International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Topological Data Analysis for Software Test Cases Generation

Mani Padmanabhan
Associate Professor- SSL
Research Guide - School of Computer Science Engineering and Information Systems
Vellore Institute of Technology (VIT), Vellore, India
e-mail: mani.p@vit.ac.in

Abstract—The escalating development of digital technologies over the last several decades has given rise to an accompanying surge in data analysis.The software based smart applications increase the importance of behaviour analysis. Software testing of expert systems such as electronic health records, health information system and software as a medical device (SaMD) refers to detect the difference between expected behaviour and actual outcome during healthcare expert systems development. Test cases are the core source of effective software testing. Test cases generation for expert systems are discover challenges of identifying the expected behaviour of the system, where the decision logic is obtained via a data –driven paradigm. In the traditional system software object oriented expected behaviour, provide the clear test cases does not change the flow such as in the healthcare system. Intelligent software test case generation approach required for smart health systems. In this research contributes key performance indicators from massive data sets, node –link diagrams from decision trees, test cases from adjacency matrix are elaborated. The experimental results of healthcare expert systems provide empirically evidence that topological data analysis are compact contribution for software requirement validation.

Keywords- Topological data analysis;Software requirement validation; Key Performance Indicators; Test Case Generation

I. INTRODUCTION

A variety of software development methodologies have been created over the past few years to support the creation of effective smart system software that fulfils requirements of ever increasing complexity [1]. While these methodologies have allowed for development of demanding software solutions in healthcare, they were not designed to support the data-driven software solutions which have recently become in high demand. A higher number of expert systems are data-specific process that uses conditional statements to provide the solution in the decision-making field [2]. Data analysis is the process of supporting decision making in the smart systems using activates of filtering, transforming, and modeling of data to discover useful solution. Data sets are useful information to extract decision making conditions.

Measuring the performance of software products using massive data sets create software tester aware of source code-related parameters and to increase test cases through timely control statements in the coding [3]. Software source code for smart systems is appropriate massive data representation. Massive data sets transformation mainly classified into three categories: key performance indicators (KPI) based, tree based, graph based. Every node and links are different tree structure of the source code to a specific data sets in the smart systems.

The virtual data representations can then be inserted in a test cases so that they can use source code as input for test case generation. The software programming are the text encoding representing of source code for smart system. Set of instruction transfer into different representations forms for software requirement validation [7]. The massive data sets can representing in three aspects such as KPI, tree structure and graph. The transferred mapping structure are the representation of the set of instruction. Figure 1 shows the example of Python programming using conditional statement as a running example in this research paper.

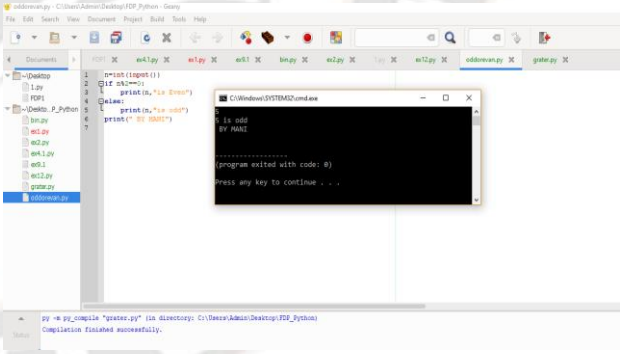


Figure 1. Sample Python code (Control Statement).

Topological Data Analysis (TDA) is to assign topological invariants to data. TDA representation treats code as free text. Thus, it converts the code into a list of key performance indicators, such as alphanumeric characters is consider as KPI (eg. ‘for’, if , ‘else’) . The special character also consider as KPI KPI (eg. ‘{’, “” , ‘( )’). The special character is not the part of the word. The set of instructions for smart system software is mapped to one KPI, then KPI values are resembles a topological representation. Hence the topological flow is often taken into the field of software test case generation.

