applied sciences

MDPI

Review

Systematic Literature Review on Test Case Selection and Prioritization: A Tertiary Study

Shweta Singhal [1], Nishtha Jatana [2], Bharti Suri [3], Sanjay Misra [4,*] and Luis Fernandez-Sanz [5]

[1] Department of Computer Science and Information Technology, Indira Gandhi Delhi Technological University for Women, New Delhi 110006, India; miss.shweta.singhal@gmail.com

[2] Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, New Delhi 110058, India; nishtha.jatana@gmail.com

[3] University School of Information & Communication Technology, Guru Gobind Singh Indraprastha University, New Delhi 110078, India; bhartisuri@gmail.com

[4] Department of Computer Science and Communication, Ostfold University College, 1759 Halden, Norway

[5] Department of Computer Science, University of Alcala, 28801, 28806 Madrid, Spain; luis.fernandez.sanz@uah.es

* Correspondence: ssopam@gmail.com

Abstract: Software testing is undertaken to ensure that the software meets the expected requirements. The intention is to find bugs, errors, or defects in the developed software so that they can be fixed before deployment. Testing of the software is needed even after it is deployed. Regression testing is an inevitable part of software development, and must be accomplished in the maintenance phase of software development to ensure software reliability. The existing literature presents a large amount of relevant knowledge about the types of techniques and approaches used in regression test case selection and prioritization (TCS&P), comparisons of techniques used in TCS&P, and the data used. Numerous secondary studies (surveys or reviews) have been conducted in the area of TCS&P. This study aimed to provide a comprehensive examination of the analysis of the enhancements in TCS&P using a thorough systematic literature review (SLR) of the existing secondary studies. This SLR provides: (1) a collection of all the valuable secondary studies (and their qualitative analysis); (2) a thorough analysis of the publications and the trends of the secondary studies; (3) a classification of the various approaches used in the secondary studies; (4) insight into the specializations and range of years covered in the secondary texts; (5) a comprehensive list of statistical tests and tools used in the area; (6) insight into the quality of the secondary studies based on the seven selected Research Paper Quality parameters; (7) the common problems and challenges encountered by researchers; (8) common gaps and limitations of the studies; and (9) the probable prospects for research in the field of TCS&P.

Keywords: systematic literature review (SLR); software testing and reliability; secondary studies; test case selection; test case prioritization; software testing; surveys

check for updates

Citation: Singhal, S.; Jatana, N.; Suri, B.; Misra, S.; Fernandez-Sanz, L. Systematic Literature Review on Test Case Selection and Prioritization: A Tertiary Study. *Appl. Sci.* **2021**, *11*, 12121. https://doi.org/10.3390/app112412121

Academic Editor: Agostino Foestiero

Received: 21 October 2021
Accepted: 4 December 2021
Published: 20 December 2021

1. Introduction

The key to today's technological advance is programming and the continuous delivery of software. When software is released to users, it faces new challenges of meeting unforeseen user requirements, unexpected inputs, ever-growing market competition, and new and changing user demands. These issues all have to be handled after the software development phase is over and the software is in its maintenance phase. Then, the software either becomes obsolete, new software has to be developed (which is very costly in terms of computational time and resources needed), or the existing software has to be updated. Updating software has a clear budget and time constraints, and pressure to fulfill the desired modification goals. Hence, the updating process involves retesting the modified software to maintain its correctness and accuracy. This part of software testing is known as regression testing. Officially, "regression testing is performed between two different versions of software in order to provide confidence that the newly introduced features of the Software Under Test (SUT) do not interfere with the existing features" [1]. Software failures as early as the Ariane-5 rocket launch and the Y2K problem, and as recent as the Feb 2020 Heathrow airport disruption [2], have been blamed on the failure to test the changes in software systems. Thus, the search for adequate regression testing techniques is still crucial. Many methods have been developed for efficient regression testing. Regression testing approaches can be distinguished between test suite minimization, selection, prioritization, or the hybrid approach [2]. The research work in this paper focuses on the regression test selection and prioritization techniques. Test case selection (TCS) cases are selected according to their relevance to the changed parts of the SUT, which characteristically involves a static white-box analysis of the program code. More formally, the selection problem as defined by Rothermel and Harrold [3] means finding a subset of test cases from the original test suite that fulfills the selection criteria to test the modified version of the SUT.

Moreover, test case prioritization (TCP) helps in ordering test cases for the early maximization of some desirable properties, such as the rate of fault detection. TCP techniques try to reorder a regression test suite based on decreasing priority. Test cases' priority is established using some pre-defined testing criteria, such as fault detection, maximum coverage, and reduced execution time. The test cases with higher priority are executed before those with lower priority in regression testing [4,5].

Many research approaches exist for conducting research. Evidence-Based Software Engineering (EBSE) means gathering and assessing empirical evidence grounded on primary studies' research questions. This enables healthier decisions associated with software engineering and development. The concept of EBSE was presented in 2004 by Kitchenham et al. [6]. The summary of results obtained empirically is mostly achieved via systematic literature reviews (SLRs) [7] or systematic mapping (SM) [8] for EBSE. As described by Kitchenham et al. [7], "SLRs are secondary studies (i.e., studies that are based on analyzing previous research) used to find, critically evaluate and aggregate all relevant research papers (referred to as primary studies) on a specific research question or research topic." The procedure is designed to ensure that the evaluation is neutral, rigorous, and auditable. The elementary methodology of conducting an SLR is always similar, regardless of the discipline being reviewed. There are many recent SLRs in software testing and reliability [9–13]. Reviews of secondary studies have also been conducted in the area of software testing [14]. A landscape presentation of 210 mapping studies in software engineering was also recently published in 2019 [15]. A mapping study and an SLR vary because mapping studies recognize all research on a particular topic, whereas SLRs address specific research questions [7]. Mapping studies also categorize the primary studies with a few properties and classifications. Kitchenham et al. [7] differentiated mapping studies from SLRs based on the following:

- **Research question:** The research question (RQ) of an SLR is specific and related to empirical studies' outcomes. The RQ of a SM is general and relevant to research trends.
- **Search process:** Whereas the search process of an SLR is defined by the research question, the search process of SM is defined by the research topic.
- **Search strategy requirements:** Whereas the search strategy requirements of an SLR are incredibly stringent, a mapping study's search strategy requirements are less stringent. All previous studies must be found for SLRs.
- **Quality evaluation:** Quality evaluation is crucial for SLRs, but is not essential for mapping studies.
- **Results:** Results of SLRs are answers to specific research questions. Results of SM are categories of papers related to a research topic.

When searching for research texts on online digital libraries, it is possible to miss out on essential texts that might not match the 'searched string'. One possible way to search such texts is snowballing. "Snowballing refers to using the reference list of a paper or the citations to the paper to identify additional papers. However, snowballing can benefit from looking at the reference lists and citations and complementing it with a systematic way of looking at where papers are referenced and cited. Using the references and the citations respectively are referred to as backward and forward snowballing" [16]. When performing surveys or reviews, this is a vital search procedure to avoid missing essential texts. Wohlin [16] explained the detailed procedure and guidelines for performing snowballing in surveying the literature for particular topics. Many researchers are taking advantage of this research strategy to conduct surveys. In addition to the SM techniques, snowballing was also used in searching the relevant studies in this paper.

TCS&P has attracted substantial research attention [17]. The increasing industrial demand for TCS&P techniques is a major driving force behind the research [18]. The advances in software technology and the rising complexity of systems have forced the software to be modified and retested. The amount of available time and cost for this retesting is usually minimal. Thus, the research for quicker, more accurate, and newer techniques to fulfill the changing technological demands is inevitable. It is essential to develop a new technique to justify its need: Doe such a method already exist? If it does exist, what are the possible shortcomings? Can the technique be updated to satisfy the requirements? Secondary studies are performed on existing literature to obtain a better answer to such questions. As the area of TCS&P has also generated numerous secondary studies, a tertiary study is therefore needed. This can provide a higher-level catalog of the research conducted in the area. This tertiary study is conducted by following the procedures used to perform a SM [8]. We used the guidelines provided by [8] because it provides detailed guidelines for carrying out a SM in the area of software engineering and is a highly cited paper (having more than 3000 citations). This study was also inspired from and supported by another tertiary study in the area of software testing [14]. We detected over 50 secondary studies in the field of TCS&P. Based on the inclusion and exclusion criteria discussed in later sections, only 22 secondary texts were selected for our survey; they were published from 2010 to 2020.

Contributions of this survey include:

1. To present all the valuable secondary studies available online in the field of TCS&P in one place for quick referral.
2. To analyze the basic publication information of these texts in the form of publishing journals, years, and online databases to be looked for.
3. To classify the secondary studies based on the review approach used as SLRs, mappings, or survey/reviews.
4. To analyze the breadth of search in the 22 studies.
5. To provide a quick reference to the statistical tests and tools being used in TCS&P, with a reference to the study in which they have been used.
6. To provide a detailed assessment of the quality of secondary research being conducted in TCS&P based on seven chosen Research Paper Quality parameters.
7. To support the research community with a consolidation of advances in TCS&P, describing possible shortcomings in the existing texts and the probable future scope and likely publishing targets in the area.

The paper is structured as follows. Section 2 presents the adopted research methodology. Data extraction and summarization are presented in Section 3. The results and discussion and the conclusions drawn from this survey are provided in Sections 4 and 5, respectively.

2. Research Methodology

The research process was inspired by the guidelines [8] for performing SM in software engineering. The process begins with the research questions to be answered through an exhaustive literature review. The research questions then help in defining the Research String used to search the relevant texts in the different online databases. After relevant texts are identified, they are examined according to the exclusion criteria at multiple levels. Text-based exclusions take into account the inclusion and exclusion criterion pre-determined for the research. Based on these, the final texts are selected to be considered for review. The following subsections explain the details of the implementation of these steps.

2.1. Research Questions

The research questions were described and categorized into five parts. The categories then were further divided to provide the associated research questions. These questions form the basis of the current survey analysis of the studies. The details of the development of research questions are as follows:

RQ 1. To get basic information of available secondary texts in TCS&P.

RQ 1.1: *What is the distribution of secondary texts on TCS&P over the different online databases?*

This RQ will enhance the knowledge of online databases in which secondary studies can be quickly found.

RQ 1.2: *What is the evolution of the number of secondary studies published in TCS&P over the years?*

The expansion/compaction of the research field on TCS&P can be acknowledged by increasing/decreasing secondary studies in the area. New secondary studies are performed only when the existing ones have become obsolete or show missing aspects in their analysis. In both cases, researchers work in secondary research areas for which high demand is maintained in the research community. The analysis of publication numbers over time reveals the expansion of TCS&P secondary studies.

RQ 1.3: *Which are the key publishing journals for TCS&P secondary studies?*

Secondary texts are generally larger than the primary studies and involve a different research approach from those who propose new ideas. Therefore, not all conferences and journals support the publication of secondary studies. The answer to this RQ will provide possible publication targets for researchers working on secondary texts in the future.

RQ 2. To study the characteristics of secondary studies.

RQ 2.1: *Which research approaches have been used in secondary studies?*

Several researchers have applied multiple techniques for secondary reviews on the existing texts. Different guidelines for different procedures have been presented [6,19]. We categorized the research approaches for performing secondary studies into three types: (1) systematic literature reviews, (2) mappings, and (3) reviews/surveys. Analyzing which texts use which approach and how these are distributed over time will help answer the RQ.

RQ 2.2: *What are the focus and the range of years covered in the secondary studies?*

The analysis of the range of years of publication of the selected primary studies is an indicator of the length of coverage of secondary studies. It provides an insight into the variety of primary studies considered in the analysis. Surveys performed over similar ranges should have included similar texts that are related to the surveyed topic. It also helps future researchers search for primary studies that lie outside the already searched ranges.

RQ 2.3: *How many texts form the basis of research for the selected secondary studies?*

The extent of studies can also be judged depending on the number of chosen primary studies for conducting the survey. The number of primary studies chosen can vary widely for similar survey topics depending on the selected research questions. However, the number of mappings should be comparable for surveys on similar topics conducted for primary studies chosen from the same range of years.

RQ 3. How have the secondary studies benefitted from the existing tools and tests?

RQ 3.1: *Have statistical tests been used in the secondary studies?*

Comparisons and analysis of data in a secondary study may have used specific statistical tests.

RQ 3.2: *Which tools have been used by the secondary studies?* Our study analyzes the use of tools available for research purposes.

RQ 4. How can the quality of secondary studies be compared?

RQ 4.1: *How have the research questions been used in the secondary studies?*

Research questions form the real basis of any survey, review, or mapping to be conducted. The analysis of the number and quality of research questions and whether they have been answered is part of our analysis. This will help in determining the quality of conclusions of the secondary studies.

RQ 4.2: *What is the quality of the secondary studies performed in the area?*

Certain parameters can act as quality criteria to compare secondary studies. These will also help future researchers use them as quality indicators in their future survey, review, and/or mapping.

RQ 5. What are the probable prospects in TCS&P?

It is possible to analyze information and characteristics of the secondary studies and propose probable future directions to researchers in the field of TCS&P.

RQ 5.1: *What are the TCS&P techniques as mentioned in the secondary studies?*

The field of TCS&P embraces many types of techniques. The studies we considered provide various categorizations. Thus, it is useful to tabulate the categories of techniques available in the area.

RQ 5.2: *What is the extent of evolutionary techniques in secondary studies?*

Evolutionary techniques have recently experienced a significant increase in use. These techniques are under continuous development and are used in all research fields concerning optimization problems. Thus, it is highly relevant to analyze coverage of evolutionary techniques in the TCS&P field. This will help researchers to discern if new or additional techniques can be applied in the area.

2.2. Defining the Search String

Given the research questions modeled in the previous section, our study's main topics of interest included surveys on selection and prioritization in the field of regression testing. Therefore, the search string included several spellings and synonyms valid for these topics. Search conditions were merged with the logical operators 'OR' and 'AND'. The SLR conducted by Kitchenham [7] on other existing SLRs inspired the formulated search string. Different online databases require the articulation of different search strings, depending on the advanced search options they provide. Table 1 (next section) summarizes the details of the search string and the databases. Streamlining the search results required advanced searching options for finding the terms in the abstract of the studies, given the large pool of available literature.

Moreover, we considered only English texts published in journals, conference proceedings, workshops, or book chapters. The range of search for our survey was from 1997 until May 2020. The beginning year was chosen as 1997 because the previous seminal studies [20,21] indicate that the research in the area of TCS&P began in this year. However, the first chosen secondary study in the field was published in 2008.

Table 1. Details of searching online databases.

Database	Search String	Filters Applied	No. of Studies Found
IEEE Explore [22]	((“Abstract”: Prioritization OR “Abstract”: selection OR “Abstract”: prioritisation) AND (“Abstract”: “test” OR “Abstract”: “testing”)) AND (“Abstract”: “survey” OR “Abstract”: “review” OR “Abstract”: “mapping”)	Conferences, Journals, 1997–2020	383
ACM Digital Library [23]	[[Abstract: prioritization] OR [Abstract: selection] OR [Abstract: prioritisation]] AND [Abstract: “test*”] AND [[Abstract: survey] OR [Abstract: review] OR [Abstract: mapping]]	1 January 1997 TO 30 June 2020	218

Table 1. *Cont.*

Database	Search String	Filters Applied	No. of Studies Found
Science Direct [24]	((prioritization OR selection) AND (test OR testing)) AND ((survey OR review) OR mapping)	Review article, research article, Book chapters Title, 1997–2020	9
Wiley Online Library [25]	"((prioritization + selection + prioritisation) & "test*") & ((survey + review) + mapping)"	In Abstract	25
Springer digital library [26]	'(("test" OR "testing") AND ((prioritisation OR prioritization) OR selection) AND ((survey OR review) OR mapping))'	English, Computer-Science, Article, 1997–2020	1072
Google Scholar [27]	((prioritization OR prioritisation OR selection) AND (test OR testing)) AND (survey OR review OR mapping)	1997–2020	2405

2.3. Conducting the Search

The search for secondary studies in TCS&P was accomplished in June 2020 following the PRISMA guidelines [28]. Table 1 summarizes the records published in online databases via the formulated search strings. It also shows the applied filters, the range of years of searched texts, and the number of studies resulting from the search.

