









Research Article

A Systematic Analysis of Regression Test Case Selection: A Multi-Criteria-Based Approach

Muhammad Rehan ¹, Norhalina Senan ¹, Muhammad Aamir ², Ali Samad ³,
Mujtaba Husnain ³, Noraini Ibrahim ¹, Sikandar Ali ⁴, and Hizbullah Khatak ⁵

¹Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia (UTHM),
Johor 8000, Malaysia

²School of Electronics, Computing and Mathematics, University of Derby, Derby 01332, UK

³Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan

⁴Department of Computer Science and Technology, China University of Petroleum-Beijing, Beijing 102249, China

⁵Department of Information Technology, Hazara University Mansehra, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Muhammad Aamir; m.aamir@derby.ac.uk and Sikandar Ali; sikandar@cup.edu.cn

Received 14 July 2021; Accepted 21 August 2021; Published 22 September 2021

Academic Editor: Kifayat Ullah

Copyright © 2021 Muhammad Rehan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In applied software engineering, the algorithms for selecting the appropriate test cases are used to perform regression testing. The key objective of this activity is to make sure that modification in the system under test (SUT) has no impact on the overall functioning of the updated software. It is concluded from the literature that the efficacy of the test case selection solely depends on the following metrics, namely, the execution cost of the test case, the lines of the code covered in unit time also known as the code coverage, the ability to capture the potential faults, and the code modifications. Furthermore, it is also observed that the approaches for the regression testing developed so far generated results by focusing on one or two parameters. In this paper, our key objectives are twofold: one is to explore the importance of the role of each metric in detail. The secondary objective is to study the combined effect of these metrics in test case selection task that is capable of achieving more than one objective. In this paper, a detailed and comprehensive review of the work related to regression testing is provided in a very distinct and principled way. This survey will be useful for the researchers contributing to the field of regression testing. It is noteworthy that our systematic literature review (SLR) included the noteworthy work published from 2007 to 2020. Our study observed that about 52 relevant studies focused on all of the four metrics to perform their respective tasks. The results also revealed that about 30% of the different categories of regression test case reported the results using metaheuristic regression test selection (RTS). Similarly, about 31% of the literature reported results using the generic regression test case selection techniques. Most of the researchers focus on the datasets, namely, Software-Artifact Infrastructure Repository (SIR), JodaTime, TreeDataStructure, and Apache Software Foundation. For validation purpose, following parameters were focused, namely, the inclusiveness, precision, recall, and retest-all.

1. Introduction

The cost associated with the regression testing can be curtailed by the in-depth analysis for finding the optimal solutions of the three processes, namely, the regression test selection (RTS), regression test prioritization (RTP), and the regression test reduction (RTR) [1–4]. The process of RTS deals with selecting a set of test cases from a repository of test suites that are capable enough of performing the regression

test [5–7]. Similarly, the RTP is the study of rearranging the test cases in the test suite with higher priority based on the ability of capturing faults in minimum time, cost, and effort. Furthermore, the test cases are chosen from the categories like re-usable test cases, change affecting test cases, fault affecting test cases, and redundant test cases. On the other hand, the removal of redundant test cases that failed to capture the maximum faults is known as the RTR process [8, 9].

Figure 1 shows the details of related 2776 articles retrieved from Web of Science. Similarly, Figure 1(a) depicts the articles published in the period 2007–2020 on software testing in general and test selection in specific. More than 900 articles were published in 2018, which indicates research on the software regression testing in applied software engineering field. Figure 1(b) shows the 2007–2020 citation trends for articles on various approaches used in software testing while Figure 2 represents the global distribution of studies on artificial intelligence (AI) applications in software regression testing. This research has spread across the globe, especially active in the United States, China, Germany, Japan, India, Australia, and several European countries. The conventional approaches used in RTS comprised of the following three components, namely, the test case selection model, the selection parameters, and the selection adequacy measures, as shown in Figure 1. The original target program *Prog*, its modified version *Prog'*, and test suite *TS* having test cases are given as inputs to the first component of the system. The test case selection model tries to identify the change in the code in the modified version *Prog'*. Besides, it also retrieves other relevant and important information like the code coverage, number of faults detected, and the total execution time of the test case. On completion, the model may select the test cases from *TS* and move it to *TS'* (a subset of *TS*) on the basis of the computations performed in the model. These computations play a key role in estimating the effectiveness of the algorithms used for the selection of appropriate test cases. It is worthy to note that we deduced the idea of writing such an article from a noteworthy work reported in [10].

It is pertinent to mention that these computations can be used in a twofold way in the RTS process; the first is to observe the performance and then select a set of appropriate test cases based on some selection criteria, and second these calculations play a significant role in evaluating the efficacy of RTS by comparing with other approaches. It is of significant importance that to organize these four metrics, namely, cost, coverage, fault detection, and code modifications, we have to analyze the dependencies with respect to each other. The key issues in this regard are discussed as under; the first issue is to choose the suitable and apposite type of these measures, like the subtypes of the coverage-based information, the types of the potential faults, and the degree of severity. The critical issue that we try to resolve is to acquire and explore the model that is closely related to the RTS that helps in justifying including these four parameters for the selection of test cases on the basis of their efficacy and adequacy scale. Furthermore, we also try to explore and identify the set of techniques that are capable of producing the efficient and acceptable results based on these effective measures. The primary objective of this paper is to explore the test case selection methods that perform solely on these metrics and work as an effective contributor. The second objective our study is to assess the current state-of-the-art regression test case selection frameworks and techniques developed so far and to identify the available datasets and algorithms for the solution of test case selection problems.

The paper is further organized as follows. In Section 2, the methodology adopted in this paper is discussed in detail by including the source of data and the framework adopted for analysis. In Section 3 and Section 4, we discussed the various indicators on the basis of which we analyzed the bibliographic data related to test case selection, namely, the co-cited reference network, burst references, co-occurrence keyword network, burst keywords, and dual-map overlay network. For this purpose, we make use of the information visualization software VOSviewer [11]. It is pertinent to mention that this program is equally applicable in both focusing on the structural and fundamental variations in regression testing and identifying the new trends in software testing domain. The paper is concluded in Section 5 followed by Section 6 in which the limitations and future work are given.

