A Dataset of Bot and Human Activities in GitHub

Natarajan Chidambaram
Software Engineering Lab
University of Mons
Mons, Belgium
natarajan.chidambaram@umons.ac.be

Alexandre Decan
F.R.S.-FNRS Research Associate
Software Engineering Lab
University of Mons
Mons, Belgium
alexandre.decan@umons.ac.be

Tom Mens
Software Engineering Lab
University of Mons
Mons, Belgium
tom.mens@umons.ac.be

Abstract—**Software repositories hosted on GitHub frequently use development bots to automate repetitive, effort intensive and error-prone tasks. To understand and study how these bots are used, state-of-the-art bot identification tools have been developed to detect bots based on their comments in commits, issues and pull requests. Given that bots can be involved in many other activity types, there is a need to consider more activities that they are carrying out in the software repositories they are involved in. We therefore propose a curated dataset of such activities carried out by bots and humans involved in GitHub repositories. The dataset was constructed by identifying 24 high-level activity types that could be extracted from 15 lower-level event types that were queried from GitHub's event stream API for all considered bots and humans. The proposed dataset contains around 834K activities performed by 408 bots and 655 humans involved in GitHub repositories, during an observation period ranging from 25 November 2022 to 9 March 2023. By analysing the activity patterns of bots and humans, this dataset could lead to better bot identification tools and empirical studies on how bots play a role in collaborative software development.**

Index Terms—**software development, bot activity, dataset, GitHub event stream, empirical analysis**

I. Introduction

The use of collaborative software development practices in social coding platforms such as GitHub and GitLab has become omnipresent in the last decades [1], [2]. In order to automate repetitive and error-prone tasks during such collaborative development, the use of *bots* has become prevalent [3], and *bots* even belong the top contributors in certain software projects [4].

Empirical studies on bot usage [4]–[6] have relied on state-of-the-art bot identification tools such as BoDeGHa [7] and BotHunter [8] to detect the presence of bots in software projects. These tools identify bots based on the comments provided in issues, pull requests and commits. But, just like humans, bots perform many more activities than commenting only (e.g., opening or closing pull requests and issues, reviewing code, publishing new releases) [3], [6].

This raised the need to create a dataset of specific high-level activities carried out by bots and humans in order to perform empirical studies of how bots play a role in collaborative software development. The dataset is publicly available on https://doi.org/10.5281/zenodo.7740520.

An important contribution of the dataset is that it contains historical activity data that can no longer be recovered through GitHub nor its API, and that cannot be retrieved easily from third-party datasets. The dataset is built from data queried through GitHub's Events API for individual contributors. Since this API is limited to retrieving the last 300 events of a given contributor, we iteratively queried it from 25 November 2022 to 9 March 2023 to obtain a complete list of all the events generated by 408 bots and 655 human contributors.

Another contribution of the dataset is that it exposes activities at a higher level of granularity than individual GitHub events. Mapping activities to events is not straightforward, since some activities actually correspond to sequences of distinct events and some events may correspond to multiple activities. Based on the 15 event types exposed by the API, we identified 24 activity types covering a wide variety of activities such as issues, pull requests, releases, repository management. Using these activity types, our dataset provides a consolidated and curated list of around 834K high-level contributor activities. These high-level activities aim to facilitate the characterisation of bot and human behaviour in GitHub repositories, by enabling the analysis of activity sequences and activity patterns of bot and human contributors.

II. Related Work

On bot identification. Golzadeh et al. [7] developed BoDeGHa, an ML-based bot identification tool that identifies bots that are involved in commenting issues and pull requests. The classification model was trained on a manually verified set of 5,000 GitHub accounts of which 527 were bots. To improve the bot identification capability, Chidambaram et al. [5] identified 36 additional bots by executing BoDeGHa on 1,000 repositories by leveraging the predictions from multiple repositories to determine the type of account. Abdellatif et al. [8] developed BotHunter, another ML-based bot identification tool, that extends BoDeGHa by integrating additional information from the account's profile and from the commits, issues and pull requests the account is involved in. Using this tool, they identified 679 bots in 5,000+ accounts.