In addition to these texts, additional literature in white papers, industry reports, and blogs are not present in these databases, because they are non-peer-reviewed texts. Such research is referred to as grey literature. Garousi et al. [29] propose a method for searching texts from grey literature. Due to the risks involved in studying grey literature, we restricted ourselves to technical reports and workshop papers. Kitchenham [7] used the same approach in her SLR on SLRs.

Additionally, we also considered the process of snowballing (see Section 1) by performing forward and backward snowballing of the citations and references, respectively, from selected studies found in online databases [16]. Thus, we identified studies that the search string may have missed. Twenty-four texts were initially identified due to snowballing and the grey literature search. Figure 1 shows the step-by-step procedure followed for conducting the search process based on the PRISMA ("Preferred Reporting Items for Systematic Reviews and Meta-Analyses") statement [28]. After the initial title-based search, we added abstract-based and text-based exclusions, resulting in 55 secondary studies in TCS&P. Then, we applied inclusion and exclusion criteria (Table 2), considering the research questions presented above. After all the steps, the final set included 22 texts for analysis.

2.4. Inclusion and Exclusion Criteria

Table 2 summarizes the five inclusion and seven exclusion criteria that were applied to the first set of 55 contributions leading to the final set of 22 selected secondary studies. The most critical aspect was the quality check of the possible relevant studies. The exclusion criteria EC5, EC6, and EC7 were the basis for the rejection of studies. Many texts collected primary studies but did not include a proper analysis; these were excluded. For example, [30] is a systematic review on test case prioritization; however, we could find no analysis at all in the study. Other such studies were [31–33]. We also found a small number of secondary studies with an incomplete set of surveyed texts, and these were also excluded from the current study [34–39]. A survey on many-objective optimization was presented in [40]; the study was sound and effective but found to be outside the scope for the current study. Catal presented 10 best practices in test case prioritization, but the study did not present any analysis other than APFDs [41].

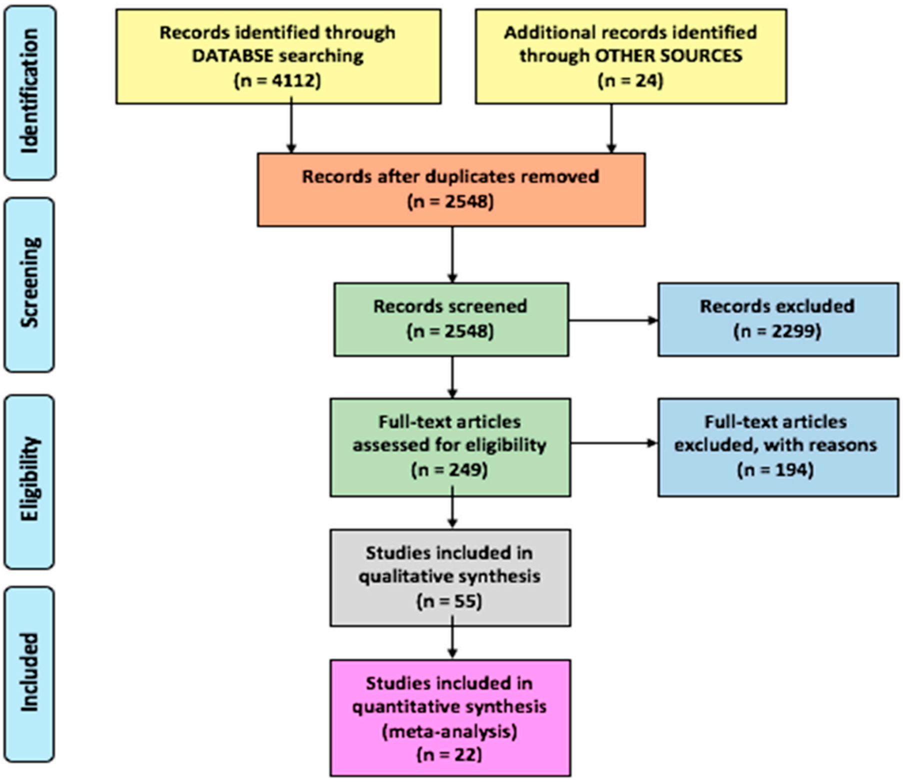
Identification
Screening
Eligibility
Included
Records identified through DATABSE searching (n = 4112)
Additional records identified through OTHER SOURCES (n = 24)
Records after duplicates removed (n = 2548)
Records screened (n = 2548)
Records excluded (n = 2299)
Full-text articles assessed for eligibility (n = 249)
Full-text articles excluded, with reasons (n = 194)
Studies included in qualitative synthesis (n = 55)
Studies included in quantitative synthesis (meta-analysis) (n = 22)

Figure 1. PRISMA flow diagram for the current search.

Table 2. Inclusion and exclusion criteria.

	Inclusion Criteria		Exclusion Criteria
IC1	Relevance to TCS&P.	EC1	Texts that are not directly relevant to TCS&P, including secondary texts in software testing not specifically mention TCS&P.
IC2	Secondary studies such as surveys, reviews, SLRs and mappings are included.	EC2	Primary texts proposing a new approach, or validating existing ones, or empirical comparison studies.
IC3	Texts published in journals, conferences, workshops, and technical reports are considered.	EC3	Texts found from magazine articles, theses, abstracts, posters, keynotes, doctoral symposiums, and non-peer-reviewed texts from blogs and comments.
IC4	Texts should be available in the English language.	EC4	Texts are not available in the English language.
IC5	Quality of the text based on evaluation of primary studies, breadth of survey and new addition to research field.	EC5	Texts that did not provide any kind of analysis for the selected primary studies.
		EC6	Texts missing many important primary texts published in the range of years reviewed.
		EC7	Texts representing mere repetition of available secondary studies already published and not adding any new knowledge to the research.

Although secondary texts that are not SLRs may not include all the relevant studies in the field, we rejected the studies that did not mention more than five relevant journal texts published in the surveyed range of years. The major reason for this is the inconsistency in the results with missing relevant information. In one of the cases, the authors presented the survey at a conference [42] and the detailed SLR was subsequently published in a journal [20]. In this case, considering EC7, specifically for the two surveys on TCS techniques by Engstrom et al. [20,42], the detailed text from the journal publication [20] was included in our selection.

3. Data Extraction and Summarization

All the 22 selected secondary studies were thoroughly examined to extract details according to RQs. Table 3 shows a summary of the selected texts. Additionally, we gathered information about the following evaluation aspects:

1. **Online Database**: Six common and mostly searched online software engineering databases (used in [17,18,20,43–51]) were selected to gather the results. These include the ACM digital library [23], IEEE Explore [22], Science Direct [24], Springer digital library [26], Wiley online library [25], and Google Scholar [27].
2. **Publication Details**: The name of the publishing journal or conference where the secondary text was published.
3. **Year of Publication**: The year in which the research was published.

Table 3. Summary of the selected 22 secondary studies.

P ID	Author	Ref.	Publication Details
S1	S.Yoo, M. Harman	[2]	Software Testing, Verification, and Reliability
S2	E.Engström, P.Runeson, M.Skogland	[20]	Information and Software Technology
S3	S.Biswas, R.Mall, M. Satpathy, S.Sukumaran	[52]	Informatica
S4	C.Catal	[17]	EAST'12
S5	C.Catal, D.Mishra	[46]	Software Quality Journal
S6	Y.Singh, A.Kaur, B.Suri, S.Singhal	[43]	Informatica
S7	E.N.Narciso, M.Delamaro, F.Nunes	[53]	International Journal of Software Engineering and Knowledge Engineering
S8	R.Rosero, O.Gómez, G. Rodríguez	[47]	International Journal of Software Engineering and Knowledge Engineering
S9	S.Kumar, Rajkumar	[48]	ICCCA'16
S11	R.Kazmi, D.Jawawi, R.Mohamad, I.Ghani	[54]	ACM Computing Surveys
S12	M.Khatisyarbini, M.Isa, D.Jawai, R.Tumeng	[49]	Information and Software Technology
S13	H.Junior, M.Araújo, J.David, R.Braga, F.Campos, V.Ströele	[55]	SBES'17
S14	R.Mukherjee, K.S.Patnaik	[18]	Journal of King Saud University
S15	O.Dahiya, K.Solanki	[50]	International Journal of Engineering & Technology
S16	Y.Lou, J.Chen, L.Zhang, D.Hao	[56]	Advances in Computers
S17	M.Alkawaz, A.Silverajoo	[57]	ICSPC'19
S18	A.Bajaj, O.P.Sangwan	[51]	IEEE Access
S19	N.Gupta, A.Sharma, M.K.Pachariya	[58]	IEEE Access
S20	P.Paygude, S.D.Joshi	[59]	ICCBI'18
S21	J.Lima, S.Virgilio	[60]	Information and Software Technology
S22	M.Cabrera, A.Dominguez, I.Bulo	[61]	SAC'20

1. **Publication Type**: The publication type was categorized into a journal paper, conference paper, or book chapter.
2. **Focus Field**: Focus of any secondary text considered for the current SM is either TCS or TCP; a few studies considered both.
3. **Survey Type**: We categorized the survey types as SLRs, mappings, or surveys/reviews. SLRs and mappings were defined in previous sections. A third category encompassed any other strategy such as literature reviews, surveys, analysis, and trends.
4. **Range of years covered in the survey**: Range of years of primary texts considered in each of the secondary studies.
5. **The number of studies**: The number of primary texts analyzed in each secondary study, which is an indicator of the length of coverage.
6. **Analysis aspects**: The type of analysis performed in each of study.

4. Results and Analysis

The detail answers to the research questions using the gathered data are presented in the following points.

4.1. RQ1-Basic Information of Available Texts in TCS&P

The three parts of the question were answered as follows:

RQ 1.1: *What is the distribution of secondary texts on TCS&P over various online databases?*

Figure 2 depicts a pie chart representing the spread of the chosen 22 studies over the different online databases. ACM and Elsevier account for 70% of the selected studies, followed by IEEE Explore and Springer digital library, respectively. Only one survey from Wiley online library was found relevant to our SLR. Thus, there is a stronger presence of TCS&P in ACM and Elsevier.

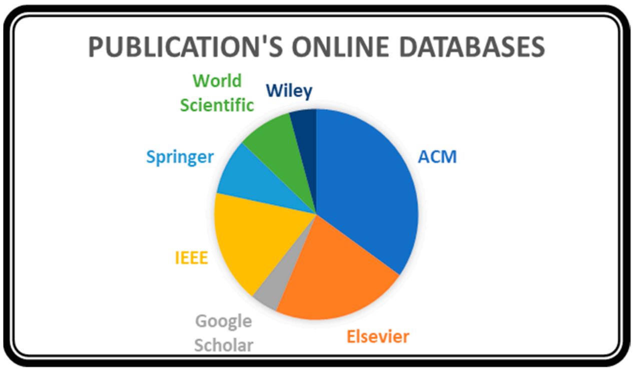

Figure 2. Spread of 22 texts over online databases.

RQ 1.2: *What is the evolution of the number of secondary studies published in TCS&P over the years?*

Growing surveys in the area reveal the growth in the research field of TCS&P. The need for new secondary studies arises from the obsolescence of existing ones or from missing aspects in the analysis. The 22 selected studies were published between Jan 2010 to May 2020. Figure 3 shows a line graph with the number of secondary studies published during this period per year. There was rapid growth in the number of surveys conducted in TCS&P after 2015. The number for 2020 is clearly lower as our study only considers the year until May; it is likely that the number for the remainder of 2020 would be consistent with the early trend. Five valuable and rigorous surveys in TCS&P were published in 2019. This demonstrates the growing interest of researchers in the field.

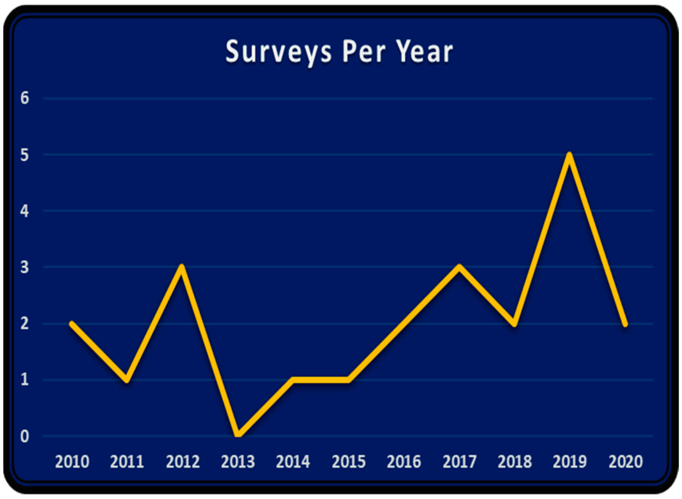

Figure 3. Number of surveys in TCS&P published per year.

RQ 1.3: *Which are the key publishing journals for TCS&P secondary studies?*

Figure 4 shows the type of publication of studies. Almost 70% of studies were journal papers, and the remainder were conference communications or book chapters. The length of text of a survey, SLR, or mapping study is generally larger than a typical general primary study. Not all journals in the software engineering field publish secondary texts due to domain and size constraints. Table 4 lists the names of the ten journals and the number of surveys on TCS&P. This list is not exhaustive for journals that might publish secondary texts; it simply shows where the selected secondary texts were published, as possible targets of publication for future contributions of researchers.

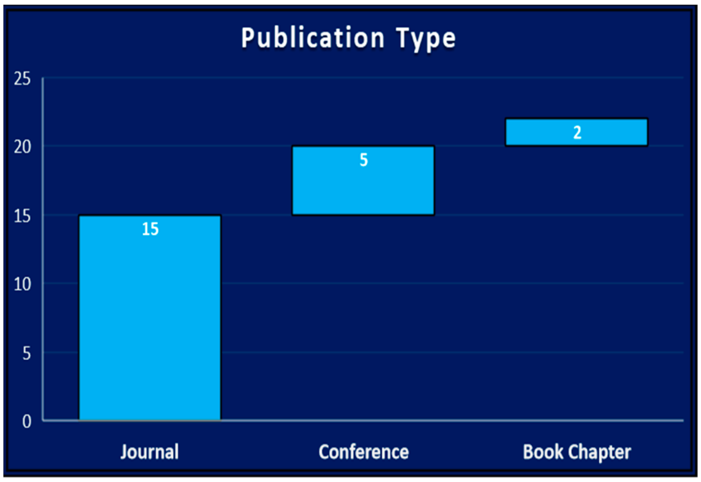

Figure 4. Publication type.

