# Bibliometric Analysis: Software Testing Based on Graph Theory

Chaimae Elasri, Fadwa Saoiabi, Nassim Kharmoum, and Soumia Ziti

*Abstract*—**As software systems become increasingly complex and development cycles accelerate, ensuring their quality becomes progressively challenging. Given its ability to identify errors and deficiencies early in the development process, software testing emerges as a cornerstone of software engineering. Traditionally performed manually, testing has been plagued by inefficiencies, inaccuracies, and resource-intensive procedures, leading to significant costs in time, money, and effort. However, contemporary testing practices have shifted towards automation, enabling testing professionals to achieve reliable results while minimizing resource expenditure. The adoption of Test-Driven Development (TDD) further reinforces this trend, advocating the creation of tests before code implementation. Integrating graph theory techniques into testing methodologies holds promise for generating intelligent test cases, thereby enhancing test quality, coverage, and precision. This study uses bibliometric analysis of intelligent software testing papers spanning from 2013 to 2023, employing tools like Biblioshiny and VOSViewer to extract insights into authorship, publications, and collaborative networks in the domain of software testing based on graph theory. Using scientific advancements and collaborative trends, this study aims to provide a global overview of developments in software testing, providing valuable insight to software engineers, architects, and researchers in the field.**

*Index Terms*—**software testing, graph theory, bibliometrics, bibliometrix package, R, biblioshiny application, VOSviewer**

## I. Introduction

SOFTWARE testing [1] is a critical aspect of software development, ensuring that software systems meet their intended specifications and perform reliably in various scenarios. Traditional testing methods often rely on exhaustive testing of individual components or paths [2] within the software. However, as software systems become more complex, these conventional approaches become increasingly inefficient and impractical.

In response to this challenge, the integration of graph theory [3] principles into software testing methodologies [4] has emerged as a promising approach. Graph theory provides a mathematical framework for modeling complex relationships and dependencies within software systems. By representing software structures and behaviors as graphs [5], testers can analyze and evaluate system functionalities more comprehensively and effectively.

This combination of software testing with graph theory offers several advantages [6]. It enables testers to identify critical paths, detect potential errors, and prioritize testing efforts based on the structural properties of the software [7]. Additionally, graph-based testing techniques facilitate the generation of test cases [8] that cover a wide range of scenarios, including both common and edge cases [9], thereby enhancing test coverage and fault detection capabilities.

In this introductory exploration of software testing based on graph theory, we delve into the fundamental concepts, methodologies, and applications of this innovative approach. We examine how graph representations can capture the intricacies of software systems [10], facilitate test case generation [11] and execution, and ultimately improve the quality and reliability of software products. Through a deeper understanding of the interaction between software testing and graph theory, our aim is to empower practitioners with valuable information and tools to navigate the complexities of modern software development [12].

Thoroughly conducted bibliometric analyses offer academics a comprehensive understanding of their research domain. They not only pinpoint areas requiring further investigation but also inspire fresh research endeavors while substantiating scholars' contributions. In addition, such studies provide a solid foundation for advancing the field in innovative directions. Leveraging bibliometric software such as Gephi, Leximancer, and VOSviewer streamline data analysis, contributing to the recent surge in scholarly interest in bibliometrics. Notably, the accessibility of vast bibliometric datasets through scientific databases like Scopus and Web of Science has greatly facilitated research in this area [13].

We are attempting to address the following research questions:

Q1. How many citations does each scientific publication typically have?

Q2. Which papers receive the most citations?

Q3. Which authors receive the most citations?

Q4. In this field, which nations produce the most research output?

Q5. Which nations typically work together?

Q6. How do nations, writers, and research papers relate to one another?

Q7. How much does research grow each year?

Q8. Which organizations and writers have made the biggest contributions to the field?

Q9. Which journals are the most popular and frequently cited?

Q10. Which institutions or groups typically work together?

Q11. Which research themes or topics are related?

Manuscript received October 8, 2024; revised July 8, 2025.

Mrs. Chaimae Elasri is a PhD candidate in the Department of Computer Science, Intelligent Processing Systems & Security (IPSS) Team, Faculty of Sciences, Mohammed V University in Rabat, Morocco (corresponding author to provide phone: 212-650422676; e-mail: elasrichaimae657@gmail.com).

Mrs. Fadwa Saoiabi is a PhD candidate in the Department of Computer Science, Intelligent Processing Systems & Security (IPSS) Team, Faculty of Sciences, Mohammed V University in Rabat, Morocco (e-mail: fadwa.saoiabi123@gmail.com).

Prof. Nassim Kharmoum is a professor in the Department of Computer Science, Intelligent Processing Systems & Security (IPSS) Team, Faculty of Sciences, Mohammed V University in Rabat, Morocco (e-mail: nkharmoum@gmail.com).

Prof. Soumia Ziti is a professor in the Department of Computer Science, Intelligent Processing Systems & Security (IPSS) Team, Faculty of Sciences, Mohammed V University in Rabat, Morocco (e-mail: ziti.soumia@gmail.com).

These research inquiries offer valuable insights into the evolution, productivity, and patterns within research output. Through the examination of citation impact across researchers, articles, and journals, scholars can discern the most influential contributions in their respective fields, guiding funding allocations and shaping future research endeavors. Furthermore, insights into collaboration dynamics among authors and organizations facilitate the identification of fruitful partnerships, enhancing research outcomes. These questions assist inform strategic research strategies and investment decisions by shedding light on new research subjects and identifying areas that need more investigation or financing, in addition to providing information on the effectiveness and influence of research output in various nations and regions.

This paper advances the subject of software testing both theoretically and practically by drawing on graph theory. In theory, it achieves three key contributions: Firstly, it discerns research trends by identifying prevalent topics, highly cited articles and authors, and influential journals, enabling researchers to find areas where the literature is lacking and shape upcoming studies agendas. Secondly, it presents a bibliometric methodology designed specifically for examining software testing studies from a graph theory perspective, which can be adapted to other fields for trend identification and literature gap analysis. Thirdly, it unveils collaboration networks among authors, institutions, and countries, fostering potential collaborations and knowledge exchange across diverse regions. On a practical level, the study pinpoints key research areas, aiding scholars in prioritizing their research efforts, and highlights influential researchers and institutions, offering valuable insights for potential collaboration opportunities.

The subsequent sections of this article are organized as follows: Section II will introduce the relevant literature in the field under study. Following this, section III will outline our data collection process and query methodology. Section IV will detail the research methodology employed in this study. The findings of our investigation will be presented in section V, while section VI will dive into a discussion of these results. Finally, section VII will present the conclusion drawn from our research.

## II. Related works

This bibliometric analysis paper [14] focuses on intelligent software testing, with an emphasis on global research production and influential researchers. VOSviewer and the Biblioshiny program were employed for data analysis and visualization. Key findings indicate a yearly growth rate of 14.87% an average citation rate of 6.54, and China, India, and the United States as top contributors. The field exhibits high collaboration (1.9% single-authored papers). Prominent sources include "IEEE Transactions on Software Engineering" and "Lecture Notes in Computer Science." The study highlights machine learning and deep learning, specifically neural networks and NLP, as heavily utilized techniques. Future research avenues include exploring other AI techniques like genetic algorithms and fuzzy logic for enhancing software testing. Additionally, there is potential for AI applications in various testing types beyond those explored in the study, such as security and usability testing. The findings provide guidance for future lines of inquiry in the ever-evolving discipline of software engineering.

Most studies contribute focusing their analysis on a subdomain of intelligent software testing. While other scholars have approached this topic through systematic literature reviews [15], [16], or simple literature reviews [17], [18], our focus distinguishes us. A broader perspective reveals articles dedicated to summarizing bibliometric data, such as [19], [20]. Some scholars have published systematic reviews focusing on specific software test types, like higher-order mutation testing [21], or software testing in general, as seen in [22]. Some researchers have delved into reviews aimed at synthesizing indexes to facilitate the extraction of pertinent data from secondary studies, promoting evidence-based decision-making in various fields of software engineering [23]. Another category of reviews revolves around automation, exploring its benefits and limitations [10], guidelines for determining when and what to automate [24], and the challenges inherent in its automation [25].

Another crucial aspect in this field involves the generation of test data. Some researchers propose an intelligent framework based on genetic algorithms [26], while others explore the integration of genetic algorithms and reinforcement learning to automate test data generation [27]. Optimization algorithms, such as particle swarm optimization [28]–[31] or the cuckoo algorithm [32], have also been employed for this purpose. Additionally, search-based methods have been identified as tools for producing test data with the specific objectives of ensuring branch coverage [33], achieving full statement coverage [34], or incorporating structural, white-box testing techniques that include the coverage of specific program structures [35].

Various artificial intelligence (AI) approaches have been associated with methods and stages of software testing [36]. Among these, the bee colony algorithm stands out as a frequently employed algorithm in the field. This algorithm emulates how honey bees operate to enhance food-finding and nectar-gathering systems [37]. Specifically, it utilizes the path coverage metric to optimize test cases within the search area [38]. The bee colony algorithm is also instrumental in prioritizing the regression test suite based on fault coverage [39] and generating test cases with the minimal required iterations and duration [40]. Another widely utilized set of algorithms in this context is genetic algorithms. These algorithms, following the principles of natural selection and heredity, aim to locate the best possible solution [41]. They mimic natural evolutionary principles to optimize a certain goal within the manual system. Genetic algorithms generate software test cases by leveraging concepts such as fossil records and proportional or evolving fitness ideas [42]. They function as a regeneration method, assessing the rate of population aging [43], or combine with mutation testing [44]. In addition, they are used to automatically generate test cases [45]–[47].

Data mining and multi-agent systems play integral roles in software testing. Data mining [48]–[50] serves as a technique for evaluating association rules among software testing metrics [51]. Additionally, it finds application in automating regression testing for data-driven software systems [52]. Multi-agent systems demonstrate collaboration capabilities in the creation of test data for software [53]. Another approach involves the development of a multi-agent testing framework [54]. Furthermore, an adaptive test automation model proves to be a valuable application, integrating the study of test requirements, the construction and execution of test cases, test planning and execution, as well as defect reporting and analysis [55]. The use of a bibliometric analysis can provide valuable insights for researchers, scientists, and graduates, aiding in more effective decision-making within their disciplines and encouraging future studies in relevant domains.

Certain scholars have explored the impact of collaborative work by tester pairs on the quality of produced test reports [56].To reduce the number of required test cases, some researchers have employed smart contract testing patterns [57]. The testing of abstract classes, which pose challenges due to their inability to be instantiated, involves the use of the template method design pattern. This pattern generates an abstract class with a concrete template method and one or more abstract primitive operations [58]. To expedite test execution, code optimization proves beneficial. An approach focused on optimizing source code using equivalent mutants utilize an algorithmic strategy that outperforms traditional compiler optimizations [59]. When an oracle is not available, test cases are created via metamorphic testing, which depends on the relationships between the inputs and outputs of several program runs. This method is used to automatically identify performance issues in combination with search-based approaches [60].

## III. Data Collection

Bibliometric analysis is widely acknowledged as a precise method for scrutinizing vast amounts of scientific data, enabling the identification of emerging fields and fostering a comprehensive understanding of field evolution [13]. The use of bibliometric software streamlines the classification and analysis of substantial data volumes collected from studies conducted over a specific period. In contrast to systematic literature reviews, which frequently employ qualitative methods and may be prone to interpretation bias due to researchers diverse academic backgrounds, both systematic review and bibliometric analysis emphasize statistical approaches, effectively reducing the likelihood of bias [13].

Various databases, including Web of Science (WoS), Scopus, PubMed, Lens, and Google Scholar offer distinct characteristics, and functionalities for extracting literature data. Google Scholar is known for its broad coverage across multiple disciplines, while Web of Science and Scopus often provide comparable results [61]. Notably, Web of Science stands out for its diverse search methods, including general, cited references, and advanced search functions. All three databases, including Scopus and Google Scholar, facilitate the monitoring, counting, processing, and analysis of cited references [62]. Web of Science boasts nearly 10,000 included journals and encompasses seven distinct citation databases that gather information from book series, conferences, journals, reports, and books [63].

Selected for its reputation as one of the most comprehensive databases, the Web of Science (WoS) database grants access to articles sourced from high-quality research literature [64]. Notably, it hosts a diverse range of reputable journals and top-tier publications that have been subject to expert evaluation within their respective disciplines [65]. Acknowledged as one of the most reliable sources for research articles [66], Web of Science was intentionally chosen for the present study. Specifically, the utilization of the Web of Science core collection was driven by its broad coverage across major scientific disciplines and its inclusion of numerous high-impact journals, ensuring a thorough representation of literature on intelligent software testing [65], [67]. The core collection's uniform indexing and citation metrics across all journals facilitate consistent and reliable comparisons among articles, authors, and journals, thereby enhancing the validity and dependability of the bibliometric analysis [68].

The papers in this study are from renowned publishers, including MDPI, IEEE, Elsevier, and Springer, known for their significant contributions to scientific research. These publications have a reputation for continuously delivering top-notch papers in a variety of disciplines, including computer science, engineering, and medical. Their widespread reputation highlights their integral role in advancing scholarly knowledge beyond traditional boundaries.

The search approach used in this study is shown in figure 1.

The selected keywords were derived from a comprehensive search encompassing software testing and graph theory related papers. Ultimately, we opted for specific terms: "software test," serving as an umbrella for variations like "software tests" or "software testing"; "test case generation" and "test prioritization," denoting specific aspects of software testing. Additionally, "graph theory" acts as an umbrella term, encompassing related keywords such as "graph" or "graph theory algorithms." This meticulous keyword selection ensures a focused exploration of relevant literature in the intersection of software testing and graph theory.

At the outset , a search was conducted using keywords from both the domains of graph theory and software testing that use every field in the Web of Science core collection and is concatenated by the AND operator. A total of 1005 documents were found in this first search.

The paper concentrates on extracting insights from the past decade. To narrow down the focus, a filter was applied, restricting the search to documents published between 2013 and 2023 (inclusive). Following this temporal filter, the result set was refined to 609 documents.

Finally the query looks like this:

("graph theory " OR " Graph" OR " theory algorithms") AND ("software testing" OR "software test" OR " software development").

## IV. Research Method

In order to evaluate the impact of a specific topic or body of work, bibliometric studies employ statistical analysis of published literature. This approach entails researching a variety of topics within the field of software testing. Evaluating the number and quality of publications related to software testing is one aspect. Finding the writers and organizations that are making major contributions to the field is another aspect. In order to identify trends and configurations within the research landscape, the bibliometric study also looks at the citation patterns of software testing papers. This kind of study is useful for finding important research topics,

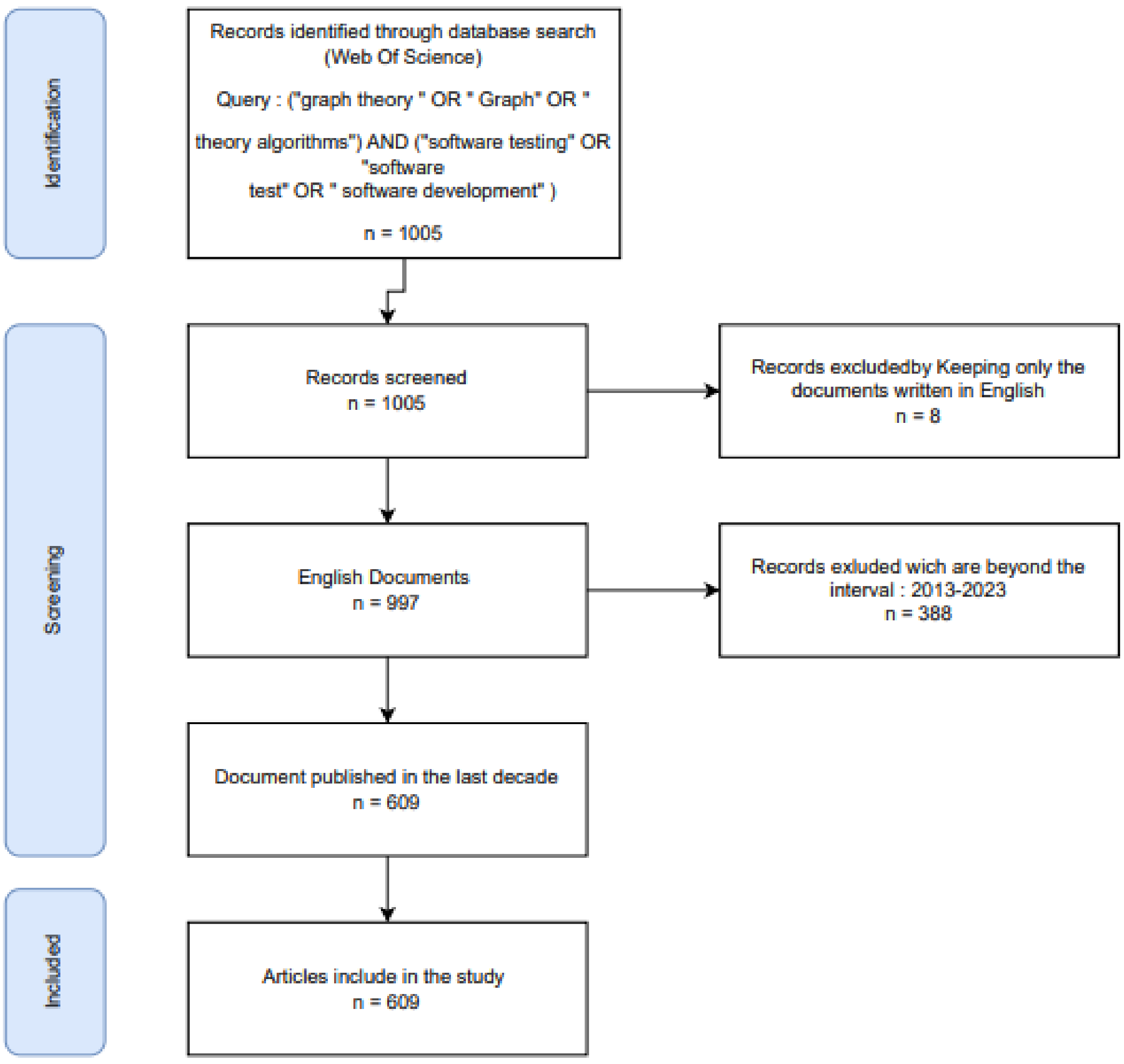

Identification
Records identified through database search (Web Of Science)
Query : ("graph theory " OR " Graph" OR " theory algorithms") AND ("software testing" OR "software test" OR " software development" )
n = 1005
Screening
Records screened
n = 1005
Records excludedby Keeping only the documents written in English
n = 8
English Documents
n = 997
Records exluded wich are beyond the interval : 2013-2023
n = 388
Document published in the last decade
n = 609
Included
Articles include in the study
n = 609

Fig. 1: Overview of the methodological search strategy

comprehending the state of the field at the moment, and spotting possible directions for further investigation.

