30 years of automated GUI testing: a bibliometric analysis

Olivia Rodríguez-Valdés[1], Tanja E.J. Vos[1,2][0000−0002−6003−9113], Pekka Aho[1], and Beatriz Marín[2][0000−0001−8025−0023]

[1] Open Universiteit, The Netherlands
[2] Universitat Politècnica de València, Spain

Abstract. *Context*: Over the last 30 years, GUIs have changed considerably, becoming everyday part of our lives through smart phones and other devices. More complex GUIs and multitude of platforms have increased the challenges when testing software through the GUI. *Objective*: To visualise how the field of automated GUI testing has evolved by studying the growth of the field; types of publications; influential events, papers and authors; collaboration among authors; and trends on GUI testing. *Method*: To conduct a bibliometric analysis of automated GUI testing by performing a systematic search of primary studies in Scopus from 1990 to 2020. *Results*: 744 publications were selected as primary studies. The majority of them were conference papers, the most cited paper was published on 2013, and the most published author has 53 papers. *Conclusions*: Automated GUI testing has continuously grown. Keywords show that testing applied to mobile interfaces will be the trend in next years, along with the integration of Artificial Intelligence and automated exploration techniques.

Keywords: Automated testing · Graphical user interface · Bibliometric analysis · Secondary study

1 Introduction

A Graphical User Interface (GUI) is a human-computer interface that includes graphical elements commonly called widgets, for example buttons, menus, textboxes, scrollbars, and icons. The first GUIs were developed in early 70s to improve the usability of operating software systems. Before GUIs, the only way to interact with the systems was through CLIs (Command Line Interfaces). GUIs allow end-users to interact with the system functionality more easily, and provide output and feedback in a graphical form based on the actions of end-users.

In GUI testing, the system is tested through the elements of the GUI and their properties. To do that, test sequences are comprised of actions (such as click, type, drag and drop) and the corresponding test oracles to check the state of the system after the execution of the actions. GUI testing is of paramount importance since it allows testing systems from the end-user's point of view.

Automated GUI testing has been researched for over three decades. The first papers on this topic are from the late 80s [8]. Automating GUI testing faces

several challenges. GUIs change frequently during the life cycle of a system (e.g., controls are removed or re-positioned, new controls are added, etc.). This has severe implications for the practice of automated testing: instead of creating new test cases to find new faults, testers struggle with repairing the old ones in order to maintain the test suite and adapt it to the changed GUI layout.

Over the last 30 years, in accordance to the evolution of programming languages from 3rd generation to the 5th, GUIs have evolved with better graphics, becoming more realistic and their graphical components more skeuomorphic. A lot of desktop applications have been replaced by web applications, representing challenges to testing in form of distributed services and systems of systems. With the rise of smartphones and other portable devices, new testing challenges arose due to a much smaller screen and more complex interactions. Mobile GUIs have to be more simple (with less elements), but at the same time, the complexity on the functionality of applications is growing.

To cover the state of the art of GUI testing, Bao et al. [1] conducted a mapping study from 1991 to 2011 that included 136 publications. The field has been growing considerably since then. To understand the community, publication patterns and trends in automated GUI testing, this paper presents a bibliometric study [17]. As far as we know, this paper presents the first bibliometric analysis on this field over the last 30 years. The main contributions are to:

1. Provide facts about the size and growth of the field.
2. Indicate the type of publications and their rankings, including most cited papers, most prolific authors, and most influential journals and conferences.
3. Show the distribution of the publications among the available sources and over the years using a spectroscopy.
4. Present and discuss the productivity and the level of collaboration among researchers in the literature.
5. Use the bibliometric laws of Bradford [3] to know the most influencing journals, and of Lotka [11] to evaluate scientific productivity of authors.
6. Show the evolution of the major research topics in the field by analysing the keywords used by the authors.
7. Make a public repository for Automated GUI testing.

The rest of the paper is organized as follows. Section 2 defines the scope of the study. Section 3 presents the methodology for the bibliometric analysis, and the results are presented in Section 4. Section 5 presents the main conclusions.