4.2. RQ2. To Study the Characteristics of Secondary Studies

RQ 2.1: *Which research approaches have been used by various secondary studies?*

Reviewing primary texts can be accomplished via various techniques. The most commonly used methods are SLR, systematic mappings, and non-systematic surveys/reviews. The difference in performing SLR and mappings is discussed in Section 1. These are systematic approaches with available guidelines to be followed for searching and selecting the studies for review. Due to the systematic approach, these techniques take a large amount of time and effort to complete the review process. Many reviews may not claim to include all the studies in the relevant field or to provide conclusions after analysis. Such texts are very helpful in providing insights into various aspects and the collection of primary texts available in the reviewed area. Moreover, they consume comparatively less time and effort. Many researchers have adopted this approach for performing literature reviews, surveying particular aspects, and analyzing and depicting trends. All of these texts were classified to the survey/review category. Figure 5 provides the classification of the reviewed 22 studies into three types: eight SLRs, three mappings, and one paper claiming to be comprise both an SLR and a mapping. Ten studies were surveys/reviews but not SLRs or mapping studies.

Table 4. Journals in which secondary studies were TCS&P published.

Name of the Journal	NS
Information & Software Technology	3
Informatica	2
International Journal of Software Engineering and Knowledge Engineering	2
IEEE Access	2
Software Testing, Verification, and Reliability	1
Software Quality Journal	1
Frontiers of Computer Science	1
ACM Computing Surveys	1
Journal of King Saud University	1
International Journal of Engineering & Technology	1

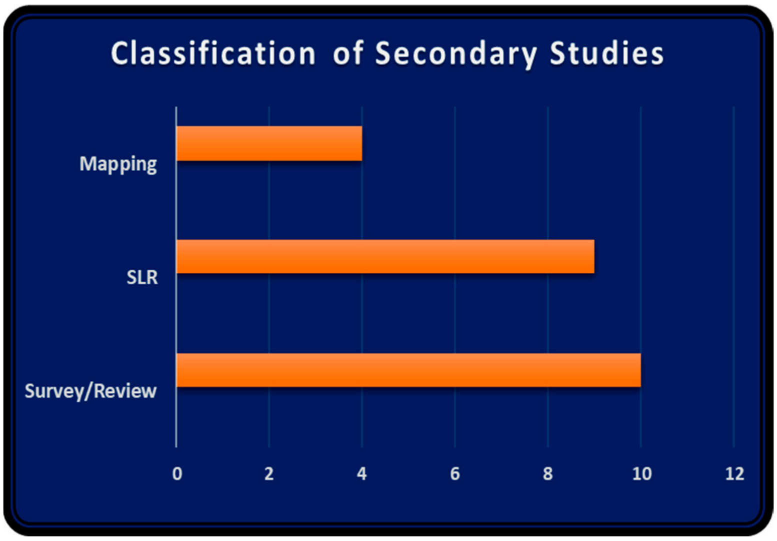

Figure 5. Classification of selected 22 secondary studies.

Figure 6 depicts publishing trends for these three types of surveys in TCS&P. Publishing trends for mappings are marked as 0 or 1 in a year. There was an apparent increase in the number of SLRs and reviews/surveys in TCS&P after 2015. The growing interest of the research community in the field of TCS&P can be inferred from these publication trends.

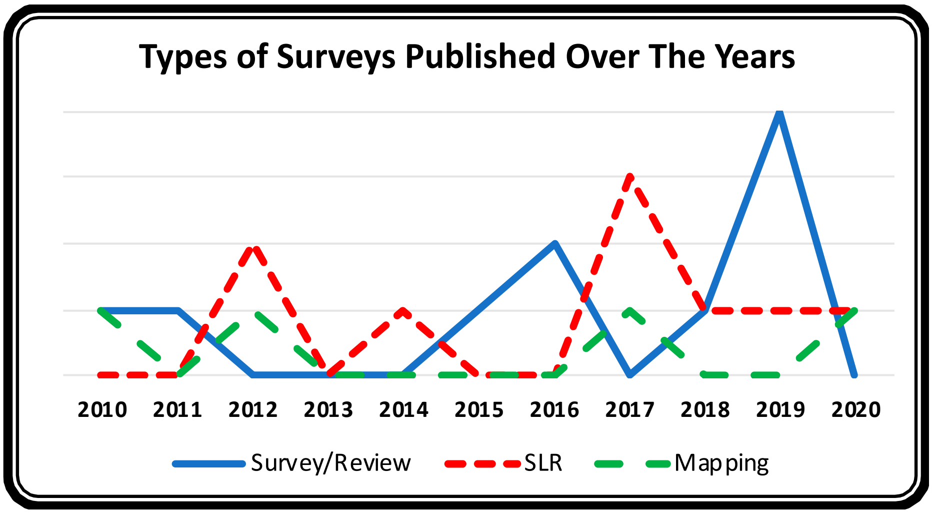

Figure 6. Types of surveys published by year.

RQ 2.2: What are the focus and the range of years covered in the secondary studies?

Figure 7 uses the radar plot to represent the same years covered by the studies. It shows the beginning and the ending year of respective searches conducted in the studies (identified as S1 to S22). It was found that all surveys or review papers (fitting our inclusion criteria) were published after 2010. Hence the outer circle, which represents the end year of the search, encompasses a decade (2010–2020). One of the studies (S22) was undertaken after another existing study; hence the search years ranged only from 2017 to 2019.

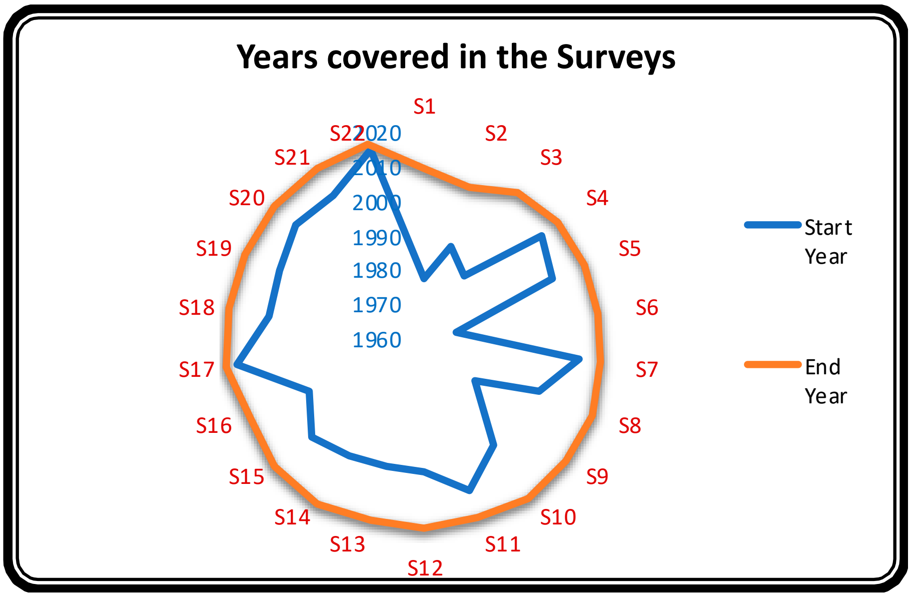

Figure 7. Years covered in the 22 secondary studies for the survey.

Surveys performed over comparable year ranges must include almost the same texts. Similarly, future researchers can explore the primary studies lying outside the outer plot presented in Figure 7.

The 22 studies either focused on TCS, TCP, or a combination of both. Figure 8 shows how the majority of the secondary texts (82%) focused on TCP, whereas only 32% focused on TCS.

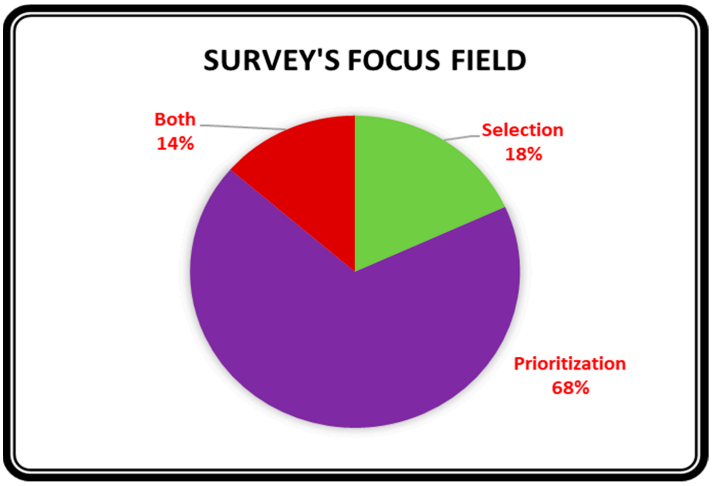

Figure 8. Distribution of research field over the 22 studies for the survey.

Research in the area of TCP began in 1997 [3], whereas that of TCS began in 1988 [62]. The surveys conducted after these studies used information gathered in these papers, so they needed to cover a reduced number of years. As is evident from Figure 7, most of the surveys were been conducted over the last two decades (2000–2020).

RQ 2.3: *How many texts form the basis of research for the selected secondary studies?*

The number of chosen primary studies for conducting the survey also measures the scope of studies. Depending directly on the selected research questions, there can be significant variations in the number of primary studies selected for SLRs. However, mapping studies must include literature that exists in the relevant topic being reviewed. Thus, the number of primary studies selected by surveys on similar topics chosen from the same range of years must be comparable.

Figure 9 plots the number of primary studies reviewed by the 22 studies (S1–S22) except for S3, where the total number of reviewed texts was unavailable. The number varied from the lowest (seven) to the highest (320) in primary texts for the remainder of the studies. Mappings recorded almost similar studies from a similar range of years, as expected. S22 reported a maximum of 320 primary texts and only in 3 years (2017–2019), although the specific links to each of the studies were missing. All other papers provided links to the primary texts. Thus, researchers can find almost all primary texts until May 2020 in the TCS&P field directly from these 22 studies.

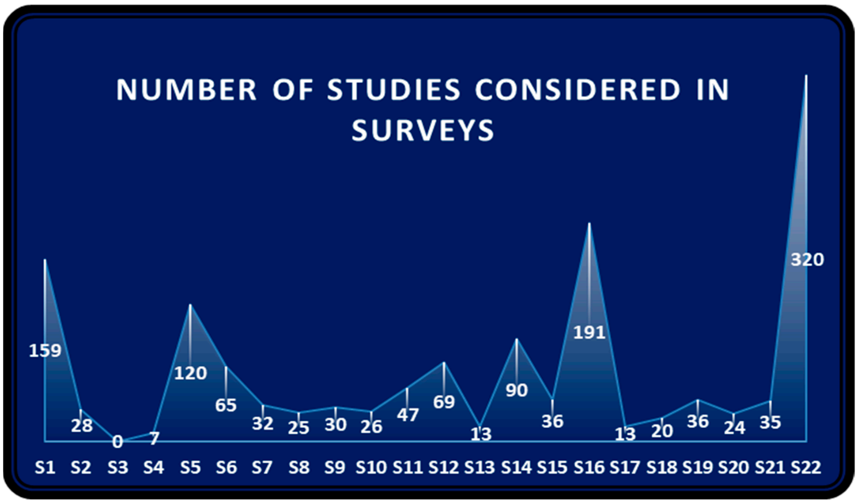

Figure 9. The number of studies surveyed in the 22 studies for the survey.

4.3. RQ3. How Have the Secondary Studies Benefitted from the Existing Tools and Tests?

RQ 3.1: *Have statistical tests been used in the secondary studies?*

Table 5 shows the details of the statistical tests in each study. The details include the area or purpose of the test, the reference of the studies where it was used, and the number of times it was used in the 22 surveyed papers.

Table 5. Statistical tests used in secondary studies summary.

S. No.	Test Name	Area	References	Count
1	Bonferroni Test	Multiple Comparison for analyzing data	[1,4,63–65]	5
2	Chi-Square	Extract the features of a test case	[66]	1
3	Vargha and Delaney Test	Performance Analysis of the algorithm and Effect Size Measurement	[67–74]	8
4	Mann–Whitney	Pairwise comparison for Performance Analysis of the algorithm	[67–70,75,76]	6
5	Spearman's Correlation	Performance Analysis of the algorithm	[77–79]	3
6	Welch's test	Performance Analysis of the algorithm	[72]	1
7	Wilcoxon t-test	Performance Analysis of the algorithm	[72]	1
8	Two-Way permutation	Evaluation of execution time of algorithms	[80]	1
9	Kruskal Wallis	To Study the statistical difference of running time of search algorithms and Multiple Comparison for the Performance Analysis of the algorithm	[76,77]	2
10	Anova	Multiple Comparison for the Performance Analysis of the algorithm	[81–87]	7
11	LSD	Performance Analysis of the algorithm	[81]	1
12	Dunn's Test	Multiple Comparison for the Performance Analysis of the algorithm	[76]	1
13	Scott–Knott Analysis	Multiple Comparison for the Performance Analysis of the algorithm	[74]	1
14	Cliff's Delta	Effect Size Measurement	[74]	1
15	Turkey's HSD	Multiple Comparison for the Performance Analysis of the algorithm	[83,87]	2

Figure 10 represents a bar graph revealing the scarcity in the usage of statistical tests in these secondary studies in TCS&P. Only four of the 22 surveyed studies mentioned the statistical tests used in the area.

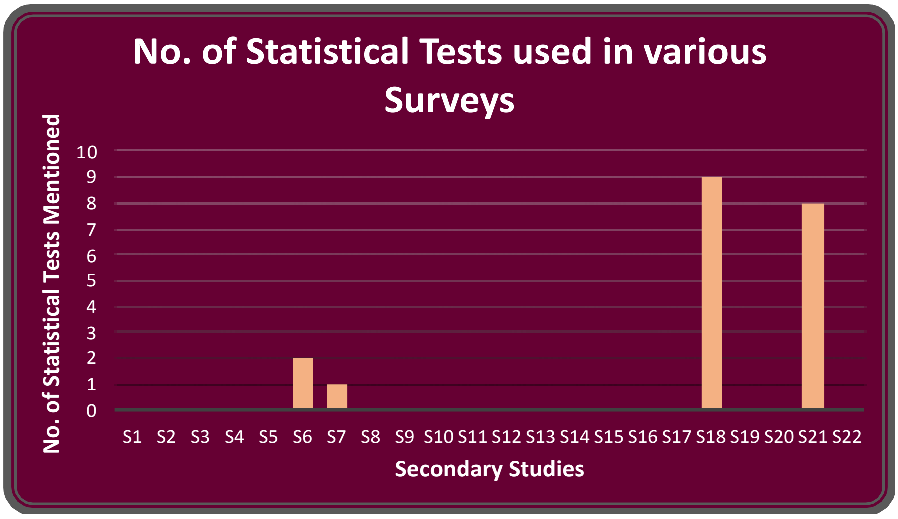

Figure 10. The number of statistical tests used in various surveys.

Figure 11 shows a doughnut graph to provide better insight into the application of statistical tests. The performance analysis of the techniques proposed by the researchers used most of the tests (approximately 74%). The graph depicted in Figure 12 clearly shows that ANOVA, Mann–Whitney, Vargha and Delaney, and Bonferroni tests were the most used in TCS&P. Table 4 provides a quick reference to the studies where the usage of these tests was recorded.

