases. Test Environment - Test an integrated environment to test the product during execution – real-time validation of the product and detection of errors. Selenium is a web browser and an open source tool for automation. Ruby, Java, Node JS, PHP, Perl and Python etc Write test scripts in programming languages like it provides a single interface that allows you to #. Functional testing is predetermined whether the software is working as per requirements QAs are a deterministic process. It's a black box using experimental techniques, where the tester has no knowledge of internal system logic. What are ELECTRE methods? ELECTRE methods are a family of decision support methods combining elimination and choice expressing reality, partial integration based on the structure of relationships of comparisons of the performance of each pair of solutions. A special case is MCDM, Priority is called MCDM. The latest over the years, MCDM in the context of priority criteria How to deal with problems is increasing.

II. Software Testing Techniques

Software testing is the detection of errors in the developed computer or software, indicating the detection of errors or missing requirements. So, this is an investigation. It is accurate Provides knowledge of product quality to stakeholders. Software testing is risk-based Can also be considered action. During the testing process what is important is that a large number of software testers understand how to reduce tests into manageable tests. [1] Among the main problems in the area of software testing one is suitable for testing a software system: How to get a collection of cases. This package maximizes performance with a minimal number of test cases to confirm. There are now many experimental techniques for developing [2] Testing conducted in software development This is software testing called Test or The main purpose of software testing Diagnosing bugs in software. An error is a SOFTWARE OR THE PROGRAM OR IN THE CONDUCT OF THE SOFTWARE error or mistake that occurs application. You can test the software and Check if the software is available is correct

- All mentioned in the design phase Also fulfills the requirements
- Gives correct output for different inputs
- Deadline or acceptable Work can be completed within time.
- Works in different environments.[3]

Software testing and error detection processes challenge the software community. Software testing and error detection activities although imprecise and poorly understood, software they are critical to the success of the project. This article is a controlled study that presents the uncertainty of how to effectively test there. An experimental method was used to address software. [4] Despite this work, research on the experimental oracle problem As a fragmented activity among researchers and social practitioners has First providing a detailed analysis and review of the work By crossing this fragmentation in this important part of software testing The role of this paper. Test oracle problem. [5] The task of software testing is the actual result and the system checks that the expected result is consistent. This is to ensure that it is delivered without errors. A There are several techniques to help verify that the system is free; Approaches and tools are proposed for defects [6]. A methodological framework that is sufficiently specific does not make any assumptions about software testing techniques and the adequacy of testing for general evaluation and pilot projects about the technique or subjects being evaluated. [7] Allowing only cost effectiveness to be assessed But Techniques by the characteristics of trained engineers Using, thus our human factors calculates. After all, humans still can't be software tested. However, an important one is the above statistics of problem error detection rates associated with nature. [8] Software testing is undoubtedly Cost consumption is Standard practice in the industry. According to NIST, Industry-standard in software testing infrastructure Impacts are approximately 59.5 per year for the American community Software costing billions of US dollars The cost of testing is generally estimated as existing, 40 to 80% of the total development cost of safety-critical systems will be more extensive.[9] Supporting replication For controlled trials, test factors and exert control over the environment; This Units of analysis and context are more complex and It is difficult to achieve. Software testing A controlled experiment with techniques while doing, there are many replicable challenges. [10] As a research community, in software testing experience Careful consideration of the specific validity issues that arise Seeing them structured and accurate Categorize methodically and help solve them Creating routines is important. [11] Software testing is in software development. It is an essential and costly process, therefore More on the question of how to automate it Research efforts have been devoted to [12] Applications of Outsourcing, cloud and crowd based Testing is a cost-effective testing of mobile Actionable solutions. Software testing services Companies can expect to deliver as a service special offers Affordable mobile apps that are critical to complete testing capabilities and labs.[15] Regarding the Test software for delivery Quality of product or services under test a platform to stakeholders is used. of software implementations It is a framework for understanding risks provides Software testing is the verification and The validation process is defined as validation [16] Most software testing techniques are few Test data that meets the criteria create Instead of generating test data Analysis of Variance is slightly in measuring experimental data is different. [17] Search-based software testing (SBST) which makes testing an optimization problem Search Based Software Engineering (SBSE) Applying computational search techniques from the field can be attacked. [19] To achieve a certain coverage in software testing Large-scale use of genetic methods There is research, however not on the met heuristic tabu search technique.[20].