2 Scope: automated GUI testing

To make the scope of the study clear, this section explains the definition of *automated GUI testing* that was used to decide which papers should be included in this bibliometric study. Executing sequences of events on the GUI widgets of a system under test (SUT) and checking test oracles is called *GUI testing*. The goal of executing these tests – like in any other type of testing – is finding failures, reducing risks, and analysing and increasing the quality of the SUT.

term	family
automated	automated OR automatic OR automatically OR automation OR automating OR automate OR generation OR generate OR generating OR generator
GUI	GUI OR UI OR "graphical user interface"
testing	testing OR test OR tested

Table 1: Family of words for the search string

Evidently, it is possible to **automate the execution** of these test sequences and call it *automated GUI testing*. However, more activities related to GUI testing can be automated. To be able to define clear inclusion/exclusion criteria for the papers of this bibliometric analysis, the definition of automated GUI testing was refined to include also other activities of GUI testing, as follows:

Automating the creation of test sequences: Test sequences in GUI testing consist of sequences of GUI actions/events on widgets together with input values. Test sequences are made to cover some test goal of the SUT (e.g., checking some specific functionality or finding a failure). Test sequence defines which path through the SUT should be taken (which *states* should be visited), i.e., *what* actions will be executed, and in which *order*.

Automating the definition or checking of the oracles: Oracles [2] are procedures that distinguish between the correct and incorrect behavior of the SUT. Since test cases in GUI testing are sequences, we can check the oracles after each action (test step) during the execution (online oracle), just one time at the end of each sequence, or analyse the results after the execution (offline oracle). Test oracle automation is important for removing the current bottleneck that inhibits greater overall test automation [2]. Without test oracle automation, a human has to determine whether observed behaviour is correct.

Automating the analysis of test results: This consists of analysing, for example, the failures that were found in a specific SUT, or evaluating the quality of the test cases that were executed, using a set of defined metrics.

When at least one of these activities is automated, it will be considered *automated GUI testing* (even when the test execution is done manually), and therefore, the corresponding papers will be included in this study.

3 Methodology

In this study, we follow the workflow for bibliometric analysis defined in [6].

3.1 Data retrieval

We used Scopus for the search process since it is the largest database of peer-reviewed literature with the largest coverage in comparison to other scientific repositories, such as WoS [20]. The search string evolved from the initial terms "Automated GUI testing" – to reduce the probability of missing relevant papers, a family of words was derived from every term (see Table 1).

The complete search query is shown in Figure 1. As can be seen in lines 1-3, the terms must appear in the article's title, abstract or keywords since the

```
1:  (TITLE-ABS-KEY((Automated W/5 Testing) AND GUI)
2:    OR TITLE-ABS-KEY ((Automated W/5 GUI) AND Testing)
3:    OR TITLE-ABS-KEY((GUI W/5 Testing) AND Automated)  )
4: AND  LIMIT-TO(LANGUAGE , "English")
5: AND  PUBYEAR > 1989 AND PUBYEAR < 2021
6: AND
7:  (LIMIT-TO(DOCTYPE , "cp") OR LIMIT-TO(DOCTYPE , "ar") OR
8:   LIMIT-TO(DOCTYPE , "ch") OR LIMIT-TO(DOCTYPE , "Undefined")  )
9: AND
10:  (LIMIT-TO(SUBJAREA , "COMP") OR LIMIT-TO(SUBJAREA , "ENGI")
11:   OR LIMIT-TO(SUBJAREA , "MATH") )
```

Fig. 1: The used Search Query

Scopus operator TITLE-ABS-KEY is used. To fine-tune the results, a minimum distance of terms was established, using the "W/" operator. The distance was set to 5 after several tests observing the results. In Figure 1, each family of words is represented by its main term. Each term was replaced by the derived family of terms, using the OR operator to accept the appearance of at least one of the terms within its family. Using the Scopus facilities, papers were also excluded according to their type, language and publication date, excluding works that:

exC1: are not written in English (in line 4, using the Scopus Document field code: LANGUAGE and limiting it to "English")

exC2: are published before year 1990 and after 2020 (in line 5 using the Scopus Publication field code: PUBYEAR)

exC3: are not conference, workshop, journal publications or book chapters (in lines 7 and 8) using the Scopus Document field code: DOCTYPE and limiting it to types Conference Paper-"cp", Article-"ar", Book Chapter-"ch" and "Undefined"). The last one was included because some documents that have been accepted for publication, but have not yet been assigned to a journal or conference, so that they are temporarily indexed as "Undefined".

exC4: do not belong computer science area (in lines 10 and 11) using the Scopus subject areas: COMP, ENGI and MATH.

The search was performed on January 2021. The total amount of papers retrieved was 2240.

3.2 Pre-processing

First of all, we manually excluded in Scopus the papers belonging to other fields, reducing the total amount of papers to 1233. This was needed because, for instance, a document can be classified as Computer Science and Social Science because it describes a social science study using some computational system. Since these papers are also categorized as COMP, ENG or MATH, they were retrieved by the search query, even if they also belonged to other fields. The papers that were clearly off-topic were manually rejected.