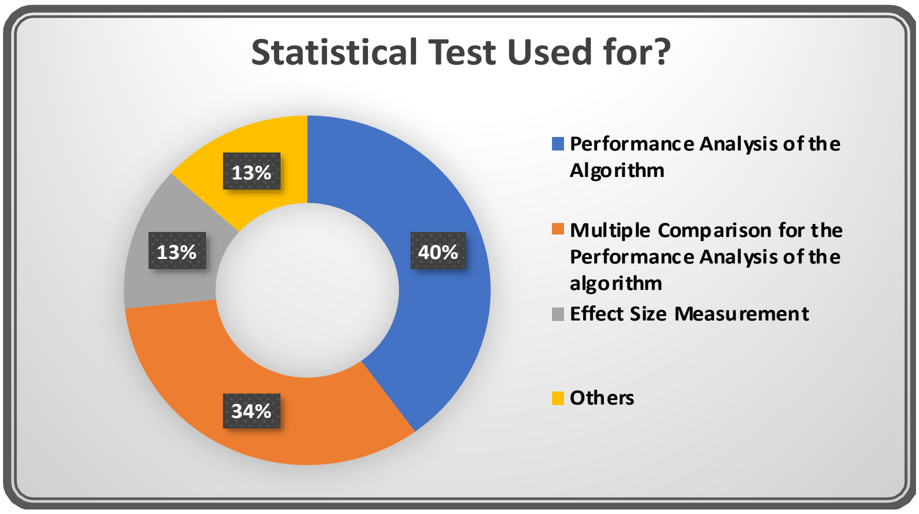

Figure 11. Usage of statistical tests.

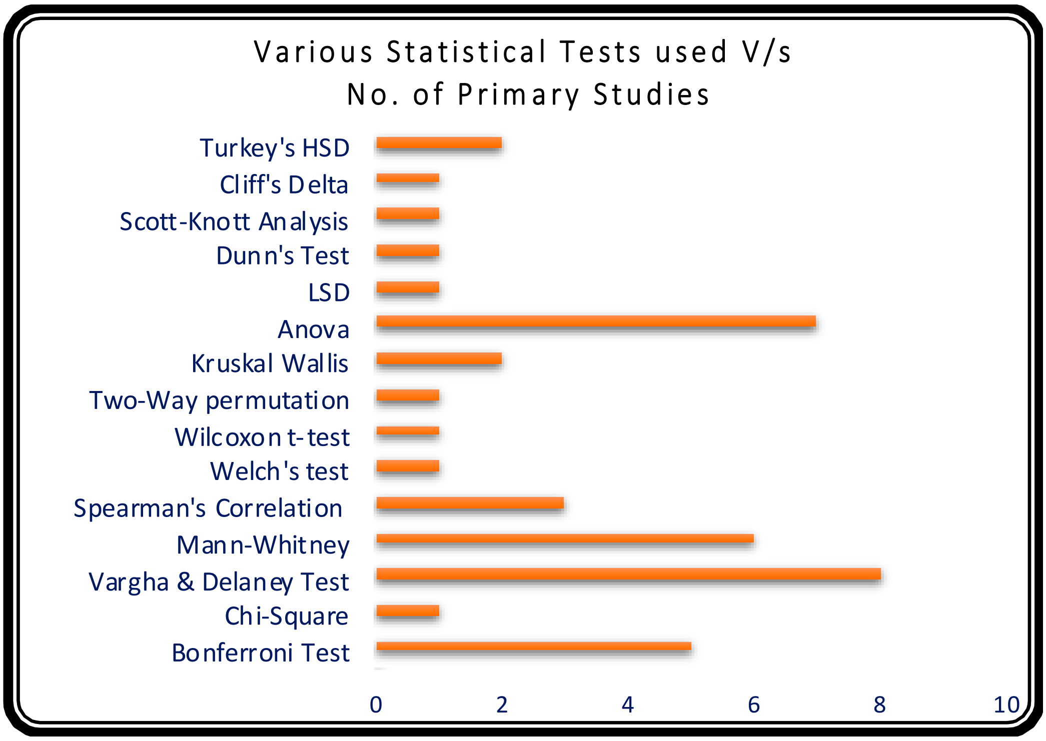

Figure 12. Various Statistical Tests used V/s Number of Primary Studies.

Unfortunately, this analysis indicates a major gap in the usage of statistical tests in regression TCS&P. This provides an opportunity for researchers who could make future use of statistical tests for qualitative assessment, validation, and comparison of the techniques used in TCS&P.

RQ 3.2: *Which tools have been used by the secondary studies?*

Table 6 summarizes the information about the open-source behavior, available download links, and the referring studies of the mentioned research tools.

Table 6. Summary of tools mentioned in the 22 surveyed studies.

S. No	Tool Name	Reference Study	Categorization/Type/Purpose	Open Source	REF
1	Aristotle Program Analysis System	S6, S13	Program Analysis/control ƒflow graph information	No	1
2	Sofya	S6, S13	ByteCode Analysis Tool for Java/Bytecode manipulation/analysis	Yes	2
3	Emma	S6, S13, S18	Code Coverage Tool for Java	Yes	3
4	Unix Based tools_(e.g., Unix Diff)	S6, S13, S18	Process tracking, collecting dynamic coverage information/Change analysis	No	-
5	SPSS	S6, S13	Data Analysis Tool	Yes	4
6	MATLAB	S6, S13, S18	Analysis and Programming Tool	Yes	5
7	Proteum Mutation System	S6, S13	Mutation System	Yes	6
8	Galileo	S6, S13	Prototype software tool for dynamic fault tree analysis/Bytecode manipulation/analysis	Yes	7
10	CodeCover	S13	Code Coverage Information	Yes	9
11	gcov	S13, S18	Code Coverage Information	Yes	10
12	Corbertura	S13	Code Coverage Information	Yes	11
13	JaCoCo	S13	Code Coverage Information	Yes	12
14	Cantata++ or Cantata	S13	Code Coverage Information	Yes	13
15	xSuds	S13	Code Coverage Information	Yes	14
16	LDRA Testbed	S13	Code Coverage Information	Yes	15
17	ATAC	S13	Code Coverage Information	Yes	16
18	Fault Tracer	S13	Code Coverage Information/Bytecode manipulation/analysis	Yes	17
19	PIT	S13	Source code mutation	Yes	18
20	MuJava	S13	Source code mutation	Yes	19
21	MutGen	S13	Source code mutation	Yes	20
22	Javalanche	S13	Source code mutation	Yes	21
23	Jumble	S13	Source code mutation	Yes	22
24	Major	S13	Source code mutation	Yes	23
25	Jester	S13	Source code mutation	Yes	24
26	Zoltar	S13	Source code mutation	Yes	25
27	xSuds	S13	Prioritization tool/framework/test analysis	Yes	26

Table 6. *Cont.*

S. No	Tool Name	Reference Study	Categorization/Type/Purpose	Open Source	REF
28	Apros	S13	Prioritization tool/framework	Yes	27
29	MOTCP	S13	Prioritization tool/framework	Yes	28
30	SLOCCount	S13	Source code metrics information	Yes	29
31	ckjm	S13	Source code metrics information	Yes	30
32	CLOC	S13	Source code metrics information	Yes	31
33	SWT-Metrics	S13	Source code metrics information	No	-
34	valgrind	S13	Execution trace information	Yes	32
35	daikon	S13	Execution trace information	Yes	33
36	AspectJ	S13	Execution trace information	Yes	34
37	ASM	S13	Bytecode manipulation/analysis	Yes	35
38	Celadon	S13	Change analysis	Yes	36
39	ILOG CPLEX	S13	Linear integer programming solver	Yes	37
40	GUROBI optimization	S13	Linear integer programming solver	Yes	38
41	AutoBlackTest	S13	Test scripts generation/creation	No	-
42	Traceclipse	S13	Artifacts traceability	Yes	39
43	Jmetal	S13	Multi-objective optimization framework	Yes	40
44	Vulcan	S13	Analysis of dynamic binary code	No	-
45	Indri toolki	S13	Information retrieval framework	Yes	41
46	Smile Library	S13	BayesianNetwork framework	Yes	42
47	LDRA Testbed	S13	fault history	Yes	43
48	ref-Finder	S13	refactoring diff	Yes	44

References for the tools mentioned in Table 6 are given in Table 7.

Table 7. Web links for tools (last accessed on 15 November 2021).

REF	Web Link
1	http://gamma.cc.gatech.edu/aas.html
2	https://github.com/SofyaTool/SofyaTool
3	http://emma.sourceforge.net/
4	https://www.ibm.com/in-en/products/spss-statistics
5	https://www.mathworks.com/products/matlab.html
6	https://github.com/magsilva/proteum
7	https://www.cse.msu.edu/~cse870/Materials/FaultTolerant/manual-galileo.htm
8	http://sandmark.cs.arizona.edu/index.html
9	http://codecover.org/
10	https://gcc.gnu.org/onlinedocs/gcc/Gcov.html
11	https://cobertura.github.io/cobertura/
12	https://www.eclemma.org/jacoco/
13	https://www.qa-systems.com/tools/cantata/
14	https://stellar.cleanscape.net/products/testwise/help/introduction.html
15	https://www.mathworks.com/products/connections/product_detail/ldra-tool-suite.html
16	https://invisible-island.net/atac/atac.html

Table 7. *Cont.*

REF	Web Link
17	https://personal.utdallas.edu/~lxz144130/ftracer.html
18	https://pitest.org/
19	https://cs.gmu.edu/~offutt/mujava/
20	https://github.com/fhcrc/mutgen
21	https://github.com/wenxcs/javalanche
22	http://jumble.sourceforge.net/
23	https://mutation-testing.org/
24	https://sourceforge.net/projects/jester/
25	https://github.com/ncfxy/zoltar
26	https://stellar.cleanscape.net/products/testwise/help/introduction.html
27	http://www.apros.fi/en/
28	https://www.scribd.com/document/210345795/Motcp
29	https://dwheeler.com/sloccount/
30	https://www.spinellis.gr/sw/ckjm/
31	http://cloc.sourceforge.net/
32	https://valgrind.org/
33	https://plse.cs.washington.edu/daikon/download/
34	https://www.eclipse.org/aspectj/downloads.php
35	https://gitlab.ow2.org/asm/asm
36	https://01.org/projectceladon/
37	https://www.ibm.com/support/pages/downloading-ibm-ilog-cplex-optimization-studio-v1290
38	https://www.gurobi.com/downloads/gurobi-optimizer-eula/
39	http://www.cs.wm.edu/semeru/traceclipse/
40	http://jmetal.sourceforge.net/
41	http://www.lemurproject.org/download-archive.php
42	https://github.com/haifengl/smile
43	https://www.mathworks.com/products/connections/product_detail/ldra-tool-suite.html
44	https://github.com/fulxie/RefFinder

A few studies mention other tools in addition to those mentioned in Table 5. The secondary study S2 refers to a text [88] that contains a small section of the survey of early tools developed for regression testing, listing their advantages and disadvantages. S6 mentions tools created to automate their own proposed techniques. Some of the tools, such as Vulcan, BMAT, Echelon, déjà vu, GCOV, Test runner, Winrunner, Rational test suite, Bugzilla, and Canatata++, were only mentioned in one study each. S8 mentions two tools: RTSEM and MISRA-C. S9 mentions three more tools: SPLAR, Feature IDE, and MBT.

As is evident from Figure 13, only 14% of the studies mentioned the tools used in the area of regression TCS &P. These tools are listed in Table 5, with the study in which they were referenced, the categorization/purpose of the tool (if mentioned), and the download link (if the tool is open source).

The downloadable links and referring study links can help future researchers to obtain a quick reference to the tools that have already benefitted the TCS&P field.

Figure 14 graphically shows that 25% of the tools are used for analysis purposes, 21% of tools are used for providing code coverage information, 19% of tools are used for stipulating the mutation adequacy score, 8% of tools are used for delivering source code metrics, and the remainder are used for miscellaneous purposes. In general, the TCS&P area lacks the usage of standard tools, and the tools used are concentrated in the analysis of techniques and code coverage. Thus, this observation suggests there is scope for finding ways of standardizing the set of possible tools to encourage more automation in the area of regression TCS&P.

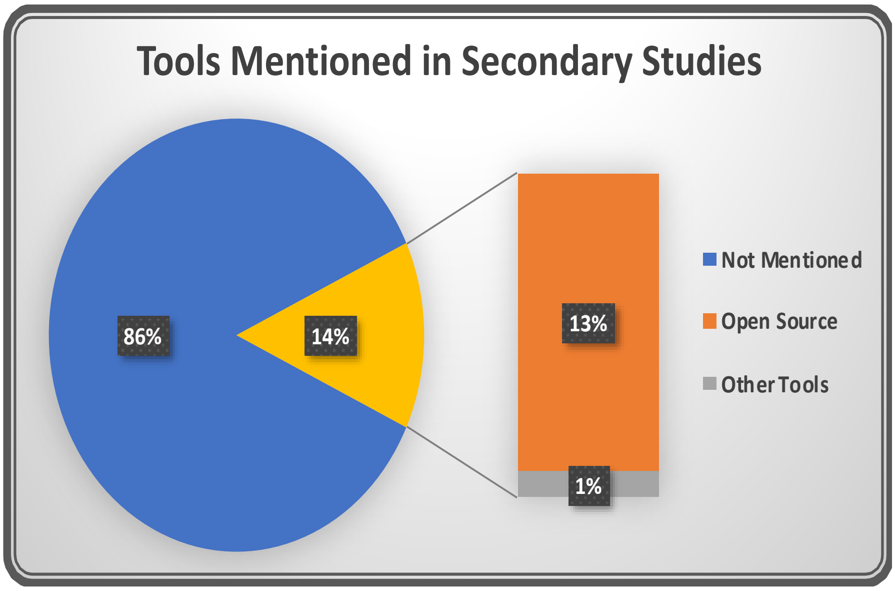

Figure 13. Tools mentioned in secondary studies.

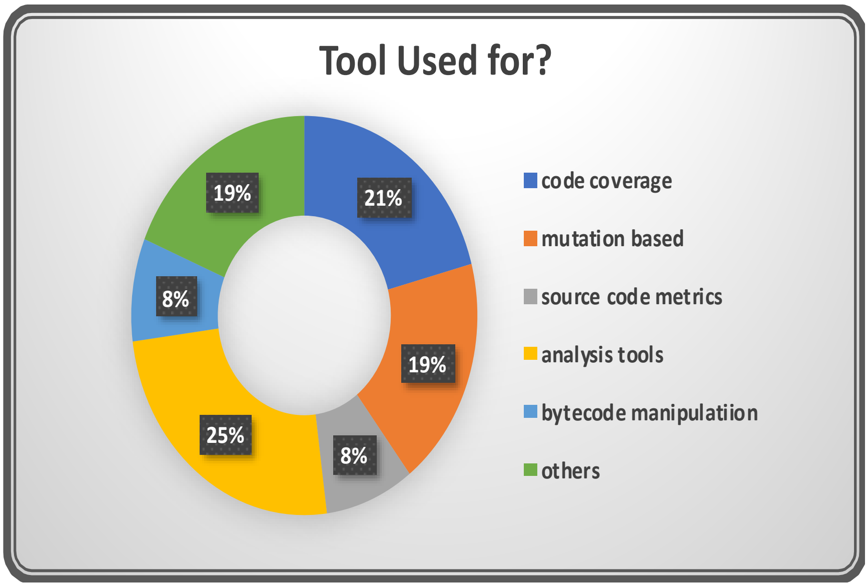

Figure 14. Domain of the tools used.

4.4. RQ4. How Can the Quality of Secondary Studies Be Compared?

The categorization presented for various TCS&P techniques in the secondary studies was recorded and is plotted in Figure 15. The highest percentage of studies categorize the techniques based on the approach, whereas a few studies specifically chose genetic algorithms (GA), and 27% of the studies chose other unique categorization techniques.

