2016 IEEE/ACM 13th Working Conference on Mining Software Repositories

Findings from GitHub: Methods, Datasets and Limitations

Valerio Cosentino
Atlanmod, Inria, Mines Nantes, LINA, Nantes, France
valerio.cosentino@mines-nantes.fr

Javier Luis Cánovas Izquierdo
UOC, Barcelona, Spain
jcanovasi@uoc.edu

Jordi Cabot
ICREA – UOC, Barcelona, Spain
jordi.cabot@icrea.cat

ABSTRACT

GitHub, one of the most popular social coding platforms, is the platform of reference when mining Open Source repositories to learn from past experiences. In the last years, a number of research papers have been published reporting findings based on data mined from GitHub. As the community continues to deepen in its understanding of software engineering thanks to the analysis performed on this platform, we believe it is worthwhile to reflect how research papers have addressed the task of mining GitHub repositories over the last years. In this regard, we present a meta-analysis of 93 research papers which addresses three main dimensions of those papers: i) the empirical methods employed, ii) the datasets they used and iii) the limitations reported. Results of our meta-analysis show some concerns regarding the dataset collection process and size, the low level of replicability, poor sampling techniques, lack of longitudinal studies and scarce variety of methodologies.

CCS Concepts

•**General and reference** → **Surveys and overviews;** •**Software and its engineering** → *Software system structures;*

Keywords

Systematic review; GitHub; Meta-analysis

1. INTRODUCTION

In the last years, a number of works ([9, 14, 16, 11] among others) have been focused on mining GitHub, an online code hosting platform that relies on Git and additionally provides collaborative and social features (e.g., pull-request support and following users). The platform has become more and more popular and currently stores more than 35 million of projects. Such popularity, social and collaborative features plus the availability of its metadata made it a perfect candidate for data mining researchers.

MSR'16, May 14-15 2016, Austin, TX, USA

ISBN 978-1-4503-4186-8/16/05…$15.00

DOI: http://dx.doi.org/10.1145/2901739.2901776

The increasing number of works targeting GitHub provides the basis to analyze how they have performed the mining process. We believe that studying how research papers have mined GitHub can be useful for the community to understand the current situation regarding the analysis of the platform and tackle potential perils.

In this paper, we analyze and discuss how research papers have addressed the task of mining GitHub repositories over the last years. In particular, we analyze the empirical methods employed, the datasets they used and the limitations reported. We collect a number of papers from the main digital libraries and complement the collection with manual retrieval of works published in the last editions of a set of conferences and journals relevant of the research field. We select 93 papers according to a criteria and analyze them using a grounded theory approach and a manual open coding analysis to identify possible concerns. Results show some concerns regarding the dataset collection process and size, the low level of replicability, poor sampling techniques, lack of longitudinal studies and scarce variety of methodologies.

The paper is structured as follows. Section 2 describes the methodology used to identify and classify the works. Section 3 shows the results obtained. Section 4 discusses the concerns found. Section 5 reports on possible threats to validity. We end the paper by commenting related work in Section 6 and providing further work and conclusions in Section 7.

2. METHODOLODY

We describe the methodology we followed to identify and classify relevant works for our study. The methodology covers: (1) the digital libraries checked, (2) the collection process, and (3) the selection criteria and screening process.

2.1 Digital libraries

The selection of the digital libraries was driven by several factors, specifically: (1) number of works indexed, (2) update frequency, and (3) facilities to execute advanced queries, navigate the citation and the reference networks. We selected 8 digital libraries (shown in Tab. 1) that represented a good mix of the desired factors.

2.2 Collection Process

We performed a three-phased selection process to make the set of collected works as complete as possible. The first phase consisted in defining the search query and its execution on the digital libraries. All works that contained in the title, abstract, author keywords or index terms the word

Digital Library	URL	Adv.Query	Cit. Nav.	Ref. Nav.
Google Scholar	scholar.google.com	×	✓	×
DBLP	dblp.uni-trier.de	✓	×	×
ACM	dl.acm.org	✓	✓	✓
IEEE Xplore	ieeexplore.ieee.org	✓	×	✓
ScienceDirect	sciencedirect.com	✓	×	×
CiteSeerX	citeseerx.ist.psu.edu	✓	✓	✓
SpringerLink	link.springer.com	✓	×	✓
Web of Science	webofknowledge.com	✓	✓	✓

Table 1: Digital libraries selected.