Driven by our additional goal to create a GUI testing research repository, we searched for a simple and flexible environment that, besides assisting our work, would allow future interactions with the extracted papers. Thus, we decided to

use BUHOS [4], an open source web-based paper management system. We uploaded the 1233 papers in BUHOS, we defined additional exclusion criteria (exC5 and exC6 below), and manually applied these exclusion criteria by screening the title and abstract of each paper.

exC5: clearly off topic, i.e. not at all related to the scope (Section 2)
exC6: not a primary study

The 1233 papers were divided among the authors, who, after reading the title and abstract, marked them as included, excluded or undecided. Next, a collective analysis was carried out to make a final decision on the undecided papers, resulting 720 papers. Then, a backward snowballing [21] on the 720 papers resulted 50 new papers that were screened based on the title and abstract. This added 24 papers, resulting in the total of 744 included publications.

3.3 Analysis and Visualization

CRExplorer [19] and Biblioshiny[3] were used to analyse and visualize the data. These tools were selected because they have specific functionalities to visualize bibliometric maps. In addition, Scopus was used in conjunction with Excel to generate the charts. Before the analysis, normalization was required on the keywords using a thesaurus of synonyms[4], and the author's names by taking accents and different formatting into account. Related to the conferences, it was necessary to split the description in order to properly obtain the name of the conference separately from the publisher and the year of publication.

4 Results

4.1 Size of the area and growth

The number of publications in a field over time is a central piece of information to investigate its growth and development. In Figure 2 the evolution of the growth per year along with the trend is depicted. The first decade covered by our study only has 18 papers related to field. There are even two years (1992 and 1993) with no papers at all. In the second decade of our study this increased to 170 works. And, in the third decade we found 556 works. Since a 41.4% of all documents have been published in the last 5 years, we expect that the automated GUI testing field continues to grow like it did in the last decade.

Between 2009 and 2013 we see an increase in the amount of papers that deviates from the trend. Reasons for this could be various. In 2008 the first edition of the ICST conference was held, being the first international conference entirely dedicated to software testing. Moreover, in 2009 the first edition of the TESTBEDS workshop was celebrated at ICST. There was also an increase in papers related to web testing, this can be related to the fact that in 2009, it

[3] https://www.bibliometrix.org/Biblioshiny.html
[4] https://gui-testing-repository.testar.org/keywords

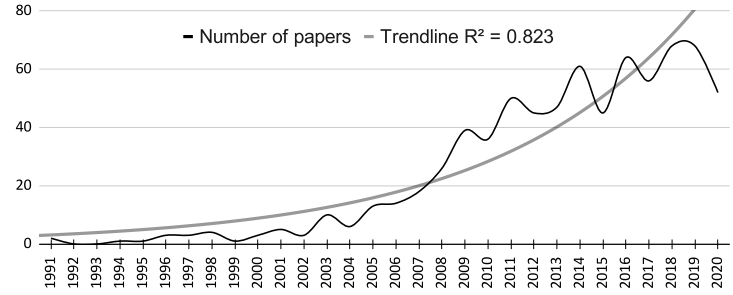

Fig. 2: Evolution of the number of publications

Total 1991-2020	J	C	W	B
744	122	528	87	7

Table 2: Papers in journals (J), conferences (C), workshops (W) and chapters (B).

was decided to merge Selenium RC and Webdriver and called the new project Selenium WebDriver, or Selenium 2.0. A third reason might be that Sikuli started in 2009 [15]. Sikuli is a visual approach to search GUIs using screenshots, allowing users to take a screenshot of a GUI element (such as a toolbar button, icon, or dialog box) and query a help system using the screenshot instead of the element's name. Finally, in 2009 there is an increase on papers related to mobile testing. This is probably related to the fact that in July 2008 the Apple's App Store went live and in August, the Android Market.

During 2020, we observe that the number of publications decreases, this could be explained by the pandemic since several conferences were canceled, mobility was reduced and therefore the research outcomes could be affected.

4.2 Types of publications and their ranking

We found papers published in journals, conferences, workshops and as book chapters. Table 2 and Figure 3 show the amont of papers of each kind.

We can observe that the majority of papers have been published in conference proceedings. This make sense since conferences provide feedback to researchers more quickly than journals. Moreover, in many cases papers describing part of a larger solution are presented in conferences in order to obtain feedback and validate each piece of work and later the entire proposal is presented in a journal. This is also the behavior in the entire Computer Science field [7].

