*Journal of Information & Optimization Sciences*
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 45 (2024), No. 4, pp. 1189–1197
DOI : 10.47974/JIOS-1702

# Recent trends on data flow testing : A review

Ankita [§]
*Department of Information Technology*
*Indira Gandhi Delhi Technical University for Women*
*Delhi 110006*
*India*

Preeti Rathee [†]
*Department of Information Technology*
*Maharaja Surajmal Institute of Technology*
*Janakpuri*
*New Delhi 110058*
*India*

Nisha Rathee*
Pooja Gera [‡]
Abhigya Verma [@]
Nishtha Goyal [#]
*Department of Information Technology*
*Indira Gandhi Delhi Technical University for Women*
*Delhi 110006*
*India*

**Abstract**

The research aims to comprehensively examine the current state of knowledge in data flow testing (DFT) to identify knowledge gaps and inform future research. The authors undertook this research question to advance the practice of DFT and improve software systems' quality and reliability. The authors analyze several state-of-the-art techniques, including the correlation tree concept and particle swarm optimization algorithm, leveraging the data flow knowledge during test execution, as well as neural networks and genetic algorithms. The authors also discuss the methods used to evaluate the effectiveness and

[§] *E-mail:* **ankita@igdtuw.ac.in**
[†] *E-mail:* **preetirathee09@gmail.com**
* *E-mail:* **nisharathee@igdtuw.ac.in (Corresponding Author)**
[‡] *E-mail:* **pooja005btit19@igdtuw.ac.in**
[@] *E-mail:* **abhigya006btit19@igdtuw.ac.in**
[#] *E-mail:* **nishtha025btit19@igdtuw.ac.in**

accuracy of the various techniques, including case studies, simulations, and test data-generating techniques. The authors aim to provide a superficial understanding of the field's current state and note that future work could focus on in-depth analysis of specific areas within DFT. Future researchers can use this research to gain a deeper understanding of current algorithms and work on improving them. This is particularly important as DFT is an essential part of the testing process for any software.

***Subject Classification:*** *68U99.*

***Keywords:*** *Software development, Software testing, Data flow testing, Algorithms.*

## 1. Introduction

DFT is one of the subcategories of Structural Testing [1]. DFT is a methodology that analyses data progression throughout a system, aiming to identify and rectify defects and bugs. A fundamental component of DFT is the utilization of "du-paths" and "du-pairs," which pertain to the correlation between the definition and usage of variables within a program. A du-path constitutes a sequence of statements in which a variable is defined and subsequently employed, while a du-pair represents a specific instantiation of a definition and usage of a variable. By analysing these associations, DFT can detect potential defects, such as utilising uninitialized variables or manipulating memory in an insecure manner.

## 2. Research Methodology

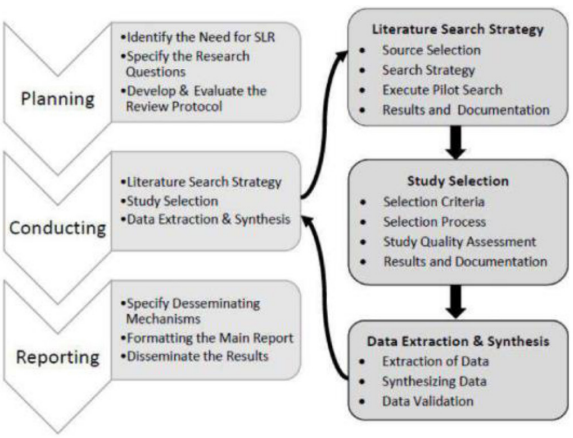


**Figure 1**

**The Process Depicting the SLR Method**

2.1 *Search Strategy*

Figure 1 shows the SLR process and Table 1 shows the repositories and their corresponding search strings.