Our methodology involved leveraging the Bibliometrix R package (biblioshiny v4.0.1) for a comprehensive scientific mapping analysis subsequent to exporting the dataset from the WoS database [69]. This facilitated the creation of the majority of graphical representations. Additionally, we employed VOSviewer, a freely available computer tool designed for crafting and visualizing bibliometric maps [70], to generate the remaining graphical representations.

We began by giving a summary of the dataset that included the following items:

- Key information: A table listing the key characteristics of our dataset;
- Scientific output: A figure illustrating the yearly quantity of scientific output in the field being studied;
- Citations annually: A graphic representation of the increase in citations over time;
- Relationship between nations, writers, and titles: An example that demonstrates the interactions between countries, authors, and article titles and offers details on the connections between these ties.

The next segment of the results delves into the domain of science mapping. This concluding phase involves a comprehensive examination and visualization of the general domain. A science mapping project typically incorporates various components, including a cluster of scientific writings, an array of metrics, indicators, and scientometric tools for observation and analysis. These components collectively shed light on emerging trends and patterns, offering valuable insights into noteworthy developments. Additionally, they play a pivotal role in exploring transformative ideas within the scientific landscape, guiding conceptual frameworks, and facilitating the analysis and interpretation of cycles [71]. During this phase, our attention was directed towards the following elements:

- High-performing and highly cited journals: The top ten most active journals are indicated in a graph according to the number of papers they published, and the top

cited journals are arranged according to the number of citations they received.
- Leading organizations: Illustrated through a graph, the top 10 most prolific institutions are showcased, determined by the quantity of documents they generated.
- Prominent Authors: Highlighted in a graph, the top 10 authors are featured based on the volume of articles they created.
- Top developing nations by corresponding authors: Presented in a graph, the top 10 most relevant countries are showcased based on the quantity of articles produced, whether in single-country publication or multiple-country collaboration.
- Globally cited authors: The top 10 most cited authors are shown in a graph that uses all of the Web of Science database's citations.
- Globally cited documents: Using all citations in the Web of Science database, a graph showcasing the top 10 most referenced papers is presented.
- Commonest terms: The most frequently used terms in the abstracts, titles, and keywords of the texts are displayed in a word cloud and graph.
- Co-occurrence network of keywords: This type of graph shows how keywords relate to one another, breaking them up into smaller groups.
- Collaboration world map and network: The interaction between nations in terms of collaboration is depicted through a world map and a graph.
- Organizations co-authorship: Illustrated in a graph, the network of collaborations between organizations is showcased.
- Authors co-citation network: Shown as a graph, this network highlights the relationships among writers based on co-citations.
- Journals co-citations network: The network of relationships between journals is displayed based on co-citations and is represented graphically by a graph.

Network analysis is the subject of the third section. Network analysis, which has its roots in network theory or graph theory, examines the characteristics of networks and the relationships between their vertices and segments. The proximity and betweenness centrality measures are the two main metrics used in network analysis. The centrality index serves as a representation of the social power of a vertex, assigning each vertex in the network a ranking based on its role and position in network communications [72]. In this segment of the study, our attention was directed toward the following components:
- Thematic map: This involves generating a network graph, known as a thematic map, based on the connections among keywords. The labels for each thematic map are determined by the most frequently appearing keyword in the associated topic [73].
- Multiple correspondence analysis (MCA): Utilized for both visual and mathematical analysis of such data, multivariate categorical analysis (MCA) serves as a method for multidimensional analysis [74].
- Correspondence analysis (CA): CA is a visual method designed to comprehend the connections between items in a frequency table. It evaluates relationships between qualitative variables and is an advancement of principal component analysis, particularly for categorical data [75].
- Multidimensional scaling: Similar to MCA and CA, multidimensional scaling is employed as a dimensionality reduction technique to construct a map of the network under study using normalized data ‘ [69].

All of these components will be thoroughly discussed in section 5 of this article.

Figure 2 provides a summary of the research approach employed in this article.

## V. Results

### A. Overview

*1) Main Information:* Figure 3 and figure 4 present the primary details pertaining to intelligent software testing, derived from the dataset extracted from the Web of Science database. Key aspects covered include the research timeframe, journals and documents included, average citation count, references cited, and the annual growth rate. Additionally, the table includes crucial information on content such as author-provided keywords and the unique keywords Plus attribute from the Web of Science database. Authorship data is categorized into authors and single-authored documents, while collaboration metrics encompass the count of single-authored documents, average authors per document, and the rate of international co-authorship.

*2) Annual Scientific Production:* Figure 5 depicts the annual scientific output in software testing research. The overall trend reveals significant variations in production over the years, with a general upward trajectory. For instance, in 2014, there were only 35 documents published, reflecting relatively low activity. By contrast, the number of publications peaked at 68 articles in 2021, showcasing heightened interest and advancements in the field. Despite minor fluctuations, the scientific output demonstrates a growing commitment to software testing research, culminating in 64 articles by 2023.

*3) Average Citation Per Year:* Figure 6 illustrates the average number of document citations per year. In 2013, the average was 0.8 citation per document. By 2020, the average citations per document had increased to 2.1, but was notably reduced to 0.2 per document by 2023. Although these findings reflect important citation trends, additional in-depth analyses could reinforce the significance of these results for journal-level publication.

*4) Relationship between Countries, Authors, and Titles:* The relationship among nations, authors, and document keywords in software testing publications is visualized using a three-field plot. The diagram consists of rectangles representing authors (AU), countries (AU_CO), and keywords (TI_TM), with varying heights indicating the cumulative relationships of the elements they depict. The thickness of the connecting lines symbolizes the intensity of these associations. Figure 7 underscores the dominance of authors from China, India, and the USA in driving significant research in intelligent software testing. Notably, keywords such as ’software,’ ’testing’ and ’development’ emerge as central themes, highlighting the global research focus on advancing methodologies and applications in this field.

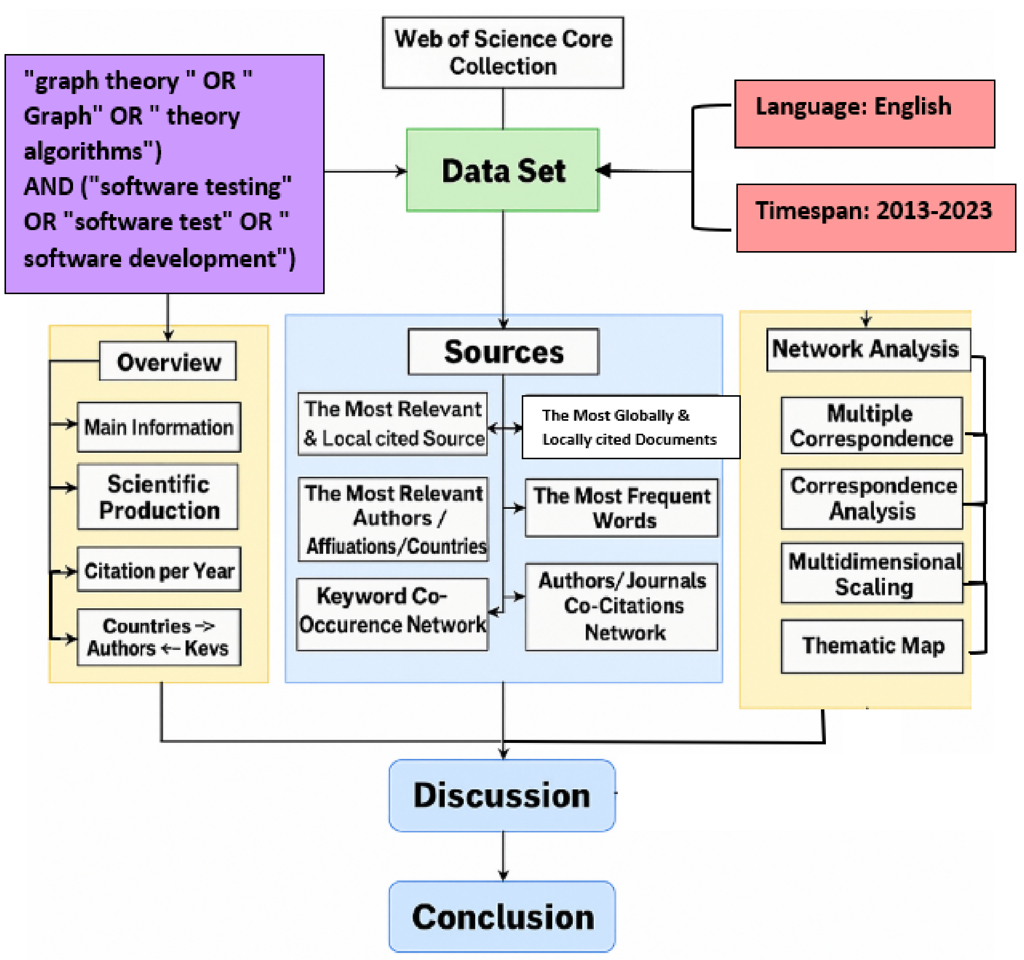
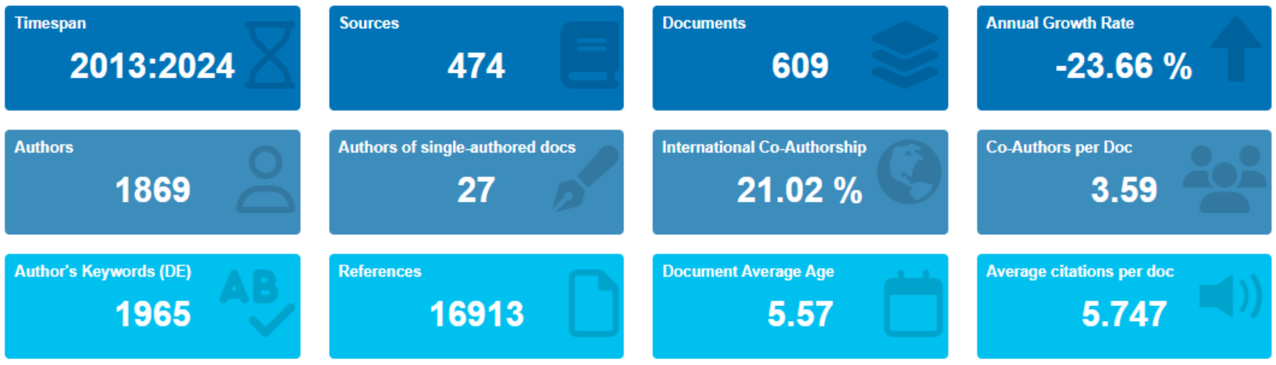

Web of Science Core Collection
"graph theory " OR " Graph" OR " theory algorithms") AND ("software testing" OR "software test" OR " software development")
Data Set
Language: English
Timespan: 2013-2023
Overview
Main Information
Scientific Production
Citation per Year
Countries -> Authors <- Keys
Sources
The Most Relevant & Local cited Source
The Most Globally & Locally cited Documents
The Most Relevant Authors / Affiuations/Countries
The Most Frequent Words
Keyword Co-Occurence Network
Authors/Journals Co-Citations Network
Network Analysis
Multiple Correspondence
Correspondence Analysis
Multidimensional Scaling
Thematic Map
Discussion
Conclusion

Fig. 2: Overview of the bibliometric analysis steps.



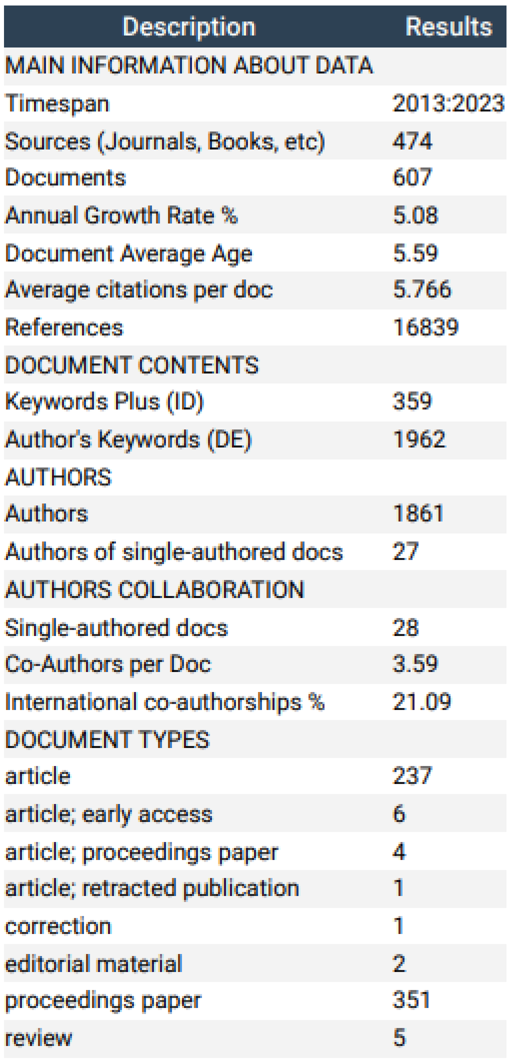
| | | | |
|---|---|---|---|
| Timespan 2013:2024 | Sources 474 | Documents 609 | Annual Growth Rate -23.66 % |
| Authors 1869 | Authors of single-authored docs 27 | International Co-Authorship 21.02 % | Co-Authors per Doc 3.59 |
| Author's Keywords (DE) 1965 | References 16913 | Document Average Age 5.57 | Average citations per doc 5.747 |

Fig. 3: Main information.

### B. Sources

*1) The Most Relevant Sources and Local Cited Sources:* Figure 8 presents a list of the top ten sources, highlighted in red, that have contributed the most papers on intelligent software testing. The analysis indicates that the highest number of publications on intelligent software testing appears in IEEE Access, which leads with 12 documents. This is followed by the Journal of Systems and Software with 11 publications, and the IEEE Transactions on Software Engineering with 9. Figure 9 unveils the top 10 most-cited sources for works in intelligent software testing. According to bibliometric analysis, the lecture notes in Computer Science book series emerges as the most cited source with 721 citations, followed by the IEEE transactions on software engineering with 601 citations. The third place, with 362 citations, is held by the proceedings of the international conference on software.

*2) The Most Relevant Affiliations:* The research output of institutions and author affiliations involved in the field of intelligent software testing was analyzed, as illustrated in figure 10. Leading the list, the chinese academy of sciences produced ten documents. While this clearly identifies leading affiliations, incorporating detailed insights into their specific contributions or collaboration patterns would further strengthen the significance of these results.



| Description | Results |
|---|---|
| MAIN INFORMATION ABOUT DATA | |
| Timespan | 2013:2023 |
| Sources (Journals, Books, etc) | 474 |
| Documents | 607 |
| Annual Growth Rate % | 5.08 |
| Document Average Age | 5.59 |
| Average citations per doc | 5.766 |
| References | 16839 |
| DOCUMENT CONTENTS | |
| Keywords Plus (ID) | 359 |
| Author's Keywords (DE) | 1962 |
| AUTHORS | |
| Authors | 1861 |
| Authors of single-authored docs | 27 |
| AUTHORS COLLABORATION | |
| Single-authored docs | 28 |
| Co-Authors per Doc | 3.59 |
| International co-authorships % | 21.09 |
| DOCUMENT TYPES | |
| article | 237 |
| article; early access | 6 |
| article; proceedings paper | 4 |
| article; retracted publication | 1 |
| correction | 1 |
| editorial material | 2 |
| proceedings paper | 351 |
| review | 5 |

Fig. 4: Main information.