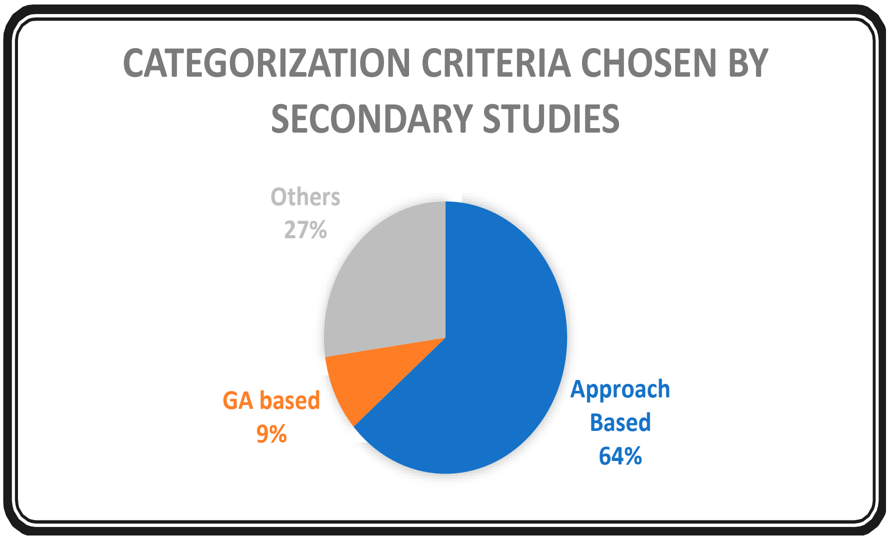

Figure 15. Categorization criteria chosen by secondary studies.

The categorization trend can be justified by the evident change in the upcoming techniques over time. In addition to this, a few studies focused on a specific area of TCS&P rather than on the general analysis of TCS&P. Therefore, categorization is not sufficient for quality assessment.

RQ 4.1: *How have the research questions been used in the secondary studies?*

The different approaches used in secondary studies clearly highlight the importance of research questions when performing a secondary study. Hence, the research questions (RQs) covered by secondary studies in TCS&P were analyzed. Figure 16 represents the number of RQs considered by the studies. Most of the studies mentioned between three and five RQs. The count of RQs includes the sub-questions of the RQs, as mentioned in the study. Publication trends determine the most found RQs. The majority of the secondary studies included RQs that are factual in nature. Only five studies analyzed the collected facts and discussed their findings from the analysis. No peculiar trend could be seen in terms of increasing RQs or in the improvement in the quality of RQ.

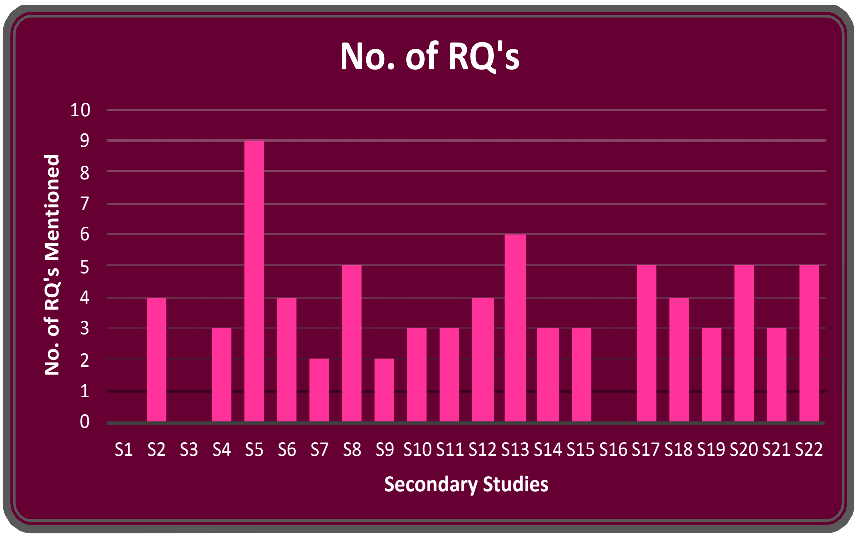

Figure 16. Number of research questions in the 22 studies for the survey.

Figure 17 shows that 63 % of the surveys in TCS&P focused on illuminating the 'facts' of working in the area. Only 23% of the secondary studies also focused on the analysis part of the work accomplished in the area. This highlights the deficit in analytically surveying or reviewing the TCS&P field over time. This also motivated us to propose detailed analytical RQs as much as possible within the logical technical constraints and limitations of time and cost.

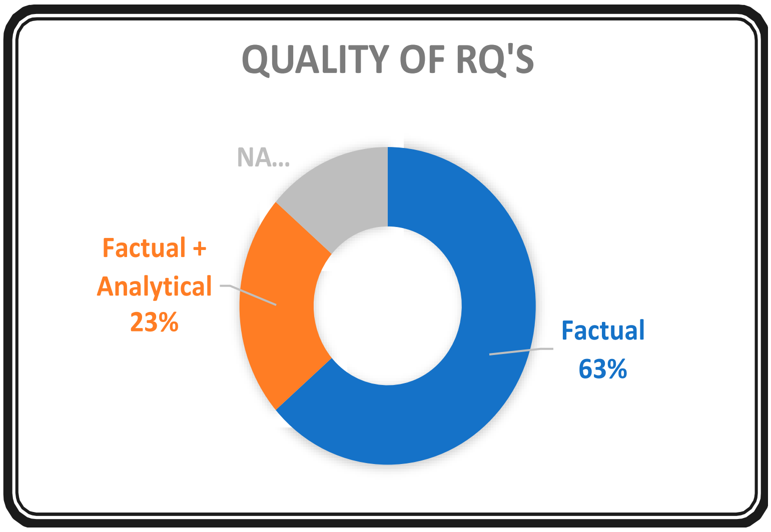

Figure 17. Quality of research questions considered in the 22 studies for the survey.

Figure 18 indicates that the majority (77%) of the surveys clearly and substantially answered the defined RQs. The remainder of the surveys either lack a pre-defined objective for the study or did not substantially answer the defined RQs. Although 23% of the studies did not answer their RQs, it is relevant to highlight the importance of proper usage of RQs in secondary studies.

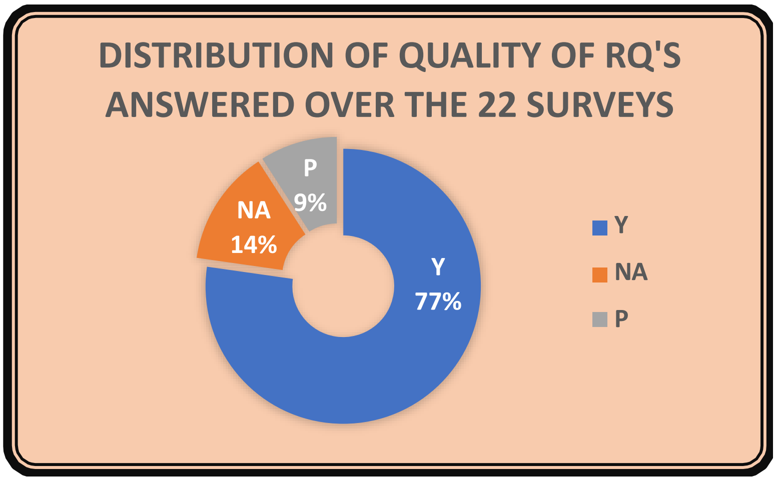

Figure 18. Distribution of quality of research questions answered over the 22 studies for the survey.

RQ 4.2: *What is the quality of the secondary studies performed in the area?*

We worked with a total of seven Research Paper Quality (RPQ) parameters in order to provide a qualitative analysis of the secondary studies published in the area of regression TCS&P. These RPQs were designed to derive their inspiration from seminal studies [20,21] in the area of TCS&P. Table 8 lists the RPQ values allocated to the 22 surveyed studies during research and analysis.

Table 8. Research paper quality markings and total for the reviewed 22 secondary studies.

S. No.	RPQ1	RPQ2	RPQ3	RPQ4	RPQ5	RPQ6	RPQ7	QUALITY
S1	0	0	1	0	0	1	1	3
S2	1	1	1	0	0	1	1	5
S3	0	0	1	0	0	1	1	3
S4	1	0.5	1	0	0	0.5	1	4
S5	1	0.5	1	0	0	1	1	4.5
S6	1	1	1	1	1	1	0.5	6.5
S7	1	0.5	1	1	0	0.5	1	5
S8	1	0.5	0	0	0	0.5	0.5	2.5
S9	1	0.5	0	0	0	0	0	1.5
S10	1	0.5	1	0	0	0	0	2.5
S11	1	0.5	1	0	0	1	1	4.5
S12	1	0.5	0	1	0	1	0.5	4
S13	1	1	0	0	1	0.5	0.5	4
S14	1	0.5	1	0	0	1	0	3.5
S15	1	0.5	0	0	0	0.5	0.5	2.5
S16	0	0	1	0	0	1	0.5	2.5
S17	1	0.5	0	0	0	0	0	1.5
S18	1	0.5	1	1	1	1	1	6.5
S19	1	0.5	0	0	0	1	0.5	3
S20	1	0.5	0	0	0	0	0	1.5
S21	1	1	1	1	0	1	1	6
S22	1	1	0	0	0	0	0.5	2.5

RPQ1 (RQ) denotes whether the study mentions and answers the RQ or not. A study was assigned the value '1' if it was found to mention the RQs; else, it was assigned the value '0'.
RPQ2 (RQ Quality) represents the quality of the RQ research in the secondary study. Value '0' means that no research questions appeared in the secondary study, so they were not answered. Secondary studies having RQs framed to provide only descriptive analysis of the existing data, without providing exploratory, inferential, predictive, or diagnostic analysis of the primary studies, were assigned the value '0.5'. Value '1' means high-quality RQs were specified and answered.
RPQ3 (Future Prospects): A study that mentions the scope of future prospects or gives directions to contribute further to the area received the value of '1' whereas value '0' was assigned to the remainder.
RPQ4 (Statistical Test): If statistical tests used in the area by different researchers were present in the secondary study, the value was '1'; otherwise, the value was '0'.
RPQ5 (Tools available): If the tools used in the area of TCS&P were mentioned in the study, then the value '1' was assigned; else the value '0' was assigned.
RPQ6 (Detailed Analysis): Adopts value '1' for a study that provided a detailed and in-depth analysis of the research work accomplished in the area of TCS&P. A study with a value '0.5' included only limited descriptive analysis of the research work such as publication years and details. Value '0' was assigned to a study that was a mere collection of facts and did not contain any analysis of the research conducted in the area.
RPQ7 (Novel Contribution): Value '1' means that the study made a novel contribution in the analysis of the research conducted in TCS&P. Value '0.5' appears when the contribution or analysis was a mixture of some novel findings and the repetition of some findings already available in the literature, and value '0' implies that the contribution did not include any significant or novel analysis of the research.

Total (RPQ) sums the values of the RPQ to provide a comprehensive qualitative analysis of the secondary studies. Our qualitative analysis shows that S6, S18, and S21 contributed most significantly to the area, and S2, S4, S5, S7, S11, S12, and S13 also contributed significantly to the area.

Even after a significant increase in the number of secondary studies in TCS&P, it was not accompanied by an improvement in quality as measured by the seven RPQ parameters. This suggests an opportunity for improving the quality of secondary studies in the future.

4.5. RQ5. What Are the Probable Prospects in TCS&P?

RQ 5.1: *What are the TCS&P techniques as mentioned in the secondary studies?*

Here we present categories of TCS&P techniques in a tabular manner (Table 9) to highlight the kinds of techniques available in the area. The second column indicates the studies which mention the technique. The techniques (Coverage-based, History-based, Requirement-based, Model-based, and Fault-based) were categorized into over 10 surveys; however, all of these are old techniques. In addition, we observed that Evolutionary techniques (which include Search-based, GA-based, ACO-based, ABC and BA, and hill climbing) saw a significant rise in development after 2015. Hence, to keep up with the current research, it would be informative to identify the trend in research on evolutionary techniques. The subsequent RQ analyzes and addresses this aspect.

Table 9. Category of TCS&P techniques mentioned in the studies.

S. No.	Category	Studies in Which Mentioned
1	Coverage-based approach	S1, S5, S6, S7, S10, S12, S13, S14, S17, S20, S21, S22
2	Distribution-based approach	S1, S5, S21
3	Human-based approach	S1, S5, S21
4	Probabilistic approach	S1, S5, S21
5	History-based	S1, S5, S6, S8, S12, S13, S14, S17, S21, S22
6	Requirement-based	S1, S5, S6, S12, S13, S14, S15, S17, S20, S21, S22
7	Model-based	S1, S5, S8, S13, S14, S16, S17, S19, S20, S21
8	Component-based	S3
9	Cost-aware approach	S5, S14, S15, S17, S21
10	Interface-contract mutation	S5
11	Relevant slices	S5
12	Call tree-paths	S5
13	Composite	S6
14	Faut-based	S6, S8, S12, S13, S15, S16, S17, S19, S20, S22
15	Genetic algorithm-based	S1, S4, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S17, S18, S19, S20
16	AI-based	S8
17	Risk-based	S12, S14, S17, S22
18	Search-based or evolutionary	S12, S13, S14, S15, S17, S19, S20, S22
19	Similarity-based	S13
20	Program structure-based	S3, S13
21	Oracle-based	S13
22	Time aware prioritization	S14, S19
23	GUI/Web application-based	S3, S14
24	ACO-based	S12, S14, S15, S17, 20
25	Clustering	S14

Table 9. *Cont.*

S. No.	Category	Studies in Which Mentioned
26	Test input-based	S3, S16
27	Change/modification-based	S6, S13, S20
28	Bayesian network-based	S17

RQ 5.2: What is the extent of Evolutionary techniques in secondary studies?

Recent research has witnessed the advent of many evolutionary techniques, such as GA, ACO, and hill climbing. New evolutionary techniques are continuously being developed and applied to various optimization problems, including software engineering problems [89–91]. TCS&P is also an optimization problem and hence has benefitted from various evolutionary techniques.

Table 10 summarizes the details of the evolutionary techniques used in TCS&P in the 22 secondary studies. Twelve evolutionary techniques were detected as part of the secondary studies. Table 8 shows the details of the studies where they appear, with quick references to the primary studies in which the techniques were applied to TCS&P.

Table 10. Details of evolutionary techniques surveyed in the 22 secondary studies. (GA—genetic algorithms, PSO—particle swarm optimization, AVM—alternating variable method, EA—evolutionary algorithms, HC—hill climbing, ACO—ant colony optimization, SD—string distance, BA—bacteriological algorithm, BCO/ABC—bee colony optimization/artificial bee colony).

S. No.	Evolutionary Technique	Study	Reference
1	GA	S1(2010), S4(2012), S6(2012), S7(2014), S8(2015), S9(2016), S10(2016), S11(2017), S12(2017), S13(2017), S14(2018), S15(2018), S17(2019), S18(2019), S19(2019), S20(2019)	[2,17,51,67–72,77–80,92–129]
2	NSGA-II	S1(2010), S19(2019)	[71,77,80,130]
3	vNSGA-II	S1(2010)	[130]
4	PSO	S7(2014), S11(2017), S15(2018), S20(2019)	[77,131–137]
5	AVM(1+1)	S9(2016)	[78]
6	Evolutionary Algorithm	S9(2016)	[78]
7	Hill Climbing	S10(2016), S14(2018)	[81]
8	ACO	S12(2017), S14(2018), S15(2018), S17(2019), S20(2019)	[21,138–144]
9	String Distance	S12(2017)	[145]
10	Bacteriological Algorithm	S15(2018)	[146]
11	Bee Colony Optimization	S15(2018)	[141]
12	ABC	S19(2019)	[147]

The initial secondary text in TCS&P with an evolutionary technique (GA) was S1. Subsequently, many more techniques have been reviewed. Figure 19 depicts the number of evolutionary techniques surveyed per year from 2010 until 2020. After 2015, the number of surveyed evolutionary techniques experienced rapid growth. This implies an even more rapid increase in the number of available primary texts using evolutionary techniques for TCS&P. This will provide support to researchers who use evolutionary techniques in TCS&P in future works.