The second representation captures the decision tree of the source code. Decision trees are a kind of topological representation. In the software test case generation methodology

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

is commonly used the tree representation for transformation of source code. Topological decision trees are a powerful and popular code transformation technique for classification kinds of problems in software testing. A decision tree is a topological data analysis where an internal node corresponds to attributes. The branch (or edge) represents a decision rule, and each leaf node denotes the classification outcome. Test case is a series of input values, implementation prerequisites and expected results, generated for testing objectives viz., to execute path(s) in a program or to check compatibility with software requirement. It requires the execution of a set of test paths in the program under testing. Figure 2 shows the electronic health test path in the topological decision trees classification based on age group. The decision trees classification is adopt in this research paper for test cases generation.

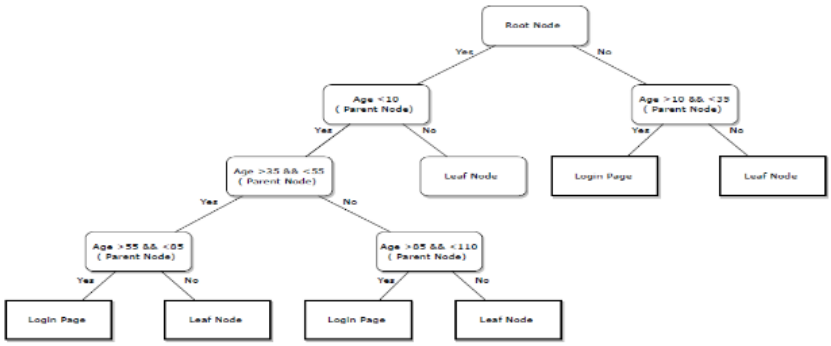


Figure. 2 Test path in the decision trees classification.

Source code to be represents as a graph with different levels. The levels of graph shows the control statements in the programming instruction. Thus, the graph levels is determined by conditional statements, e.g., in the figure 1 shows the if statement for conditional testing of given number in python coding. In the control flow graph, nodes denote statements and conditions in the each level of programming, and directed edges connect them to indicate the transfer of control among the user defined functions.

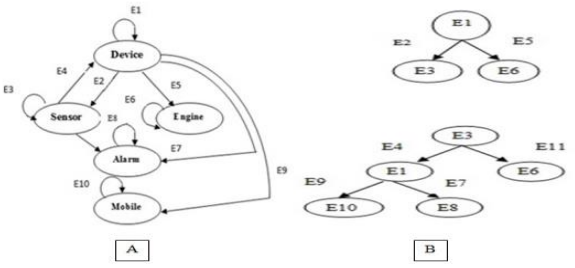


Figure 3. (A) CFG with Self-Loop (B) Control Flow Tree

This research paper is structured as follows. Section 2 is the literature review, In Section 3 describes the healthcare datasets and decision tree algorithm. Node-link extraction methodology. Examines the process of proposed software test case generation using three end-end healthcare software are elaborated with results are described in the graphical representation in section 4. In Section 5 outlines the conclusions and plans for future work on the topics of this paper.

II. Related Work and Research Direction

This section provides a description of the previous research in the field of software test case generation. Relevant systematic review were identified by searching in the e-library database this search string written related to the proposed approach 253 papers in total after removing duplicates research specific relevant paper are identified in total 28 the majority of the papers focused on adaptation of agile methods like global software development. Software test case generation bases on white box testing are limited in the number. Alkhazi et al used multiple objective search for test case selection this approach identity the conditional factor maximum rule coverage and minimum execution time the Pareto algorithm was validated using different transmission of programs but in this proposed research paper aims to be identify the maximum test cases based on the key performance indicators (KPI).Figure 3 describe the control flow graph.

Honfi et al. presented a method for testing the artificial intelligence based robots this method uses optimization for selecting the minimal test cases using method data driven development in this MDD model identity the domain specific language. The prototype tool supports test case identification approach based on domain specific language however in this proposed research paper is focused on applying Artificial Bee Colony (ABC) algorithm for transmission of node –link diagram to adjacency matrix.