2. Review Methodology

In this section, we will discuss the methods that we used to retrieve the related articles. In this review, we collected the noteworthy articles from a number of databases associated with Web of Science (WoS). The WoS platform is sufficient enough to provide and explore a number of useful links that are related to the search query and criteria that the researcher may use for collecting the data. Furthermore, the repository of the research articles offers a variety of search criteria to search the items with complete bibliographic information, previous versions of the article, the reference data that are cited according to the current article, and the links to full texts. The WoS repository is different from other sources in a way that it adds about 20,000 articles along with an average of 500, 000 cited references every other week.

Furthermore, in this survey, we focused to include the related literature as much as possible. To perform this task, we first quantified the compactness of the retrieved articles by using a number of combinations of the relevant keywords. Furthermore, we also tried to verify manually the articles retrieved from the WoS. We applied the following query to search (QtS) and retrieved the related articles.

QtS = “(Regression Testing OR Test case selection OR Test case Prioritization OR Test Case Reduction) AND (Multi-level OR Multi-criteria OR Multi-purpose OR multi-dimensional OR multi-directional)”.

It is clear that our search query has two parts: (1) regression testing related terms and (2) multi-criteria-based related terms. The query uses 9 keywords that cover most of the fields of software regression testing and test case selection. The key purpose of the query is to search the articles that reported the results on test case selection while considering multiple criteria at the same time. The bibliographic metadata of each paper contain the following information: the title of the article, list of authors indicating the corresponding author, its abstract, the keywords, digital object identifier (DOI), journal (or conference) name, and references. It is noteworthy that our systematic review is solely based on the articles retrieved through Science Citation Index Expanded (SCIE) dataset. The dataset contains about 2776 records which is about 31% of the literature reported in the domain of regression testing.

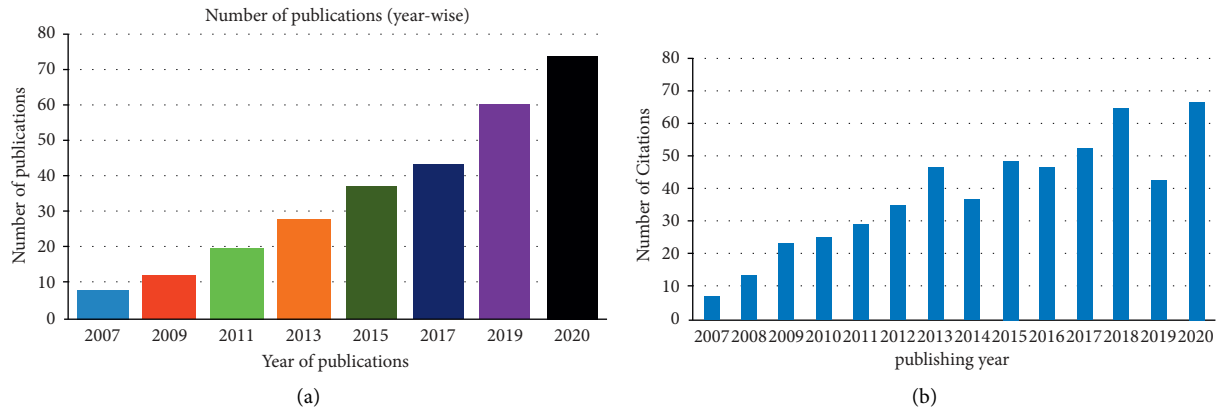


FIGURE 1: Trend in research on software regression testing in applied software engineering: (a) published articles each year; (b) article citations each year.



FIGURE 2: The heatmap for studies on AI applications in software regression testing indicating the regions that are highly active in this domain.

2.1. Scientometric Analysis. With the rapid growth in the literature published in a number of domains, the visualization tools were made efficient enough to process the publications' metadata for better understanding. The relevant terms are depicted in Table 1.

The most widely used software for mapping the scientific literature, namely, the CiteSpace V [12], is one of the tools that is capable enough of generating a number of mappings like the network representing the references that were found co-cited in the literature, the frequency of the keywords appeared, and so on using dual-map overlay network. Furthermore, it is also helpful in identifying the changes and modifications in the hotspots by computing the cluster size from the associated network to determine foremost and frontier topics. This task is performed by calculating the annual publications published on average using its cluster. Furthermore, we also make use of the VOSviewer [11] to perform the analysis at co-authorship level. The VOSviewer is one of the software tools to construct the visualization of the bibliometric or citation-based graph models. The VOSviewer is equally applicable

in text mining related tasks for the same purpose mentioned above.

The analysis of these network visualizations produced useful findings and observations about the emerging trends in software test case selection using different approaches while focusing on multiple criteria. Furthermore, a disciplined framework is adopted to perform functions on the dual-map overlays to ascertain the target hotspots with the recent trends in the field of software testing.

3. Scientometric Analysis and Visualization for Detecting the Emerging Trends, Structural Changes, and Future Work in Software Test Case Selection

The contributors in the scientometric analysis concluded that the network-based mapping containing the links representing the co-citations among the document pairs is capable enough to reveal the research focus and knowledge base of a domain [13–15].

TABLE 1: Basic terms and their explanation along with the respective purposes.