Table 3 shows that IEEE Transactions on Software Engineering (TSE) has been the top one journal with 12 published articles on the field. Even though the automated GUI testing field has been steadily growing during the last 3 decades, STVR is the first journal that launched a special issue entirely dedicated to this field in only 2020. Papers included in that special issue have not been counted for our study because they were not yet published in 2020. By examining the data in Table 3, Bradford's Law [3] can be applied. This law establishes that the total set of journals in a research field can be divided into 3 categories or zones, each containing approximately one third of the total papers in the field. The

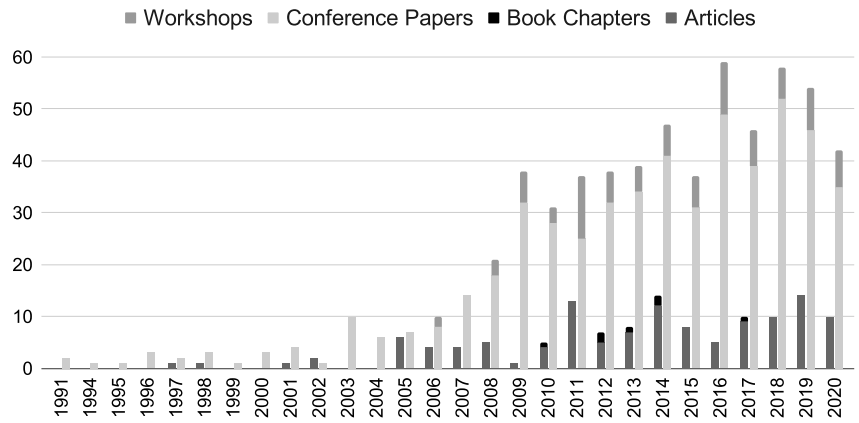

Fig. 3: Number of papers published in (journals + books) vs (conferences + workshops)

Journal name	**# papers**	**%**	**SJR**
Transactions On Software Engineering (TSE)	12	9,83%	1.19
Information And Software Technology (IST)	8	6,55%	0.78
Software Quality Journal (SQJ)	7	5,73%	0.36
IEEE Software	6	4,92%	0.81
Transactions On Software Engineering And Methodology (TOSEM)	5	4,10%	0.76
Software Testing Verification And Reliability (STVR)	5	4,10%	0.31
Empirical Software Engineering (ESE)	4	3,28%	1.08
Information Technology Journal	4	3,28%	0.11
ACM SIGPLAN Notices	3	2,46%	4.90
IEEE Access	3	2,46%	3.90
Innovations In Systems And Software Engineering	3	2,46%	1.90
Remaining 54 from the total of 65 journals	62	50,82%	
Total number of papers	**122**	**100%**	

Table 3: Top 11 of most contributing Journals

first category is related to articles which are published in an small number of journals, called core journals. The second category corresponds to the journals with an average of papers. And the last category corresponds to several journals that publish few papers.

From Table 3 we can derive that the top 6 journals are the core journals, since they correspond to 43 articles, which is 35.2% of all 122 journal papers. The next group is found in the next 16 journals (37 articles or 30.3%). In order to represent the last articles, the 42 remaining journal are necessary. The Bradford relation for journals is 6:16:43 and the details per zone can be found in Table 4.

The 528 papers were presented at 386 conferences of which 4.15% has CORE ranking A*, 16.32% CORE A, 16.84% CORE B, 10.36% CORE C and 37.31% has no CORE ranking. The remaining 14.51% conferences were in years when no CORE ranking was given (yet). The 87 workshop papers were presented at 56 workshops of which 37.50% was co-located at a CORE A* conference, 28.57% at a CORE A conference, 3.57% at CORE B conference and 12.50% at

Zones	# journals	# papers
Zone 1	6	43
Zone 2	16	37
Zone 3	43	42
Total	65	122

Table 4: Bradford's Law zones applying Leimkuhler model [9]

Conference name	# papers	%
International Conference on Software Engineering (ICSE)	37	7,01%
International Conference on Software Testing, Verification and Validation (ICST)	36	6,81%
International Conference on Automated Software Engineering (ASE)	27	5,11%
International Symposium on Software Testing and Analysis (ISSTA)	26	4,92%
Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)	22	4,17%
IEEE International Symposium on Software Reliability Engineering (ISSRE)	15	2,84%
International Conference on Software Maintenance (ICSM)	14	2,65%
International Computer Software and Applications Conference (COMPSAC)	11	2,08%
International Conference on Software Engineering and Knowledge Engineering (SEKE)	9	1,70%
International Conference on Software Quality, Reliability and Security (QRS)	9	1,70%
Remaining 45 conferences from the total of 56 conferences	322	60,98%
Total number of papers	**528**	**100%**

Table 5: Top 10 of most influential Conferences

CORE C conference, 5.36% at conferences with no CORE ranking, and 5.36% at workshops not co-located with any conference. The remaining 7.15% workshops were in years when no CORE ranking was given (yet). Table 5 shows the most contributing conferences. ICSE and ICST are almost even at the top while in 2020 ICSE had celebrated 42 editions and ICST only 13.