III. ELECTRE Methods

Within this framework, we attempt to separately analyze the performance of State-owned, private in Turkey Owned and foreign banks. ANP and ELECTRE I methods are used to achieve the objective.[1] ELECTRE Multi-criteria decision analysis methods Family. ELECTRE methods are two main Includes steps. First, detail each pair of actions. Ranking relationships for comparison are developed step by step. The second step is a review process on the first step Elaborates recommendations based on results obtained through [2] The ELECTRE method is among the most popular outranking models One, one can be used to solve the MCDM problem. One of the simplest logic methods because it is very Excellent, but some studies A-IFS ELECTRE this method used properly, Based on the estimated information provided by the decision Contains pair wise comparison of alternatives.[3] To make the evaluations more rational and efficient, of the proposed concept of source reliability Basically discount system is provided in DST. Additionally, along with the family of MCDM models, ELECTRE The method is to rank the set of alternatives Famous for improved relationships.[4] ELECTRE Used to select issues, ELECTRE for TRI and to assign those issues For ELECTRE II, III and IV grade problems. Post hoc synchronization and pair

wise comparison Using uses discord indices.[5] ELECTRE determines the high-level relationships between distinct bidding schemes, through consistency and contradiction analysis, and continuously searches for a subset of inferior schemes, ultimately deciding on a First class auction plan. In this approach, the production units Consider the actual situation and payment option Consists of and subsidiary in evaluation indicators overcame the status quo, [6] The main advantage of using outranking is that methods are those that can take into account completely normal parameters without the need to change the original, an arbitrary imposition, an abbreviation, a range of parameters, thus maintaining the original concrete verbal meaning. The ELECTRE method is based on numerical spacing consists of the following steps:[7] The Design using given input data Selection of An optimal solution defined by parameters study attempts to provide the ELECTRE method and Methods based on Pareto Optimality concept. [8] Various MCDA methods over the years have been created. Advanced approaches, especially the ELECTRE family the methods have been around for over 40 years, but one within MCDA continues to be a popular field of research. In this paper English based on ELECTRE et al A comprehensive collection of scholarly papers Literature review is done. Areas of electricity and various electrical How to based methods Our aim is to explore how they are perceived [9] The ELECTRE method is a highly developed multi-criteria approach Attitude analysis model decision making process is objective of the examiner to systematically use outlier relations. [11] By having complex business problems Classified, conflict of values results in certain types of decisions to be inclusive, this is multiple and often counterproductive creates a process of crossing perspectives. In this context, with participatory approaches Electrified methods are not fully featured in the MCDA literature. [12]

IV. Analysis and Discussion

This table 1 is Alternative: Equal distribution, Boundary Value Analysis, Chaos, Cause- Effect graph, orthogonal array Test, all pair test. Evaluation Preference: Control flow testing, branch testing, Basic path testing, data flow Tests Equivalence partitioning it is seen that basis path testing is showing the highest value for data flow testing is showing the lowest value. Boundary value analysis is referred to as branch testing showing the highest value for data flow testing is showing the lowest value. Fuzzing it is seen that branch testing is showing the highest value for basis path testing is showing the lowest value. Cause-effect graph it is seen that data flow testing is showing the highest value for control flow testing is showing the lowest value. Orthogonal array testing it is seen that data flow testing is showing the highest value for control flow testing is showing the lowest value. All pair testing it is seen that data flow testing is showing the highest value for basis path testing is showing the lowest value.

White Box Check:

Equal sharing
Boundary value analysis
Phasing
Cause-effect diagram
Orthogonal array test
All pairs test.

Black Box Testing:

Control flow Test
Branch test
Basic path testing
Data flow is tested.