GitHub or variations of it (e.g., *github*, *git hub*) were collected. At the end of this phase, 184 works were collected.

The second phase took the previous set of works and applied a breadth-first search approach using backward and forward snowball methods by navigating their citations and references. We relied on the citation links provided by the digital libraries when available (otherwise we made it manually). New works were added only if they fulfill our search query. By following a breadth-first search approach, the snowball methods were applied iteratively to the new works until no more works were identified. The second phase was able to identify 47 new works, thus having a total of 231 collected works.

In the third phase, we performed an issue-by-issue analysis of main conference proceedings and journals in software engineering from January 2009 until October 2015. Our goal was to complete the list of the initial works and assess the completeness of the collection obtained so far. We selected 24 top venues (16 conferences and 8 journals, see Tab. 2) including topics as empirical studies, open source and software analysis, development and evolution. All the works identified in this phase were already included in our collection.

Conf.	CSCW	CSMR	MSR	ICSM(E)
	ICSE	FSE	ISSRE	APSEC
	SANER	WCRE	ESEM	SEKE
	IST	OSS	SAC	EASE
Journ.	TOSEM	TSE	SoSym	Software
	JSS	ESE	IST	SCP

Table 2: Selected venues.

2.3 Selection Criteria and Screening Process

We defined a selection criteria to identify relevant works. The main inclusion criteria was that only research efforts focused on GitHub were considered. In particular, they had to leverage on GitHub metadata (i.e., project and user information such as issues, pull requests, watchers and followers) in order to shed some light on OSS dynamics, software development practices (e.g, testing, forking), project features (e.g., popularity, licenses) or project communities (e.g., participation, composition). If a work published in a journal or conference was deemed a more complete study of a previous version of the work published by the same authors, the extended version was included and the previous one was discarded. As exclusion criteria, all works i) not written in English or ii) being Master/PhD thesis were excluded. We applied the screening process using this selection criteria and selected 93 out of the 231 collected works[1].

Table 3 shows the distribution along the years of the number of works collected/selected (and the publication type for the selected works). They span from 2010 to (October) 2015 and, as can be seen, there is an increasing trend in the works published along the past 5 years.

[1]The list of collected and selected works is available at http://tinyurl.com/GitHub-SystRev-Papers

	Collected	Selected		Techn. rep.	Work.	Conf.	Journ.
2010	4	1	=	1	0	0	0
2011	3	0	=	0	0	0	0
2012	12	6	=	0	0	5	1
2013	43	17	=	1	0	15	1
2014	93	41	=	3	3	35	0
2015	76	28	=	2	2	20	4
Total	231	93	=	7 (7.5%)	5 (5.4%)	75 (80.7%)	6 (6.4%)

Table 3: Distribution of collected/selected works along the years.

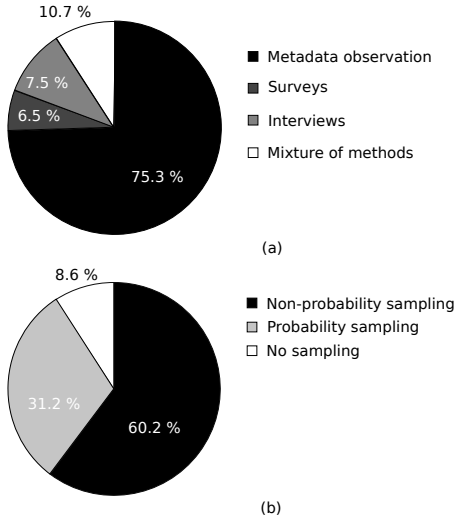

Figure 1: (a) Empirical methods and (b) sampling techniques employed.

3. RESULTS

In this section we present the results of our analysis in terms of three main dimensions, namely: (1) the empirical methods employed, (2) dataset size and how it was collected, and (3) limitations reported by the selected works.

3.1 Empirical Methods Employed

Fig. 1a shows the results of the study of empirical methods employed. As can be seen, the great majority of the works (75.3%) rely on the direct observation of GitHub metadata. The use of surveys and interviews was detected in 14% of the works. The remaining 10.7% of the works combine pairs of the previous research methods (e.g., metadata observation and interviews). It is worth noting that only 5.4% of the selected works applied longitudinal studies[2] (e.g., co-evolution of documentation and popularity [1]).