Naslavsky et al presented the test case generation approach based on modification to the model rather than extract the source code. The approach identity the connectivity between the model elements using UML class diagram and sequence diagram for extracting the test case. Our previous research work 2018,2019,2022 test case are generates based on specification diagrams for embedded system, test cases for chatbot, test cases for smart system agriculture devices and test cases for web applications are used the model transmissions approach. In past research achieves 98% of test case.

Related traceability approach was implemented by Troya et al this approach for debugging for fault identification in the traceability approach when the test cases failed the approach

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

change the rules according to the fault identification. They purpose an approach for automatically test the ATL model when the rule is executed a traceability can be automatically identified the trace links according to the rules.

The majority of the massive date sets based software development testing are used the specification model and traceability approach covers the code based testing and high coverage criteria in the system software. In this proposed research paper implemented the traceability matrix approach. The source of inputs are optimized using key performance indicators (KPI).

To evaluate the proposed approach in this paper conducted several experiment using three different health care software programmes this section starts with research questions that the experiments attempt to address the research question. The objective of the research question validate the need of presented approach in the emerging field of data sets based health care software development.

RQ1	How does proposal approach select the path for test case generation?
RQ2	What is the size of the time effective in the proposed approach?
RQ3	How cost effective during the regression test selection?

Investing research questions the experiments are tested with the comparison of existing our approach with code coverage cost effectiveness time consuming number of reduced test case. In the experiments result want to ensure that the fault deduction of test case is better than existing methodology in the following section provides test case identification methodology for health care system software, based on our unique approach to achieve the effective optimized test cases using topological data analysis.

III. PROPOSED METHODOLOGY

The proposed research work differs from the above-mentioned research on the following grounds: the proposed algorithm will observe the KPI filter components from massive data sets; then, the node –link flow graph will be consider; and finally, ABC Algorithm based test case generation track the event during the implementation, the bee flows are transferred to the test path to generate the possible test cases for the smart system software. However, the approach tackles the challenges of test case generation of smart system software with undetermined input from massive date sets.

In the massive data sets, the data is defined the flow of application operation. For example, online consultation software as a medical device in healthcare have different modules like, make an appointment, select doctor and select your preferred doctor, proceed with the booking, face –to –face video consultation, electronic health records retrieval, health information system analysis based on data specific metrics by the patient, providing treatment may be allowed or may be prohibited. These algorithms are incorporated into a type of data-driven software. In the parallel operation from same set of data sets need more potential algorithm to solve a critical stage of conditional statements. To accomplish this parallel operation, the proposed research chose to adapt of key performance indicators (KPIs) for anomaly detection approach.

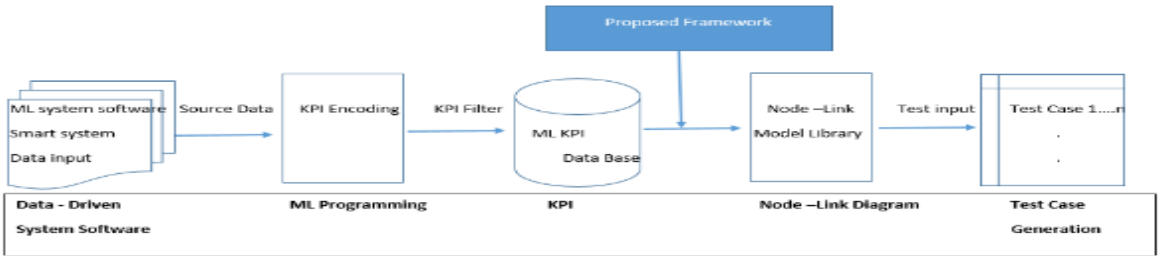


Figure. 4 Proposed approach for software test case generaton

The optimized KPI are well suitable for test case generation, node –link diagram are well-established source in the software testing, by combining a novel specification based test case generation and KPI clustering algorithm that are used for adaptive the intelligent test case generation strategy. Proposed approach for software test case generaton describe in figure 4.