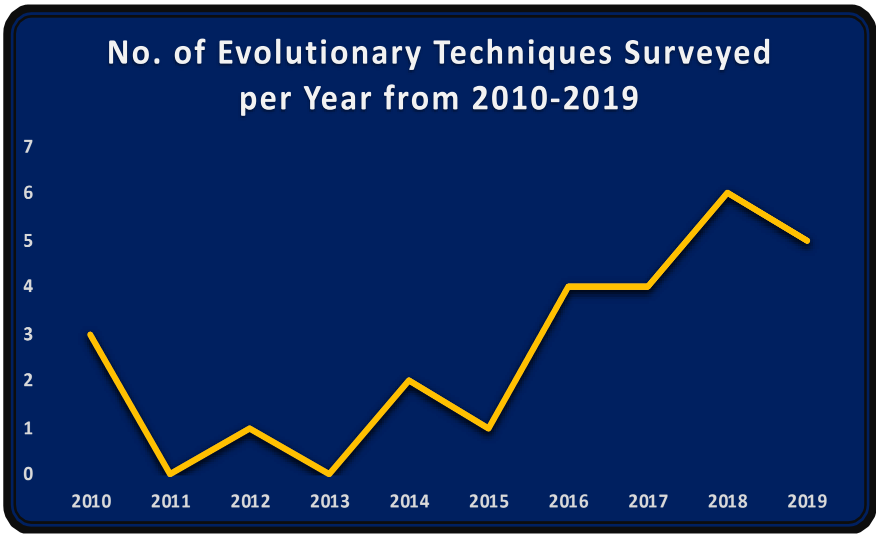

Figure 19. The number of evolutionary techniques surveyed per year from 2010 to 2019.

Notably, the number of secondary studies mentioning evolutionary techniques also rose after 2015 (see Figure 20). This is evident from the fact that evolutionary techniques increased in number for TCS&P after 2015 (Figure 20) and that the number of secondary studies conducted in TCS&P also rose post-2015 (Figure 3). It can also be easily inferred that the extensive use of evolutionary techniques in TCS&P is increasing.

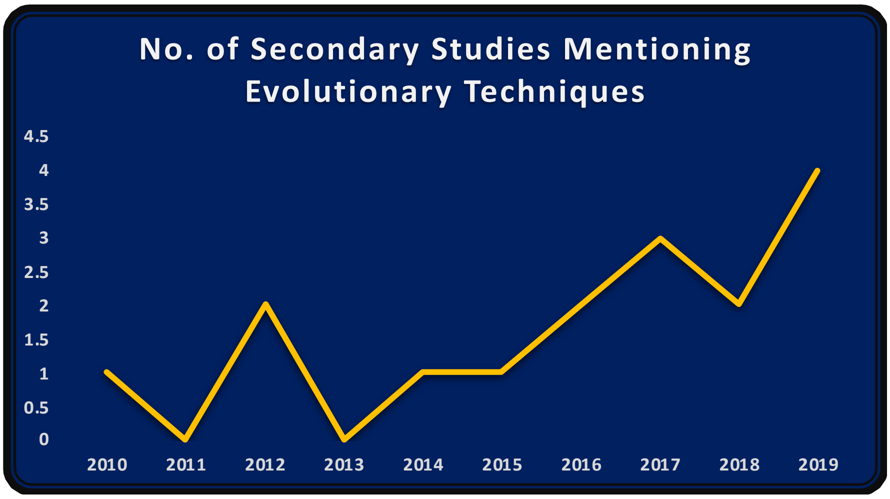

Figure 20. Number of secondary studies mentioning evolutionary techniques by year.

Another relevant finding is that GA is the most commonly applied evolutionary technique in TCS&P (see Figure 21), and has been used in 16 secondary studies. GA and its variants are popular in obtaining solutions to many other real-world problems [148,149]. The two next highest ranking are ACO and PSO. Additional new techniques are being developed while earlier approaches continue to be used. All these findings provide incentive for researchers to work on evolutionary techniques in TCS&P.

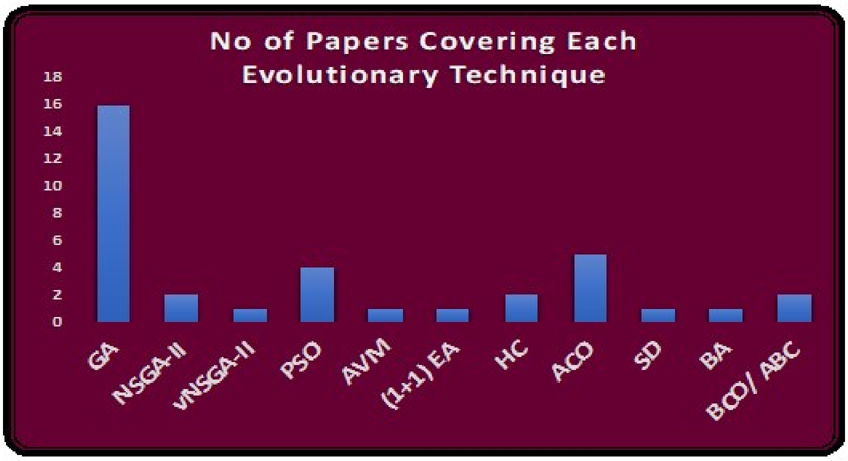

Figure 21. Number of secondary studies covering each evolutionary technique.

5. Threats to Validity

Both secondary and tertiary studies are susceptible to threats to validity [150]. We followed the categorization provided by Ampatzoglou et al. [150] for classifying the possible threats to validity present in the case of SLRs or SMs. We also mention the steps taken to mitigate the effects of these threats in the following section.

1. **Study Selection Validity:** One of the main threats in the case of SLRs or SMs is the elicitation of relevant studies. We formulated a search string for automated selection from six databases. The string was manually tested with a trial-and-error technique to check if it found all the well-known studies. In addition, we also performed snowballing to gather any study missed while performing automated retrieval of studies. However, it is possible that we missed a small number of studies that used different terminology to describe their secondary study (such as 'analysis' or 'study synthesis'or 'study aggregation', or any similar phrases or strings). Moreover, we excluded grey literature in the area by assuming that good quality grey literature is generally available in conference/journal papers. It is also desirable to use Scopus and WoS databases for the retrieval of relevant studies. In our case, we could not obtain access to these databases, so we used Google Scholar and other seminal databases (IEEE, Wiley, Science Direct, Springer, and ACM) to gather the relevant studies.
2. **Data Validity:** Data validity threats occur when performing data extraction or analysis of the extracted data. Individual bias may be a factor when locating relevant facts and subjective data from the gathered studies. The data extraction and analysis in our case was initially performed by one author. Then, another author verified the process of extracting the data for analysis to provide a check on the extracted data. The remaining authors then reviewed the overall analysis.
3. **Research Validity:** This threat is associated with the overall research design. In order to mitigate this threat, we tried to follow a research methodology that is very well formulated and recognized by researchers in the area. We thereby followed the guidelines provided by Petersen et al. [8], which specify a thorough approach to carrying out an SM specifically in the area of software engineering. Its high citation count (more than 3000 citations) ensures that these guidelines are recognized by renowned researchers working in the field. Our study was also inspired by and supported by another tertiary study in the area of software testing [14].

6. Conclusions

Our SM adopted quantitative and qualitative analyses to assemble the research findings in the area of TCS&P. The secondary studies (reviews or surveys conducted on primary studies) found in the field of TCS&P were reviewed systematically to obtain an overall picture of the recent findings in the area. This work primarily presented a general data analysis of the papers, and a detailed data analysis of the included studies corresponding to the formulated RQs. The studies were thoroughly analyzed to gather information, and to comprehensively tabulate results and findings. This study can provide a quick guide to researchers working in the area of TCS&P for clear insight into the trends of work undertaken in the area, in addition to the tools, statistical tests, limitations, and probable prospects.

One goal of our study was the detailed analysis of the 22 selected secondary studies to find common findings and limitations. Although we identified very few common findings, we tried to compile results and generate conclusions from them. These are stated as follows:

- Five of the studies [S1, S2, S13, S15, S16] report that programs downloaded from SIR (Software-artifact Infrastructure Repository) are the most commonly used benchmark programs for the evaluation of the techniques in the area.
    - Four studies [S13, S14, S16, S21] noted that APFD (Average Percentage of Faults Detected) is the most preferred metric in the assessment of techniques in regression TCS&P.
    - Three studies [S7, S18, S19] recorded that GA (genetic algorithm) is the most commonly used approach employed by the researchers working in the area.
    - Four studies [S8, S11, S14, S19] concluded that coverage-based selection and prioritization is the most preferred criterion, whereas S15 reports that there is a paradigm shift from coverage-based to nature-inspired or search-based approaches.
    - S21 reports that a history-based system is the most preferred criterion by researchers working in regression TCS&P.

We also identified some common problems, research gaps, or limitations identified by the analysis of the surveys.

- The studies S1, S10, and S11 observed that regression TCS&P techniques had been applied to limited and usually small test benches (as reported by [S4]).
- The studies S1, S3, and S19 showed that there is a lack of application of regression TCS&P techniques to testing for non-functional requirements.
- Two studies, S7 and S16, reported a lack of common tools used in TCS&P. The studies S10 and S16 state that the prioritization techniques available in the literature do not provide the execution order of the prioritized test cases.
- S19 and S21 highlight the fact that there is a gap in the literature on how to achieve tighter integration between the regression techniques and debugging techniques.
- S7 highlights that most of the regression TCS&P techniques proposed have focused on a particular application domain or are context-specific; hence, assessment of the superiority of a technique over the others is not possible. It also notes the gap in the lack of techniques for mobile applications or web services, and the lack of regression TCS&P techniques for software using GUI or complex domain software.
- S10 stated that most of the existing prioritization techniques are evaluated using APFD. This measurement, however, suffers from a large number of constraints in practice.

The analysis of the studies enables a collective report of the following prospects of future work in the area, which may also overcome limitations and gaps.

- Our analysis indicates a major gap in the usage of statistical tests in regression TCS&P. This opens an opportunity to researchers who could make use of statistical tests for qualitative assessment, validation, and comparison of the techniques used in TCS&P in the future.
- S7, S1, and S3 recommended the use of TCS&P techniques for software domains such as SOA, web services, and model-based testing, and embedded, real-time, and safety-critical software.

- S4, S5, and S18 suggest priority to public datasets rather than proprietary ones.
- S1, S7, S11, and S19 support the usage of multicriteria-based TCS&P techniques. In addition, S1 and S3 recommend the usage of regression TCS&P techniques for model-based testing.
- S10 states that it is necessary to optimize the execution of test cases, mainly due to the cost of individual test cases rather than the interest in total cost.
- S14 recommends prioritizing the order of multiple test suites rather than test cases and suggests measuring the efficiency of TCP techniques based on actual time spent on fault detection.
- S16 suggests the usage of APFDC to explicitly consider test execution time.
- Finally, S21 provides a good presentation of prospects in regression TCS&P.

The significant contributions of the accomplished study are: (1) presents highlights of the publication trends and a list of the popular journals and conferences relevant to the area of TCS&P; (2) enumerates the statistical tests used in the area; (3) comprehensively provides a list of tools used in the area, with the source; (4) lists the test benches and metrics commonly used and the most frequent approaches; and (5) lists the limitations of the research conducted and the prospects for future work in the area.

The results of this SM are subject to the following limitations. Firstly, the secondary studies selected for the study are limited according to the adopted inclusion criteria and the specified RQs. Secondly, the search criteria of the selected papers were limited to only those in the English language. These conditions were essential for the feasibility of our SM. In summary, this SM reports the recent progress in the area of TCS&P to provide insight into previous research and to identify prospects for further work in the area.

Author Contributions: Conceptualization, S.S., N.J.; methodology, B.S., S.M.; validation, S.S., N.J., B.S., S.M., L.F.-S.; formal analysis, S.S., N.J.; writing—original draft preparation, S.S., N.J.; writing—review and editing, B.S., S.M. and L.F.-S.; visualization, S.S., N.J.; supervision, B.S. and S.M.; project administration, B.S.; funding acquisition, L.F.-S. All authors have read and agreed to the published version of the manuscript.

Funding: Funded by the TIFYC Research Group of University of Alcalá, Spain.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rothermel, G.; Untch, R.H.; Chu, C.; Harrold, M.J. Prioritizing test cases for regression testing. *IEEE Trans. Softw. Eng.* **2001**, *27*, 929–948. [CrossRef]
2. Yoo, S.; Harman, M. Regression testing minimization, selection and prioritization: A survey. *Softw. Test. Verif. Reliab.* **2012**, *22*, 67–120. [CrossRef]
3. Rothermel, G.; Harrold, M. Analyzing regression test selection techniques. *IEEE Trans. Softw. Eng.* **1996**, *22*, 529–551. [CrossRef]
4. Elbaum, S.; Malishevsky, A.; Rothermel, G. Prioritizing test cases for regression testing. In Proceedings of the International Symposium on Software Testing and Analysis, New York, NY, USA, 18–22 July 2020; ACM Press: New York, NY, USA, 2020; pp. 102–112.
5. Rothermel, G.; Untch, R.; Chu, C.; Harrold, M. Test case prioritization: An empirical study. In Proceedings of the IEEE International Conference on Software Maintenance, Oxford, UK, 30 August–3 September 1999; pp. 179–188.
6. Kitchenham, B. *Procedures for Undertaking Systematic Review*; TR/SE-0401 and 0400011T.1; Keele University: Keele, UK; National ICT Australia Ltd.: Sydney, Australia, 2004.
7. Kitchenham, B.A.; Brereton, P.; Turner, M.; Niazi, M.K.; Linkman, S.; Pretorius, R.; Budgen, D. Refining the systematic literature review process—Two participant-observer case studies. *Empir. Softw. Eng.* **2010**, *15*, 618–653. [CrossRef]
8. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic Mapping Studies in Software Engineering. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE), Bari, Italy, 26–27 June 2008; pp. 1–10.
9. Durelli, V.H.S.; Durelli, R.S.; Borges, S.S.; Endo, A.T.; Eler, M.; Dias, D.R.C.; Guimaraes, M.P. Machine Learning Applied to Software Testing: A Systematic Mapping Study. *IEEE Trans. Reliab.* **2019**, *68*, 1189–1212. [CrossRef]
10. Kong, P.; Li, L.; Gao, J.; Liu, K.; Bissyande, T.F.; Klein, J. Automated Testing of Android Apps: A Systematic Literature Review. *IEEE Trans. Reliab.* **2019**, *68*, 45–66. [CrossRef]
11. Galster, M.; Weyns, D.; Tofan, D.; Michalik, B.; Avgeriou, P. Variability in Software Systems—A Systematic Literature Review. *IEEE Trans. Softw. Eng.* **2014**, *40*, 282–306. [CrossRef]