Terms	Explanation	Usage
Burst references	This term depicts the number of citations of a specific research document in a specific time. The higher the value of burst references, the higher the impact of that document.	It is used to show the concern level of research communities working in the same domain
Hotspots	It gives information about the number of times a specific research document is visited	It is useful in identifying references and the keywords that appeared in most of the research articles
Centrality	This term refers to the availability of the document either located in some center of the repository or some other location	This assists in finding the relevant research articles in the neighborhood of the visualization graph (or network)
Modularity	This term is related to the overall structure of the visualization network	This information is used to identify and extract the minimum set of the related articles in the network (i.e., clique in a graph)
Co-cited references	It represents the situation when two different research items were cited in a single document	It helps in finding the set of significant research articles along with the recent trends
Co-occurrence keywords	If two relevant keywords appear simultaneously in a single document	This information is helpful in identifying the significant and focused keywords
Silhouette value	This term depicts how much similar a research document is to its cluster neighbors	Its value is in $[-1, 1]$ that refers to the degree of associativity of the research articles within a cluster
Mutual information (MI)	This refers to the information about the two different clusters	It is computed as the similarity measure of two or more clusters in a network

The graph of the network representing a model of co-cited references to extract the research on the application of test case selection can be generated by importing all the 2776 research documents to CiteSpace V. The parameters are tuned to the following values. We retrieved the related documents published in the time span of 2007–2020. Furthermore, we used the switch of “Setup per slice” for each paper included in our scientometrics analysis. If the switch is set to 100, it means that top 100 papers will be selected for visualization in the coreference tree that was mostly cited. It means that if we enter a value of 50, then CiteSpace will select the 50 most cited or occurred items from each slice to construct a network, depending on the node types we selected in the previous step. Similarly, if we selected multiple node types, then these nodes will be ranked by the number of times they appeared in the records for each slice. It is pertinent to mention that the overall structure of the resulting network is not too large to be pruned; therefore, there is no need of pruning the graph. The generated results showed that, in Figures 3–5, the co-cited references network generated by 10, 237 cited documents (from 2776 citing documents) created about 527 nodes with mutual connections of 1587 connections. It is pertinent to mention that from the publications retrieved between 2007 and 2020, a total of 18 clusters were displayed by removing the small-sized clusters. These clusters were produced by fixing the vertex index to 2.75.

The nodes in the graph depict the co-cited references while the node size represents the number of items that were cited, i.e., the larger the node size is, the higher will be the number of that item cited. As a result, this node affects significantly the impact factor of the journal in which it is published. The large-sized circles of the nodes depict the burst reference that is ultimately reflecting the research hotspots in different intervals of time. The edges in the network are the co-citation links, whereas the different colors of the links are reflecting the first ever and the latest

citations. It is concluded from Figures 3–5 that the research in the field of test case selection in software engineering attracted a number of researchers from different academic and research institutes globally. Furthermore, it also depicts that the research in this field has not been performed without involving the other methodologies from the domains.

For example, the edges are observed between the cluster “test selection” and “effective test case selection” depicting that the domain of test selection can be improvised in an effective way using the algorithms of machine learning and artificial intelligence. Some smaller node values like “the regression test case,” “test case selection method,” and “adaptive random testing” are of key importance while searching the research articles of the same kind. Furthermore, these small clusters are most likely to be the hot topics in near future in the domain of software testing.

In Table 2, the complete information about the 6 large clusters depicting the studies in software testing in software engineering is given. The clusters are numbered in the clockwise direction as depicted in Figure 3. As mentioned earlier, the cluster size reflects the number of references that ultimately refers to its hotspot status, whereas the measurement of cluster similarity is computed by the mutual information (MI) that is shared among the clusters. The average number of publications is shown by the time span on year scaling, representing that the papers that were published recently will appear on the front, indicating the emerging trends in the cluster.

It is depicted from Table 2 that the clusters such as “test selection, 2011,” “data mining, 2011,” “quality-aware test case prioritization, 2017,” and “continuous integration, 2020,” are concluded as the key hotspots in the software testing related research using different approaches including artificial intelligence (AI) and machine learning (ML) in the field of applied software engineering. Furthermore, it is also observed that the recent trends that were exercised in the field of software

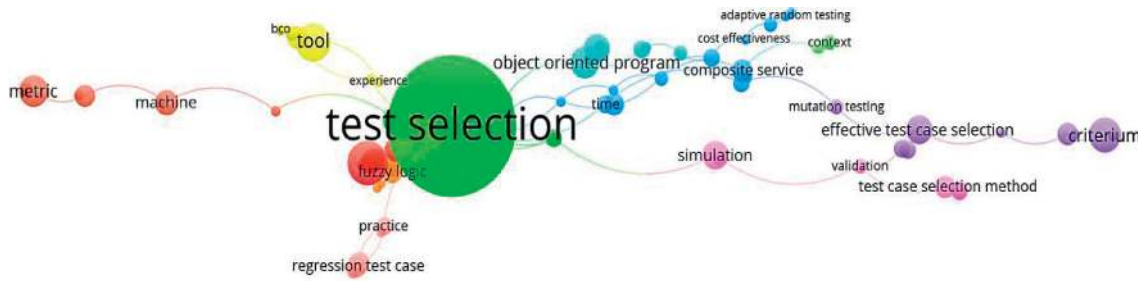


FIGURE 3: The reference network representing the references that were co-cited in the field of test case selection in applied software engineering.

TABLE 2: Basic terms and their explanation along with the respective purposes.

Cluster	Silhouette	Frequently occurred terms (FQTs)	Size	Mean (year)	Mutual information (MI)	Frequently occurred terms depicted as a likelihood ratio $p = 1.04^{-4}$
1	0.634	Regression testing, software testing, test case selection	67	2011	Effective testing, regression cycles	Genetic algorithm
2	0.748	Test suite, software testing	63	2012	Software development life cycle, software quality assurance	Swarm intelligence
3	0.897	Quality-aware test case prioritization	75	2017	Maximizing coverage	Weighted source code
4	0.715	Risk-based testing	54	2020	Software modification information	Method invocation relationship
5	0.530	Multiple testing criteria	68	2018	Cost-cognizant average percentage	Pareto-optimal clonal selection algorithm
6	0.472	Continuous integration	71	2020	Test case selection	Machine learning

testing are “multiple testing criteria, 2018,” “risk-based testing, 2020,” and “cost-cognizant, 2020.”

In last 3 years, the research community showed significant interest in adopting the machine learning concepts like k -mean clustering, regression analysis, and decision trees in predicting dynamically the type of research methodology used in the retrieved documents. This type of research is found in “continuous integration, 2020,” article.