A. KPI Encoding and filtering

Smart systems services produce a massive number of data sets in recent years. Key performance indicators (KIP) filtering is the necessary from the parallel data sets. Define parallel operation of KPI source data sets as the graph adjacency matrix A, its defines the direct sum K= QJQ^{-1} where Q= $[Q_i,Q_i\ldots\ldots Q_n]$, i=1,2,3…N are eigenvectors of adjacency matrix A.

$$\text{Matrix of GFT (Q)} = \begin{matrix} n1.. & 0 \ldots & 0 \\ 0 \ldots . & n2 & 1. \\ 0 \ldots . & 0 \ldots & n3 \end{matrix}$$

Algorithm 1: KPI Model library Mining

1: Calculate the graph adjacency matrix A
2: Identify the Graph Fourier transform of A
3: Select the first Node Q1

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

4: Q= Q1….Qn
5: **While** new node from KPI **do**
Calculate the detection error (Ne) for new GFT
Calculate new GFT node until the error =0
6: Detect the GFT model and store as the optimizing KPI library.
7 : **If** new node are normal **then**
Update the GFT model
8: **End if**
9: **End while**

KPI parallel identification and mining algorithms are based on topological data analysis. In this proposed algorithm 1 calculate the adjacency matrix from massive KPI. And select the first node and process it as Q1. The node is validated using new node condition and change to topological graph (GFT). Detect the GFT model and store as the optimizing KPI library. The GFT is updated until all the node visited in massive date sets for node-link extraction.

In this research paper propose a node-link framework, an efficient node-link graph conversion framework from massive data sets. The proposed framework consists of two stages: transfer KPI and duplicate date sets detection. GFT model undergo for the anomaly detection using clustering process. The parameters of effective (E) , deployed state (D) and termination stage(s) will be calculating the time leg (Q) for the node to node connection in the topological tree structure. In the clustering process, we extract NL models, split them into segments, and utilize Node link diagrams with hierarchical agglomerative clustering. Proposed framework for node-link extraction shows in the figure 5.

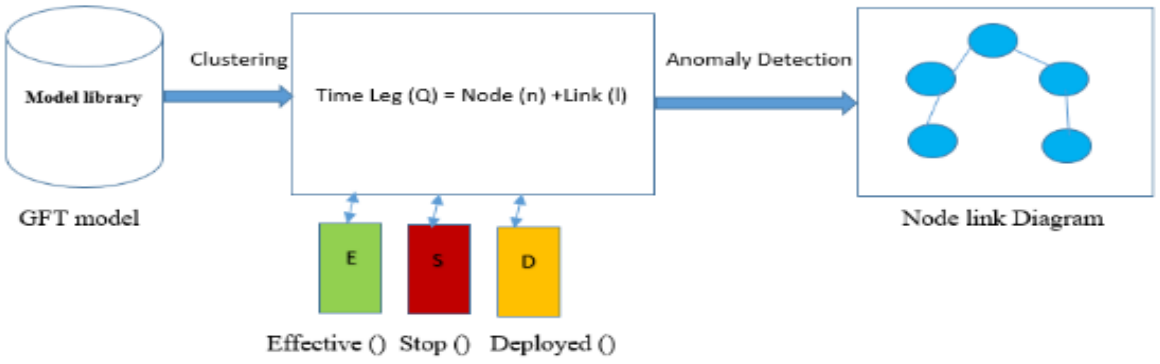


Figure. 5 Proposed framework for node-link extraction

Finally, the node link extract stage hyper parameter with topological data calcification to calculate the conditional statement in the second stage of the framework. The unique node –link topological structure is the core source for the test case generation methodology.

B. *Transformation of Node link Graph*

The first process of graph approach is to convert the topological node link to test matrix. Node link graph provides a moderate extraction of the path, branch distance and time leg with combination of nodes and edges. The Cyclomatic complexity provide the approximated number independent path. V(G) denote the Cyclomatic complexity of the topological KPI structure. E describe the number of edges in the topological KPI and N denotes number of nodes in the topological KPI tree mining.

$$V(G) = E - N + 2 \quad (1)$$

$$NC = CCi \cap CCj \quad (2)$$

Here CCi denotes the Cyclomatic complexity of the effective date sets and CCj describe the Cyclomatic complexity of the deployed data sets. NC is a metric of approximated statistics value of independent new test cases during software testing. To find the accurate number of test cases Adjacency matrix to be formulated. The below flow chart describe the process of adjacency matrix creation from node-link graph. The topological KPI node link is formulated based on the proposed algorithm the adjacency matrix denoted as Adj(Matrix of GFT).

$$Adj(Q) = \begin{matrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

The adjacency matrix is generated from node-link diagram contains all the test cases. All these test cases are equalized with pre-condition, data, post condition.