12. Jiang, B.; Zhang, Y.; Chan, W.K.; Zhang, Z. A Systematic Study on Factors Impacting GUI Traversal-Based Test Case Generation Techniques for Android Applications. *IEEE Trans. Reliab.* **2019**, *68*, 913–926. [CrossRef]
13. Bluemke, I.; Malanowska, A. Software Testing Effort Estimation and Related Problems: A Systematic Literature Review. *ACM Comput. Surv.* **2021**, *54*, 1–38. [CrossRef]
14. Garousi, V.; Mäntylä, M. A systematic literature review of literature reviews in software testing. *Inf. Softw. Technol.* **2016**, *80*, 195–216. [CrossRef]
15. Khan, M.U.; Sherin, S.; Iqbal, M.Z.; Zahid, R. Landscaping systematic mapping studies in software engineering: A tertiary study. *J. Syst. Softw.* **2019**, *149*, 396–436. [CrossRef]
16. Wohlin, C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; pp. 1–10.
17. Catal, C. On the application of genetic algorithms for test case prioritization: A systematic literature review. In Proceedings of the 2nd International Workshop on Evidential Assessment of Software Technologies, Lund, Sweden, 22 September 2012.
18. Mukherjee, R.; Patnaik, K.S. A survey on different approaches for software test case prioritization. *J. King Saud Univ.-Comput. Inf. Sci.* **2018**, *33*, 1041–1054. [CrossRef]
19. Keele, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Version 2.3 EBSE Technical Report EBSE-2007-01; Keele University: Keele, UK, 2007.
20. Engström, E.; Runeson, P.; Skoglund, M. A systematic review on regression test selection techniques. *Inf. Softw. Technol.* **2010**, *52*, 14–30. [CrossRef]
21. Suri, B.; Singhal, S. Implementing ant colony optimization for test case selection and prioritization. *Int. J. Comput. Sci. Eng.* **2011**, *3*, 1924–1932.
22. IEEE Explore. Available online: https://ieeexplore.ieee.org/Xplore/home.jsp (accessed on 31 May 2020).
23. ACM Digital Library. Available online: https://dl.acm.org/ (accessed on 31 May 2020).
24. Elsevier Science Direct Digital Library. Available online: https://www.sciencedirect.com/ (accessed on 31 May 2020).
25. Online Wiley Library. Available online: https://www.onlinelibrary.wiley.com/ (accessed on 31 May 2020).
26. Springer Link. Available online: https://link.springer.com/ (accessed on 31 May 2020).
27. Google Scholar. Available online: https://scholar.google.co.in/ (accessed on 31 May 2020).
28. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.; Group, T.P. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Med.* **2009**, *7*, 6.
29. Garousi, V.; Felderer, M.; Mäntylä, M.V. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Inf. Softw. Technol.* **2019**, *106*, 101–121. [CrossRef]
30. Jatain, A.; Sharma, G. A Systematic Review of Techniques for Test Case Prioritization. *Int. J. Comput. Appl.* **2013**, *68*, 38–42. [CrossRef]
31. Mohanty, S.; Acharya, A.A.; Mohapatra, D.P. A survey on model based test case prioritization. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *2*, 1042–1047.
32. Sultan, Z.; Abbas, R.; Bhatti, S.N.; Shah, S.A.A. Analytical Review on Test Cases Prioritization Techniques: An Empirical Study. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 293–302. [CrossRef]
33. Monica, A.S. Test Case Prioritization: A Review. *Int. J. Eng. Res. Technol.* **2014**, *3*, 5.
34. Sagar, G.P.; Prasad, P.V.R.D. A Survey on Test Case Prioritization Techniques for Regression Testing. *Indian J. Sci. Technol.* **2017**, *10*, 1–6. [CrossRef]
35. Ramirez, A.; Romero, J.R.; Ventura, S. A survey of many-objective optimisation in search-based software engineering. *J. Sys. Softw.* **2019**, *149*, 382–395. [CrossRef]
36. Thakur, P.; Varma, T. A survey on test case selection using optimization techniques in software testing. *Int. J. Innov. Sci. Eng. Technol. IJISET* **2015**, *2*, 593–596.
37. Narula, S. Review Paper on Test Case Selection. *IJCSET* **2016**, *6*, 126–128.
38. Rajal, J.S.; Sharma, S. A review on various techniques for regression testing and test case prioritization. *Int. J. Comput. Appl.* **2015**, *116*, 16.
39. Rani, M.; Singh, A. Review of Regression Test Case Selection. *Int. J. Eng. Res. Technol.* **2014**, *3*, 5.
40. Farooq, U.; Aman, H.; Mustapha, A.; Saringat, Z. A review of object-oriented approach for test case prioritization. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *16*, 429–434. [CrossRef]
41. Catal, C. The Ten Best Practices for Test Case Prioritization. In Proceedings of the Robotics, Singapore, 13 September 2012; pp. 452–459.
42. Engström, E.; Skoglund, M.; Runeson, P. Empirical evaluations of regression test selection techniques: A systematic review. In Proceedings of the (ESEM'08) ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Kaiserslautern, Germany, 9–10 October 2008.
43. Singh, Y.; Kaur, A.; Suri, B.; Singhal, S. Systematic literature review on regression test prioritization techniques. *Informatica* **2012**, *36*, 379–408.
44. Jatana, N.; Suri, B.; Rani, S. Systematic literature review on search based mutation testing. *e-Inf. Softw. Eng. J.* **2017**, *11*, 59–76.

45. Hasnain, M.; Ghani, I.; Pasha, M.F.; Jeong, S.-R. Ontology-Based Regression Testing: A Systematic Literature Review. *Appl. Sci.* **2021**, *11*, 9709. [CrossRef]
46. Catal, C.; Mishra, D. Test case prioritization: A systematic mapping study. *Softw. Qual. J.* **2013**, *21*, 445–478. [CrossRef]
47. Rosero, R.H.; Gómez, O.S.; Rodríguez, G. 15 Years of Software Regression Testing Techniques—A Survey. *Int. J. Softw. Eng. Knowl. Eng.* **2016**, *26*, 675–689. [CrossRef]
48. Kumar, S.; Rajkumar. Test case prioritization techniques for software product line: A survey. In Proceedings of the 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 29–30 April 2016; pp. 884–889.
49. Khatibsyarbini, M.; Isa, M.A.; Jawawi, D.N.; Tumeng, R. Test case prioritization approaches in regression testing: A systematic literature review. *Inf. Softw. Technol.* **2018**, *93*, 74–93. [CrossRef]
50. Dahiya, O.; Solanki, K. A systematic literature study of regression test case prioritization approaches. *Int. J. Eng. Technol.* **2018**, *7*, 2184–2191. [CrossRef]
51. Bajaj, A.; Sangwan, O.P. A Systematic Literature Review of Test Case Prioritization Using Genetic Algorithms. *IEEE Access* **2019**, *7*, 126355–126375. [CrossRef]
52. Biswas, S.; Mall, R.; Satpathy, M.; Sukumaran, S. Regression test selection techniques: A survey. *Informatica* **2011**, *35*, 289–321.
53. Narciso, E.N.; Delamaro, M.E.; Nunes, F.D.L.D.S. Test Case Selection: A Systematic Literature Review. *Int. J. Softw. Eng. Knowl. Eng.* **2014**, *24*, 653–676. [CrossRef]
54. Kazmi, R.; Jawawi, D.; Mohamad, R.; Ghani, E. Effective regression test case selection: A systematic literature review. *ACM Comput. Surv.* **2017**, *50*, 1–32. [CrossRef]
55. Junior, H.C.; Ara'ujo, M.; David, J.; Braga, R.; Campos, F.; Stroele, V. Test case prioritization: A systematic review and mapping of the literature. In Proceedings of the SBES'17 Brazilian Symposium on Software Engineering, Fortaleza, CE, Brazil, 20–22 September 2017.
56. Lou, Y.; Chen, J.; Zhang, L.; Hao, D. A Survey on Regression Test-Case Prioritization. *Adv. Comput.* **2019**, *113*, 1–46.
57. Alkawaz, M.H.; Silvarajoo, A. A Survey on Test Case Prioritization and Optimization Techniques in Software Regression Testing. In Proceedings of the 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, 13–14 December 2019; pp. 59–64.
58. Gupta, N.; Sharma, A.; Pachariya, M.K.; Kumar, M. An Insight Into Test Case Optimization: Ideas and Trends With Future Perspectives. *IEEE Access* **2019**, *7*, 22310–22327. [CrossRef]
59. Paygude, P.; Joshi, S.D. Use of Evolutionary Algorithm in Regression Test Case Prioritization: A Review. In *Lecture Notes on Data Engineering and Communications Technologies, LNDECT-31*; Springer: Cham, Switzerland, 2020; pp. 56–66.
60. Lima, J.P.; Virgilio, S. Test Case Prioritization in Continuous Integration environments: A systematic mapping study. *Inf. Softw. Technol.* **2020**, *121*, 106268. [CrossRef]
61. Castro-Cabrera, M.D.; García-Dominguez, A.; Medina-Bulo, I. Trends in prioritization of test cases: 2017–2019. In Proceedings of the ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020.
62. Harrold, M.; Souffa, M. An incremental approach to unit testing during maintenance. In Proceedings of the Conference on Software Maintenance, Scottsdale, AZ, USA, 24–27 October 1988; pp. 362–367.
63. Rothermel, G.; Elbaum, S.; Malishevsky, A.; Kallakuri, P.; Qiu, X. On test suite composition and cost-effective regression testing. *ACM Trans. Softw. Eng. Methodol.* **2004**, *13*, 277–331. [CrossRef]
64. Elbaum, S.; Malishevsky, A.; Rothermel, G. Test case prioritization: A family of empirical studies. *IEEE Trans. Softw. Eng.* **2002**, *28*, 159–182. [CrossRef]
65. Belli, F.; Eminov, M.; Gökçe, N. Coverage-oriented, prioritized testing–a fuzzy clustering approach and case study. In Proceedings of the Latin-American Symposium on Dependable Computing, Morella, Mexico, 26–28 September 2007.
66. Yu, L.; Xu, L.; Tsai, W.-T. Time-Constrained Test Selection for Regression Testing. In Proceedings of the International Conference on Advanced Data Mining and Applications, Chongqing, China, 19–21 November 2010; Springer: Singapore, 2010; Volume 6441, pp. 221–232.
67. Bian, Y.; Kirbas, S.; Harman, M.; Jia, Y.; Li, Z. Regression Test Case Prioritisation for Guava. In *Lecture Notes in Computer Science*; Springer: Singapore, 2015; pp. 221–227.
68. Pradhan, D.; Wang, S.; Ali, S.; Yue, T.; Liaaen, M. CBGA-ES+: A Cluster-Based Genetic Algorithm with Non-Dominated Elitist Selection for Supporting Multi-Objective Test Optimization. *IEEE Trans. Softw. Eng.* **2021**, *47*, 86–107. [CrossRef]
69. Pradhan, D.; Wang, S.; Ali, S.; Yue, T.; Liaaen, M. STIPI: Using Search to Prioritize Test Cases Based on Multi-objectives Derived from Industrial Practice. In Proceedings of the International Conference on Testing, Software and Systems (ICTSS 2016), Graz, Austria, 17–19 October 2016; pp. 172–190.
70. Arrieta, A.; Wang, S.; Markiegi, U.; Sagardui, G.; Etxeberria, L. Employing Multi-Objective Search to Enhance Reactive Test Case Generation and Prioritization for Testing Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1055–1066. [CrossRef]
71. Epitropakis, M.G.; Yoo, S.; Harman, M.; Burke, E.K. Empirical evaluation of pareto efficient multi-objective regression test case prioritisation. In Proceedings of the 2015 International Symposium on Software Testing and Analysis, Baltimore, MD, USA, 13–17 July 2015; ACM Press: New York, NY, USA, 2015.

72. Nucci, D.D.; Panichella, A.; Zaidman, A.; Lucia, A.D. A test case prioritization genetic algorithm guided by the hypervolume indicator. *IEEE Trans. Softw. Eng.* **2018**, *46*, 674–696. [CrossRef]
73. Haghighatkhah, A.; Mäntylä, M.; Oivo, M.; Kuvaja, P. Test prioritization in continuous integration environments. *J. Syst. Softw.* **2018**, *146*, 80–98. [CrossRef]
74. Yu, Z.; Fahid, F.; Menzies, T.; Rothermel, G.; Patrick, K.; Cherian, S. TERMINATOR: Better automated UI test case prioritization. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia, 26–30 August 2019; ACM Press: New York, NY, USA, 2019.
75. Marijan, D.; Gotlieb, A.; Liaaen, M. A learning algorithm for optimizing continuous integration development and testing practice. *Softw. Pract. Exp.* **2019**, *49*, 192–213. [CrossRef]
76. Pradhan, D.; Wang, S.; Ali, S.; Yue, T.; Liaaen, M. Employing rule mining and multi-objective search for dynamic test case prioritization. *J. Syst. Softw.* **2019**, *153*, 86–104. [CrossRef]
77. Wang, S.; Ali, S.; Yue, T.; Bakkeli, Ø; Liaaen, M. Enhancing test case prioritization in an industrial setting with resource awareness and multi-objective search. In Proceedings of the 38th International Conference on Software Engineering Companion, Austin, TX, USA, 14–22 May 2016; ACM Press: New York, NY, USA, 2016; pp. 182–191.
78. Wang, S.; Buchmann, D.; Ali, S.; Gotlieb, A.; Pradhan, D.; Liaaen, M. Multi-objective test prioritization in software product line testing: An industrial case study. In Proceedings of the 18th International Software Product Line Conference, Florence, Italy, 15–19 September 2014.
79. Arrieta, A.; Wang, S.; Sagardui, G.; Etxeberria, L. Test Case Prioritization of Configurable Cyber-Physical Systems with Weight-Based Search Algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference, Denver, CO, USA, 20–24 July 2016; ACM Press: New York, NY, USA, 2016; pp. 1053–1060.
80. Marchetto, A.; Islam, M.; Asghar, W.; Susi, A.; Scanniello, G. A Multi-Objective Technique to Prioritize Test Cases. *IEEE Trans. Softw. Eng.* **2015**, *42*, 918–940. [CrossRef]
81. Li, Z.; Harman, M.; Hierons, R.M. Search algorithms for regression for regression test case prioritization. *IEEE Trans. Softw. Eng.* **2007**, *33*, 225–237. [CrossRef]
82. Jiang, B.; Zhang, Z.; Tse, T.H.; Chen, T.Y. How Well Do Test Case Prioritization Techniques Support Statistical Fault Localization. In Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, WA, USA, 20–24 July 2009; pp. 99–106.
83. Jiang, B.; Zhang, Z.; Chan, W.; Tse, T.; Chen, T.Y. How well does test case prioritization integrate with statistical fault localization? *Inf. Softw. Technol.* **2012**, *54*, 739–758. [CrossRef]
84. Jiang, B.; Chan, W. On the Integration of Test Adequacy, Test Case Prioritization, and Statistical Fault Localization. In Proceedings of the 2010 10th International Conference on Quality Software, Zhangjiajie, China, 14–15 July 2010; pp. 377–384.
85. Jiang, B.; Chan, W.K.; Tse, T. On Practical Adequate Test Suites for Integrated Test Case Prioritization and Fault Localization. In Proceedings of the 2011 11th International Conference on Quality Software, Madrid, Spain, 13–14 July 2011; pp. 21–30.
86. Abdullah, A.; Schmidt, H.W.; Spichkova, M.; Liu, H. Monitoring Informed Testing for IoT. In Proceedings of the 2018 25th Australasian Software Engineering Conference (ASWEC), Adelaide, SA, Australia, 26–30 November 2018; pp. 91–95.
87. Chen, J.; Lou, Y.; Zhang, L.; Zhou, J.; Wang, X.; Hao, D.; Zhang, L. Optimizing test prioritization via test distribution analysis. In Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2018; pp. 656–667.
88. Hartmann, J.; Robson, D. Approaches to regression testing. In Proceedings of the Conference on Software Maintenance, Scottsdale, AZ, USA, 24–27 October 1988.
89. Jatana, N.; Suri, B.; Misra, S.; Kumar, P.; Choudhury, A.R. Particle Swarm Based Evolution and Generation of Test Data Using Mutation Testing. In Proceedings of the International Conference on Computational Science and Its Applications, Beijing, China, 4–7 July 2016; Springer: Singapore, 2016; pp. 585–594.
90. Jatana, N.; Suri, B. An Improved Crow Search Algorithm for Test Data Generation Using Search-Based Mutation Testing. *Neural Process. Lett.* **2020**, *52*, 767–784. [CrossRef]
91. Jatana, N.; Suri, B. Particle Swarm and Genetic Algorithm applied to mutation testing for test data generation: A comparative evaluation. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 514–521. [CrossRef]
92. Walcott, K.R.; Soffa, M.L.; Kapfhammer, G.M.; Roos, R.S. TimeAware test suite prioritization. In Proceedings of the 2006 International Symposium on Software Testing and Analysis-ISSTA'06, Portland, ME, USA, 17–20 July 2006; ACM: New York, NY, USA, 2006; pp. 1–12.
93. Li, S.; Bian, N.; Chen, Z.; You, D.; He, Y. A Simulation Study on Some Search Algorithms for Regression Test Case Prioritization. In Proceedings of the 2010 10th International Conference on Quality Software, Zhangjiajie, China, 14–15 July 2010; pp. 72–81.
94. Huang, Y.-C.; Huang, C.-Y.; Chang, J.-R.; Chen, T.-Y. Design and Analysis of Cost-Cognizant Test Case Prioritization Using Genetic Algorithm with Test History. In Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference, Seoul, Korea, 19–23 July 2010; pp. 413–418.
95. Last, M.; Eyal, S.; Kandel, A. Effective Black-Box Testing with Genetic Algorithms. In Proceedings of the Haifa Verification Conference, Haifa, Israel, 13–16 November 2005; Springer: Singapore, 2006; Volume 3875, pp. 134–148.