The variance in the trends of the changing patterns of the research associated with software testing is observed throughout the years proving that the structural changes emerged as a continuous trend in applied software engineering with the emergence of artificial intelligence algorithms.

3.1. Burst References as an Indicator. The increasing number of citations of a specific research object represents its dynamic properties that eventually reflect the reputation of the journal in which it is published. This characteristic property is also referred to as “burst references,” as mentioned in Table 1. It helps the research community to diverge their attention towards the concerned research articles based on their respective burst reference graph.

In order to have close tabs on the recent and important research areas specific to some domain, we make use of tracing process to track the timely progress trends in burst references. This activity assists in finding the duration of the burst intensities in a specific time duration to extract the prominent features that play a key role in assigning the ranks to the research objects. In Figure 6, we depict the structural and operational modifications that appeared constantly with the intervention of AI-assisted algorithms in software testing process. It is also observed that the AI-related algorithms and framework played a key role in generating effective results in software regression testing since 2013. It is concluded from the literature that the most of the researchers showed interest in using modern algorithms of machine learning (ML) in 2013 and 2014 while dealing with high dimensional data since the conventional approaches failed to extract the key features from the huge data. The experiments showed that ML- and AI-based models outperformed the conventional approaches while performing software testing at the unit level. The top 8 research objects with significant burst reference values are shown in Figure 6, which depicts the results in some self-explanatory way in capturing the research hotspots and evolutionary trends in the recent studies of AI applications in software testing.

It is evident from Figure 6 that the quality of these research objects solely depends on the data mining and machine learning techniques of AI in applied software engineering that attracted a number of researchers since 2014. Among these research articles, last two articles were ranked on top in late 2019 and early 2020. The authors of these articles also recommend use of Weka and Python libraries to perform a comprehensive and disciplined series of experiments using machine learning procedures to pre-process and then classify the test case on the basis of the criteria set by the practitioners. These tools are also capable of rapid application development and comparing the results with state-of-the-art approaches working on new datasets.

4. Role of Co-Occurrence Keyword Network and Burst Keywords

It is a common observation that the two keywords are considered as potentially important and relevant if they appear in some document concurrently in the same order. A network that represents the co-occurrence keywords or potentially important indicating terms is capable of showing the emergent and importance of the important and key terms in a research area. Moreover, this network also assists in indicating the hotspots and the recent trends in a specific domain [18, 19]. Figure 7 depicts the time zone mapping of the co-occurrence network for the recent noteworthy work in software testing.

In Figure 7, the co-occurrence network depicts 271 key terms represented by the nodes with more than 1800 links showing powerfully build connections among the keywords that were mostly used in the articles published between 2007 and 2020. The size of the nodes indicates how frequent the term appeared while the edges are the links of co-occurrence, variation in the line colors depicts the pioneer connection, and thickness of the link corresponds to the co-occurrence strength. Figure 7 also indicates that hotspots in the recent studies including AI in the domain of software engineering in the time interval of 2009–2020 involve the following terms: “test-suite selection,” “test-case selection,” “test-case ordering,” “software testing,” “test case prioritization,” “test-case reduction,” “*k*-means clustering,” “machine learning models,” “unit-testing,” “regression-testing,” “high-dimensional data,” “fault detection,” “code coverage,” “execution cost,” “pattern matching,” and “mutation testing.”

4.1. Overlay Network in Dual Mapping as Indicator. Two other important mappings are also drawn in overlay mode in CiteSpace V that bridges the gap between the maps generated by the program (aka as overlay) with the map depicting the basic knowledge known as base layer. The base layer is associated with the information of some publishing discipline listed in the Journal Citation Reports (JCR). It helps in detecting the changes in the journal metrics time to time in a very disciplined and comprehensive way.

In Figure 8, a base mapping network is shown that is generated using the information provided by the publication(s) in a specific journal having higher number of

citations. CiteSpace is capable enough to support visualization of these mappings via the same interface. The number of articles published in a specific journal is mentioned in front of each journal name. The edges among the journals and the academic discipline with different colors show the information about cite-in and cite-out occurrence. In summary, Figure 8 signifies the key disciplines and significant journals among the researchers that are working in the domain of software testing using a number of approaches including ML and AI.

4.2. Most Trending Disciplines and Key Journals. Figure 8 shows that the journal “IEEE Transactions on Software Engineering” making a red cluster (in the middle of the network) is one of the significant journals that covered the most trending disciplines in the field of software testing. In addition, the following studies and science disciplines also played a key role in the related studies: “web services, computer networks, atomic rules, and web-based software.” In summary, it is concluded that AI-based algorithms play a key role along with other approaches while performing the software testing at the unit level. Furthermore, these algorithms provide an in-depth picture of the software testing using the statistical and applied mathematical models that also helped in increasing citation of the journal (or the article).

4.3. Country Implications. In Figure 9, a country collaboration network is drawn showing the collaborating countries between 2007 and 2020 in software testing. It is obvious that only three countries around the world are contributing more than 90% from rest of the world. Among these, USA is at the top with 764 publications so far. It is one of the interesting facts that a number of other regions and countries contribute in a relatively very low percentile.

4.4. Author Contribution. One of the key metrics in designing a comprehensive systematic review is exploiting the co-authorship information among the retrieved literature related to some specific domain. The same type of co-authorship network in this field is depicted in Figure 10 that is generated by using VOSviewer. It is observed that the minimum number of publications out of a total 12,405 authors, only 695 publications satisfied the abovementioned criteria.

The analysis at cluster level of the co-author map detects that there are 12 clusters with variant colors. The most important cluster, “Chen Tsong Yueh,” is shown in sea-green on the left, showing 54 published articles with 267 citations and link strength of 109. The second most important cluster “Cai Kai-Yuan” is shown in light green with 54 articles with 124 citations and link strength of 37. The authors with higher degrees of centrality are more central in the network structure and tend to have a greater capacity to influence others. The co-authorship analysis indicates that most of the authors of papers on software testing are densely connected with each other in terms of collaboration and citations.