TABLE 1. Software Testing Techniques in data set

	Equal sharing	Boundary value analysis	confusion	Cause-effect diagram	Orthogonal array test	All pairs test
control flow testing	750	450	89	240	73	150
branch testing	880	470	98	242	85	120
basis path testing	960	390	86	540	78	105
data flow testing	670	380	88	276	88	212

Alternative: Equal distribution, Boundary Value Analysis, Chaos, Cause- Effect graph, orthogonal array Test, all pair test. Evaluation Preference: Control flow testing, branch testing, Basic path testing, data flow Tests.

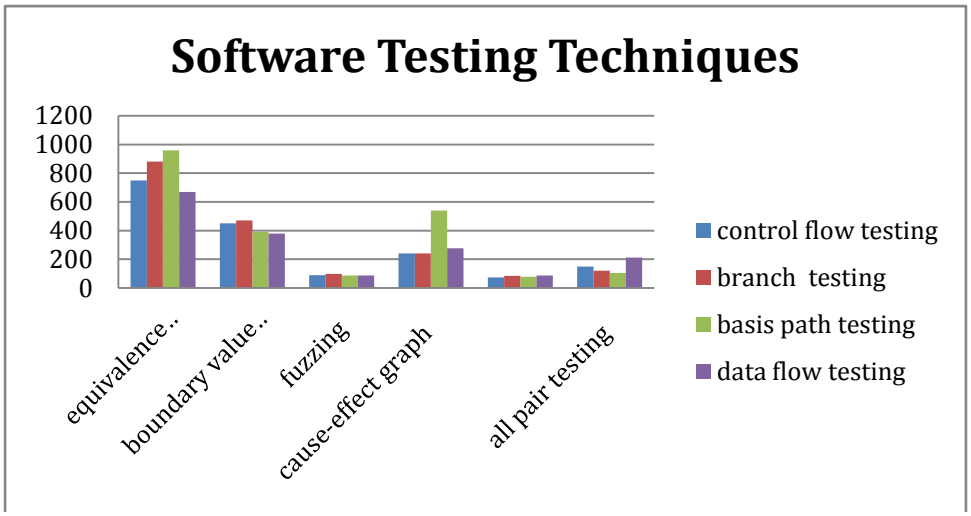

FIGURE 1. Software Testing Techniques in data set

Alternative: Equal distribution, Boundary Value Analysis, Chaos, Cause- Effect graph, orthogonal array Test, all pair test. Evaluation Preference: Control flow testing, branch testing, Basic path testing, data flow Tests.

TABLE 2. Software Testing Techniques SUM & SQRT

	Equal sharing	Boundary value analysis	confusion	Cause-effect diagram	Orthogonal array test	All pairs test
control flow testing	562500	202500	7921	57600	5329	22500
branch testing	774400	220900	9604	58564	7225	14400
basis path testing	921600	152100	7396	291600	6084	11025
data flow testing	448900	144400	7744	76176	7744	44944
SUM	2707400	719900	32665	483940	26382	92869
SQRT	1645.418	848.4692	180.7346	695.658	162.4254	304.7442

Table 2. Shows the Software Testing Techniques SUM & SQRT value of Alternative: Equal distribution, Boundary Value Analysis, Chaos, Cause- Effect graph, orthogonal array Test, all pair test. Evaluation Preference: Control flow testing, branch testing, Basic path testing, data flow Tests. This table mention the SUM & SQRT value equivalence partitioning SUM=2707400, SQRT=1645.418. Boundary value analysis SUM=719900, SQRT=848.4692. Fuzzing SUM=32665, SQRT=180.7346. Cause-effect graph SUM=483940, SQRT=695.658. Orthogonal array testing SUM=26382, SQRT= 162.4254. All pair testing SUM=92869, SQRT=304.7442.

TABLE 3. Normalized Data Matrix

Normalized DM						
	Equal sharing	Boundary value analysis	confusion	Cause-effect diagram	Orthogonal array test	All pairs test
control flow testing	0.455811	0.530367	0.492435	0.344997	0.449437	0.492216
branch testing	0.534819	0.553939	0.542232	0.347872	0.523317	0.393773
basis path testing	0.583438	0.459651	0.475836	0.776244	0.480221	0.344551
data flow testing	0.407191	0.447865	0.486902	0.396747	0.541787	0.695666

Table 3. Shows the Normalized Data Matrix of Alternative: Equal distribution, Boundary Value Analysis, Chaos, Cause-Effect graph, orthogonal array Test, all pair test. Evaluation Preference: Control flow testing, branch testing, Basic path testing, data flow Tests.