**Table 1**

**Repositories and their corresponding search strings.**

| Publication | Research String | Paper Count |
|---|---|---|
| Springer | "software testing" | 1902359 |
| | software AND testing, AND data AND flow AND testing filter: 2012-2022 | 471428 |
| | "data flow testing" within 2012 - 2022 | 137 |
| | "data flow testing"' within Conference Paper 2012 - 2022. | 57 |
| | "data flow testing" in title within 2012 - 2022 | 7 |
| IEEE | Searched "software testing" | 100961 |
| | "data flow testing" | 11170 |
| | "("All Metadata": software testing) AND ("All Metadata": data flow testing) | 1602 |
| | ("Document Title": data flow testing) AND ("All Metadata":software testing) AND Filters Applied: 2010-2023 | 25 |
| | Final Abstract based shortlisting | 8 |
| ACM | "software testing" | 489180 |
| | [Title: data flow testing] AND [Publication Date: (01/01/2012 TO 12/31/2022)] | 28158 |
| | [Title: data flow testing] AND [Publication Date: (01/01/2012 TO 12/31/2022)] with "short paper" filter | 1835 |
| | [Title: data flow testing] AND [Abstract: techniques] AND [Publication Date: (01/01/2012 TO 12/31/2022)] with "short paper" filter | 337 |
| | [Title: "data flow testing"] AND [Publication Date: (01/01/2012 TO 12/31/2022)] | 5 |
| | Final Abstract based shortlisting | 4 |
| Science Direct | "software testing" | 7210 |
| | "data flow testing" | 204 |
| | [Title, abstract, keywords:"data flow testing"] AND [Year:2012-2022] | 5 |
| | Final Abstract based shortlisting | 2 |

A trial search was conducted on the selected online databases. By retrieving the testing process - sample papers, it was confirmed that the search results are relevant and comprehensive. The number of outcomes from the pilot searches gauged the labour needed for the full search procedure. The literature from 2012 until January 31, 2022, was extracted