We also study the kind of sampling techniques used to build datasets out of GitHub (i.e., subsets of projects and users). Fig. 1b shows the results of this analysis. Most of the works (60.2%) use non-probability sampling, while around a third (31.2%) rely on probability sampling. Interestingly enough, stratified random sampling[3], which takes into ac-

[2]A longitudinal study is a correlational research study that concerns repeated observations of the same variables over long periods of time.

[3]The stratified random sampling involves the division of

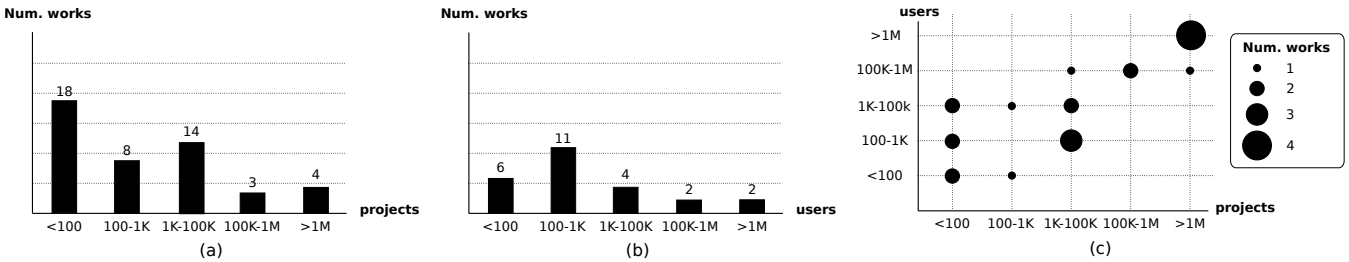

Figure 2: Number of works reporting the size of their datasets according to (a) the number of projects, (b) number of users and (c) both.

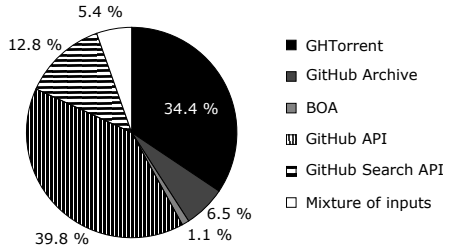

Figure 3: How the data was collected.

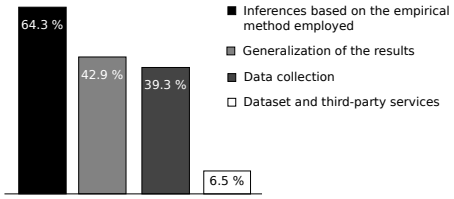

Figure 4: Classification of the limitations reported.

count the diversity of projects and users in GitHub[10], is used just in 3.2% of the works. It is also interesting to note that only 8.6% of the analyzed works do not use sampling techniques at all and prefer to build a dataset covering all the information provided by GitHub.

3.2 Datasets Used

The way the analyzed papers report the size of their datasets is in terms of number of projects and/or users (total number of users included in the dataset). In total, 50.5% of the works reported the dataset size in terms of projects, while 26.9% of them used the number of users. Only 22.6% of the works provided the two dimensions. Fig. 2 summarizes the number and size of the datasets according to the number of projects (see Fig. 2a), number of users (see Fig. 2b) and both (see Fig. 2c).

Regarding how the data was collected, we noted the usage of 4 solutions, namely: (1) curated dataset mirroring GitHub's data (i.e., GHTorrent[6], GitHub Archive and BOA[5]), (2) GitHub API, (3) GitHub Search API and (4) a mixture of the previous ones. Fig. 3 shows the results of this analysis. The majority of the works used curated datasets and, among the existing datasets, GHTorrent is the most popular (34.4% of the works). The use of the GitHub API and GitHub Search API was spotted in 39.8% and 12.8% of papers, respectively. It is worth noting that the use of the search API was mainly used to collect user' contacts in those works reporting on surveys and interviews. Finally, the 5.4% of the works leveraged on a mix of the previous solutions.

With respect to the availability of the datasets to replicate the findings, only 31.2% of the works either provide a link to download the datasets used or use datasets freely available on the Web. The remaining 68.8% of the works although explaining how the datasets were collected and treated, do not provide any link to replicate their findings.

population into smaller groups (strata), that share same characteristics. It produces characteristics in the sample that are proportional to the overall population.