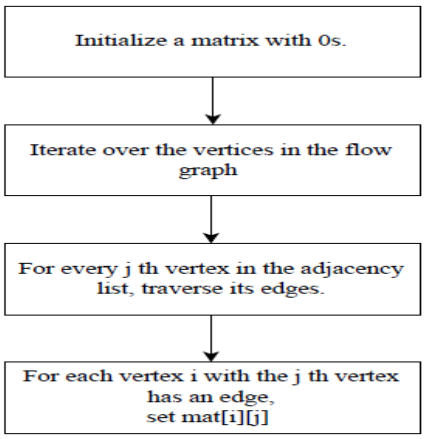
C. *Test cases generation*

Test case expected output generation is often made more difficult by decisions made during requirements specification and software design. A method of assessing the testability of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

requirements specifications based on adjacency matrix help to further reduce the time and cost of software testing.

A single objective finder need in the adjacency matrix that is artificial bee colony (ABC) algorithm, the random searches in the adjacency matrix carried out by employee bee. The actual outcome and expected outcome are identify by onlooker bee. The scout bee occurred the new path.



Algorithm 2: ABC Algorithm based test case generation

Step 1: Initialization of programming flow

Step 2:Scout bee are search the node (0,0) for new space

Step 3: Onlooker bee are save the flow in the expected outcome on their respective path.

Step 4: Track possible node with employed bee.

Step 5: Save the position trace (path) for test cases selection (precondition, data and post condition)

Step 6: Repeat until all the node are visited.

The proposed test case generation methodology for validating the software requirement has its advantage over other methodology. The experimental evaluation for three end –end health care software programming code in python are provides the accuracy of the proposed methodology in the next section.

IV. Experimental Results and Discussion

To demonstrate the proposed approach, three phases of health care expert system programing instructions are pre-owned. Python programming instructions has been applied for the extraction of KPI. The methodology to be divvied in the five phases of execution. The objective of the experiments provided in the table 1.

First phase KPI filter is executed. In the second phase node -link diagram extraction. Phase three adjacency matrix to be formulated. Fourth phase test cases are generated. In the final phases software requirement has been validated. In this five phases provide the full code coverage in the health care expert systems.

Table 1 Experiments and Objectives

Phase NO.	Automatic health care expert system	Objectives
1	Automated body monitoring system	Automated body monitoring system for human body identify the temperature and blood pressure.
2	Smart chatbot Discussion	If the limit exceeded chatbot dissuasion will be enable and ask for the hospital booking.
3	Hospital Appointment booking	Based on the past record of the patient nearest hospital has been booked with the stored record.

The datasets form automated body monitoring system are converted as the key performance indicators (KIP) based on the proposed approach. In the node-link diagram have the E1 to E14 unique datasets in the healthcare system. The ABC algorithm based node to be extracted for test cases. In the ABC algorithm based node provide the pre and post condition. Figure 6 shows the Sample code for AI based body monitoring system.