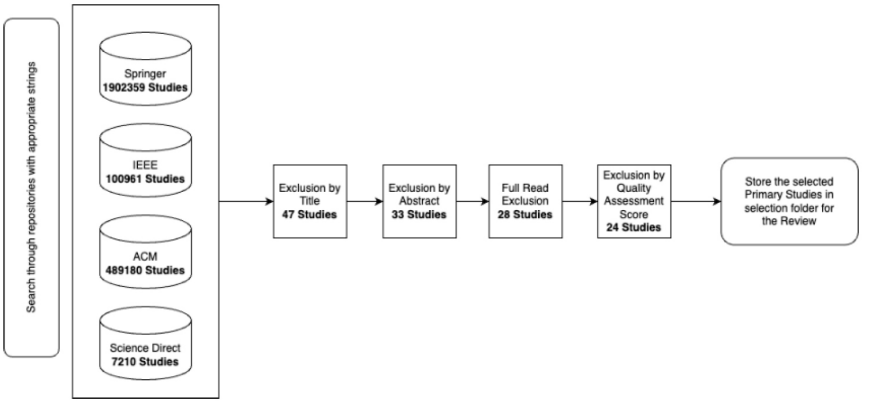


**Figure 2**

**Search and Selection Procedure**

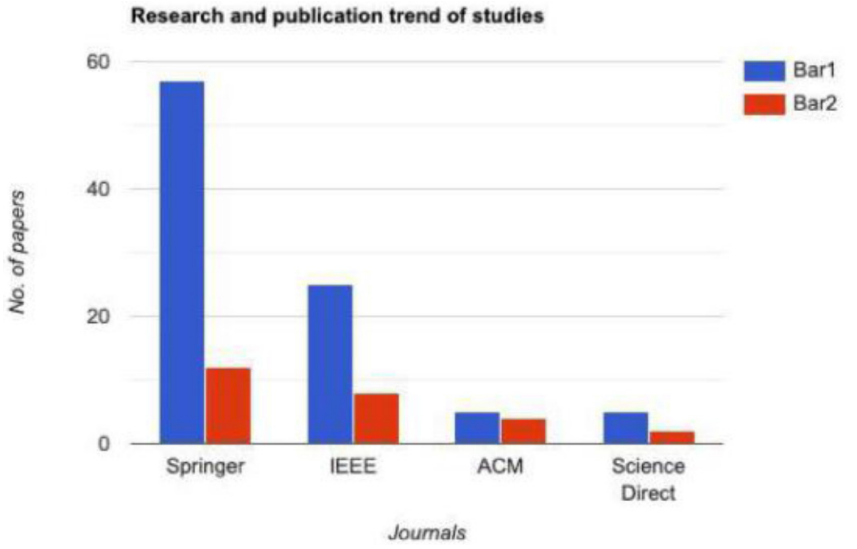


**Figure 3**

**BarPlot Depicting the initial search result counts v/s the papers considered from it for every publication [9]**

using four distinct search strings across four different databases. Many studies from the repositories were discovered after searching the research papers by title, abstract, keywords, and bibliographical data. Figure 2 shows the search and selection process. Figure 3 depict the initial search result counts v/s paper considered for the publication.

## 3. Background: Brief Review of Studies

In this section, all the studies and methodologies that were finally selected are briefly discussed to understand the status of DFT.

In [2], the author explores the possibility of intertwining two parts of DFT - generating test cases automatically and dynamic identification of data-flow elements. [3] focuses specifically on performing DFT on XACML requests. XACML policies are utilized specifically to limit or provide access to resources in a piece of software. [4] explores DFT on aspect-oriented programs. In [5], DFT has been used not only on one piece of code as it is done in the traditional approach but also on pieces of code running simultaneously to identify all the definitions and use associations for the variables of the program in question. [6] have introduced a novel approach, "Swine Influenza Models Based Optimization (SIMBO)". The intuition behind this approach is that when control flow graphs are developed, they are designed as a spanning tree where the nodes represent the test cases that might exist during the testing process. [7] introduces the gravitational search algorithm (GSA) for optimizing structural test paths inspired by gravitational forces. [8] proposes utilising data flow-based coverage criteria for black-box integration testing of distributed software systems.

## 4. Research Questions (RQ)

Based on a preliminary study of the papers, it was decided to analyze the current work. The same is discussed in below sections.

### 4.1 *RQ1: What are the different data flow testing techniques being used in the recent years?*

Observed that several state-of-the-art techniques have been used by the authors of research work selected studies were analyzed. The following techniques were found to being used for DFT:

- Gravitational search algorithm for optimization of test paths [7]
- Correlation tree concept and particle swarm optimization algorithm [9]
- Leveraging the data flow knowledge during test execution [8]
- Neural networks and genetic algorithms [10]
- Optimal histogram technique and attribute mutation methodology [11]
- DefUse data flow technique with SMT solver [12]

4.2 *RQ2: How the research effectiveness and accuracy has been evaluated?*

Various techniques used in different studies have different use cases. Based on the use cases, all the authors have used other methods to test their algorithms, which are relevant to their use cases and work. Following is a brief description of the methods of evaluation for the studies:

- In [13], the case of a safety injection system is used to analyze the FPCCTestGen scheme and its effectiveness in test case generation.
- In the study [4], only aspect-oriented (AO) programs were used to analyze the effectiveness of the proposed technique. The primary reason is that the model and method are proposed only for this program; hence, testing it on only those programs makes much sense.
- [14] used Java byte-code programs to simulate their approach and analyze its performance.
- The authors of [5] used a Java concurrent software called FactorySim (FS). This programme simulates the production of a limited number of objects a client requests.
- In the study by Singla et al. [6], a newly proposed approach's performance was evaluated on multiple programs using test data generation techniques such as SIMBO, GA, and PSO. The efficacy of the optimization algorithms SIMBO, GA, and PSO was appraised by utilising the principles of dominance trees and spanning trees.

4.3 *RQ3: What is the overall significance of the research work, and how will it affect the time to come?*

The present study endeavours to extensively explore the extant body of knowledge about DFT to identify lacunae and inform the trajectory of

future research endeavours. This inquiry aims to foster the advancement of DFT practices and augment the robustness and dependability of software systems.