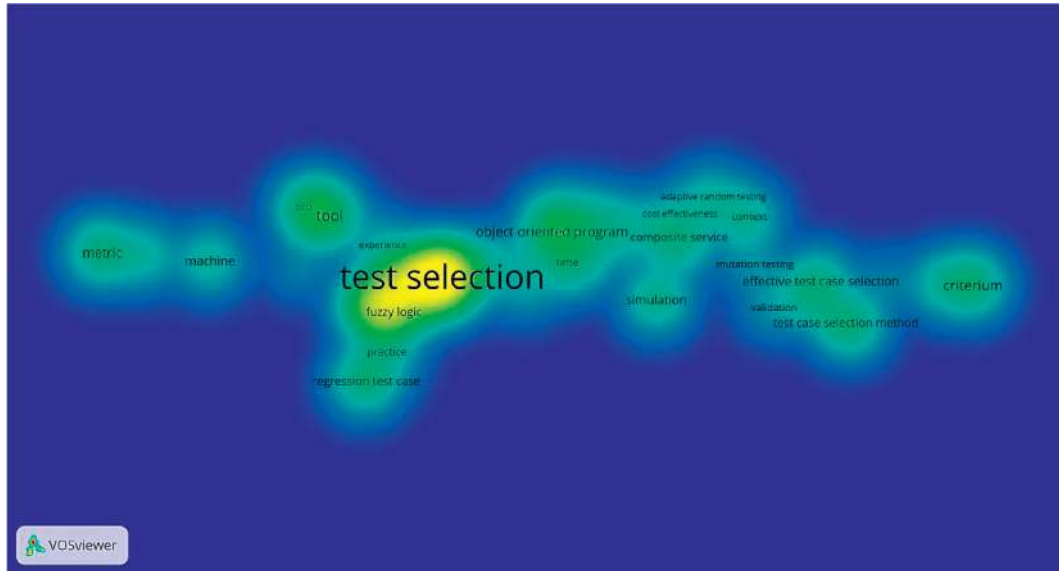


FIGURE 4: The equivalent density visualization graph representing that the most frequently appeared term is the “test selection.”

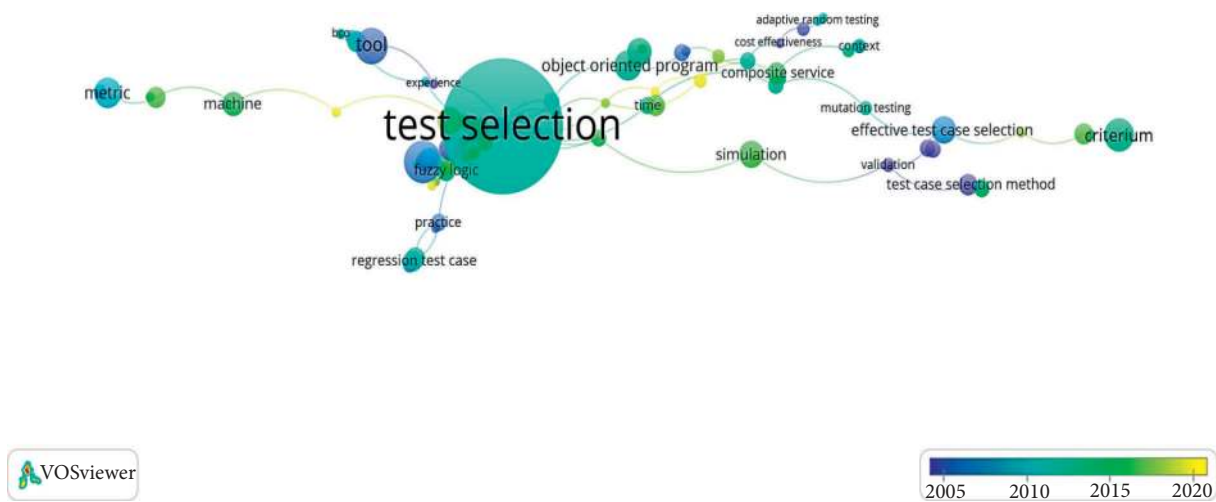


FIGURE 5: Overlay analysis of the published articles related with the test case selection, depicting the spatial data with the attribute data.

Top 8 References with the Strongest Citation Bursts

References	Year	Strength	Begin	End	2016-2019
Yoo S, 2012, Software Testing Verification and Reliability, V22, P67, DOI 10.1002/stv.430, DOI	2012	18.86	2016	2017	
Mirarab S, 2012, IEEE Transactions on Software Engineering, V38, P936, DOI 10.1109/tse.2011.56, DOI	2012	2.86	2016	2017	
Harman M, 2012, ACM Computing Surveys, V45, P11, DOI 10.1145/2379776.2379787, DOI	2012	2.2	2016	2017	
Catal C, 2012, Software Quality Journal, V21, P445, DOI 10.1007/s11219-012-9181-z, DOI	2012	2.2	2016	2017	
Johansen MF, 2012, Proceedings n - SPLC 12-Volume 1, V0,P46, DOI	2012	2.2	2016	2017	
Marijan D, 2013, 2013 IEEE Int Software Maintenance, V0, P540, DOI	2013	2.98	2017	2019	
Gotlieb A, 2014, Proceedings o nalysis-ISSTA 2014, VO, P171, DOI	2014	1.73	2017	2019	
Hao D, 2014, ACM Transaction on Software Engineering and Methodology, V24, P1, DOI 10.1145/2685614, DOI	2014	1.63	2017	2019	

FIGURE 6: Top 8 research objects having the significant burst values. The red line depicts the total burst time while the dark blue line represents the publication time.