4.3 Citations and Reference Publication Year Spectroscopy

In Table 6 we list the top 10 papers that have the most cites in Scopus, together with the year of publication, the complete reference, the number of cites retrieved by Scopus (Sc), and the number of cites retrieved by Google Scholar (GS). The cites from Scopus and Scholar differ in that Scholar has a much higher count. From [12], we learn that Scholar citation data is essentially a superset of Scopus, but with substantial extra coverage. We can see that 7 out of the top 10 most cited papers are concerned with Android testing. The remaining 3 papers are related to models (event-flow or state models) and widget detection (Sikuli).

The technique of Reference Publication Year Spectroscopy (RPYS) [13] is a quantitative method to identify the historical origins or turning points of research fields. This method analyzes the publication years of the references cited by all

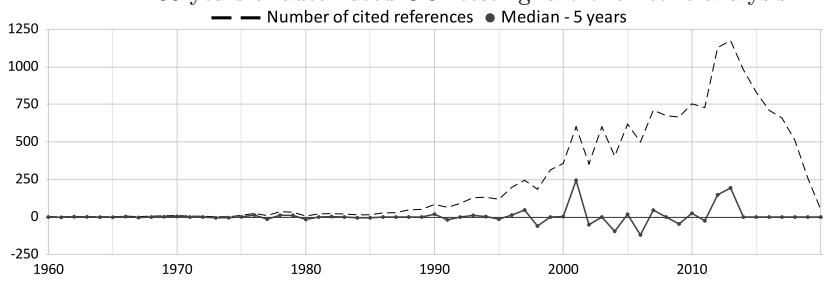

Fig. 4: Reference Publication Year Spectroscopy

Ref	Title	Author	Year	Sc	GS
[MTN13]	*Dynodroid: An input generation system for android apps*	Machiry, A., Tahiliani, R., Naik, M.	2013	397	672
[AFT+12b]	*Using GUI ripping for automated testing of android applications*	Amalfitano, D., Fasolino, A., Tramontana, P., De Carmine, S., Memon, A.	2012	343	563
[CGO16]	*Automated test input generation for android: Are we there yet?*	Choudhary S.R., Gorla A., Orso A.	2016	245	401
[ANHY12]	*Automated concolic testing of smartphone apps*	Anand, S., Naik, M., Harrold, M., Yang, H.	2012	231	428
[AOA05]	*Testing Web applications by modeling with FSMs*	Andrews A.A., Offutt J., Alexander R.T.	2005	227	477
[YCM09a]	*Sikuli: Using GUI screenshots for search and automation*	Yeh T., Chang T.-H., Miller R.C.	2009	217	400
[MHJ16]	*Sapienz: Multi-objective automated testing for android applications*	Mao K., Harman M., Jia Y.	2016	207	336
[GNAM13]	*RERAN: Timing- and touch-sensitive record and replay for Android*	Gomez L., Neamtiu I., Azim T., Millstein T. Total	2013	202	341
[Mem07]	*An event-flow model of GUI-based applications for testing*	Memon A.M.	2007	193	364
[HLN+14]	*PUMA: Programmable UI-automation for large-scale dynamic analysis of mobile apps*	Hao S., Liu B., Nath S., Halfond W.G.J., Govindan R.	2014	192	321

Table 6: Top 10 papers with most cites in Scopus (includes cites in Google Scholar) [5]

the papers in a specific field. A Reference Publication Year (RPY) is reflected in the spectogram as a pronounced peak, usually corresponding to a publication that has been referenced very frequently. These publications are of significant importance, as they may represent the origins of the research field in question.

An RPYS chart was obtained using CRExplorer and is shown in Figure 4, from 1960, although there are references up to 1901. The most influential year seems to be 2001, this is the year when Atif M. Memon finished his PhD entitled *A comprehensive framework for testing graphical user interfaces* [14]. In that year he published two final papers for his thesis. The first paper [MPS01] presents a new test case generation technique based on Artificial Intelligence Planning and using a model based on a GUI structure. Both Artificial Intelligence and Model-based Testing are trends that will guide the research field in the posterior years

[5] https://gui-testing-repository.testar.org/bibliography

Name	Total	J	C	W	B	Year of first publication
Memon, A.M.	53	18	29	5	1	1999
Paiva, A.C.R.	31	6	20	5	0	2005
Alégroth, E.	17	3	8	5	1	2013
Vos, T.E.J.	16	2	11	3	0	2012
Xie, Q.	15	4	10	1	0	2004
Fasolino, A.R.	13	4	5	4	0	2010
Zeller, A.	13	1	10	2	0	2012
Aho, P.	12	0	7	4	1	2011
Amalfitano, D.	11	3	4	4	0	2010
Coppola, R.	11	4	3	4	0	2016
Ramler, R.	11	1	8	2	0	2008