TABLE 4. Weighted Normalized matrix

	A weighted normalized matrix					
	0.2336	0.1652	0.3355	0.1021	0.0424	0.1212
	Equal sharing	Boundary value analysis	confusion	Cause-effect diagram	Orthogonal array test	All pairs test
control flow testing	0.106478	0.087617	0.165212	0.035224	0.019056	0.059657
branch testing	0.124934	0.091511	0.181919	0.035518	0.022189	0.047725
basis path testing	0.136291	0.075934	0.159643	0.079254	0.020361	0.04176
data flow testing	0.09512	0.073987	0.163356	0.040508	0.022972	0.084315

Table 4. Shows the Weighted Normalized matrix value of the Equal sharing =02336, Boundary value analysis =0.1652, confusion =0.3355, Cause-effect diagram =0.1021, Orthogonal array test =0.0424, All pairs test =0.1212. Normalized Data Matrix multiplication criterion Weights this will be going to multiply again will be constant Weighted Normalized matrix value.

TABLE 4. Concordance Interval Matrix & Discordance Interval Matrix

C12 ={2}	D12 = {1,3,4,5,6}
C13 = {3,5}	D13={1,2,4,6}
C14 = {2}	D14={1,3,4,5,6}
C21={1,3,4,5,6}	D21={2}
C23={1,3,5}	D23={2,4,6}
C24={1,4}	D24={2,3,5,6}
C31={1,2,4,6}	D31={3,5}
C32={2,4,6}	D32={1,3,5}
C34={1,2,4,6}	D34={3,5}
C41={1,3,4,5,6}	D41={2}
C42={2,3,5,6}	D42={1,4}
C43={3,5}	D43={1,2,4,6}

Table 4. Shows the concordance and discordance sets A= {a,b,c,...} a may denote a finite set of alternatives, the following Two different attribute sets in the formula concordance interval set (Cab) and discordance interval set (Dab).
Coherence to describe the dominant query the interval set is used

$$C_{ab}= \{ j|x_{aj} \geq x_{bj} \}$$

Contrast Interval Set (Dab)

$$D= \{ j|x_{aj} \geq x_{bj}\} = J\text{-}C_{ab}$$

TABLE 5. Synchronization

0	0	0	0	0	1
0	1	1	0	0	1
1	1	1	0	0	0
1	1	1	1	1	0
0	1	1	0	1	1
1	1	1	0	0	0
1	0	0	1	1	0
1	0	0	1	0	0
1	1	0	1	0	0

Table 5 Shows the Concordance =IF(I12>=I13,1,0).

TABLE 6. Concordance Interval Matrix

Concordance Interval Matrix					
	Control flow test	branch test	Basic path testing	Data flow testing	
control flow test	0	0.1652	0.3779	0.1652	0.7083
branch testing	0.8348	0	0.6115	0.3357	1.782
basis path testing	0.6221	0.3885	0	0.6221	1.6327
data flow testing	0.8348	0.6643	0.3779	0	1.877
	2.2917	1.218	1.3673	1.123	6
	c bar				0.5

TABLE 7. Concordance Index Matrix

Concordance Index Matrix				
	Control flow test	branch test	Basic path testing	Data flow testing
control flow test	0	0	0	0
branch testing	1	0	1	0
basis path testing	1	0	0	1
data flow testing	1	1	0	0