96. Conrad, A.P.; Roos, R.S.; Kapfhammer, G.M. Empirically studying the role of selection operators duringsearch-based test suite prioritization. In Proceedings of the 12th Annual Conference on Cyber and Information Security Research, Oak Ridge, TN, USA, 4 –6 April 2017; ACM Press: New York, NY, USA, 2010; pp. 1373–1380.
97. Sabharwal, S.; Sibal, R.; Sharma, C. Prioritization of test case scenarios derived from activity diagram using genetic algorithm. In Proceedings of the 2010 International Conference on Computer and Communication Technology (ICCCT), Allahabad, India, 17–19 September 2010; pp. 481–485.
98. Smith, A.M.; Kapfhammer, G.M. An empirical study of incorporating cost into test suite reduction and prioritization. In Proceedings of the 2009 ACM Symposium on Applied Computing-SAC '09, Honolulu, HI, USA, 8–12 March 2009; ACM Press: New York, NY, USA, 2009.
99. Hemmati, H.; Briand, L.; Arcuri, A.; Ali, S. An enhanced test case selection approach for model-based testing: An industrial case study. In Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering, Santa Fe, NM, USA, 7–11 November 2010.
100. Hemmati, H.; Briand, L. An Industrial Investigation of Similarity Measures for Model-Based Test Case Selection. In Proceedings of the 2010 IEEE 21st International Symposium on Software Reliability Engineering, San Jose, CA, USA, 1–4 November 2010; pp. 141–150.
101. Arcuri, A.; Iqbal, M.Z.; Briand, L. Black-Box System Testing of Real-Time Embedded Systems Using Random and Search-Based Testing. In *Lecture Notes in Computer Science*; Springer: Singapore, 2010; pp. 95–110.
102. Hemmati, H.; Arcuri, A.; Briand, L. Reducing the Cost of Model-Based Testing through Test Case Diversity. In Proceedings of the IFIP International Conference on Testing Software and Systems, Natal, Brazil, 8–10 November 2010.
103. Hemmati, H.; Arcuri, A.; Briand, L. Empirical Investigation of the Effects of Test Suite Properties on Similarity-Based Test Case Selection. In Proceedings of the 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation, Berlin, Germany, 21–25 March 2011; pp. 327–336.
104. Parsa, S.; Khalilian, A. A Bi-objective Model Inspired Greedy Algorithm for Test Suite Minimization. In Proceedings of the Computer Vision, Jeju Island, Korea, 10–12 December 2009; Springer: Singapore, 2009; pp. 208–215.
105. Biswas, S.; Mall, R.; Satpathy, M.; Sukumaran, S. A model-based regression test selection approach for embedded applications. *ACM SIGSOFT Softw. Eng. Notes* **2009**, *34*, 1–9. [CrossRef]
106. Maia, C.L.B.; de Carmo, R.A.F.; de Freitas, F.G.; de Camp, G.A.L.; de Souza, J.T. Automated test case prioritization with reactive GRASP. *Adv. Softw. Eng.* **2010**, *2*, 1–13. [CrossRef]
107. Mirarab, S.; Akhlaghi, S.; Tahvildari, L. Size-Constrained Regression Test Case Selection Using Multicriteria Optimization. *IEEE Trans. Softw. Eng.* **2011**, *38*, 936–956. [CrossRef]
108. Oliveira, A.A.L.D.; Camilo-Junior, C.G.; Vincenzi, A.M.R. A Coevolutionary Algorithm to Automatic Test Case Selection and Mutant in Mutation Testing. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013.
109. Panichella, A.; Oliveto, R.; Di Penta, M.; De Lucia, A. Improving Multi-Objective Test Case Selection by Injecting Diversity in Genetic Algorithms. *IEEE Trans. Softw. Eng.* **2015**, *41*, 358–383. [CrossRef]
110. Maheswari, R.; Mala, D. Combined genetic and simulated annealing approach for test case prioritization. *Indian J. Sci. Technol.* **2015**, *8*, 35. [CrossRef]
111. Lou, Y.; Hao, D.; Zhang, L. Mutation-based test-case prioritization in software evolution. In Proceedings of the 2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE), Gaithersbury, MD, USA, 2–5 November 2015; pp. 46–57.
112. Yuan, F.; Bian, Y.; Li, Z.; Zhao, R. Epistatic Genetic Algorithm for Test Case Prioritization. In *Lecture Notes in Computer Science*; Springer: Singapore, 2015; pp. 109–124.
113. Kaur, A.; Goyal, S. A genetic algorithm for fault based regression test case prioritization. *Int. J. Comput. Appl.* **2011**, *32*, 8.
114. Jun, W.; Yan, Z.; Chen, J. Test Case Prioritization Technique Based on Genetic Algorithm. *Int. Comput. Inf.* **2011**, 173–175.
115. Deb, K.; Pratab, S.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NGSA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
116. Di Nardo, D.; Alshahwan, N.; Briand, L.; Labiche, Y. Cov-erage-based regression test case selection, minimization and prioritization: A case study on an industrial system. *Softw. Test. Ver-If. Reliab.* **2015**, *25*, 371–396. [CrossRef]
117. Mishra, D.B.; Mishra, R.; Acharya, A.A.; Das, K.N. Test Data Generation for Mutation Testing Using Genetic Algorithm. *Adv. Int. Syst. Comput.* **2019**, 857–867.
118. Mishra, D.B.; Mishra, R.; Acharya, A.A.; Das, K.N. Test Case Optimization and Prioritization Based on Multi-objective Genetic Algorithm. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2018; pp. 371–381.
119. Masri, W.; El-Ghali, M. Test case filtering and prioritization based on coverage of combinations of program elements. In Proceedings of the Seventh International Workshop on Dynamic Analysis, Chicago, IL, USA, 26 October 2009.
120. Huang, Y.-C.; Peng, K.-L.; Huang, C.-Y. A history-based cost-cognizant test case prioritization technique in regression testing. *J. Syst. Softw.* **2012**, *85*, 626–637. [CrossRef]
121. Ahmed, A.A.; Shaheen, M.; Kosba, E.; Shaheen, M. Software testing suite prioritization using multi-criteria fitness function. In Proceedings of the 2012 22nd International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, 13–15 October 2012; pp. 160–166.

122. Parejo, J.A.; Sánchez, A.B.; Segura, S.; Ruiz-Cortés, A.; Lopez-Herrejon, R.E.; Egyed, A. Multi-objective test case prioritization in highly configurable systems: A case study. *J. Syst. Softw.* **2016**, *122*, 287–310. [CrossRef]
123. Khanna, M.; Chauhan, N.; Sharma, D.; Toofani, A.; Chaudhary, A. Search for Prioritized Test Cases in Multi-Objective Environment During Web Application Testing. *Arab. J. Sci. Eng.* **2018**, *43*, 4179–4201. [CrossRef]
124. Raju, S.; Uma, G.V. Factors oriented test case prioritization technique in regression testing using genetic algorithm. *Eur. J. Sci. Res.* **2012**, *74*, 389–402.
125. Khurana, N.; Chillar, R. Test Case Generation and Optimization using UML Models and Genetic Algorithm. *Procedia Comput. Sci.* **2015**, *57*, 996–1004. [CrossRef]
126. Mahajan, S.; Joshi, S.D.; Khanaa, V. Component-Based Software System Test Case Prioritization with Genetic Algorithm Decoding Technique Using Java Platform. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Washington, DC, USA, 26–27 February 2015; pp. 847–851.
127. Nagar, R.; Kumar, A.; Kumar, S.; Baghel, A.S. Implementing test case selection and reduction techniques using meta-heuristics. In Proceedings of the 2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence), Noida, India, 25–26 September; pp. 837–842.
128. Islam, M.; Marchetto, A.; Susi, A.; Kessler, F.B.; Scanniello, G.; Islam, M. MOTCP: A tool for the prioritization of test cases based on a sorting genetic algorithm and Latent Semantic Indexing. In Proceedings of the 2012 28th IEEE International Conference on Software Maintenance (ICSM), Trento, Italy, 23–28 September 2012; pp. 654–657.
129. Sabharwal, S.; Sibal, R.; Sharma, C. A genetic algorithm based approach for prioritization of test case scenarios in static testing. In Proceedings of the 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), Allahabad, India, 15–17 September 2011; pp. 304–309.
130. Yoo, S.; Harman, M. Pareto efficient multi-objective test case selection. In Proceedings of the 2007 International Symposium on Software Testing and Analysis-ISSTA '07, London UK, 9–12 July 2007; pp. 140–150.
131. de Souza, L.S.; de Miranda, P.B.; Prudencio, R.B.; Barros, F.D.A. A multi-objective particle swarm optimization for test case selection based on functional requirements coverage and execution effort. In Proceedings of the 23rd International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, 7–9 November 2011; pp. 245–252.
132. De Souza, L.S.; Prudêncio, R.B.C.; Barros, F.D.A. A Hybrid Binary Multi-objective Particle Swarm Optimization with Local Search for Test Case Selection. In Proceedings of the 2014 Brazilian Conference on Intelligent Systems, São Paulo, Brazil, 18–22 October 2014; pp. 414–419.
133. De Souza, L.S.; Prudencio, R.; Barros, F.D.A. A comparison study of binary multi-objective Particle Swarm Optimization approaches for test case selection. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 2164–2171.
134. De Souza, L.S.; Prudêncio, R.B.; Barros, F.D.A.; Aranha, E.H.D.S. Search based constrained test case selection using execution effort. *Expert Syst. Appl.* **2013**, *40*, 4887–4896. [CrossRef]
135. Hla, K.; Choi, Y.; Park, J.S. Applying Particle Swarm Optimization to Prioritizing Test Cases for Embedded Real Time Software Retesting. In Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops, Washington, DC, USA, 8 – 11 July 2008; pp. 527–532.
136. Maheswari, R.U.; Jeyamala, D. A novel approach for test case prioritization. In Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research, Enathi, India, 26–28 December 2013; pp. 1–5.
137. Tyagi, M.; Malhotra, S. Test case prioritization using multi objective particle swarm optimizer. In Proceedings of the International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, India, 12–13 July 2014.
138. Solanki, K.; Singh, Y.; Dalal, S.; Srivastava, P. Test case prioritization: An approach based on modified ant colony optimization. In *Emerging Research in Computing, Information, Communication and Applications*; Springer: Singapore, 2016.
139. Gao, D.; Guo, X.; Zhao, L. Test case prioritization for regression testing based on ant colony optimization. In Proceedings of the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 September 2015; pp. 275–279.
140. Noguchi, T.; Washizaki, H.; Fukazawa, Y.; Sato, A.; Ota, K. History-Based Test Case Prioritization for Black Box Testing Using Ant Colony Optimization. In Proceedings of the 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), Graz, Austria, 13–17 April 2015; pp. 1–2.
141. Suri, B.; Mangal, I.; Srivastava, V. Regression Test Suite Reduction using an Hybrid Technique Based on BCO And Genetic Algorithm. *Int. J. Comput. Sci. Inform.* **2013**, *2*, 26–33. [CrossRef]
142. Suri, B.; Singhal, S. Literature survey of Ant Colony Optimization in software testing. In Proceedings of the 2012 CSI Sixth International Conference on Software Engineering (CONSEG), Indore, India, 5–7 September 2012; pp. 1–7.
143. Vescan, A.; Pintea, C.-M.; Pop, P.C. Solving the Test Case Prioritization Problem with Secure Features Using Ant Colony System. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2019; pp. 67–76.
144. Ansari, A.; Khan, A.; Mukadam, K. Optimized regression test using test case prioritization. *Procedia Comput. Sci.* **2016**, *79*, 152–160. [CrossRef]
145. Ledru, Y.; Petrenko, A.; Boroday, S.; Mandran, N. Prioritizing test cases with string distances. *Autom. Softw. Eng.* **2011**, *19*, 65–95. [CrossRef]

146. Baudry, B.; Fleurey, F.; Jézéquel, J.-M.; Traon, Y.L. Automatic Test Cases Optimization: A Bacteriological Algorithm. *IEEE Softw.* **2005**, *22*, 76–82. [CrossRef]
147. Joseph, A.K.; Radhamani, G. Fuzzy C Means (FCM) Clustering Based Hybrid Swarm Intelligence Algorithm for Test Case Optimization. *Res. J. Appl. Sci. Eng. Technol.* **2014**, *8*, 76–82. [CrossRef]
148. Folino, G.; Forestiero, A.; Spezzano, G. A Jxta Based Asynchronous Peer-to-Peer Implementation of Genetic Programming. *J. Softw.* **2006**, *1*, 12–23. [CrossRef]
149. Sangeeta; Sitender. Comprehensive analysis of hybrid nature-inspired algorithms for software reliability analysis. *J. Stat. Manag. Syst.* **2020**, *23*, 1037–1048. [CrossRef]
150. Ampatzoglou, A.; Bibi, S.; Avgeriou, P.; Ver-Beek, M.; Chatzigeorgiou, A. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Inf. Softw. Technol.* **2019**, *106*, 201–230. [CrossRef]