Table 7: Ranking of author by number of publications in journals (J), conferences (C), workshops (W) and book chapters (B)

papers	1	2	3	4	5	6	7	8	9	10	11	12	13	15	16	17	31	53
authors	1128	198	60	36	21	14	8	3	3	6	3	1	2	1	1	1	1	1

Table 8: Distributions of number of author per number of publications

to this publication, as we explain later on in Section 4.7. In the second paper, Memon et al. [MSP01] introduce different coverage criteria for GUI testing and evaluate them through a case study, for the first time.

In addition, years 2012 and 2013 appear as peaks in the Spectroscopy chart. Five publications [AFT+12a, MTN13, CNS13, YPX13, AN13] appear among the most cited within the field. All of them have one common topic: Android testing.

4.4 Most influential authors

The 744 documents that integrate this study have been written by a total of 1,488 authors. Table 7 shows the 11 most prolific authors, among them contributing 203 publications (27.28 %). For this ranking we count all authors of each paper, not only the first one. One notable fact is that 7 of the 11 authors published their first paper in the field since 2010, and only one published before 2000.

The distribution of the number of publications among authors is presented in Table 8. The largest group consists on authors who published a single paper, representing 75.81%. As show in the table, as the number of publications increases, the number of authors tends to decrease. The Lotka's law describes this behavior and states that the number of authors y publishing a certain amount of papers x is in inversely proportional to x, as $y = \frac{c}{x^n}$, where n and c are two constants to be estimated for every data set. We used the software Lotka [16], to apply the Maximal Likelihood method and estimate the parameters for this study, resulting in $n \approx 2.59$ and $c \approx 0.77$ i.e., our data set follows Lotka's general law as $y = \frac{0.77}{x^{2.59}}$. To assess the fitness between this hypothesized Lotka model and the actual distribution of the data, the Kolmogorov-Smirnov statistical test was applied. Even for a level of significance of 0.2, the results support the hypothesis.

4.5 Productivity and funding

As shown in Figure 5, there is a large gap between the most contributing country, United States, and the rest. China published its first papers in 2006 and since

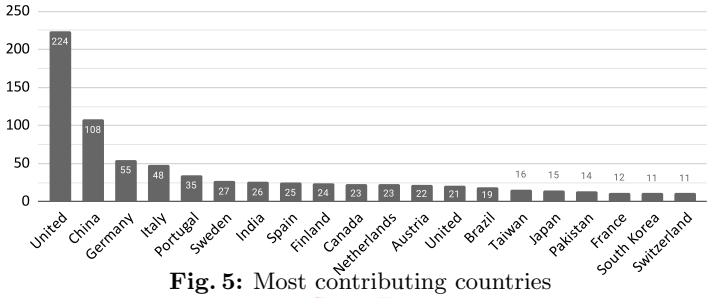

Fig. 5: Most contributing countries

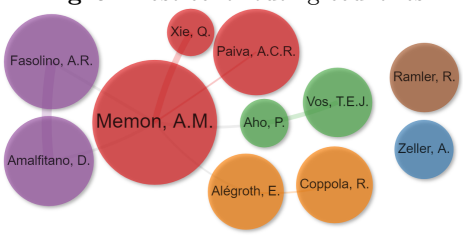

Fig. 6: Collaboration network of authors

then has contributed 108 publications, keeping a rate of 7.2 publications per year, similar to that of United States, with 7.5 annual papers since 1991.

Although China and the US are the main contributing countries to the field, the European region has had a boost in the last decade, and since 2015 occupies the first place with 308 publications. The Asian continent has contributed 242 publications, closely following North America with 245 publications so far.

A 21% of the papers included funding information. From all the mentions, 9,7% came from private funding by big companies such as Google, Microsoft, Amazon Web Services, and Boeing, amongst others. Asia is the continent that provides most funding resources for the majority of sponsored works (33,7%), followed by Europe (28,6%) and North America (27,1%). The leading funding agency in Asia is the National Natural Science Foundation of China. Likewise, the leading funding agencies in Europe and North America are the European Commission and the National Science Foundation, respectively. It is worth to mention that the only South American country that has funding is Brazil.

4.6 Collaboration

In Figure 6 we depict the collaboration between the most prolific authors in the field from Table 7. Six authors have been co-authors with Atif M. Memon, who can also been related with another two authors through those six. Only 2 of the 11 authors do not have co-authorship with any of the most contributing authors.