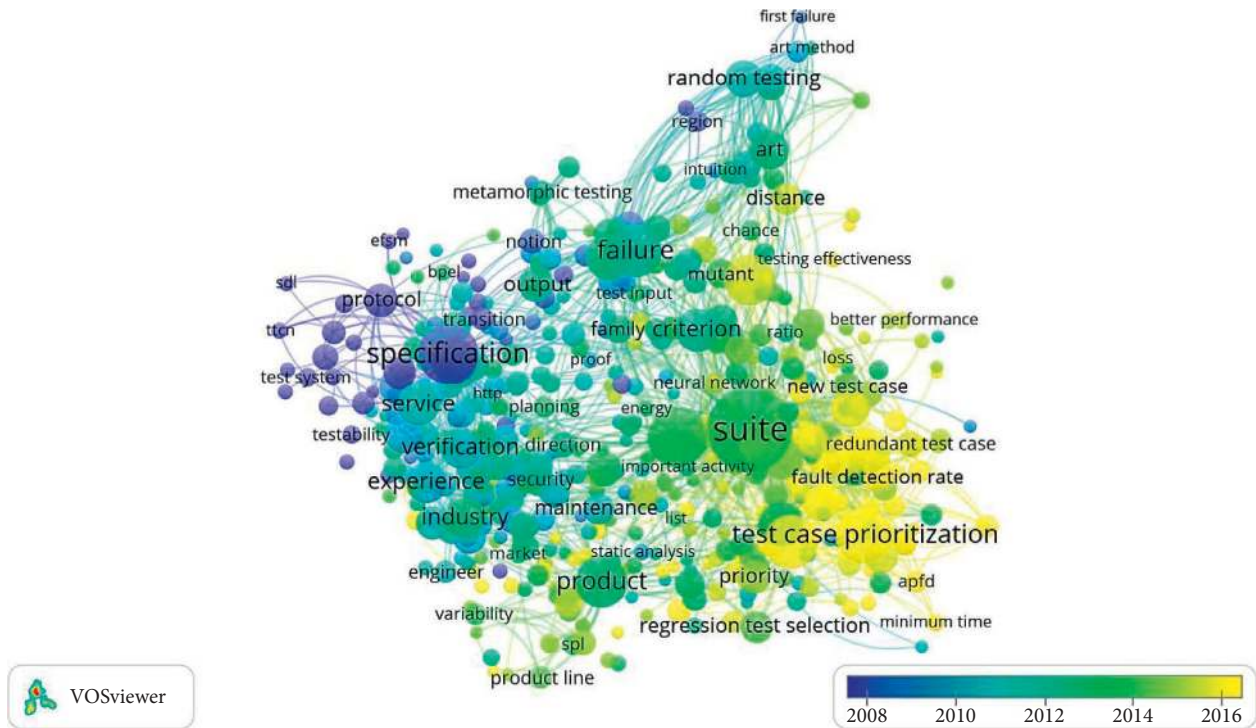


FIGURE 7: Mapping of co-occurrences of keywords used in the studies of software testing. The nodes in the mapping depict the keywords.

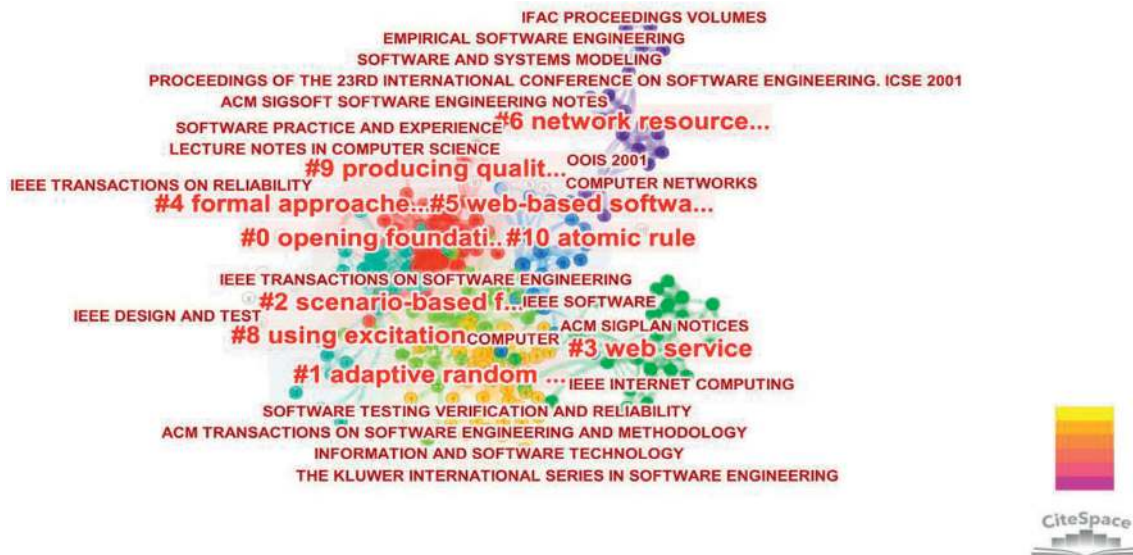


FIGURE 8: A dual-map overlay network showing both the citing journals and the cited journals in the domain of software regression testing.

The application of both the classical and AI-based approaches in software regression testing attracted a number of research communities globally; specifically, the institutes from the United States, China, and India emerged as the growing and prominent institutes compared to other regions. Figure 9 depicts that USA is prominent among other regions and is densely connected to other countries in the research on regression testing in applied software engineering with the largest number of

publications with 654 articles. In continental terms, the Asian research institutes dominate the research in software regression testing. Similarly, the figure shows that “IEEE Transactions on Software Engineering” is among one of the top research journals with a total of 649 published articles related to regression testing.