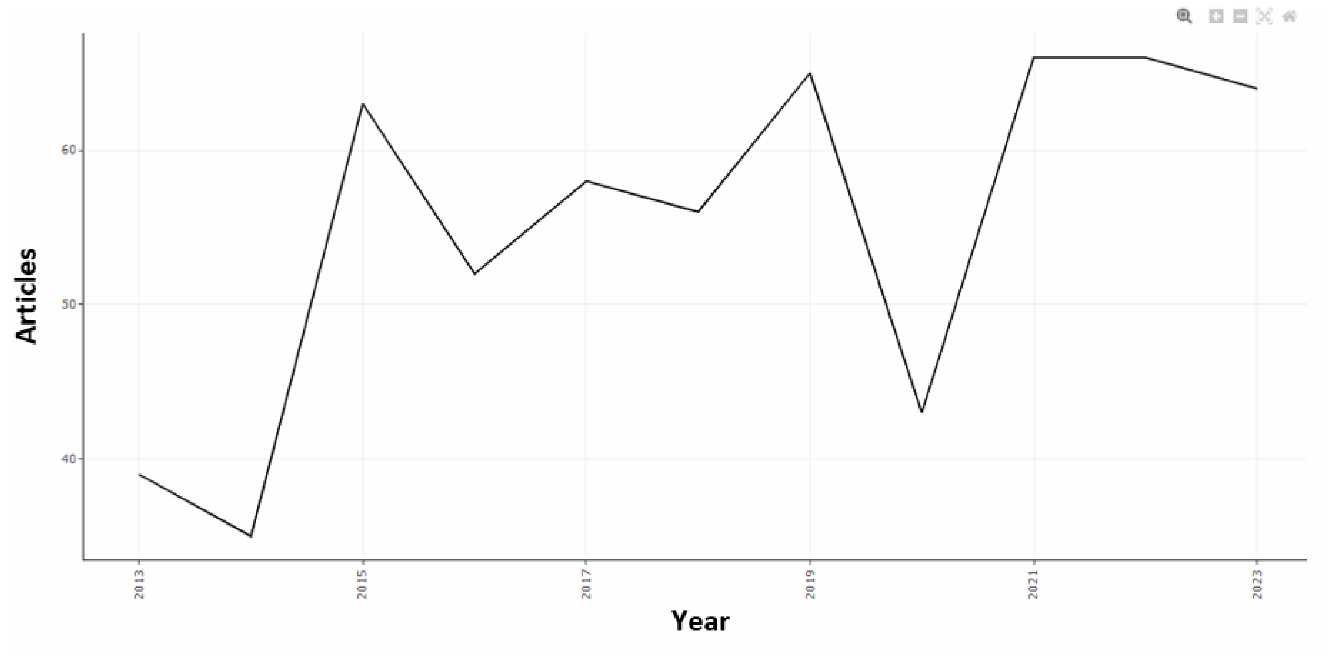


Fig. 5: The annual scientific production.

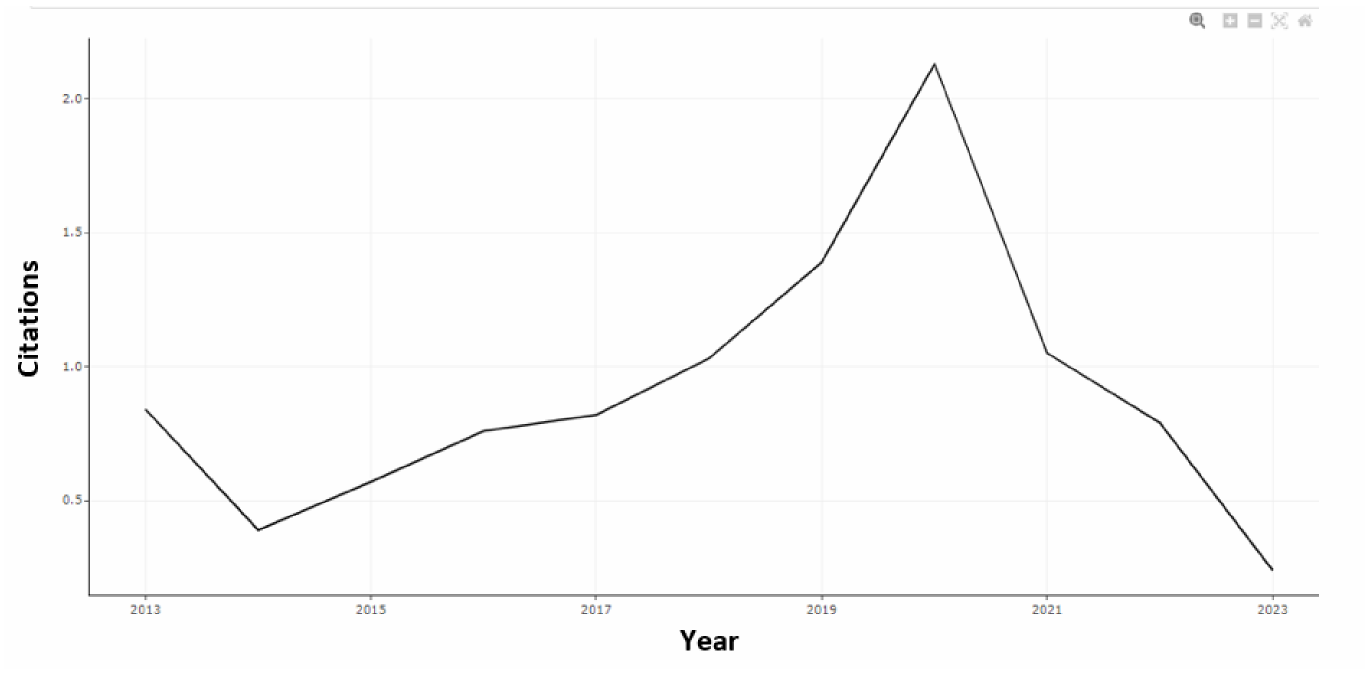


Fig. 6: The average citations per year.

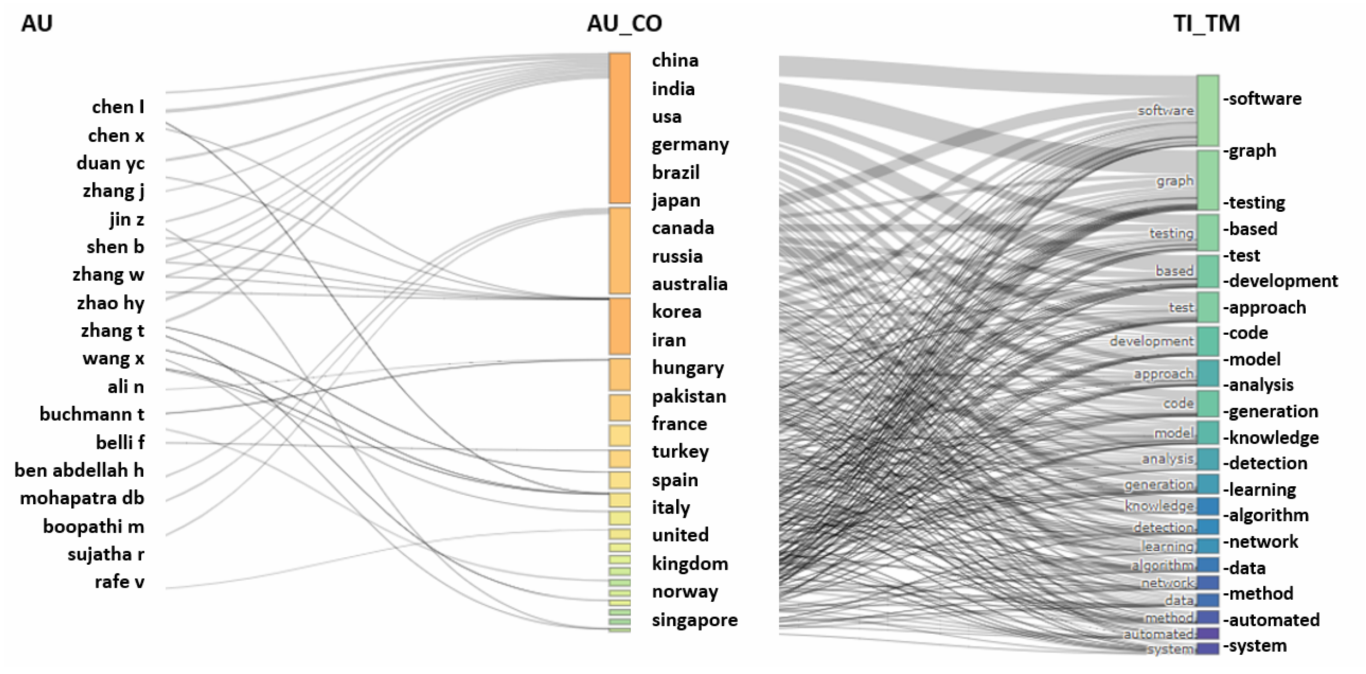


Fig. 7: Countries, authors, and document title keywords in a three-field layout.

*3) The Most Relevant Authors:* Figure 11 highlights the top 10 most influential authors in the domain of software testing research. MOHAPATRA DP stands out as the most productive author, contributing 6 publications with a fractionalized value of 2.33, underscoring a significant impact. BUCHMANN T, CHEN X, and DUAN YC follow closely with 5 documents each and fractionalized values of 2.67, 1.14, and 0.93, respectively. Other contributors, including BELLI F and BOOPATHI M, with 4 publications each, reflect the collaborative and distributed efforts driving advancements in this research field.

*4) The Most Relevant Countries by Corresponding Authors:* This study also took into account the countries in which the corresponding authors' publications in the subject of intelligent software testing were published. as seen in figure 12 and figure 13. China was ranked one with 119 single-nation publications, 28 multi-nation publications, and maximum frequency of 0.242. China, India, and USA are among the top three scientifically most productive countries globally.

*5) The Most Globally Cited Documents:* The phrase "globally cited documents" describes how many citations in the entire WoS core collection a given document has earned from other papers. The documents that are most frequently cited worldwide are shown in figure 14. With 98 citations, a work written by YAMAGUCHI and published in the IEEE

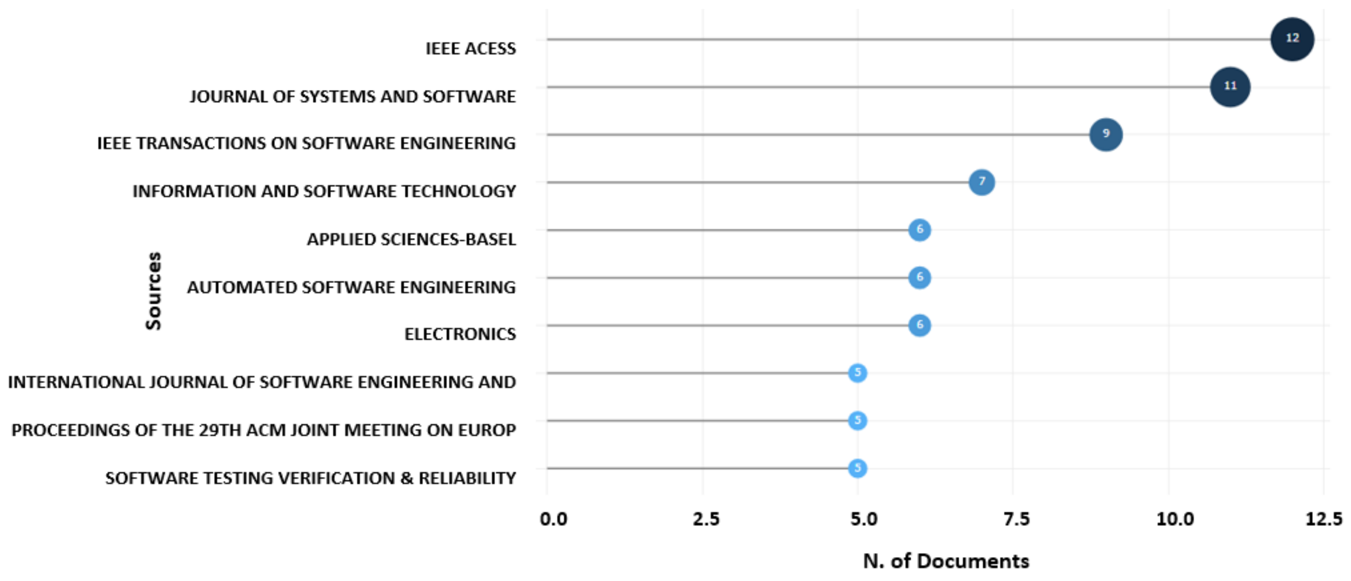


Fig. 8: The most pertinent references.

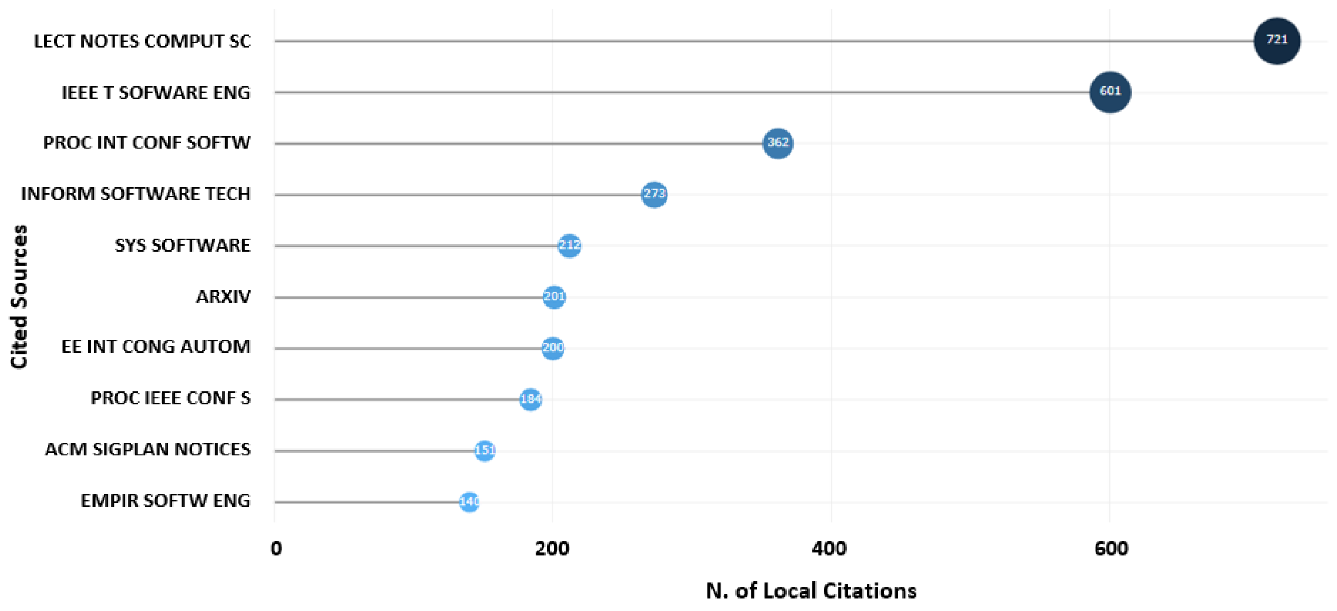


Fig. 9: The sources most frequently cited locally.

Symposium on Security and Privacy in 2015 is ranked first. This finding might be made more thorough and in-depth by adding additional talks about how these documents affect ongoing research or real-world applications.

*6) The Most Locally Cited Documents:* With an emphasis on intelligent software testing in our situation, the phrase "local citation count" refers to the quantity of citations a certain document has acquired from other papers within the dataset under investigation. The most locally cited documents are shown in figure 15. The article "Optimal test sequence generation using firefly algorithm," written by SRIVATSAVA PR and published in 2013, comes in first place with five citations. Additional contextual material or illustrated assessments of these significant local documents may improve the thoroughness required for journal-level publishing.

*7) Lotka's Law, or the Frequency Distribution of Scientific Productivity:* The Lotka's law parameters for the software testing of intelligent articles are computed by this bibliometric indicator. Lotka's Law explains the relationship between authors and the number of articles published. Lotka's Law describes how writers are distributed across time or within particular informatics subject areas. [75]. Figure 16 illustrates how the quantity of publications and the frequency of writers in the subject matter closely adhere to Lotka's law. Figure 17 illustrates Lotka's Law's frequency distribution of scientific output.

*8) The Most Frequent Words:* With a combined total of 300 occurrences, the phrases "software" and "graph" were used by authors the most, followed by "testing," which appeared 82 times. The most often used terms in the subject of graph-theoretic software testing are shown in figure 18 . Additionally, a word cloud displaying the most often used keywords in studies pertaining to the field under study is shown in figure 19.

*9) Collaboration World Map and Network:* Figures 20 and 21 illustrate that China occupies a central position

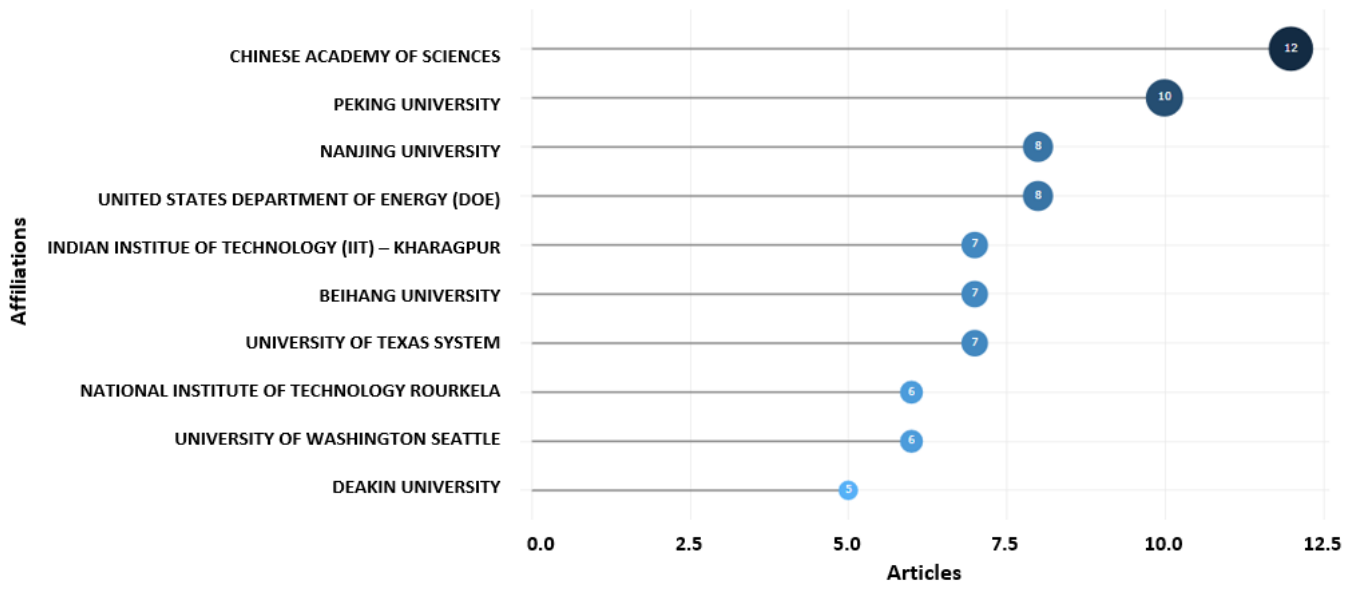


Fig. 10: Ten most important associations.