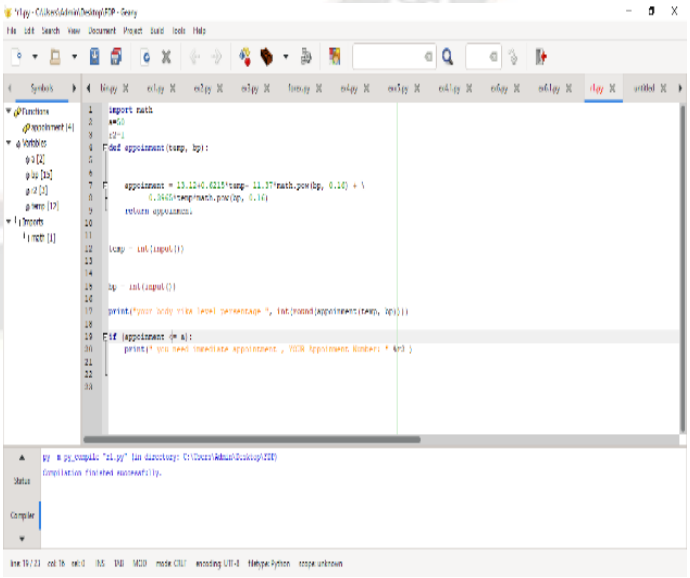


Figure 6 Sample code for AI based body monitoring system.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Table 2 **KPI Encoding and filtering**

Data Sets ID	Test Scenario	Test case ID	Precondition(Q1)	Input(new node)	Output(GFT)
Path _1 : Temperature identification	Input value : Temperature	Test Case1	Waiting Signal	identify the limit	Compare the limit .
Path _2 : BP monitoring	BP range	Test Case2	Sensor Input	Sensor IP(x)	Speed cal(x)
Path _3 : Record searching	Search based on unique ID	Test Case3	Get signal(x)	Sorting	Capture/ find
Path _4 : Smart chatbot discussion	Limit trigger: voice all	Test Case4	Message	If message > yes	Call message
Path _5 : Hospital appointment booking	Message:/Accept/ Reject	Test Case5	Get signal(x)	If (Accepted) booking; Else; Wait()	Store the record

TABLE 3(a) Test Cases for Path _1

Test case ID	Node	Pre	Data	Post
TS1	(0,0)	E1	Trace1= (M>10)	E2
TS2	(3,1)	E2	Trace 2=(M<10)	E3
TS3	(0,5)	E5	Invalid data	E7

TABLE 3(b) Test Cases for Path _2

Test case ID	Node	Pre	Data	Post
TS1	(0,0)	E4	Trace1= (bP>90)	E5
TS2	(5,1)	E6	Trace 2=(bp<130)	E7
TS3	(1,6)	E8	Bp(<40>50)	Invalid data

TABLE 3(c)Test Cases for Path _3

Test case ID	Node	Pre	Data	Post
TS1	(3,0)	E1	Trace1= (ID>' ')	E2
TS2	(1,5)	E2	Trace 2=(X<10)	E6
TS3	(1,2)	E5	ID not match	E8

TABLE 3 (d) Test Cases for Path _5

Test case ID	Node	Pre	Data	Post
TS1	(4,0)	E1	Trace1= (Num.>10)	E4
TS2	(2,1)	E2	Trace 2=(Num.<10)	E4
TS3	(3,2)	E7	Book ID	E2

The following adjacency matrix based sample test cases are provided in order to assess the outcome of the experiments and hence they were recorded for every path in the healthcare system. Table 2 provide the KPI Encoding and filtering. Table 3 shows the sample test cases and table 4 are the Node –link extraction for Path 1 to Path 3. The temperature identificaton based on the proposed algorithms of ABC emplyee bee saved the position trace (path) for test cases selection (precondition, data and post condition) are provided in the table 3

Table 4 Node –link extraction for Path 1 to Path 5

Data sets: ID	ABC based node visit
Path _1 : Temperature identification	E2→E5→E9→E7→E4→E11
Path _2 : BP monitoring	E5→E6→E7→E8→E4→E11
Path _3 : Record searching	E5→E6→E7→E8→E11

In order to verify the performance of the proposed approach in this research paper, Three different healthcare

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

experiments were conducted with five path. Literature shows that the Control Flow Graph (CFG) testing is more robust than testing like statement and branch, but it has some limitations too. Testing all paths in a complex program is practically difficult to achieve. Instead of testing all paths, testing a subset of paths based on coverage criteria helps in achieving the testing goal with lesser efforts required to be put in.