The retrieved data from the WoS platform depicts that the number of published and cited papers on regression testing grew rapidly in recent years. As mentioned earlier,

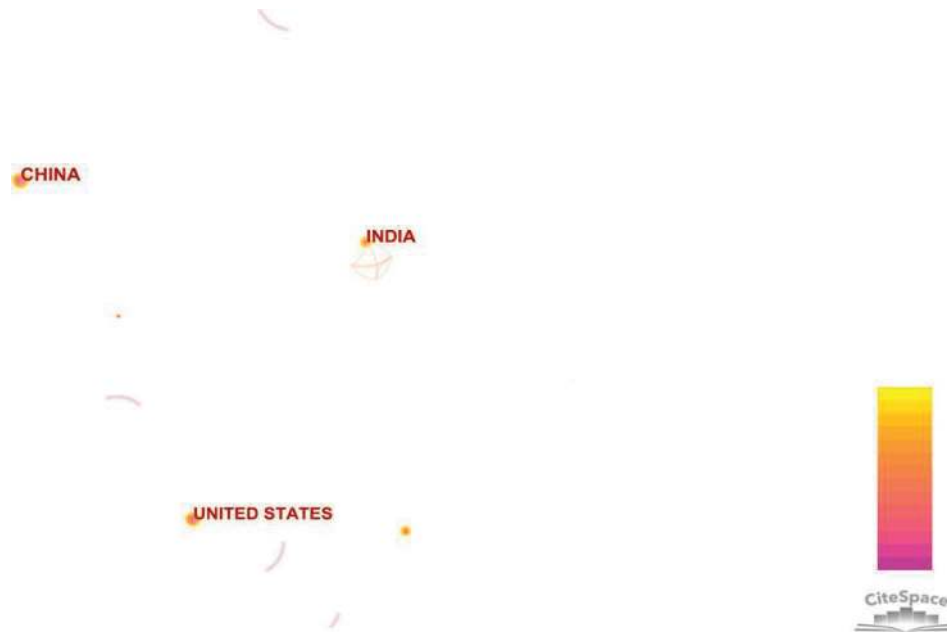


FIGURE 9: A knowledge panorama of studies on software testing using different approaches.

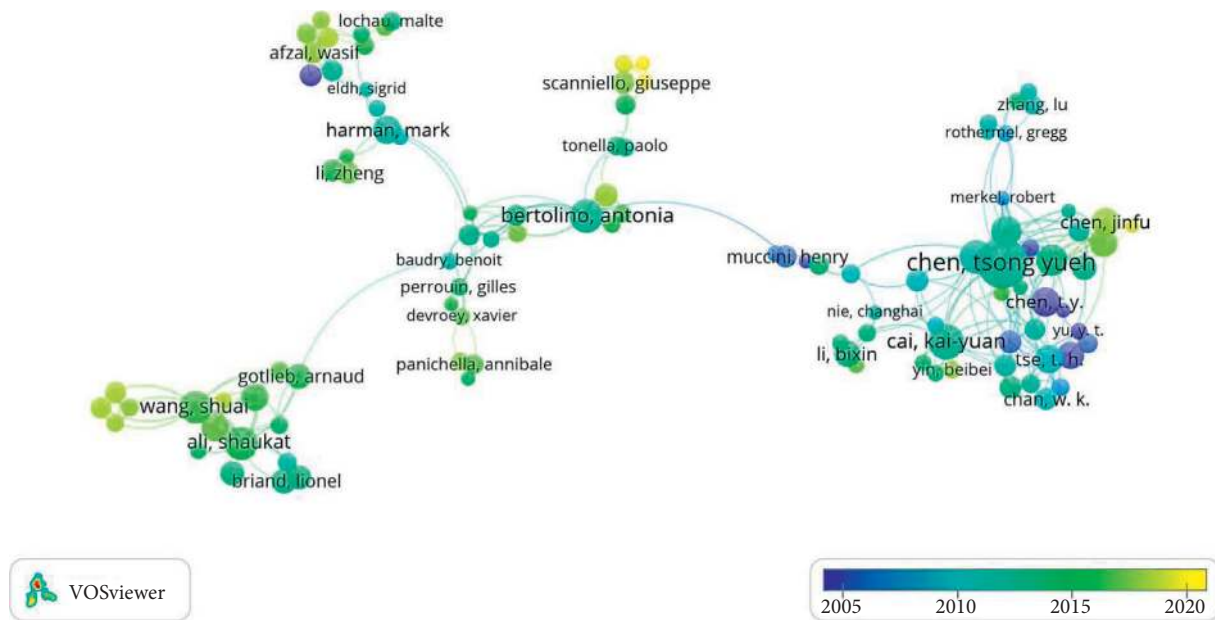


FIGURE 10: The co-authorship network.

USA-based institutes are heavily connected in terms of both journals and collaboration. The journal Information and Software Technology is ranked second, with 604 articles, and it is distributed globally including 17 countries including France and the Netherlands. Regarding the authorship, Figure 10 shows that Chen Tsong Yueh is one of the authors who is highly active in collaborative research activities.

It is noteworthy that according to the Journal Citation Report (JCR) published in 2020 that the authors of the articles discussed in our paper produced the research articles with a ratio of 0.56 in 2017 to 2019, while the productivity rates of the research articles from the same authors were

0.43, 0.35, and 0.28, respectively, in 2007–2008, 2001–2011, and 2012–2013, respectively.

4.5. *Revolutionary Changes in Regression Testing with the Intervention of AI-Based Approaches.* The following indicative points show the changes in the basic structure that emerged with the advent of AI-based approaches in regression testing:

- (1) According to the indicating reference provided by the co-cited reference network in Table 1, the initial research in regression testing mainly focused on the

basic metrics including the code coverage and execution time of the individual test case that solely depends on the data provided by the previously executed processes. In recent years, amid the impacts of machine learning and deep network, the research in regression testing gave rise to a next new level representing the new aspect of the testing process by considering the training data. The thematic patterns in the scientific literature differ over time indicating the structural changes in the field since the AI-based algorithms behave differently than the conventional approaches.

- (2) In Figure 7, the time zone map of the co-occurrence keyword network indicates that from 2007 to 2010, most of the literature of both the conventional and AI-based approaches in software regression testing was reflected in the following keywords: “system,” “regression testing,” “unit testing,” “model checking,” “machine learning,” “classification,” and “*k*-means clustering.”
- (3) The timeline map of the co-occurrence keyword network (see Figure 7) shows the structural changes in the field of regression testing in software engineering. It is clear that the “test-case prioritization” was researched earliest and “specification” was the starting point. Moreover, the “failure” hotspot “metamorphic testing” promoted the emergence of “artificial neural networks.” The divergent directions of the curves of “computational cost” and “genetic algorithms” show that it gave birth to today’s research on intelligent regression testing.