TABLE 8. Discordance

	Equal sharing	Boundary value analysis	confusion	Cause-effect diagram	Orthogonal array test	All pairs test
D12	0.018456	0.003894	0.016707	0.000294	0.003133	0.011931
	1					
D13	0.029814	0.011682	0.005569	0.04403	0.001305	0.017897
	1					
D14	0.011358	0.013629	0.001856	0.005284	0.003916	0.024658
	1					
D21	0.018456	0.003894	0.016707	0.000294	0.003133	0.011931
	0.210991					
D23	0.011358	0.015576	0.022276	0.043737	0.001827	0.005966
	1					
D24	0.029814	0.017523	0.018563	0.00499	0.000783	0.036589
	1					
D31	0.029814	0.011682	0.005569	0.04403	0.001305	0.017897
	0.12648					
D32	0.011358	0.015576	0.022276	0.043737	0.001827	0.005966
	0.509315					
D34	0.041171	0.001947	0.003713	0.038747	0.00261	0.042555
	0.087243					
D41	0.011358	0.013629	0.001856	0.005284	0.003916	0.024658
	0.55273					
D42	0.029814	0.017523	0.018563	0.00499	0.000783	0.036589
	0.814818					
D43	0.041171	0.001947	0.003713	0.038747	0.00261	0.042555
	1					

TABLE 9. Discordance Index matrix

Discordance Interval Matrix					
	Control flow test	branch test	Basic path testing	Data flow testing	
control flow testing	0	1	1	1	3
branch testing	0.210991	0	1	1	2.210991
basis path testing	0.12648	0.509315	0	0.087243	0.723038
data flow testing	0.55273	0.814818	1	0	2.367548
	0.890201	2.324133	3	2.087243	8.301577
d bar					0.691798

TABLE 10. Discordance Index matrix

Discordance Index matrix				
	Control flow test	branch test	Basic path testing	Data flow testing
control flow testing	1	0	0	0
branch testing	1	1	0	0
basis path testing	1	1	1	1
data flow testing	1	0	0	1

TABLE 11. Net superior value & Rank

	Net superior value (Concordance Interval Matrix)	Rank	Net Inferior Value (Discordance Interval Matrix)	Rank
control flow testing	-1.5834	4	2.109799	4
branch testing	0.564	2	-0.11314	2
basis path testing	0.2654	3	-2.27696	3
data flow testing	0.754	1	0.280305	1

Table 11 Shows the Net superior value & Rank of the Net superior value (Concordance Interval Matrix) Rank control flow testing is in 4^{td} rank, branch testing is in 2nd rank, basis path testing is in 3rd rank, data flow testing is in 1st rank. Net Inferior Value (Discordance Interval Matrix) Rank control flow testing is in 4^{td} rank, branch testing is in 2nd rank, basis path testing is in 3rd rank, data flow testing is in 1st rank.

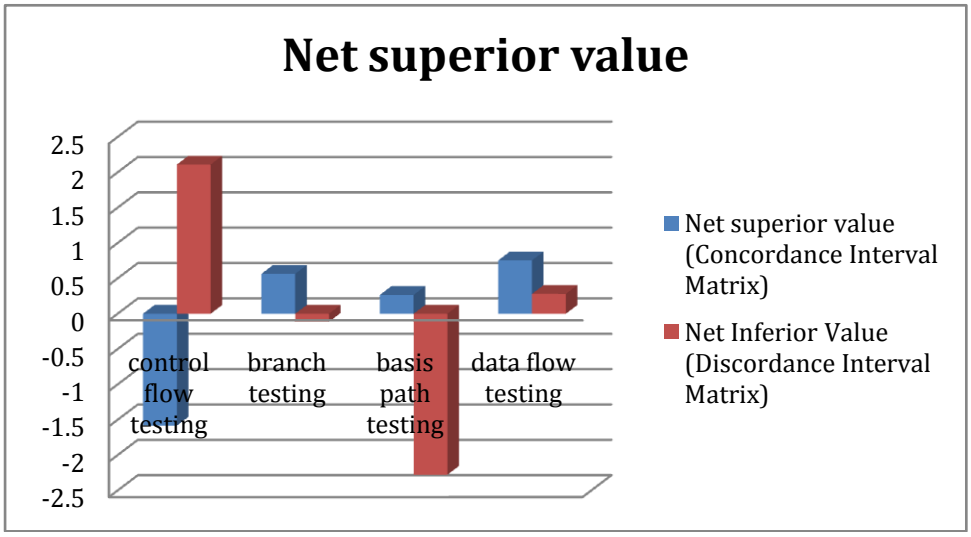

FIGURE 2. Net superior value

Figure 3 shows the graphical representation Net superior value of the Net superior value (Concordance Interval Matrix) Control flow test -1.5834, branch test 0.564, Basic path testing 0.2654, and data flow testing 0.754. Net Inferior Value (Discordance Interval Matrix) Control flow test 1.378762, branch test -0.817, Basic path testing 0.083059, Data flow testing -0.64482