- The problem of scalability is being tackled by the automatic generation of test cases [2].
- Fault tolerance for XACML policies can be assessed by generating automatic test cases and utilising them with the X-CREATE framework [3].
- Random test case generation for aspect-oriented programming by focusing on the associations between the aspects as in control flow graphs can help identify faults [4].
- Bitwise algorithms and efficient data structures can reduce RAM utilisation and decrease testing costs [14].
- Utilising a minimum spanning tree to decrease the number of tests for DFT[6].

## 5. Discussion and Conclusion

DFT is one of the essential parts of the software testing process. The SLR was an attempt to superficially understand the various techniques and algorithms developed so far in the field of DFT; hence, an in-depth analysis of a particular field inside it was done.

In the future, work can be done to understand the research and algorithms developed for a specific area in the DFT rather than considering the whole area as such. This will help in getting more details and in-depth knowledge of the research work done so far. Future researchers can use this research to understand and work on the current DFT algorithms.

## 6. Conflict of Interest

The lead author asserts that no conflicting interests are present on behalf of all contributors.

## References

[1] Ntafos, S.C.: A comparison of some structural testing strategies. *IEEE Transactions on Software Engineering* 14(6), 868–874 (1988).

[2] Vivanti, M.: Dynamic data-flow testing. In: Companion Proceedings of the 36th International Conference on Software Engineering, pp. 682–685 (2014).

[3] Limaye, S., Zhang, Y.: Combining algorithm-based data flow testing approach for xacml. In: Proceedings of the Third ACM Workshop on Attribute-Based Access Control, pp. 25–31 (2018).

[4] Wedyan, F., Ghosh, S., Vijayasarathy, L.R.: An approach and tool for measurement of state variable based data-flow test coverage for aspect- oriented programs. *Information and Software Technology* 59, 233–254 (2015).

[5] Souza, P.S., Souza, S.S., Rocha, M.G., Prado, R.R., Batista, R.N.: Data flow testing in concurrent programs with message passing and shared memory paradigms. *Procedia Computer Science* 18, 149–158 (2013)

[6] Singla, S., Kumar, R., Kumar, D.: Natural computing for automatic test data generation approach using spanning tree concepts. *Procedia Computer Science* 85, 929–939 (2016).

[7] Rathee, Nisha, and Rajender Singh Chhillar. "Gravitational search algorithm: A novel approach for structural test path optimization." *Journal of Interdisciplinary Mathematics* 23.2 : 471-480 (2020).

[8] Hellhake, D., Schmid, T., Wagner, S.: Using data flow-based coverage criteria for black-box integration testing of distributed software systems. In: 2019 12th Ieee Conference on Software Testing, Validation and Verification (icst), pp. 420–429. IEEE (2019).

[9] Singh, Vijander, et al. "Prediction of COVID-19 corona virus pandemic based on time series data using Support Vector Machine." *Journal of Discrete Mathematical Sciences and Cryptography* 23.8 : 1583-1597 (2020).

[10] Ji, S., Chen, Q., Zhang, P.: Neural network-based test case generation for data-flow oriented testing. In: 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), pp. 35–36. IEEE (2019).

[11] Zhang, Y.-P., Zhang, B.-B., Wang, Z.-W.: Research of a data flow test method for attribute-based access control systems. In: 2017 Interna-

tional Conference on Network and Information Systems for Computers (ICNISC), pp. 167–171. IEEE (2017).

[12] Bhatnagar, Vaibhav, et al. "Descriptive analysis of COVID-19 patients in the context of India." *Journal of Interdisciplinary Mathematics* 24.3 : 489-504 (2021).

[13] Wu, Y.-C., Fan, C.-F.: Automatic test case generation for structural testing of function block diagrams. *Information and Software Technology* 56(10), 1360–1376 (2014).

[14] Chaim, M.L., De Araujo, R.P.A.: An efficient bitwise algorithm for intra- procedural data-flow testing coverage. *Information Processing Letters* 113(8), 293–300 (2013).