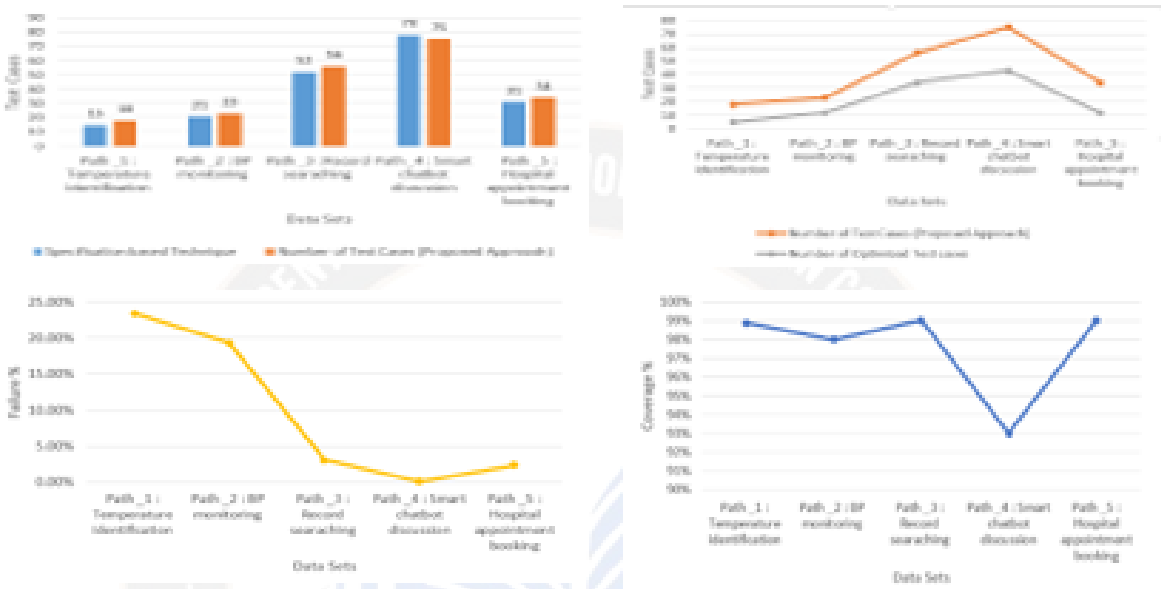


Figure. 7 Effitiveness of proposed approach.

The proposed approach for generating test case based on topological data analysis and ABC alogorthm using node – link exteractoin approach yields efficient test cases. It also shows how to created Node –link extraction for saved the position trace (path) from KPI provded more code coverage. Effitiveness of proposed approach provided in figure 7. Some concluding observations from the research are given below.

- The proposed algorithm is very effective for converting KPI Model library Mining.
- The method for test case generation based on ABC algorithm achieves high test coverage from massive data sets.
- It has been illustrated with the various experiments of smart systems by using massive data sets which further shows the behaviour of the system.
- The outcome of the proposed approach indicates the possible test path to generate the test cases for smart sysems.

It is observed that the proposed approach with validation technique, presented in this paper is able to generate test cases with high requiremnt coverage, optimized test cases and low failure test cases. The detailed concluding observations from the experiments are given in the next session.

VI Conclusion

In this research contributes topological data analysis using key performance indicators from massive data sets, node –link diagrams from decision trees. Effective test cases from adjacency matrix are elaborated. The proposed approach provide the node-link based test case generation. Proposed methodology to select a subset of paths (known as unique paths) from the set of all paths of a smart systems. The experimental results of healthcare expert systems provide empirically evidence that optimized data sets are a compact contribution for software requirement validation. Test case generation based on topological data analysis technique achieves 99.9% of the code coverage. In future, the test case generation of proposed work could be tried with other datasets with fully automate process.

References

[1] L. V. Sartori, H. Waeselynck, and J. Guiochet, “Pairwise Testing Revisited for Structured Data With Constraints,” in 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland: IEEE, Apr. 2023, pp. 199–209. doi: 10.1109/ICST57152.2023.00027

[2] G. H. Subramanian, P. C. Pendharkar, and D. R. Pai, “An Examination of Determinants of Software Testing and Project Management Effort,” Journal of Computer Information Systems, vol. 57, no. 2, pp. 123–129, Apr. 2017, doi: 10.1080/08874417.2016.1183428.