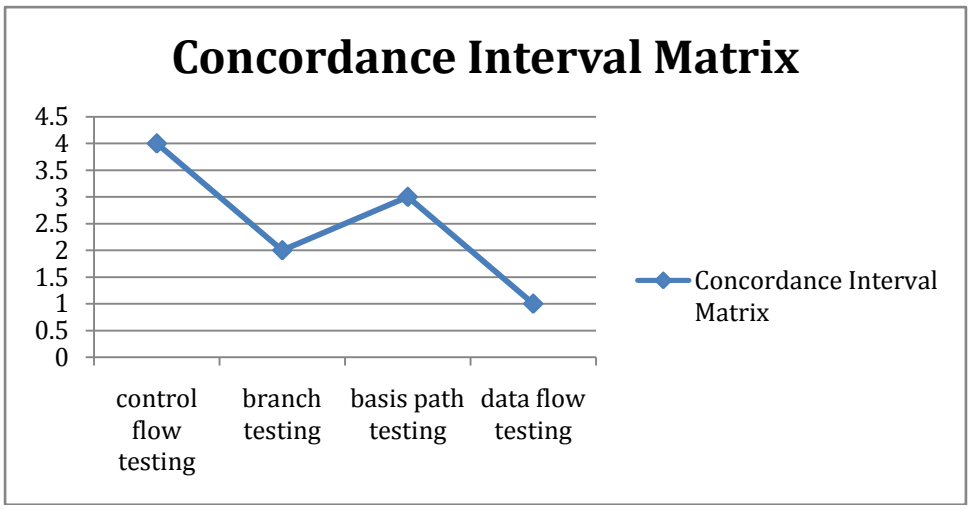

FIGURE 3. Concordance Interval Matrix

Figure 4 shows the graphical representation Concordance Interval Matrix Rank value of the control flow testing is in 4^{td} rank, branch testing is in 2nd rank, basis path testing is in 3rd rank, data flow testing is in 1st rank.

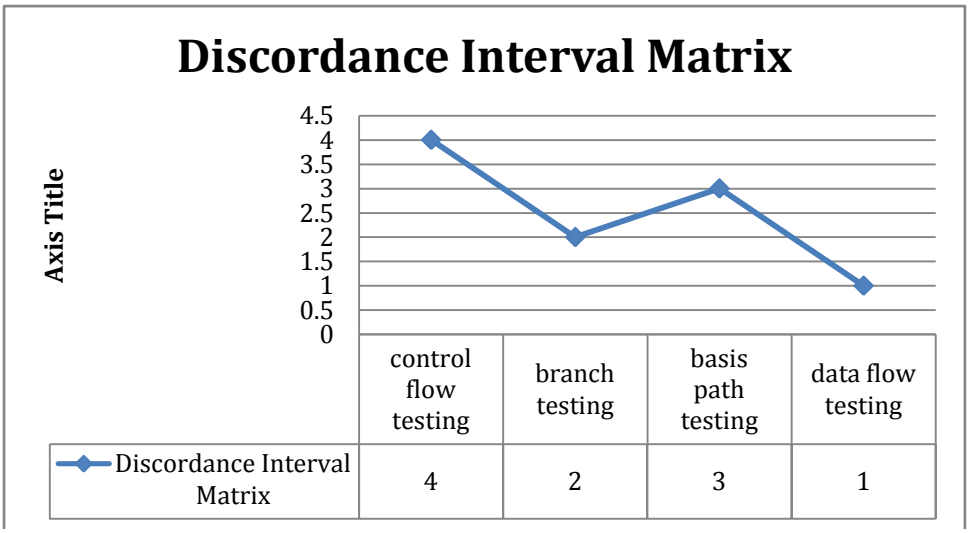

FIGURE 4. Discordance Interval Matrix

Figure 4 shows the graphical representation Discordance Interval Matrix Rank value of the control flow testing is in 4^{td} rank, branch testing is in 2nd rank, basis path testing is in 3rd rank, data flow testing is in 1st rank.