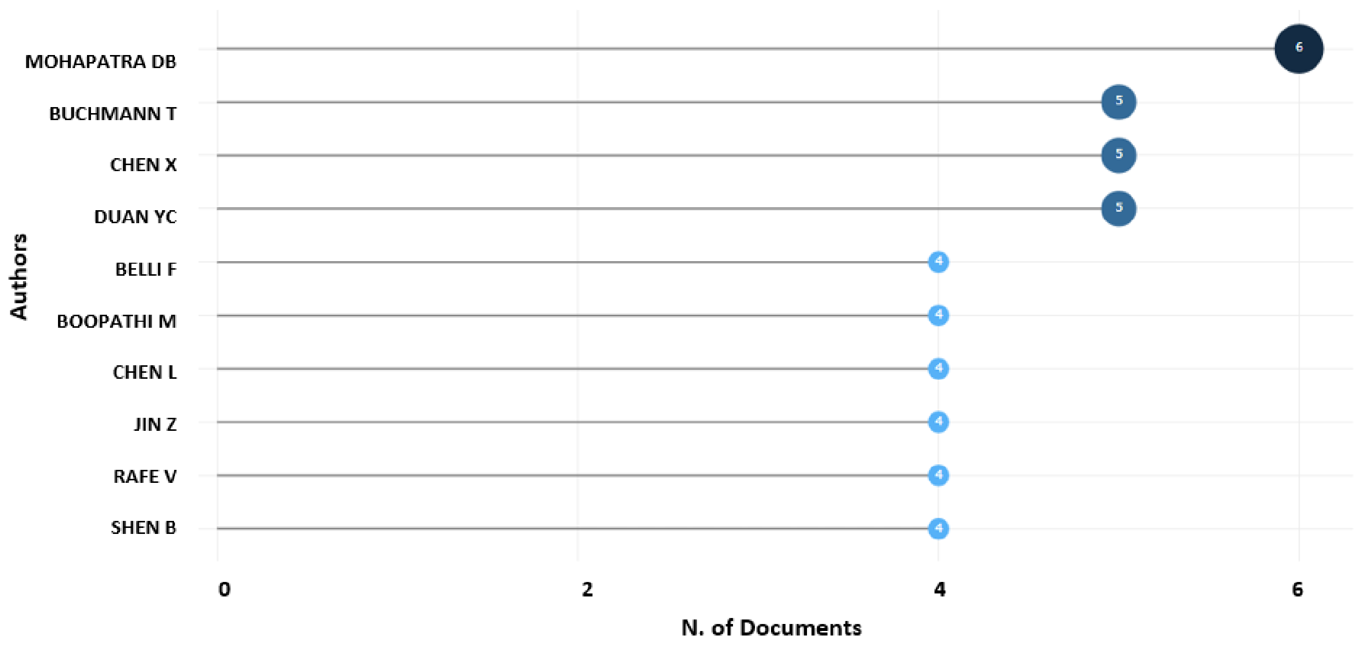


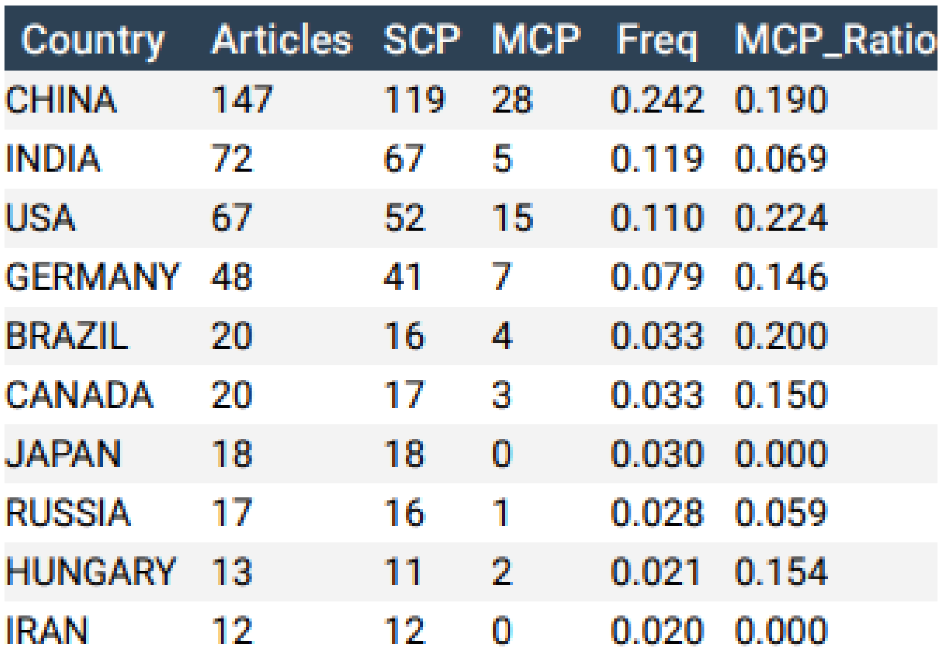
Fig. 11: The ten most pertinent authors.

within the largest collaboration cluster, with India and USA following closely behind. Another significant cluster emerges among Germany, Brazil, Russia, Spain, Italy, and Finland. Additionally, a distinct cluster is observed involving Canada, South Korea, Pakistan, Hungary, Malaysia, Saudi Arabia, and the Czech Republic. Future inclusion of specific examples or outcomes from these international collaborations could further enhance the comprehensiveness of these results.

*10) Keywords Co-Occurrence Network:* In figure 22, the keyword co-occurrence network is depicted. Each node in the network represents a keyword, with the size of the node indicating the frequency of the keyword occurrence. Links between nodes illustrate the co-occurrence of terms, with the thickness of the link representing the frequency of such co-occurrences. Different thematic clusters are distinguished by colors. Among these clusters, keywords such as software testing, software, graph theory, control flow graph, and deep learning emerge as the most frequent.

*11) Network of Co-Citations by writers:* Figure 23 shows a network of co-citations by writers, divided into four different clusters that illustrate relationships between authors. M is the most often referenced author in cluster 1, Allamanis. Chen, Ty is the author most frequently cited in Cluster 2. In cluster 3, Rothermel, G. is the most referenced author, whereas in cluster 4, Harman, M. is the most cited author. The journal publication's universality would be improved with more background information or an illustrative overview of these authors' contributions to the area.

*12) Journals Co-Citations Network:* Displayed in figure 24 is a detailed network illustrating co-citations among journals in the domain of software testing research. The visualization identifies four distinct clusters, representing



| Country | Articles | SCP | MCP | Freq | MCP_Ratio |
|---|---|---|---|---|---|
| CHINA | 147 | 119 | 28 | 0.242 | 0.190 |
| INDIA | 72 | 67 | 5 | 0.119 | 0.069 |
| USA | 67 | 52 | 15 | 0.110 | 0.224 |
| GERMANY | 48 | 41 | 7 | 0.079 | 0.146 |
| BRAZIL | 20 | 16 | 4 | 0.033 | 0.200 |
| CANADA | 20 | 17 | 3 | 0.033 | 0.150 |
| JAPAN | 18 | 18 | 0 | 0.030 | 0.000 |
| RUSSIA | 17 | 16 | 1 | 0.028 | 0.059 |
| HUNGARY | 13 | 11 | 2 | 0.021 | 0.154 |
| IRAN | 12 | 12 | 0 | 0.020 | 0.000 |

Fig. 12: The most Corresponding Author’s Countries.

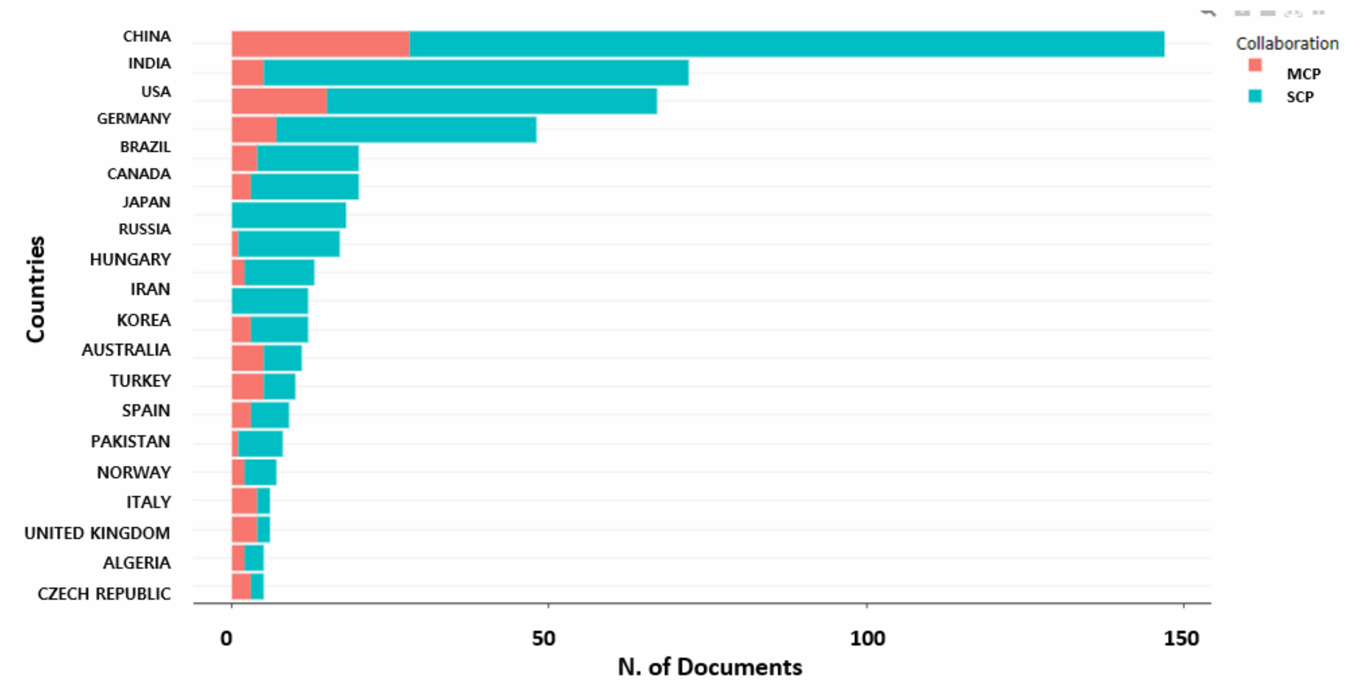
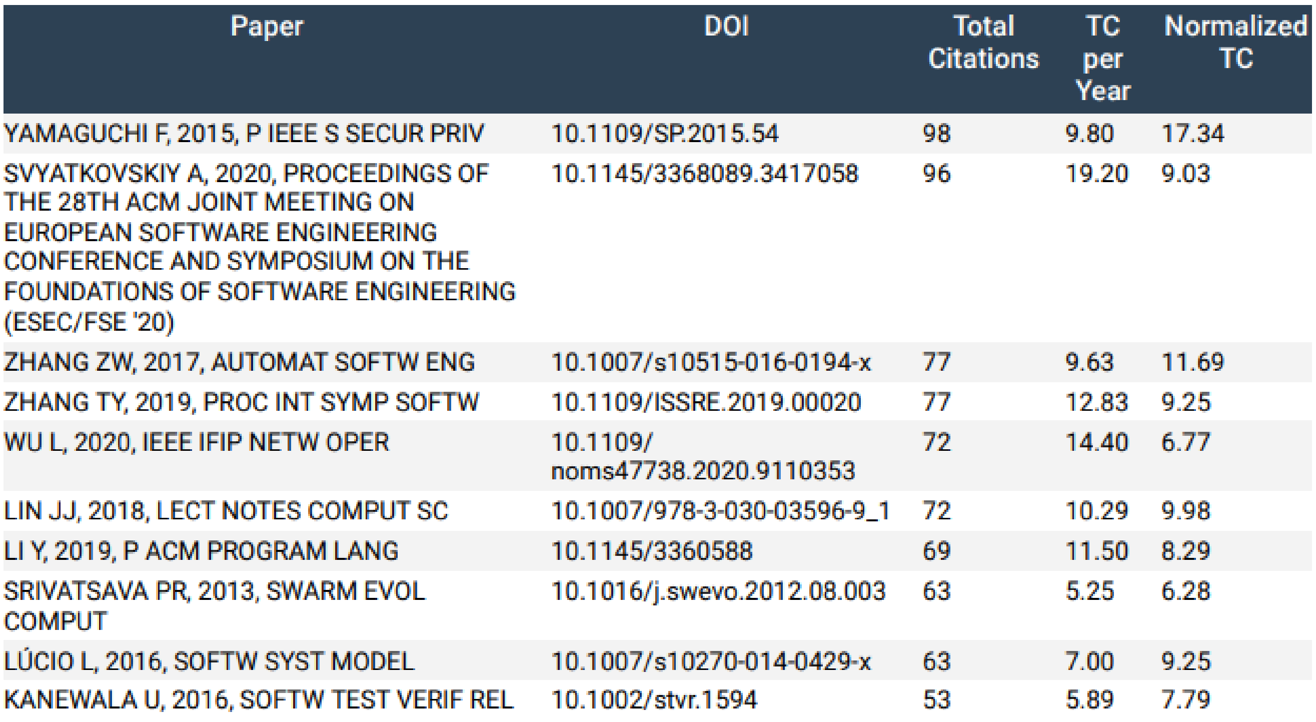


Fig. 13: The most Corresponding Author’s Countries.

groups of journals with closely related content and citation patterns. The IEEE Transactions on Software Engineering journal emerges as the most frequently cited, highlighting its central role in the field. It is followed by the lecture notes in computer science book series, which bridges multiple domains, and the information and software technology journal, noted for its interdisciplinary connections. These clusters reveal the diverse yet interconnected nature of research in this area.

### *C. Network Analysis*

*1) Thematic Map:* Based on their density and centrality, thematic maps show groups of keywords grouped in a circle and represented as a two-dimensional image. As seen in figure 25, each thematic map is separated into quadrants according to where they are located. The terms with the highest level of development and relevance are found in the upper-right quadrant, which is occupied by motor themes. The lower-right quadrant contains basic themes, which are



| Paper | DOI | Total Citations | TC per Year | Normalized TC |
|---|---|---|---|---|
| YAMAGUCHI F, 2015, P IEEE S SECUR PRIV | 10.1109/SP.2015.54 | 98 | 9.80 | 17.34 |
| SVYATKOVSKIY A, 2020, PROCEEDINGS OF THE 28TH ACM JOINT MEETING ON EUROPEAN SOFTWARE ENGINEERING CONFERENCE AND SYMPOSIUM ON THE FOUNDATIONS OF SOFTWARE ENGINEERING (ESEC/FSE '20) | 10.1145/3368089.3417058 | 96 | 19.20 | 9.03 |
| ZHANG ZW, 2017, AUTOMAT SOFTW ENG | 10.1007/s10515-016-0194-x | 77 | 9.63 | 11.69 |
| ZHANG TY, 2019, PROC INT SYMP SOFTW | 10.1109/ISSRE.2019.00020 | 77 | 12.83 | 9.25 |
| WU L, 2020, IEEE IFIP NETW OPER | 10.1109/ noms47738.2020.9110353 | 72 | 14.40 | 6.77 |
| LIN JJ, 2018, LECT NOTES COMPUT SC | 10.1007/978-3-030-03596-9_1 | 72 | 10.29 | 9.98 |
| LI Y, 2019, P ACM PROGRAM LANG | 10.1145/3360588 | 69 | 11.50 | 8.29 |
| SRIVATSAVA PR, 2013, SWARM EVOL COMPUT | 10.1016/j.swevo.2012.08.003 | 63 | 5.25 | 6.28 |
| LÚCIO L, 2016, SOFTW SYST MODEL | 10.1007/s10270-014-0429-x | 63 | 7.00 | 9.25 |
| KANEWALA U, 2016, SOFTW TEST VERIF REL | 10.1002/stvr.1594 | 53 | 5.89 | 7.79 |

Fig. 14: The most globally cited document.