Figure 7 shows the evolution of author's collaboration over the 30 years. Single author publications have historically remained low, while publications of more than 4 authors have been increasing. However, only 18.95% of the papers have been the result of collaboration among affiliations from different countries.

4.7 Trends in keywords

By analysing the keywords provided by the authors, we aim to reveal the details of a domain's major research topics and their introduction into the field. This is

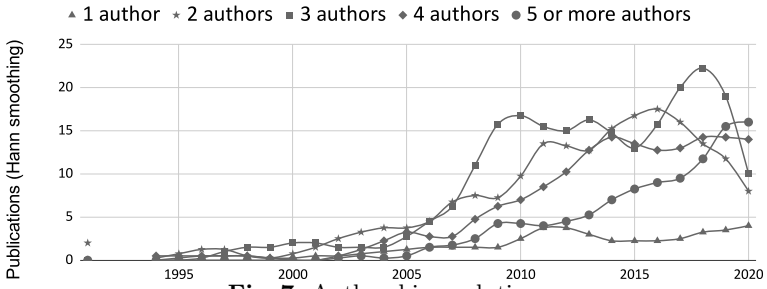

Fig. 7: Authorship evolution

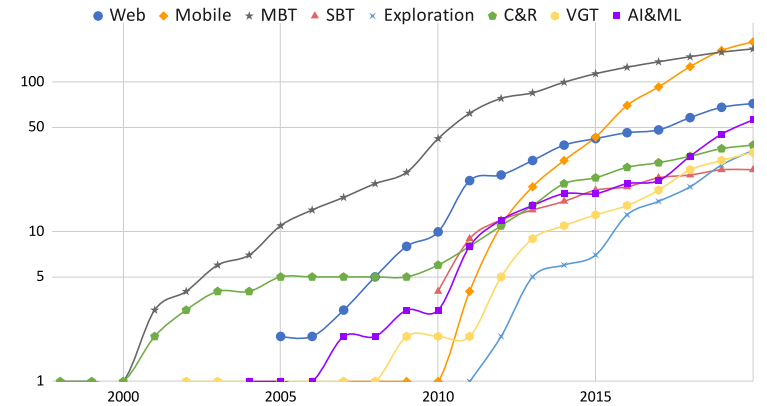

Fig. 8: Cumulative frequency of keywords

not as easy as just counting the most used keywords [5, 18]. Many keywords do not give specific information on the details of the field because they are inherent to it, e.g. software testing, GUI testing, tools, regression testing, etc. In addition, different words can be used for describing the same concept, and thus we had to group them. We started standardizing plural forms into their singular form, by means of NLTK [10]. In order to group the keywords, the authors of this paper studied all the available keywords, and each made their individual classification. We set-up two brain-storming sessions to come to the following classification as a representation of relevant research themes in the domain that we want to study:

> web, mobile, model-based testing (MBT), search-based testing (SBT), visual-based testing (VBT), Artificial Intelligence and Machine Learning (AI&ML), Capture and Replay (C&R) and Automated Exploration

The objective is to study: *Web* and *mobile*: to distill the trend in the types of SUTs that are tested **;** *MBT*, *SBT*, *VBT*, *AI&ML*: to visualize the timeline of the pick-up of different technologies into automated GUI testing**;** *C&R*: to investigate the evolution of the trend where the focus was on these tools; and *Automated Exploration*: for the shift from scripted to scriptless testing using random testing, traversal techniques and crawling. Figure 8 shows the cumulative frequency values per each group of keywords, annually. We see that both MBT and C&R have their first appearance in 1998. Since then MBT has been the main topic of the field, until Mobile reached a greater number of papers in 2019. As of 2010, two topics were introduced: SBT and Automated Exploration.

Publications mentioning Web-based SUT have remained constant. It is remarkable that 50% of MBT papers have been published as of 2014, being MBT one of the first topics in the field. C&R has seen a decrease in its frequency, coinciding with the considerable increase in VBT. This might indicate that C&R is being replaced by Image Recognition or Image Comparison techniques.

AI&ML has appeared in 56 papers: by 2013, it had appeared in 48 papers (26.79%) and it took 5 years to reach 50% of its total frequency. However, just one year was needed for AI&ML to reach 75%. In the last two years AI&ML appeared in as many papers as in the entire previous history of the field.

4.8 Discussion

To avoid internal validity threats, we use Scopus, the largest database of peer-reviewed scientific literature; we define a search string and we validate the results with a small set of relevant works. Since computer science works are mainly published in English, we advocate that we found the majority of works even though we aware that some works are not retrieved due to they are published in a different language.

Regarding the replicability of the study, we clearly define a protocol and documented all the process to mitigate this threat. We use the metadata of the works to perform the analysis to mitigate the threat that results may be biased by researchers' judgement. In order to deeper understand the techniques used for automated GUI testing, we propose to follow this work with a mapping review in order to establish the trends in the area.