V. Conclusion

Despite this work, research on the experimental oracle problem as a fragmented activity among researchers and social practitioners has First providing a detailed analysis and review of the work By crossing this fragmentation in this important part of software testing The role of this paper. Test oracle problem. The task of software testing is the actual result and The system checks that the expected result is consistent This is to ensure that it is delivered without errors. There are several techniques to help verify that the system is free, Approaches and tools are proposed for defects ELECTRE determines the high-level relationships between distinct bidding schemes, through consistency and contradiction analysis, and continuously searches for a subset of inferior schemes, ultimately deciding on a First class auction plan. In this approach, the production units Consider the actual situation and payment option Consists of and subsidiary in evaluation indicators overcame the status quo From the result it is seen that data flow testing is given the first rank whereas is the control flow testing is having the lowest rank.

Reference

[1] Jamil, Muhammad Abid, Muhammad Arif, Normi Sham Awang Abubakar, and Akhlaq Ahmad. "Software testing techniques: A literature review." In 2016 6th international conference on information and communication technology for the Muslim world (ICT4M), pp. 177-182. IEEE, 2016.

[2] Vegas, Sira, and Victor Basili. "A characterisation schema for software testing techniques." Empirical Software Engineering 10, no. 4 (2005): 437-466.

[3] Sneha, Karuturi, and Gowda M. Malle. "Research on software testing techniques and software automation testing tools." In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 77-81. IEEE, 2017.

[4] Basili, Victor R., and Richard W. Selby. "Comparing the effectiveness of software testing strategies." IEEE transactions on software engineering 12 (1987): 1278-1296.

[5] Barr, Earl T., Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. "The oracle problem in software testing: A survey." IEEE transactions on software engineering 41, no. 5 (2014): 507-525.

[6] Bansal, Anju. "A Comparative study of software testing techniques." Int. J. Comput. Sci. Mob. Comput 36, no. 6 (2014): 579-584.

[7] Vos, Tanja EJ, Beatriz Marín, Maria Jose Escalona, and Alessandro Marchetto. "A methodological framework for evaluating software testing techniques and tools." In 2012 12th international conference on quality software, pp. 230-239. IEEE, 2012.

[8] Briand, Lionel, and Yvan Labiche. "Empirical studies of software testing techniques: Challenges, practical strategies, and future research." ACM SIGSOFT Software Engineering Notes 29, no. 5 (2004): 1-3.

[9] Eldh, Sigrid, Hans Hansson, Sasikumar Punnekkat, Anders Pettersson, and Daniel Sundmark. "A framework for comparing efficiency, effectiveness and applicability of software testing techniques." In Testing: Academic & Industrial Conference-Practice And Research Techniques (TAIC PART'06), pp. 159-170. IEEE, 2006.