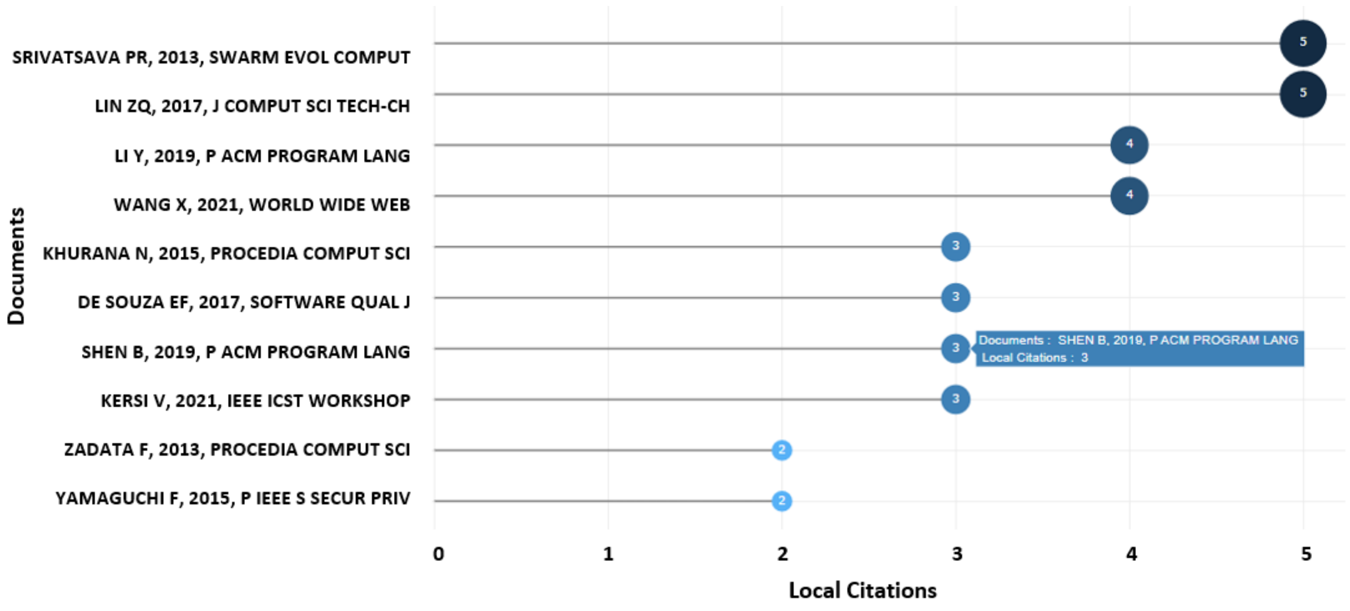
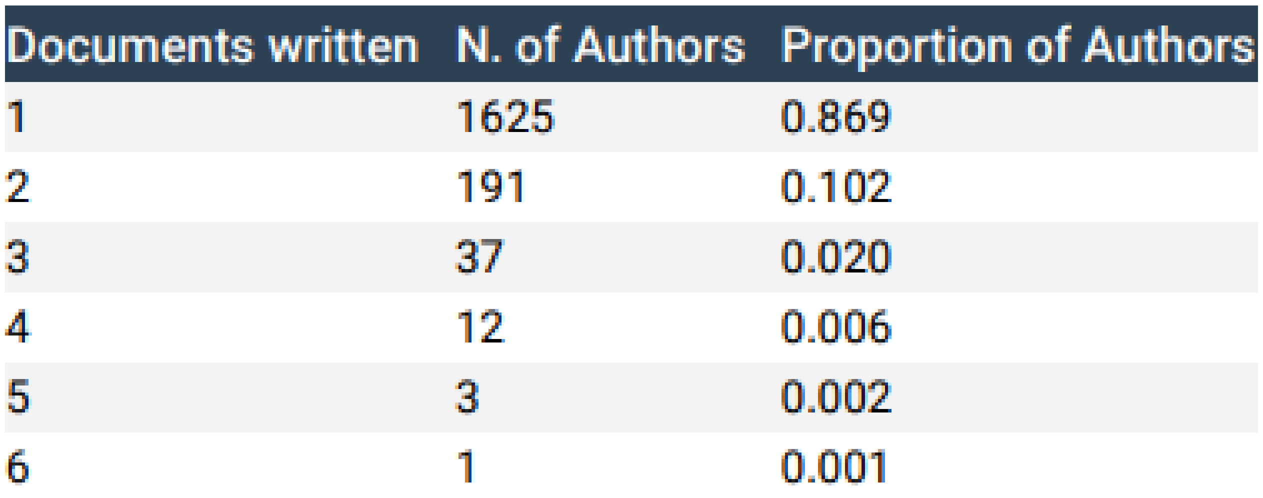


Fig. 15: The most locally cited document.

made up of highly relevant terms with different levels of development. The lower-left quadrant, which represents terms of moderate relevance and development, is where emerging or declining themes are located. The upper-left quadrant is occupied by niche themes, which are defined by keywords that have significant levels of development but may be less relevant.

2) *Multiple Correspondence Analysis:* Described in section IV, multiple correspondence analysis (MCA) serves to graphically and mathematically analyze multivariate categorical data. It aims to uncover latent variables or factors by assessing the interrelationships among various categorical data points. Interpretation of results relies on the relative positions and distributions of points along dimensions depicted in figure 26, closer proximity of words in these figures indicates greater similarity in distribution.

Figure 26 depict the conceptual structure map, respectively, generated from the keywords plus of each paper. The y-axis indicates height, representing the distance between



| Documents written | N. of Authors | Proportion of Authors |
|---|---|---|
| 1 | 1625 | 0.869 |
| 2 | 191 | 0.102 |
| 3 | 37 | 0.020 |
| 4 | 12 | 0.006 |
| 5 | 3 | 0.002 |
| 6 | 1 | 0.001 |

Fig. 16: Description of the frequency distribution of scientific productivity (Lotka's Law).

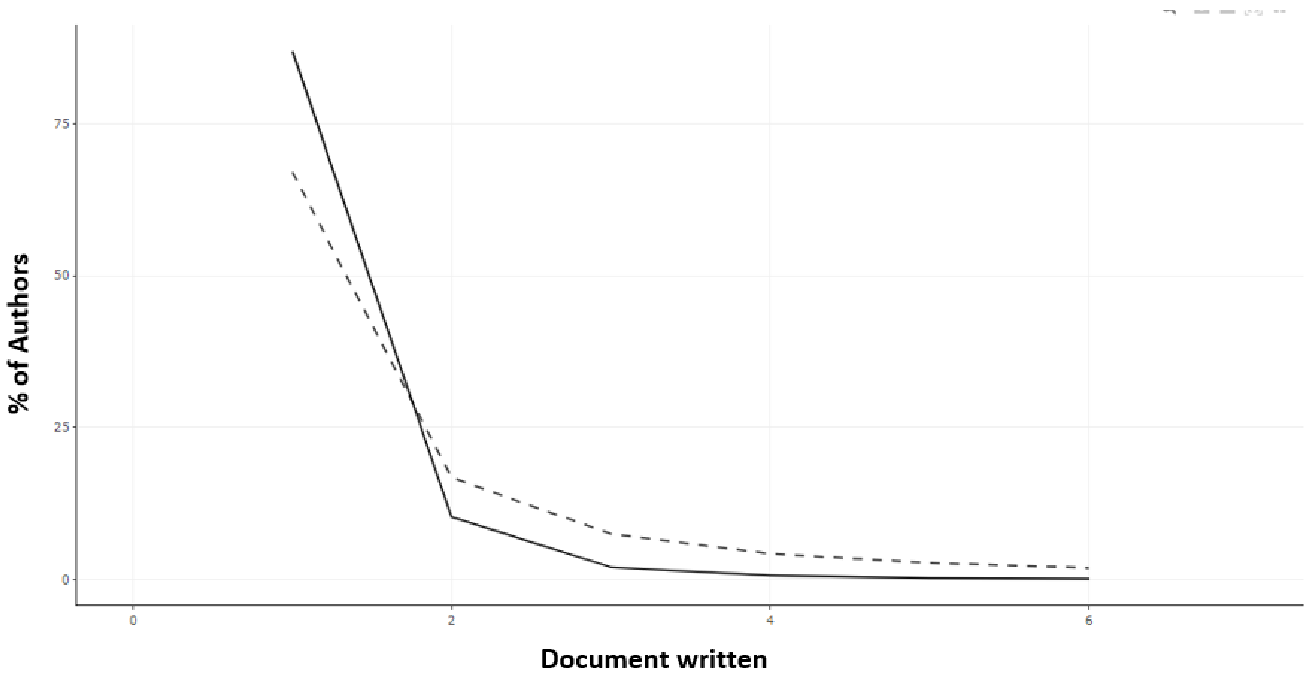


Fig. 17: Description of the frequency distribution of scientific productivity (Lotka's Law).

words. Groups of words clustered together signify association with the same topic.

*3) Correspondence Analysis:* In section IV, we outline the definition and applications of correspondence analysis. It serves as a visual technique to explore the relationships between variables within a contingency table. Correspondence analysis facilitates the reduction and representation of datasets through two-dimensional graphs. Figure 27 demonstrates its aim of generating a comprehensive data visualization for interpretation. Figures 27 and 28 present the subject dendrogram and conceptual structure map respectively, derived from the title of each paper. The y-axis for both numbers refers to height, which is the distance between words. The clusters of words grouped together represent an association with the same topic.

*4) Multidimensional Scaling:* Figure 29 demonstrates the application of multidimensional scaling (MDS), a method discussed in section 4 for analyzing multivariate data. MDS is used to visualize similarities and dissimilarities among samples by plotting points in two-dimensional plots. The MDS algorithm utilizes input data from a dissimilarity matrix, which indicates the distances between pairs of objects. This approach allows for a visual representation of the relationships between samples based on their pairwise dissimilarities, aiding in the exploration and interpretation of complex datasets. In figure 30, the topic dendrogram was generated based on the author's keywords from each paper.

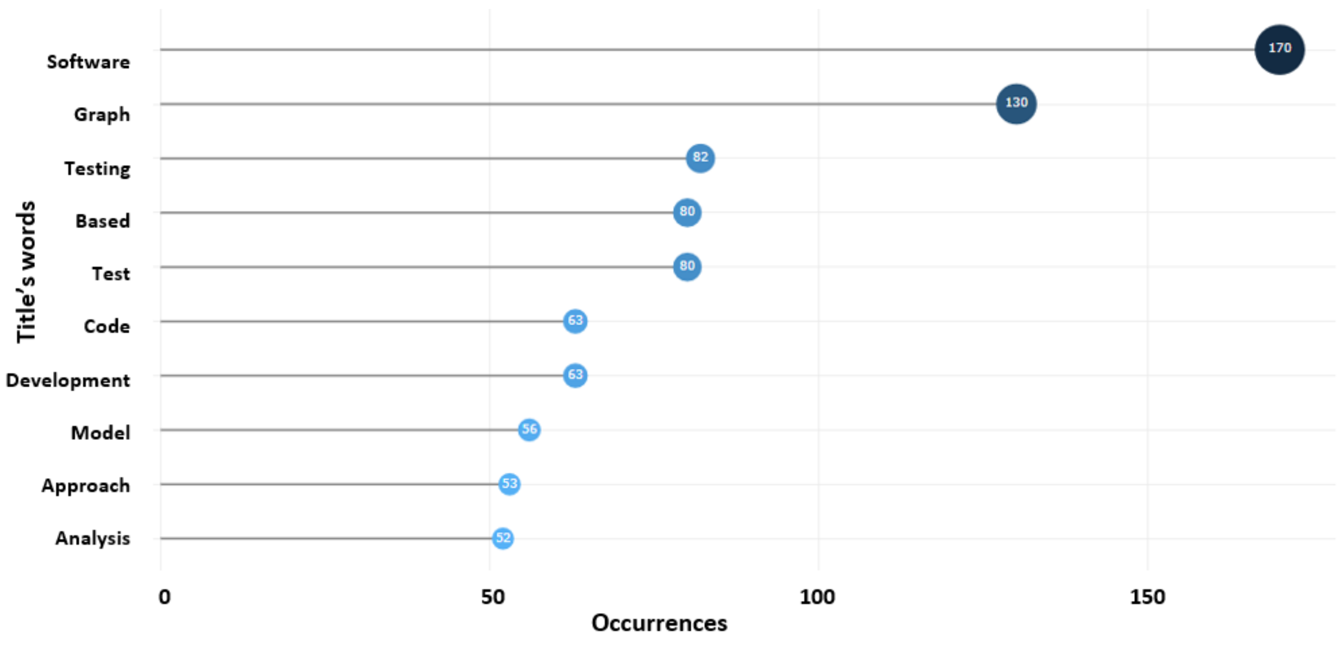


Fig. 18: The most pertinent terms from graph theory research used in software testing.

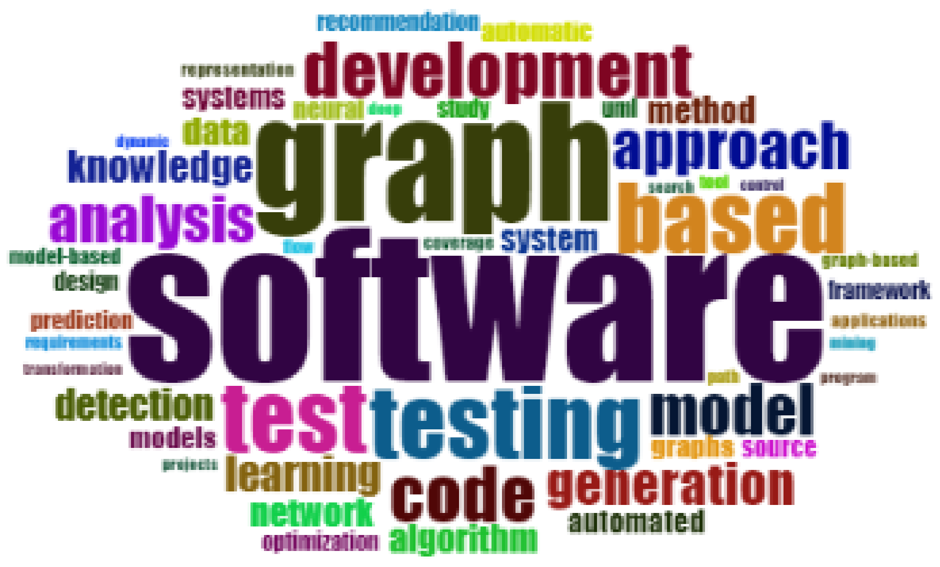


Fig. 19: Word cloud of the most frequently used keywords in software testing research.

## VI. Discussion

The outcomes of both bibliometric analysis and content studies have sparked considerable debate by yielding numerous inferences and implications. This comprehensive investigation spans 609 papers concerning intelligent software testing, authored by 1869 individuals, over the period from 2013 to 2023. With an average citation rate of 5.75, the study indicates significant academic engagement. In particular, most of the documents were co-authored, with only 1.44% being single-authored, underscoring the pronounced collaborative nature of research in this domain.

The three-field plots, employing three crucial metadata fields, provide valuable insights into the interrelationships among domains. These plots facilitate linking authors' contributions to particular keywords and the countries engaged in the research field. For instance, authors Wang, X., Belli, F., and Boopathi, M. from China, Germany, and India respectively, wielded considerable influence on the study of intelligent software testing. Moreover, our findings underscore a significant correlation between the research topics studied in China, India, and USA.

The findings also revealed seven clusters of keywords, each comprising eight or more words. Cluster 1 encompasses 14 words related to graph analysis, task analysis, and various

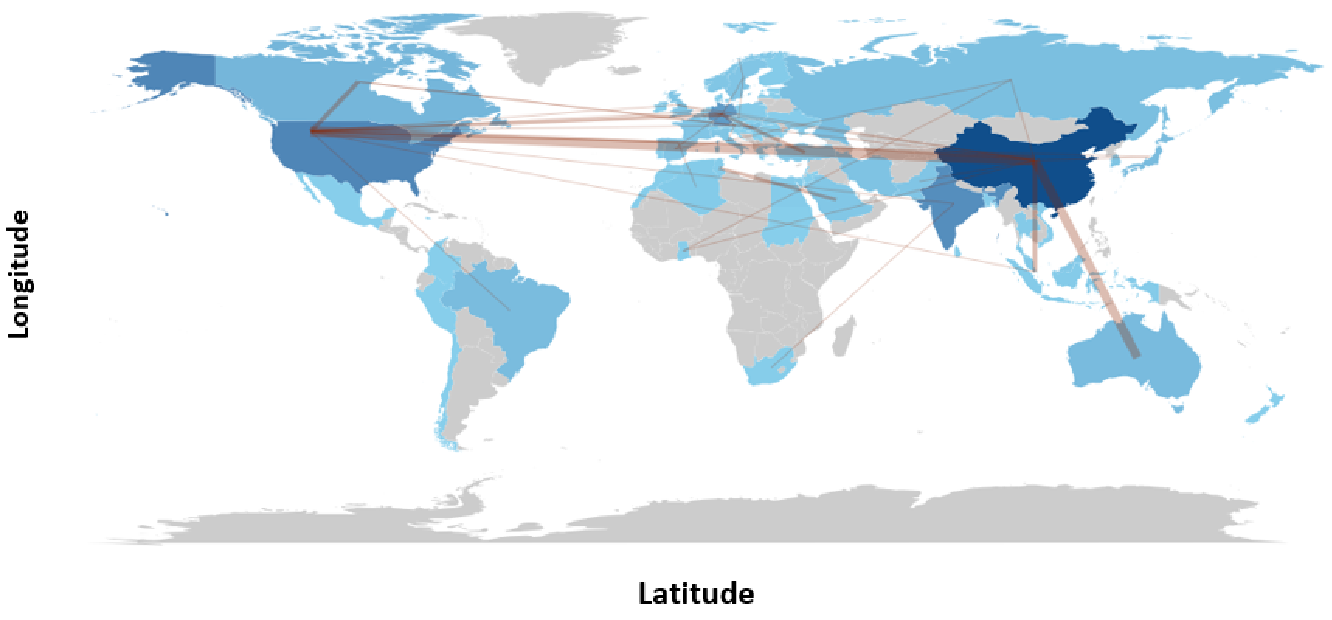


Fig. 20: Country collaboration world map.

types of graph. Cluster 2 consists of 12 keywords related to software development and other aspects of software engineering. Cluster 3 focuses on software testing, test cases, and algorithms, comprising 11 keywords. Cluster 4, closely associated with software architecture, contains 11 keywords. Cluster 5, centered around model-based testing and UML, contains 9 keywords. Keywords related to test data generation populate cluster 6. Lastly, cluster 7 discusses machine learning and mutation testing. These results underscore the widespread utilization of graph theory to enhance the quality of testing in software engineering.