3.3 Limitations Reported

By analyzing the limitations reported in the selected papers, we can assess their degree of self-awareness regarding potential threats to validity. Interestingly enough, 39.8% of them did not report any issue on this respect. For the rest, we identified four categories of limitations related to (1) the inferences based on the empirical method employed, (2) the data collection process, (3) the generalization of the results and (4) the dataset and use of third-party services. Fig. 4 shows the results. Note that works may report limitations covering several categories.

Most of the works (64.3% of the works reporting limitations) reported issues about the inferences based on the empirical method employed, which included potential errors and bias introduced by the authors, techniques and tools used The issues regarding the generalization of the results (42.9%) was mainly due to the non-probability sampling techniques chosen. It is worth noting that this category does not include the generalization of the results for Open Source, since many works justify the use of GitHub data due to its representativeness of the current status of Open Source. With respect to the data collection (39.3%), it is interesting to note that half of them explicitly commented on problems with the GitHub API (e.g., limited quota of requests or events not properly returned). Finally, we detected a few works (6.5%) which reported problems with datasets and third-party services mirroring GitHub, mostly related to the datasets size and data freshness.

4. DISCUSSION

In this section we report on the concerns we derive from our study. We believe that, if tackled, they can be useful to enhance the generality, quality and confidence of the findings when mining GitHub.

Data Collection. A common requirement of the works

analyzed is the acquisition of up-to-date and curated data from GitHub. Data is generally obtained through the platform API or existing datasets from third-party services (as reported in Sect. 3.2). However these solutions are currently antithetic with respect to its freshness and curation, thus requiring researchers to evaluate this trade-off. On the one hand, the GitHub API allows developers to get a limited amount of fresh and not curated data (e.g., latest snapshot of a project) due to API requests limit and inconsistencies in the returned data (as reported in Sect. 3.3). On the other hand, existing datasets and services provide curated data (e.g., *GHTorrent* and *BOA*), but usually out-of-date with respect to the original one on GitHub.

Dataset size. Most of the analyzed papers use datasets of small-medium size (as reported in Sect. 3.2). Among those works using datasets composed of less than 100 projects, there are even 5 with less than 50 projects, which may provide less evidences when generalizing of the results.

Replicability. Replicability is the assumption that a valid scientific study can be repeated and will yield the same results. However, more than two thirds of the selected works does not make available neither the datasets used nor the code to collect the data and replicate the studies (see Sect. 3.2), thus creating a barrier to compare previous and new findings from different studies, as well as reducing the confidence of the corresponding results. For instance, the works presented in [15] and [2] report conflicting results on StackOverflow and GitHub (i.e., the former claims that active GitHub contributors are also active on StackOverflow while the latter denies it), as the work in [2] does not provide the dataset used, it is not possible to discern potential discrepancies.

Sampling. Sampling represents a methodological concern that can bias the generalization of the findings, where representativeness is important. Only few works take care of building diverse samples (i.e., datasets) using techniques such as stratified random sampling (as reported in Sect. 3.1). Around two thirds of the samples are obtained using non-probability sampling, that most of times handpicks successful projects (in terms of popularity, code contributions, etc.), thus hampering the generalization of the results. We believe that proper methods to build representative samples for GitHub is still missing.

Longitudinal studies. Only a tiny percentage of the selected works conducted longitudinal studies, a concern that seems to persist in the analysis of Open Source [4, 9]. We believe that longitudinal studies could be useful to shed some lights on social aspects such as the change of motivation of single contributors, the interactions between users as well as the evolution of project in terms of popularity.

Variety of methodologies. Despite the growing popularity that GitHub has catalysed from the research community, some research methodologies have been overlooked. For instance, replication and comparative studies, which have not been found during this study, could enhance the findings obtained from GitHub. In particular, the former could be used to assure that previous results are reliable and valid, while the latter could inspire new research efforts by comparing previous findings from related studies.

5. THREATS TO VALIDITY

The main threats to the validity of this study concern the collection process and the selection criteria. In particular, we relied on a set of digital libraries and their query and citation support. Some works might have been ignored since they were not available or not properly linked to other works. To mitigate this issue we included an issue-by-issue browsing of top-level conferences and journals containing relevant works for our analysis. We may also have ignored some relevant works that did not fit in our selection criteria. However, given the large number of retained works after applying the selection process, we believe that the points of discussions we report in Sect. 4 are valid and can be applied to many of the works that mine GitHub. Finally, other threats to validity arise from our subjectivity for selecting the digital libraries, defining the selection criteria and understanding the original authors' point of view of the studied papers.