[10] Do, Hyunsook, Sebastian Elbaum, and Gregg Rothermel. "Infrastructure support for controlled experimentation with software testing and regression testing techniques." In Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04., pp. 60-70. IEEE, 2004.
[11] Briand, Lionel C. "A critical analysis of empirical research in software testing." In First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), pp. 1-8. IEEE, 2007.
[12] Fraser, Gordon, and Andrea Arcuri. "Sound empirical evidence in software testing." In 2012 34th International Conference on Software Engineering (ICSE), pp. 178-188. IEEE, 2012.
[13] Muccini, Henry, Antonio Di Francesco, and Patrizio Esposito. "Software testing of mobile applications: Challenges and future research directions." In 2012 7th International Workshop on Automation of Software Test (AST), pp. 29-35. IEEE, 2012.
[14] Whittaker, James A., and Michael G. Thomason. "A Markov chain model for statistical software testing." IEEE Transactions on Software engineering 20, no. 10 (1994): 812-824.
[15] Khan, Mumtaz Ahmad, and Mohd Sadiq. "Analysis of black box software testing techniques: A case study." In The 2011 International Conference and Workshop on Current Trends in Information Technology (CTIT 11), pp. 1-5. IEEE, 2011.
[16] DeMillo, Richard A., Dany S. Guindi, W. M. McCracken, A. Jefferson Offutt, and Kim N. King. "An extended overview of the Mothra software testing environment." In Workshop on Software Testing, Verification, and Analysis, pp. 142-143. IEEE Computer Society, 1988.
[17] Harman, Mark, Yue Jia, and Yuanyuan Zhang. "Achievements, open problems and challenges for search based software testing." In 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), pp. 1-12. IEEE, 2015.
[18] Díaz, Eugenia, Javier Tuya, and Raquel Blanco. "Automated software testing using a metaheuristic technique based on tabu search." In 18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings., pp. 310-313. IEEE, 2003.
[19] Dinçer, Hasan, Ümit Hacıoğlu, and Serhat Yüksel. "Managerial and market-based appraisal of agriculture banking using ANP and ELECTRE method." Management and Organizational Studies 3, no. 3 (2016): 29.
[20] Lian, Jiunn-Woei, and Chih-Kun Ke. "Using a modified ELECTRE method for an agricultural product recommendation service on a mobile device." Computers & Electrical Engineering 56 (2016): 277-288.
[21] Wu, Ming-Che, and Ting-Yu Chen. "The ELECTRE multicriteria analysis approach based on Atanassov's intuitionistic fuzzy sets." Expert Systems with Applications 38, no. 10 (2011): 12318-12327.
[22] Fei, Liguo, Jun Xia, Yuqiang Feng, and Luning Liu. "An ELECTRE-based multiple criteria decision making method for supplier selection using Dempster-Shafer theory." IEEE Access 7 (2019): 84701-84716.
[23] Dammak, Fatma, Leila Baccour, Abdelkarim Ben Ayed, and Adel M. Alimi. "ELECTRE method using interval-valued intuitionistic fuzzy sets and possibility theory for multi-criteria decision making problem resolution." In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6. IEEE, 2017.
[24] Hassan, Muhammad Firdhaus Che, Mohd Uzair Mohd Rosli, and Muhammad Afiq Mohd Redzuan. "Material selection in a sustainable manufacturing practice of a badminton racket frame using Elimination and Choice Expressing Reality (ELECTRE) Method." In Journal of Physics: Conference Series, vol. 1020, no. 1, p. 012012. IOP Publishing, 2018.
[25] Khandan, Mohammad, Maryam Maghsoudipour, and Shahram Vosoughi. "Ranking of working shift groups in an Iranian petrochemical company using ELECTRE method based on safety climate assessment results." Journal of the Chinese Institute of Industrial Engineers 28, no. 7 (2011): 537-542.
[26] Stefanović-Marinović, Jelena, Marko D. Petković, and Ivan P. Stanimirović. "Application of the ELECTRE method to planetary gear train optimization." Journal of Mechanical Science and Technology 29, no. 2 (2015): 647-654.
[27] Govindan, Kannan, and Martin Brandt Jepsen. "ELECTRE: A comprehensive literature review on methodologies and applications." European Journal of Operational Research 250, no. 1 (2016): 1-29.
[28] Chen, Chen-Tung, and Wei-Zhan Hung. "Applying fuzzy linguistic variable and ELECTRE method in R&D project evaluation and selection." In 2008 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 999-1003. IEEE, 2008.
[29] Zandi, Faramak, Madjid Tavana, and David Martin. "A fuzzy group Electre method for electronic supply chain management framework selection." International Journal of Logistics Research and Applications 14, no. 1 (2011): 35-60.
[30] Hatami-Marbini, Adel, and Madjid Tavana. "An extension of the Electre I method for group decision-making under a fuzzy environment." Omega 39, no. 4 (2011): 373-386.