[3] A. Nayak and D. Samanta, "Automatic Test Data Synthesis using UML Sequence Diagrams.," The Journal of Object Technology, vol. 9, no. 2, p. 115, 2010, doi: 10.5381/jot.2010.9.2.a2.

[4] M. Chen, P. Mishra, and D. Kalita, "Efficient test case generation for validation of UML activity diagrams," Design Automation for Embedded Systems, vol. 14, no. 2, pp. 105–130, Jun. 2010, doi: 10.1007/s10617-010-9052-4.

[5] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege, "A Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation," IEEE Transactions on Software Engineering, vol. 36, no. 6, pp. 742–762, Nov. 2010, doi: 10.1109/TSE.2009.52.

[6] M. S. AbouTrab, M. Brockway, S. Counsell, and R. M. Hierons, "Testing Real-Time Embedded Systems using Timed Automata based approaches," Journal of Systems and Software, vol. 86, no. 5, pp. 1209–1223, May 2013, doi: 10.1016/j.jss.2012.12.030.

[7] Mani Padmanabhan and M. Prasanna, "Validation of automated test cases with specification path," Journal of Statistics and Management Systems, vol. 20, no. 4, pp. 535–542, Jul. 2017, doi: 10.1080/09720510.2017.1395173.

[8] G. H. Subramanian, P. C. Pendharkar, and D. R. Pai, "An Examination of Determinants of Software Testing and Project Management Effort," Journal of Computer Information Systems, vol. 57, no. 2, pp. 123–129, Apr. 2017, doi: 10.1080/08874417.2016.1183428.

[9] P. M. Jacob and P. Mani, "Software architecture pattern selection model for Internet of Things based systems," IET softw., vol. 12, no. 5, pp. 390–396, Oct. 2018, doi: 10.1049/iet-sen.2017.0206.

[10] M. Azizi and H. Do, "Graphite: A Greedy Graph-Based Technique for Regression Test Case Prioritization," in 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Memphis, TN: IEEE, Oct. 2018, pp. 245–251. doi: 10.1109/ISSREW.2018.00014.

[11] Mani Padmanabhan, "Rapid medical guideline systems for COVID-19 using database-centric modeling and validation of cyber-physical systems," in Cyber-Physical Systems, Elsevier, 2022, pp. 161–170. doi: 10.1016/B978-0-12-824557-6.00012-1.

[12] Mani Padmanabhan, Prasanna M, "Test Case Generation For Embedded System Software Using UML Interaction Diagram", Journal of Engineering Science and Technology, Vol.12, NO.4, pp. 860-874, 2017.

[13] Mani Padmanabhan," Test Case Generation for Arduino Programming Instructions using Functional Block Diagrams. Trends in Sciences, 19(8), 3472. https://doi.org/10.48048/tis.2022.3472.

[14] Mani Padmanabhan, Prasanna M, "Test Case Generation for Real-Time System Software using Specification Diagram", Journal of Intelligent Engineering and Systems, Vol.10, No.1, pp. 166 – 175, 2017

[15] Mani Padmanabhan, "Regression Test Case Optimization Using Jaccard Similarity Mapping of Control Flow Graph," in Innovations in Computational Intelligence and Computer Vision, Lecture Notes in Networks and Systems, vol. 680. , Singapore: Springer Nature Singapore, 2023, pp. 545–558. doi: 10.1007/978-981-99-2602-2_41.

[16] A. Shahbazi and J. Miller, "Black-Box String Test Case Generation through a Multi-Objective Optimization," IEEE Transactions on Software Engineering, vol. 42, no. 4, pp. 361–378, Apr. 2016, doi: 10.1109/TSE.2015.2487958.

[17] G. H. Subramanian, P. C. Pendharkar, and D. R. Pai, "An Examination of Determinants of Software Testing and Project Management Effort," Journal of Computer Information Systems, vol. 57, no. 2, pp. 123–129, Apr. 2017, doi: 10.1080/08874417.2016.1183428.

[18] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," Knowledge-Based Systems, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.