6. RELATED WORK

The problem of collecting and interpreting the data extracted from repositories has been studied in other platforms (e.g., GitHub [9]), Git [3] or Sourceforge [8]). These works highlight promises and perils that arise when performing the mining process and provide some recommendations to help researchers to face the process. Instead, our work presents a meta-analysis on the papers using data mined from GitHub. In this sense, our findings acknowledge that some perils reported in [9] (e.g., third-party services limitations) are actually a concern in the studied works.

We found a few works presenting meta-analysis similar to ours [13, 4, 7, 12], none of them focusing on GitHub. Also, they consider different timeframes in their studies. Interestingly enough, some of their findings are aligned to ours in GitHub. Thus, the work presented in [13] also reports on possible data inconsistencies (due to possible modifications of the Git version control system's history). In [4], authors report on methodological issues related to sampling strategies and lack of longitudinal studies. Finally, the works in [7] and [12] review the papers published in the proceedings of MSR and acknowledge that data and tool sharing are often overlooked in MSR papers, thus making harder to replicate and generalize the results.

7. CONCLUSION

In this paper, throughout a combination of systematic searches, pruning of non-relevant works and comprehensive forward and backward snowballing processes, we have identified 93 relevant works that have been analyzed in order to highlight the status of the research conducted on GitHub. Our analysis raises some concerns about the dataset collection processes and sizes, the low level of replicability, poor sampling techniques, lack of longitudinal studies and scarce variety of methodologies. To mitigate these issues, we believe that researchers should share their datasets and provide clear instructions to enable the replication of their studies, thus allowing to validate and compare previous and new results each other. Additionally, choosing a proper sampling technique is also crucial to leverage on representative GitHub projects. Finally, we believe that researchers should address other typologies of study that have been overlooked (e.g., longitudinal studies). We believe these improvements would enhance the generality, quality and confidence of the findings derived from GitHub.

8. REFERENCES

[1] K. Aggarwal, A. Hindle, and E. Stroulia. Co-evolution of project documentation and popularity within GitHub. MSR, pages 360–363, 2014.

[2] A. S. Badashian, A. Esteki, A. Gholipour, A. Hindle, and E. Stroulia. Involvement, contribution and influence in GitHub and StackOverflow. CSSE, pages 19–33, 2014.

[3] C. Bird, P. Rigby, and E. Barr. The promises and perils of mining git. In *MSR conf.*, pages 1–10, 2009.

[4] K. Crowston, K. Wei, J. Howison, and A. Wiggins. Free/libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, 44(2):7, 2012.

[5] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. ICSE, pages 422–431, 2013.

[6] G. Gousios and D. Spinellis. Ghtorrent: GitHub's data from a firehose. MSR, pages 12–21, 2012.

[7] H. Hemmati, S. Nadi, O. Baysal, O. Kononenko, W. Wang, R. Holmes, and M. W. Godfrey. The msr cookbook: Mining a decade of research. MSR.

[8] J. Howison and K. Crowston. The perils and pitfalls of mining SourceForge. In *MSR conf.*, pages 7–11, 2004.

[9] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian. An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering*, pages 1–37, 2015.

[10] M. Nagappan, T. Zimmermann, and C. Bird. Diversity in software engineering research. ESEC/FSE, pages 466–476, 2013.

[11] R. Padhye, S. Mani, and V. S. Sinha. A study of external community contribution to open-source projects on GitHub. In *11th Working Conference on Mining Software Repositories*, pages 332–335, 2014.

[12] G. Robles. Replicating msr: A study of the potential replicability of papers published in the mining software repositories proceedings. MSR, pages 171–180, 2010.

[13] A. Serebrenik and T. Mens. Challenges in software ecosystems research. ECSAW, pages 40:1–40:6, 2015.

[14] F. Thung, T. F. Bissyande, D. Lo, and L. Jiang. Network Structure of Social Coding in GitHub. In *17th European Conference on Software Maintenance and Reengineering*, pages 323–326, 2013.

[15] B. Vasilescu, V. Filkov, and A. Serebrenik. Stackoverflow and GitHub: associations between software development and crowdsourced knowledge. SocialCom, pages 188–195, 2013.

[16] J. Xavier and A. Macedo. Understanding the popularity of reporters and assignees in the GitHub. In *26th International Conference on Software Engineering and Knowledge Engineering*, pages 484–489, 2014.