5. Conclusions and Discussion

In this paper, the noteworthy contribution in the field of software testing is visualized by considering different aspects of the research like co-cited reference network analysis, co-occurrence keyword network analysis, burst reference analysis, and so on. Furthermore, the recent hotspots and evolving developments in regression testing using conventional and modern AI approaches are recognized in a very disciplined and comprehensive way. Moreover, the involvement of the global research community is also highlighted.

In the literature reviewed in this article, it is observed that most of the selection methods focus their tabs on the application of some specific software domain that limits our effort in retrieving any significant evidence to analyze and assess the superiority and dominance of one method over the other. Some relevant topics for future works are (1) to evaluate the selection methods and metrics surveyed in different contexts to prove their effectiveness, efficiency, applicability, and scalability; (2) to develop selection methodologies that can be extended to different software domains; and (3) to develop frameworks or tools to support the test case selection. Furthermore, while analyzing the empirical results generated in the experiment reported in the

literature, it is concluded that most of the work was performed in software developed by the researchers and classified by them as small. Analysis and validation of results in test case selection experiments are still limited. Despite the availability of important sources, such as SIR and the Free Software Foundation, it is necessary to expand the experiments in terms of size, application (e.g., industrial use), and software complexity, as replicating studies using different test suites may reveal different patterns of effectiveness and efficiency as well as help prove the viability and applicability of the proposed selection methods.

6. Limitations and Future Work

Although the WoS core collection was chosen as the study’s data source, we may have missed some important research publications on AI and conventional approaches in the software regression analysis and testing. To ensure high data quality, this study selected only articles from the SCIE database that may also have led us to omit some important research results (e.g., books, Ph.D. theses, and SSCI database).

Furthermore, the “Top 100 per slice” was set as the standard for data extraction using CiteSpace V that may have some different effect on the analyses when compared to some other slice value. In the future, we will perform a detailed analysis of studies on some genetic and deep learning applications in validation and verification of the developed software before launching or handing over to the clients. Furthermore, we also plan to use a latent Dirichlet allocation (LDA) model for text clustering in the future. We plan to extend the analysis by running correlations between number of citations with respect to the number of primary studies and quality measures of secondary studies. We may even be able to build a regression model to predict citations to secondary studies.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We acknowledge the support of Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia.

References

- [1] J. Anderson, S. Salem, and H. Do, “Improving the effectiveness of test suite through mining historical data,” in *Proceedings of the 11th Working Conference on Mining Software Repositories*, Hyderabad, India, May 2014.
- [2] A. Celik, M. Vasic, A. Milicevic, and M. Gligoric, “Regression test selection across jvm boundaries,” in *Proceedings of the*

- 2017 11th Joint Meeting on Foundations of Software Engineering, Paderborn, Germany, September 2017.
- [3] H. Cibulski and A. Yehudai, "Regression test selection techniques for test-driven development," in *Proceedings of the 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, IEEE, Berlin, Germany, March 2011.
- [4] H. Do, S. Elbaum, and G. Rothermel, "Supporting controlled experimentation with testing techniques: an infrastructure and its potential impact," *Empirical Software Engineering*, vol. 10, no. 4, pp. 405–435, 2005.
- [5] R. Kazmi, D. N. A. Jawawi, R. Mohamad, and I. Ghani, "Effective regression test case selection," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–32, 2017.
- [6] N. Mansour, H. Takkoush, and A. Nehme, "Uml-based regression testing for oo software," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 23, no. 1, pp. 51–68, 2011.
- [7] B. Li, D. Qiu, H. Leung, and D. Wang, "Automatic test case selection for regression testing of composite service based on extensible bpel flow graph," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1300–1324, 2012.
- [8] G. Xu and A. Rountev, "Regression test selection for aspectj software," in *Proceedings of 29th International Conference on Software Engineering (ICSE'07)*, IEEE, Minneapolis, MN, USA, May 2007.
- [9] J. Zheng, L. Williams, B. Robinson, and K. Smiley, "Regression test selection for black-box dynamic link library components," in *Second International Workshop on Incorporating COTS Software into Software Systems: Tools and Techniques (IWICSS'07)*, IEEE, Minneapolis, MN, USA, May 2007.
- [10] N. N. Abbas, T. Ahmed, S. H. U. Shah, M. Omar, and H. W. Park, "Investigating the applications of artificial intelligence in cyber security," *Scientometrics*, vol. 121, no. 2, pp. 1189–1211, 2019.
- [11] N. J. Van Eck and L. Waltman, "Software survey: vosviewer, a computer program for bibliometric mapping," *Scientometrics*, vol. 84, no. 2, pp. 523–538, 2010.
- [12] C. Chen, "Citespace ii: detecting and visualizing emerging trends and transient patterns in scientific literature," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 359–377, 2006.
- [13] H. J. Kim, Y. K. Jeong, and M. Song, "Content- and proximity-based author co-citation analysis using citation sentences," *Journal of Informetrics*, vol. 10, no. 4, pp. 954–966, 2016.
- [14] H. Small and E. Greenlee, "Citation context analysis of a co-citation cluster: recombinant-dna," *Scientometrics*, vol. 2, no. 4, pp. 277–301, 1980.
- [15] Y. Jin and X. Li, "Visualizing the hotspots and emerging trends of multimedia big data through scientometrics," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1289–1313, 2019.
- [16] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.
- [17] S. Yazdani and M. Montazeri-Gh, "A novel gas turbine fault detection and identification strategy based on hybrid dimensionality reduction and uncertain rule-based fuzzy logic," *Computers in Industry*, vol. 115, Article ID 103131, 2020.
- [18] S. Li and Y. Sun, "The application of weighted co-occurring keywords time gram in academic research temporal sequence discovery," *Proceedings of the American Society for Information Science and Technology*, vol. 50, no. 1, pp. 1–10, 2013.
- [19] H.-N. Su and P.-C. Lee, "Mapping knowledge structure by keyword co-occurrence: a first look at journal papers in technology foresight," *Scientometrics*, vol. 85, no. 1, pp. 65–79, 2010.