5 Conclusions

This paper provides facts about automated GUI testing field. Publications have increased continuously, with exponential growth in the last decade. Lotka's Law and Bradford's Law were found applicable to the field. Analysis of author's collaboration, keywords and the geographic dispersion of the field was provided. The most common type of publication is the conference papers. The 6 core journals were identified, as well as the most prolific authors. A repository [6] was developed, with all the 744 referenced papers and further bibliometric results.

We conclude that this study offers relevant information for the field, its evolution over 30 years and trending topics for future research.

6 Acknowledgements

We thank Fernando Pastor for his valuable contribution. This research has been funded by DECODER (decoder-project.eu), iv4XR (iv4xr-project.eu), and IVVES (ivves.weebly.com) projects.

[6] https://gui-testing-repository.testar.org

References

1. Banerjee, I., Nguyen, B., Garousi, V., Memon, A.: Graphical user interface testing: Systematic mapping and repository. IST **55**(10), 1679–1694 (Oct 2013)
2. Barr, E.T., Harman, M., McMinn, P., Shahbaz, M., Yoo, S.: The oracle problem in software testing: A survey. TSE **41**(5), 507–525 (2015)
3. Bradford, S.C.: Sources of information on specific subjects. Eng. **137**, 85–86 (1934)
4. Bustos, C., Malverde, M., L., P., Díaz-Mujica, A.: Buhos: A web-based systematic literature review management software **7** (11 2018)
5. Chen, G., Xiao, L.: Selecting publication keywords for domain analysis in bibliometrics: A comparison of three methods. J. of Informetrics **10**, 212–223 (02 2016)
6. Cobo, M., López-Herrera, A., Herrera-Viedma, E., Herrera, F.: Science mapping software tools: Review, analysis, and cooperative study among tools. Journal of the American Society for Information Science and Technology **62**(7), 1382–1402 (2011)
7. Franceschet, M.: The role of conference publications in cs. Communications of the ACM **53**(12), 129–132 (2010)
8. Johnson, M.: Automated testing of user interfaces. In: Pacific North West Software Quality conference. pp. 285–293 (1987)
9. Leimkuhler, F.: An exact formulation of bradford's law. J. of Documentation (1980)
10. Loper, E., Bird, S.: Nltk: The natural language toolkit. arXiv 0205028 (2002)
11. Lotka, A.J.: The frequency distribution of scientific productivity. Journal of the Washington Academy of Sciences **16**(12), 317–323 (1926)
12. Martín-Martín, A., Orduna-Malea, E., Thelwall, M., Delgado López-Cózar, E.: Google scholar, web of science, and scopus: A systematic comparison of citations in 252 subject categories. Journal of Informetrics **12**(4), 1160–1177 (2018)
13. Marx, W., Bornmann, L., Barth, A., Leydesdorff, L.: Detecting the historical roots of research fields by reference publication year spectroscopy (rpys). Journal of the Association for Information Science and Technology **65**(4), 751–764 (2014)
14. Memon, A.M.: A comprehensive framework for testing graphical user interfaces. Ph.D. (2001), advisors: Mary Lou Soffa and Martha Pollack; Committee members: Prof. Rajiv Gupta (University of Arizona), Prof. Adele E. Howe (Colorado State University), Prof. Lori Pollock (University of Delaware)
15. Paulos, E.: The rise of the expert amateur: Diy culture and citizen science. In: Proceedings of the 22nd annual ACM symposium on User interface software and technology. pp. 181–182 (2009)
16. Rousseau, B., Rousseau, R.: Lotka: A program to fit a power law distribution to observed frequency data. Cybermetrics: International Journal of Scientometrics, Informetrics and Bibliometrics (4), 4 (2000)
17. Small, H.: Visualizing science by citation mapping. Journal of the American Society for Information Science **50**(9), 799–813 (1999)
18. Su, H.N., Lee, P.C.: Mapping knowledge structure by keyword co-occurrence: a first look at journal papers in Technology Foresight. Scientometrics **85**(1), 65–79 (October 2010). https://doi.org/10.1007/s11192-010-0259-8
19. Thor, A., Marx, W., Leydesdorff, L., Bornmann, L.: Introducing citedreferencesexplorer (crexplorer): A program for reference publication year spectroscopy with cited references standardization. Journal of Informetrics **10**(2), 503–515 (2016)
20. Vieira, E.S., Gomes, J.A.N.F.: A comparison of Scopus and Web of Science for a typical university. Scientometrics **81**(2), 587–600 (2009)
21. 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. pp. 1–10 (2014)