Using knowledge frameworks, thematic maps are effective tools for displaying the dynamic and structural components of a research subject. They support the development of conceptual frameworks that outline the main ideas, topics, and theoretical tenets influencing an author's influence in the scientific community. These frameworks provide a thorough summary of important developments and trends in software testing research, allowing for the analysis of how concepts or situations change over time. Additionally, this method facilitates a targeted investigation of particular issues by giving academics lists of the most influential papers for each subject category. Furthermore, by allowing for projections about the possible future growth of certain subjects, scientific maps offer insights into the significance of topics based on centrality and density.

The findings of the bibliometric analysis reveal that a select group of authors contribute to the most prominent works in the field. With the majority of papers being open access, contributions are swiftly disseminated to the public, leading to the emergence of numerous writers as the topic progresses. Notably, the increasing citation rates, with the most cited paper accumulating 721 citations, underscore the current significance of the topic. China, India, and USA stand out as the top three contributors to scientific output in this domain, consistent with their leading positions in worldwide scientific productivity across all categories [76]. These results highlight the interdisciplinary nature of the study, as even the most productive researchers employ a range of approaches and expertise. To ensure effective knowledge dissemination, journals must be efficient and comprehensive. Our analysis identified "The IEEE Access" and "Journal of System and Software" as the primary sources with the highest citations on the subject, emphasizing their importance as platforms for disseminating research in the field.

This study specifically focused on articles related to intelligent software testing that were indexed in the Web of Science core collection database. While comparing datasets from multiple databases wasn't within the scope of this study, it's worth noting that searching different databases may yield varying sets of items, potentially leading to different outcomes in the analysis [77].

During the conduct of this study, the authors diligently sought to comprehensively gather all pertinent material associated with the topic. Nevertheless, the study faces several constraints, including a restricted search period spanning from 2013 to 2023; dependence on the WoS database, which might not encompass all indexed journals and could overlook relevant publications; limitations in the search process that could result in false positive or negative outcomes; and a focus solely on english language papers, potentially introducing bias towards english-speaking nations [78], [79].

Future research should address these limitations by expanding the search to multiple databases, incorporating articles in other languages, and considering broader timeframes. Additionally, further exploration of emerging technologies such as AI-driven data mining tools could enhance the depth and precision of bibliometric analyses in this domain [80]–[83].

## VII. Conclusion

This bibliometric analysis paper examines 609 publications spanning from 2013 to 2023 in relation to 11 research questions. Drawing data from the Web of Science database, the paper compares and evaluates worldwide research production concerning software testing, particularly focusing on

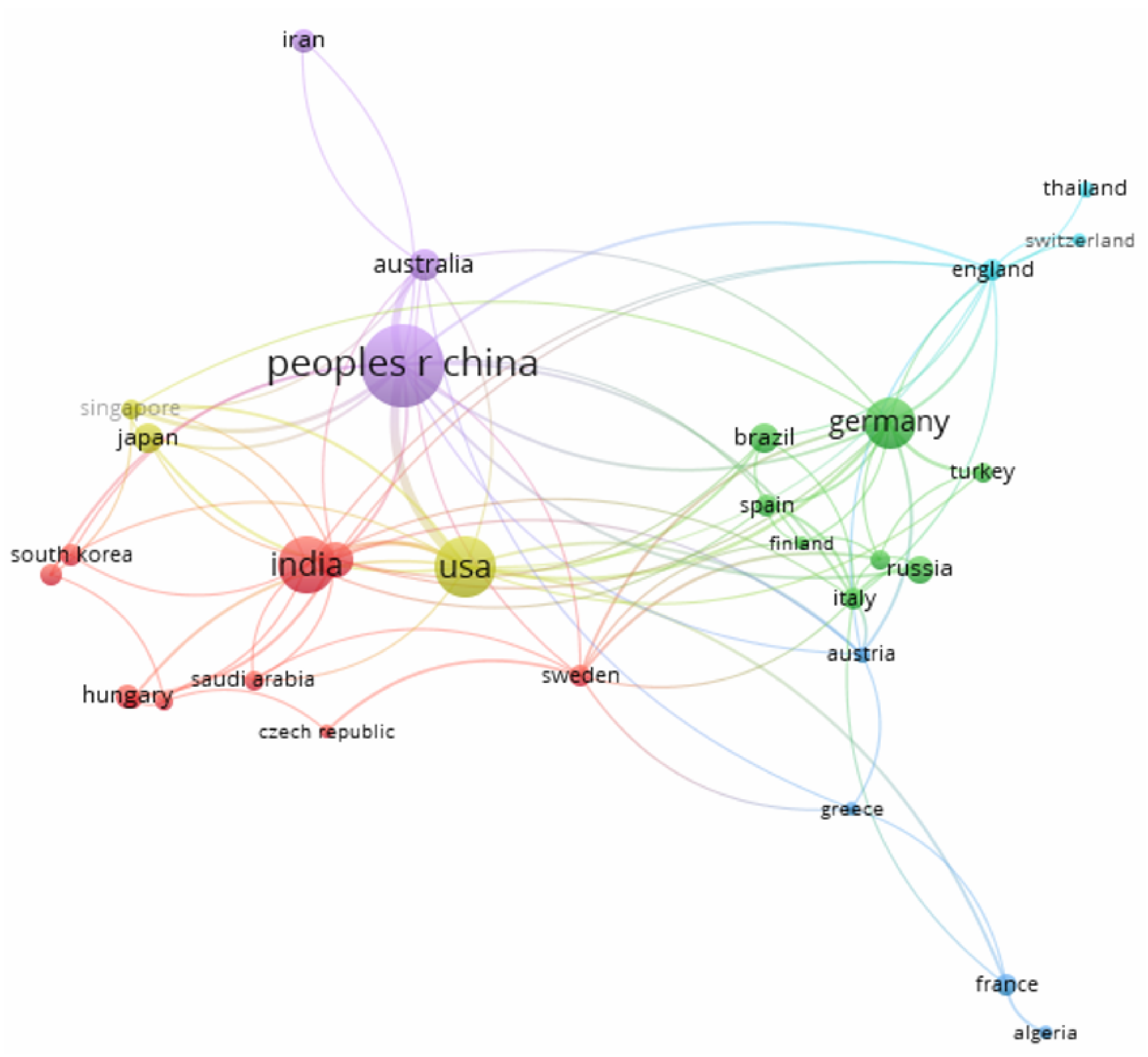


Fig. 21: Country collaboration world map via VOS Viewer.

graph theory. It identifies the most influential researchers globally and maps their geographical distribution and publications. To analyze the data and provide comprehensive visualizations for insight, we employed VOSviewer and the Biblioshiny program from the Bibliometrix package for R. Scientific publication trends indicate an average citation rate of 5.75. China, India, and the United States emerge as the top three countries in scientific production within this field. Collaboration is prevalent, as evidenced by only 1.44% of single-authored papers. Notably, "The IEEE Access" and "Journal of System and Software" are the two sources with the highest number of citations on the subject.

Our findings aim to illuminate potential future research directions and viewpoints in the swiftly changing discipline of software engineering. Giving a comprehensive overview of trends related to research in software testing based on graph theory, we hope to provide valuable insights for future investigations. There exist numerous opportunities for substantial future work in this domain.

Future work in software testing based on graph theory could focus on developing advanced algorithms for test case generation, prioritization, and fault location, leveraging the structural information inherent in software graphs. This could involve exploring dynamic graph-based testing techniques to handle evolving software systems, integrating machine learning approaches to enhance testing effectiveness, and addressing scalability challenges associated with large-scale software architectures. Additionally, research could investigate the application of graph-based testing methodologies in emerging domains such as cybersecurity, IoT, and cloud computing, while also focusing on automation and integration with DevOps practices to enable continuous testing and feedback loops. Empirical evaluations and case studies could further validate the effectiveness and practical applicability of these techniques in real-world software development scenarios, ultimately contributing to improving software quality and reliability.

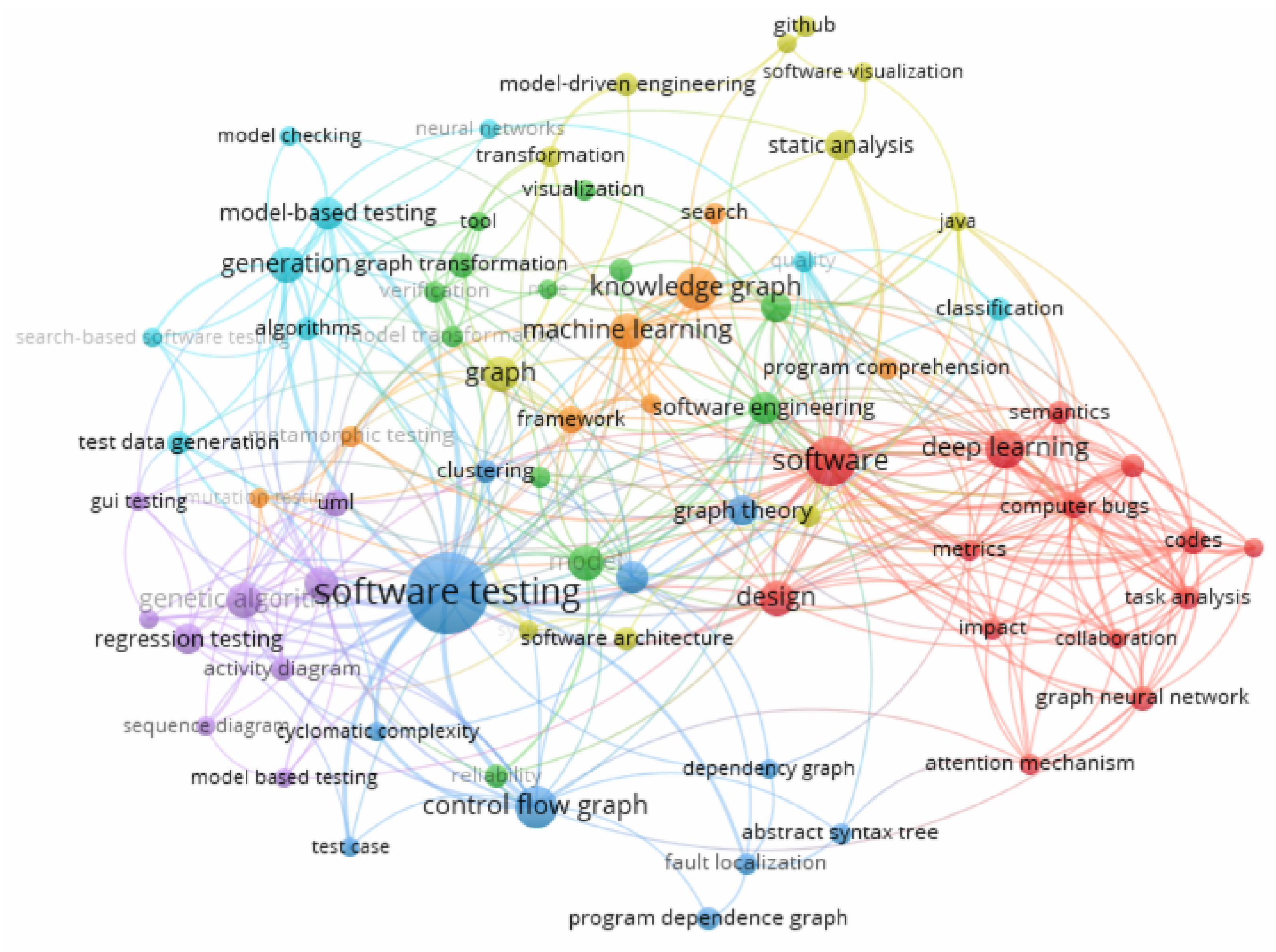


Fig. 22: Keywords co-occurrence network.

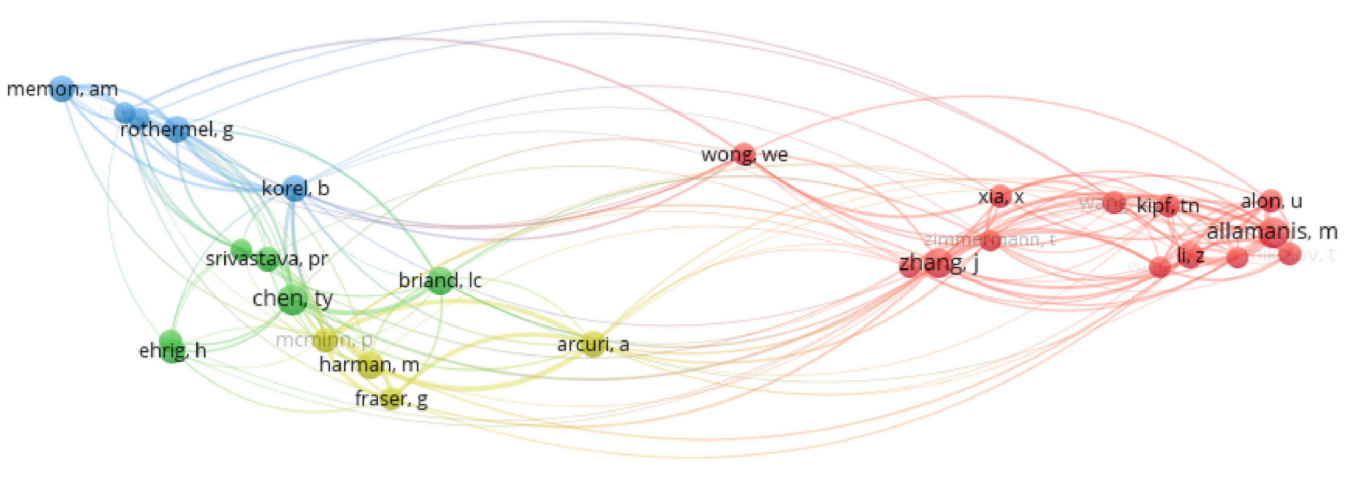


Fig. 23: Authors co-citation network.

## References

[1] I. T. Elgendy, M. R. Girgis, and A. A. Sewisy, "A ga-based approach to automatic test data generation for asp .net web applications," *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp. 557–564, 2020.

[2] A. Desiani, Erwin, B. Suprihatin, S. Yahdin, A. I. Putri, and F. R. Husein, "Bi-path architecture of cnn segmentation and classification method for cervical cancer disorders based on pap-smear images," *IAENG International Journal of Computer Science*, vol. 48, no. 3, pp. 782–791, 2021.

[3] J. A. Bondy and U. S. R. Murty, *Graph theory*. Springer Publishing Company, Incorporated, 2008.

[4] A. Z. AKTAŞ, E. YAĞDERELİ, and D. SERDAROĞLU, "An introduction to software testing methodologies," *Gazi University Journal of Science Part A: Engineering and Innovation*, vol. 8, no. 1, pp. 1–15, 2021.

[5] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.

[6] C. Elasri, N. Kharmoum, F. Saoiabi, M. Boukhlif, S. Ziti, and W. Rhalem, "Applying graph theory to enhance software testing in medical applications: A comparative study," in *International Conference on Advanced Intelligent Systems for Sustainable Development*.

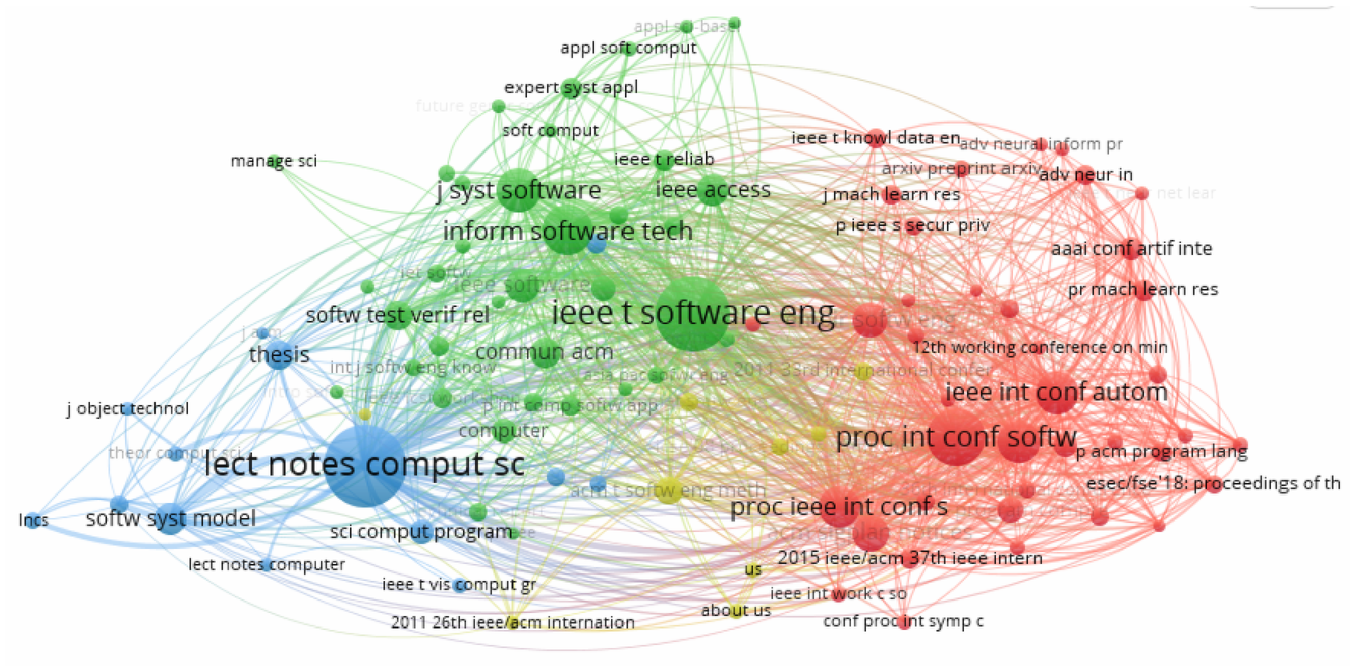


Fig. 24: Journals co-citation network.

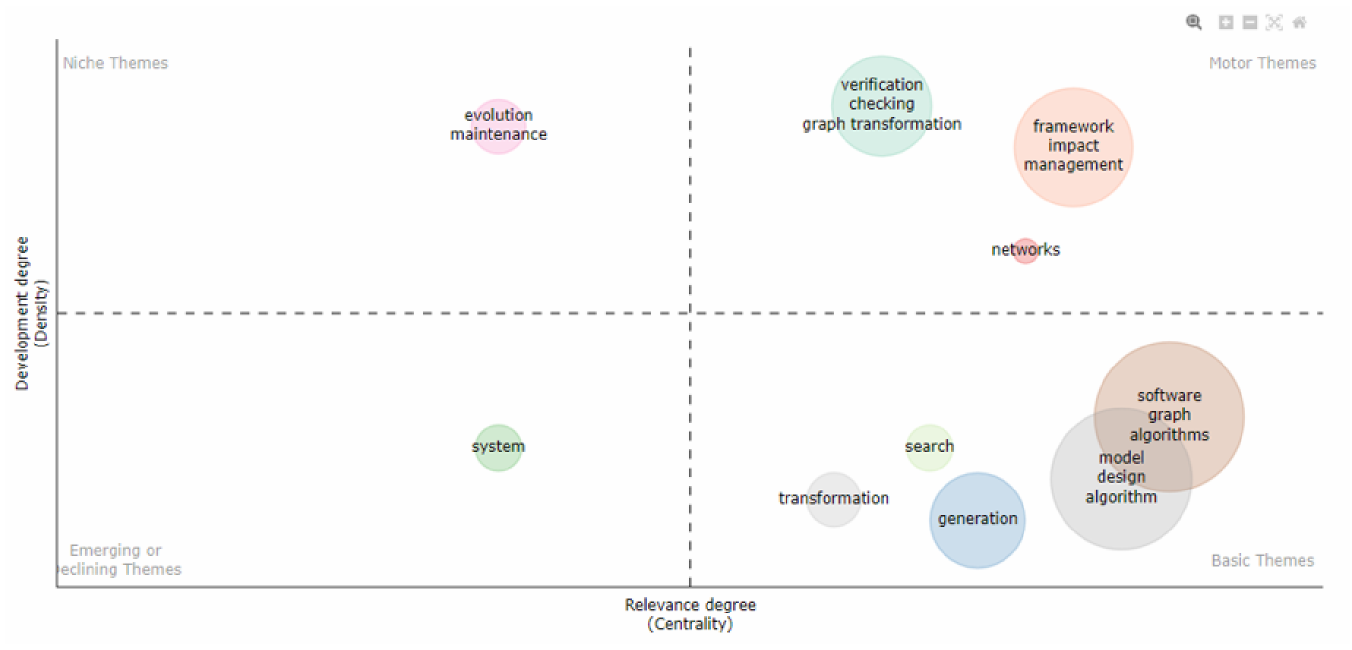


Fig. 25: Thematic map.

Springer, 2023, pp. 70–78.

[7] D. Nunez-Agurto, W. Fuertes, L. Marrone, and M. Macas, "Machine learning-based traffic classification in software-defined networking: A systematic literature review, challenges, and future research directions," *IAENG International Journal of Computer Science*, vol. 49, no. 4, pp. 1002–1015, 2022.

[8] C. Catal and D. Mishra, "Test case prioritization: a systematic mapping study," *Software Quality Journal*, vol. 21, pp. 445–478, 2013.

[9] P. Bourgade, H.-T. Yau, and J. Yin, "The local circular law ii: the edge case," *Probability Theory and Related Fields*, vol. 159, pp. 619–660, 2014.

[10] R. Elliott, J. Glauert, V. Jennings, and R. Kennaway, "An overview of the sigml notation and sigmlsigning software system," *sign-lang@ LREC 2004*, pp. 98–104, 2004.

[11] 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, 2009.

[12] H. D. Mills, "Software development," *IEEE Transactions on Software Engineering*, no. 4, pp. 265–273, 1976.

[13] N. Donthu, S. Kumar, D. Mukherjee, N. Pandey, and W. M. Lim, "How to conduct a bibliometric analysis: An overview and guidelines," *Journal of business research*, vol. 133, pp. 285–296, 2021.

[14] M. Boukhlif, M. Hanine, and N. Kharmoum, "A decade of intelligent software testing research: A bibliometric analysis," *Electronics*, vol. 12, no. 9, p. 2109, 2023.

[15] A. Trudova, M. Dolezel, and A. Buchalcevova, "Artificial intelligence in software test automation: A systematic," 2020.

[16] R. Lima, A. M. R. da Cruz, and J. Ribeiro, "Artificial intelligence applied to software testing: A literature review," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2020, pp. 1–6.

[17] E. Serna M, E. Acevedo M, and A. Serna A, "Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review," *Journal of Software: Evolution and Process*, vol. 31, no. 7, p. e2159, 2019.

[18] V. Garousi, S. Bauer, and M. Felderer, "Nlp-assisted software testing: A systematic mapping of the literature," *Information and Software*

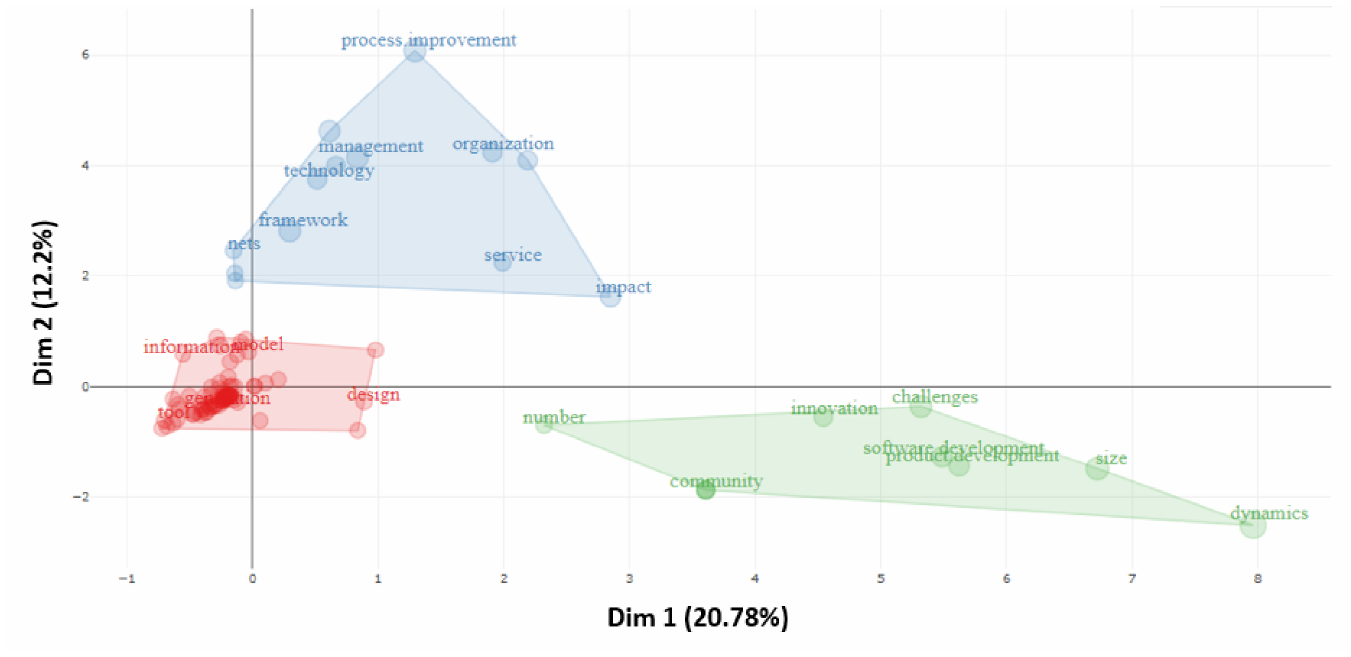


Fig. 26: Conceptual structure map using multiple correspondence analysis.

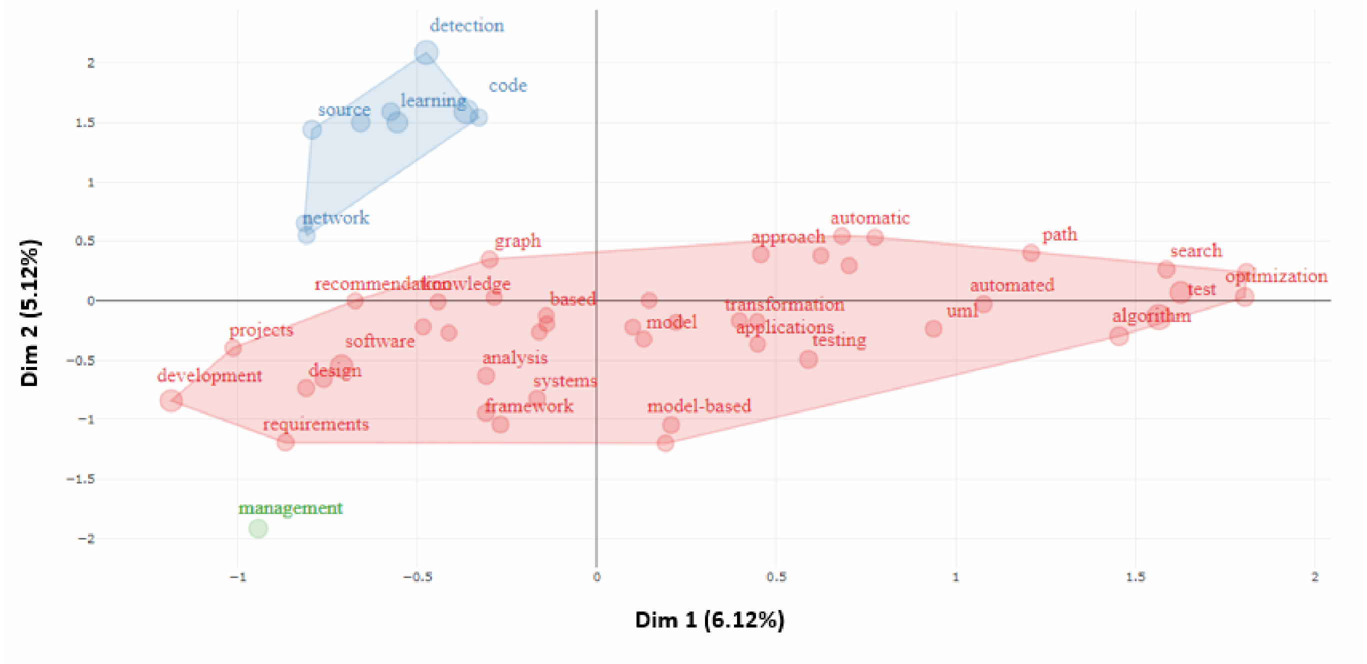


Fig. 27: Conceptual structure map using correspondence analysis.

*Technology*, vol. 126, p. 106321, 2020.

[19] V. Garousi, M. Felderer, and F. N. Kılıçaslan, “A survey on software testability,” *Information and Software Technology*, vol. 108, pp. 35–64, 2019.

[20] S. Zardari, S. Alam, H. A. Al Salem, M. S. Al Reshan, A. Shaikh, A. F. K. Malik, M. Masood ur Rehman, and H. Mouratidis, “A comprehensive bibliometric assessment on software testing (2016–2021),” *Electronics*, vol. 11, no. 13, p. 1984, 2022.

[21] A. S. Ghiduk, M. R. Girgis, and M. H. Shehata, “Higher order mutation testing: A systematic literature review,” *Computer Science Review*, vol. 25, pp. 29–48, 2017.

[22] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, “Software testing techniques: A literature review,” in *2016 6th international conference on information and communication technology for the Muslim world (ICT4M)*. IEEE, 2016, pp. 177–182.

[23] V. Garousi and M. V. Mäntylä, “A systematic literature review of literature reviews in software testing,” *Information and Software Technology*, vol. 80, pp. 195–216, 2016.

[24] ——, “When and what to automate in software testing? a multi-vocal literature review,” *Information and Software Technology*, vol. 76, pp. 92–117, 2016.

[25] K. Wiklund, S. Eldh, D. Sundmark, and K. Lundqvist, “Impediments for software test automation: A systematic literature review,” *Software Testing, Verification and Reliability*, vol. 27, no. 8, p. e1639, 2017.

[26] D. Zhang, “Machine learning in value-based software test data generation,” in *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’06)*. IEEE, 2006, pp. 732–736.

[27] M. Esnaashari and A. H. Damia, “Automation of software test data generation using genetic algorithm and reinforcement learning,” *Expert Systems with Applications*, vol. 183, p. 115446, 2021.

[28] X.-m. Zhu and X.-f. Yang, “Software test data generation automatically based on improved adaptive particle swarm optimizer,” in *2010 International Conference on Computational and Information Sciences*. IEEE, 2010, pp. 1300–1303.

[29] C. Koleejan, B. Xue, and M. Zhang, “Code coverage optimisation in genetic algorithms and particle swarm optimisation for automatic

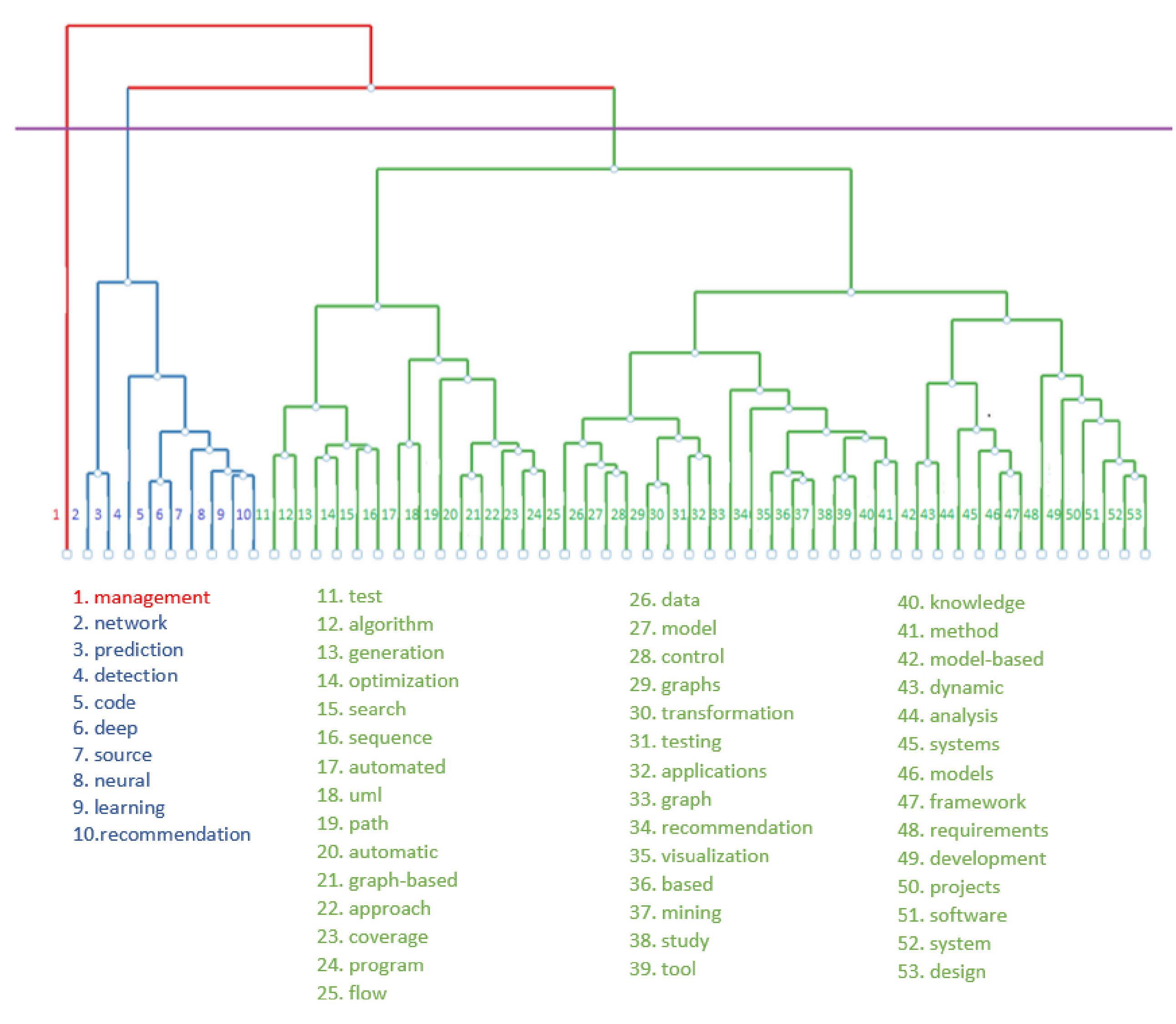


Fig. 28: Topic Dendrogram using correspondence analysis.

software test data generation," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 1204–1211.
[30] M. Huang, C. Zhang, and X. Liang, "Software test cases generation based on improved particle swarm optimization," in *Proceedings of 2nd International Conference on Information Technology and Electronic Commerce*. IEEE, 2014, pp. 52–55.
[31] C. Liu, "Research on software test data generation based on particle swarm optimization algorithm," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2021, pp. 1375–1378.
[32] M. Khari and P. Kumar, "A novel approach for software test data generation using cuckoo algorithm," in *Proceedings of the second international conference on information and communication technology for competitive strategies*, 2016, pp. 1–6.
[33] M. Alshraideh and L. Bottaci, "Search-based software test data generation for string data using program-specific search operators," *Software Testing, Verification and Reliability*, vol. 16, no. 3, pp. 175–203, 2006.
[34] S. Mairhofer, R. Feldt, and R. Torkar, "Search-based software testing and test data generation for a dynamic programming language," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 1859–1866.
[35] P. McMinn, "Search-based software test data generation: a survey," *Software testing, Verification and reliability*, vol. 14, no. 2, pp. 105–156, 2004.
[36] L. F. de Lima, L. M. Peres, A. R. A. Grégio, and F. Silva, "A systematic literature mapping of artificial intelligence planning in software testing." in *ICSOFT*, 2020, pp. 152–159.
[37] S. S. Dahiya, J. K. Chhabra, and S. Kumar, "Application of artificial bee colony algorithm to software testing," in *2010 21st Australian software engineering conference*. IEEE, 2010, pp. 149–154.
[38] D. J. Mala, V. Mohan, and M. Kamalapriya, "Automated software test optimisation framework–an artificial bee colony optimisation-based approach," *IET software*, vol. 4, no. 5, pp. 334–348, 2010.
[39] K. Karnavel and J. Santhoshkumar, "Automated software testing for application maintenance by using bee colony optimization algorithms (bco)," in *2013 International Conference on Information Communication and Embedded Systems (ICICES)*. IEEE, 2013, pp. 327–330.
[40] P. Lakshminarayana and T. SureshKumar, "Automatic generation and optimization of test case using hybrid cuckoo search and bee colony algorithm," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 59–72, 2020.
[41] W. Lijuan, Z. Yue, and H. Hongfeng, "Genetic algorithms and its application in software test data generation," in *2012 International Conference on Computer Science and Electronics Engineering*, vol. 2. IEEE, 2012, pp. 617–620.
[42] D. Berndt, J. Fisher, L. Johnson, J. Pinglikar, and A. Watkins, "Breeding software test cases with genetic algorithms," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. IEEE, 2003, pp. 10–pp.
[43] S. Yang, T. Man, J. Xu, F. Zeng, and K. Li, "Rga: A lightweight and effective regeneration genetic algorithm for coverage-oriented software test data generation," *Information and Software Technology*, vol. 76, pp. 19–30, 2016.
[44] R. Khan and M. Amjad, "Automatic test case generation for unit software testing using genetic algorithm and mutation analysis," in *2015 IEEE UP section conference on electrical computer and electronics (UPCON)*. IEEE, 2015, pp. 1–5.
[45] Y. Dong and J. Peng, "Automatic generation of software test cases based on improved genetic algorithm," in *2011 International Conference on Multimedia Technology*. IEEE, 2011, pp. 227–230.

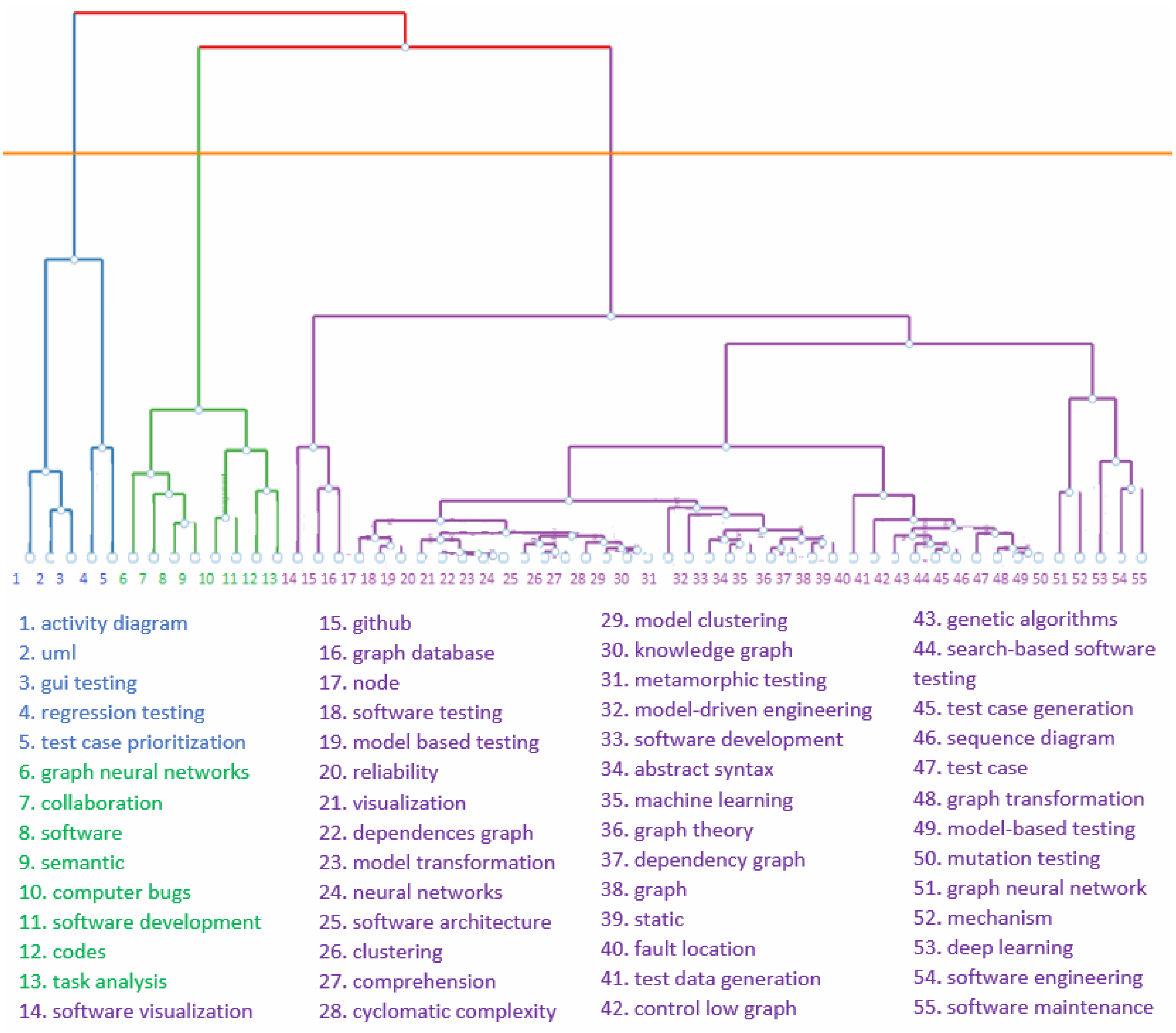


Fig. 29: Topic Dendrogram using correspondence analysis.

[46] A. Bouchachia, "An immune genetic algorithm for software test data generation," in *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*. IEEE, 2007, pp. 84–89.

[47] Y. P. Peng and B. Zeng, "Software test data generation for multiple paths based on genetic algorithms," *Applied Mechanics and Materials*, vol. 263, pp. 1969–1973, 2013.

[48] K. B. Cohen, L. E. Hunter, and M. Palmer, "Assessment of software testing and quality assurance in natural language processing applications and a linguistically inspired approach to improving it," in *Trustworthy Eternal Systems via Evolving Software, Data and Knowledge: Second International Workshop, EternalS 2012, Montpellier, France, August 28, 2012, Revised Selected Papers 2*. Springer, 2013, pp. 77–90.

[49] I. Ahsan, W. H. Butt, M. A. Ahmed, and M. W. Anwar, "A comprehensive investigation of natural language processing techniques and tools to generate automated test cases," in *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, 2017, pp. 1–10.

[50] A. Ansari, M. B. Shagufta, A. S. Fatima, and S. Tehreem, "Constructing test cases using natural language processing," in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. IEEE, 2017, pp. 95–99.

[51] H. Wang, L. Bai, M. Jiezhang, J. Zhang, and Q. Li, "Software testing data analysis based on data mining," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2017, pp. 682–687.

[52] X. Wang, S. Cui, and Q. Ai, "Data-driven channel estimation for otfs systems affected by hpa," *IAENG International Journal of Computer Science*, vol. 52, no. 1, pp. 103–110, 2025.

[53] S. Yu, J. Ai, and Y. Zhang, "Software test data generation based on multi-agent," in *Advances in Software Engineering: International Conference on Advanced Software Engineering and Its Applications, ASEA 2009 Held as Part of the Future Generation Information Technology Conference, FGIT 2009, Jeju Island, Korea, December 10-12, 2009. Proceedings*. Springer, 2009, pp. 188–195.

[54] P. Dhavachelvan and G. Uma, "Complexity measures for software systems: Towards multi-agent based software testing," in *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing, 2005*. IEEE, 2005, pp. 359–364.

[55] J. Tang, "Towards automation in software test life cycle based on multi-agent," in *2010 International Conference on Computational Intelligence and Software Engineering*. IEEE, 2010, pp. 1–4.

[56] S. Alyahya, "Collaborative crowdsourced software testing," *Electronics*, vol. 11, no. 20, p. 3340, 2022.

[57] T. Górski, "The k+ 1 symmetric test pattern for smart contracts," *Symmetry*, vol. 14, no. 8, p. 1686, 2022.

[58] A. Bijlsma, H. J. Passier, H. Pootjes, and S. Stuurman, "Template method test pattern," *Information Processing Letters*, vol. 139, pp. 8–12, 2018.

[59] J. López, N. Kushik, and N. Yevtushenko, "Source code optimization using equivalent mutants," *Information and software technology*, vol. 103, pp. 138–141, 2018.

[60] S. Segura, J. Troya, A. Durán, and A. Ruiz-Cortés, "Performance metamorphic testing: A proof of concept," *Information and Software Technology*, vol. 98, pp. 1–4, 2018.

[61] A.-W. Harzing and S. Alakangas, "Google scholar, scopus and the web of science: a longitudinal and cross-disciplinary comparison,"

*Scientometrics*, vol. 106, pp. 787–804, 2016.

[62] M. Norris and C. Oppenheim, "Comparing alternatives to the web of science for coverage of the social sciences' literature," *Journal of informetrics*, vol. 1, no. 2, pp. 161–169, 2007.

[63] A. A. Chadegani, H. Salehi, M. M. Yunus, H. Farhadi, M. Fooladi, M. Farhadi, and N. A. Ebrahim, "A comparison between two main academic literature collections: Web of science and scopus databases," *arXiv preprint arXiv:1305.0377*, 2013.

[64] X.-W. Feng, M. Hadizadeh, and J. P. G. Cheong, "Global trends in physical-activity research of autism: Bibliometric analysis based on the web of science database (1980–2021)," *International journal of environmental research and public health*, vol. 19, no. 12, p. 7278, 2022.

[65] I. Skute, "Opening the black box of academic entrepreneurship: a bibliometric analysis," *Scientometrics*, vol. 120, no. 1, pp. 237–265, 2019.

[66] D. Zhao and A. Strotmann, *Analysis and visualization of citation networks*. Morgan & Claypool Publishers, 2015.

[67] X. Ma, L. Zhang, J. Wang, and Y. Luo, "Knowledge domain and emerging trends on echinococcosis research: A scientometric analysis," *International journal of environmental research and public health*, vol. 16, no. 5, p. 842, 2019.

[68] Y.-C. Lee, C. Chen, and X.-T. Tsai, "Visualizing the knowledge domain of nanoparticle drug delivery technologies: a scientometric review," *Applied Sciences*, vol. 6, no. 1, p. 11, 2016.

[69] M. Aria and C. Cuccurullo, "bibliometrix: An r-tool for comprehensive science mapping analysis," *Journal of informetrics*, vol. 11, no. 4, pp. 959–975, 2017.

[70] I. Suryana, D. Chaerani, A. S. Abdullah, A. S. Prabuwono, K. R. A. Muslihin, and A. Z. Irmansyah, "Systematic literature review for optimization system with advection-diffusion-reaction non-linear equation in water quality," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 12, pp. 2541–2554, 2024.

[71] C. Chen, "Science mapping: a systematic review of the literature," *Journal of data and information science*, vol. 2, no. 2, pp. 1–40, 2017.

[72] M. B. Negahban and N. Zarifsanaiey, "Network analysis and scientific mapping of the e-learning literature from 1995 to 2018." *Knowledge Management & E-Learning*, vol. 12, no. 3, pp. 268–279, 2020.

[73] M. J. Cobo, A. G. López-Herrera, E. Herrera-Viedma, and F. Herrera, "An approach for detecting, quantifying, and visualizing the evolution of a research field: A practical application to the fuzzy sets theory field," *Journal of informetrics*, vol. 5, no. 1, pp. 146–166, 2011.

[74] J. Matute and L. Linsen, "Evaluating data-type heterogeneity in interactive visual analyses with parallel axes," in *Computer Graphics Forum*, vol. 41, no. 1. Wiley Online Library, 2022, pp. 335–349.

[75] H. Ejaz, H. M. Zeeshan, F. Ahmad, S. N. A. Bukhari, N. Anwar, A. Alanazi, A. Sadiq, K. Junaid, M. Atif, K. O. A. Abosalif *et al.*, "Bibliometric analysis of publications on the omicron variant from 2020 to 2022 in the scopus database using r and vosviewer," *International Journal of Environmental Research and Public Health*, vol. 19, no. 19, p. 12407, 2022.

[76] K. Jaffe, E. Ter Horst, L. H. Gunn, J. D. Zambrano, and G. Molina, "A network analysis of research productivity by country, discipline, and wealth," *Plos one*, vol. 15, no. 5, p. e0232458, 2020.

[77] F. Saoiabi, N. Kharmoum, C. Elasri, M. E. Boukhari, S. Ziti, and W. Rhalem, "Agile software engineering in medical environments: Challenges and opportunities," in *International Conference on Advanced Intelligent Systems for Sustainable Development*. Springer, 2023, pp. 79–87.

[78] N. Kharmoum, S. Ziti, Y. Rhazali, and F. Omary, "A method of model transformation in mda approach from e3value model to bpmn2 diagrams in cim level," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 599–615, 2019.

[79] N. Kharmoum, K. El Bouchti, N. Laaz, W. Rhalem, and Y. Rhazali, "Transformations' study between requirements models and business process models in mda approach," *Procedia Computer Science*, vol. 170, pp. 819–824, 2020.

[80] C. Elasri, N. Kharmoum, F. Saoiabi, S. N. Lagmiri, and S. Ziti, "Analyzing generative ai's impact on graph theory and software testing: A comparative study," in *Smart Business and Technologies*, S. N. Lagmiri, M. Lazaar, and F. M. Amine, Eds. Cham: Springer Nature Switzerland, 2025, pp. 305–314.

[81] F. Saoiabi, N. Kharmoum, C. Elasri, S. N. Lagmiri, and S. Ziti, "Generative ai in software engineering: Enhancing development and innovation," in *Smart Business and Technologies*, S. N. Lagmiri, M. Lazaar, and F. M. Amine, Eds. Cham: Springer Nature Switzerland, 2025, pp. 315–323.

[82] N. Kharmoum, S. Retal, K. E. Bouchti, W. Rhalem, M. Z. Es-Sadek, S. Ziti, and M. Ezziyyani, "Agile user stories' driven method: a novel users stories meta-model in the mda approach," in *International Conference on Advanced Intelligent Systems for Sustainable Development*. Springer, 2022, pp. 145–154.

[83] M. Boukhlif, M. Hanine, N. Kharmoum, A. R. Noriega, D. G. Obeso, and I. Ashraf, "Natural language processing-based software testing: A systematic literature review," *IEEE Access*, 2024.

**Chaimae ELASRI** received her Master Degree in Data Engineering and Software Development from the Faculty of Sciences Rabat, Morocco in 2022. Currently, she is a Software Engineering at the Multinational Company Cegedim Business Services in Morocco. She is a member of the IPPS (Intelligent Processing Systems & Security) team, and she prepares her Ph.D. degree at the Faculty of Sciences, Mohammed V University in Rabat, Morocco. Her research interest focuses on software engineering, software testing, artificial intelligence, analysis and conceptual modeling..

Fadwa SAOIABI received her Master Degree in Data Engineering and Software Development from the Faculty of Sciences Rabat, Morocco in 2022. Currently, she is a Software Engineering at the Multinational Company Cegedim Business Services in Morocco. She is a member of the IPPS (Intelligent Processing Systems & Security) team, and she prepares her Ph.D. degree at the Faculty of Sciences, Mohammed V University in Rabat, Morocco. Her research interest focuses on software engineering, artificial intelligence, software development, analysis and conceptual modeling.

Nassim KHARMOUM, Ph.D. and university Professor. He received his diploma of doctor degree in Computer Science from the Mohammed V University in Rabat. Currently, he is Professor at the National Center for Scientific and Technical Research in Morocco. He proposed different validated methods in scientific articles published in international journals, books, and conferences. His research interest focuses on software engineering, artificial intelligence, analysis and conceptual modeling.

Soumia ZITI, is a Franco-Moroccan teacher-researcher lady at the Faculty of Sciences of Mohammed V University in Rabat since 2007. She holds a PhD in computer science in the field of graph theory, and a diploma in advanced studies in fundamental computer science, both obtained at the University of Orleans in France. His areas of expertise and research are graph theory, information systems, artificial intelligence, data science, software development engineering, modeling of relational databases and big data, cryptography and numerical methods and simulations in spintronics. She has over than sixty publications in high-level international journals and conferences in several research areas. In addition, she coordinates or participates in several educational or socio-economic projects.