On developer activities. Wang et al. [6] manually inspected the activities carried out by 230 bots in GitHub repositories. They notably found that bots are involved in and perform multiple and varied tasks such as checking pull requests, assigning and labelling issues and reviewing changes. Young et al. [9] analysed the code and non-code contributions made by the top 100 contributors present in 2,855 extracted repositories and grouped the contributions into 5 coarse categories (code, code review, issue, maintenance, Wiki/Docs). Onoue et al. [10] analysed the characteristics of Github developer activities based on their latest 300 events. They categorized developers based on measures such as whether they prefer communication by coding or comments, or whether they are specialists or generalists. Ma et al. [11] created a dataset of contributors that are involved in git commits. They cloned and extracted the git objects for 62M repositories to find relationships between projects, contributors and files.

On activity datasets. Gousios et al. [12] developed GHTorrent, a dataset mirroring large parts of the data available on GitHub. This dataset contains data about repositories, issues, pull requests, contributors, etc. While it can be used to detect some of the activities made by contributors, it does not cover all the activities that can be identified from the events generated by them. Moreover, the service is no longer actively maintained. More related to our own proposed dataset, GH Archive[1] provides a collection of all public events present in the GitHub events timeline. While this dataset can be used to reconstruct the activities made by contributors in GitHub repositories, doing so for large period of time is tricky. Indeed, hundreds of events are generated every second on GitHub and GH Archive aggregates them into hourly archives whose size ranges from 100Mb to 1000Mb. Finding all the events generated over the last 60 days, for example, would imply processing more than 1Tb of data.

III. Dataset Construction

This section details the process that was followed to create the activity dataset. Fig 1 provides a high-level summary of this process, decomposed in three steps: (A) Curating contributors, (B) Querying events, and (C) Generating activities.

A. Curating contributors

Since our goal was to gather activities made by human contributors and by bots, we needed to come up with a list of human contributors and a list of bots. To do so, we relied on four curated datasets that were used for training the bot identification tools BoDeGHa [7] and BotHunter [8] and for analysing bot usage in collaborative software development [5], [6]. These datasets were published in 2021 or 2022, hence providing a quite recent list of (manually verified) bots and human contributors. We combined all bots identified in these ground-truth datasets and removed duplicates, leading to 890 distinct bots, and we randomly selected a similar number of human contributors. Figure 2 shows the distribution of these 890 bots among the four curated datasets we relied on.

The bots in our list correspond either to GitHub accounts mimicking human contributors; or to GitHub Apps taking actions via the API on their own behalf using their own identity, without needing to maintain a separate user account for them.[2] GitHub automatically adds the *[bot]* suffix to the displayed name of GitHub Apps (e.g., *renovate[bot]*) and requires the use of this suffix when querying its API. For each bot name in our list, we queried the GitHub API to determine whether it exists as a GitHub App. If not, we checked for the presence of GitHub account corresponding to the bot name. Out of the 890 considered bots, 165 are implemented as GitHub Apps, 685 are acting through a GitHub accounts, and 40 were dismissed as they no longer exist on GitHub.

B. Querying events

Given the name of a contributor, GitHub's Events API provides all recent events that were generated by the contributor (e.g., CreateEvent when a repository is created, or IssueCommentEvent when commenting an issue). We relied on these events to generate the high-level activities. However, the Events API can only be used to retrieve up to 300 events, and only those that were generated during the last 90 days. This second limitation is not really impactful since most contributors require less than 90 days to generate 300 events. The first limitation, however, has important implications for contributors that generate more than 300 events in short periods of time, since any older events will no longer be retrieved by the API.

Since our goal was to provide all the activities performed by the considered contributors during a period of 105 days, between 25 November 2022 and 9 March 2023, we needed to iteratively and frequently query the API to ensure that no event is missed. Therefore, we queried the API every 6 hours for each contributor. To ensure that no event was missed between two consecutive calls A and B, we checked whether the oldest returned event in B was part of the events returned in A (i.e., there was no "gap" between both event sequences). In case an event was missed, we removed the corresponding contributor from our list. We found 28 human contributors and 69 bots in such a situation. We also excluded 74 human contributors and 373 bots that did not generate any event during the considered period of time. By doing so, we retrieved 1M+ distinct events for 408 bot contributors and 655 human contributors.

C. Generating activities

The third step consists in generating the activities made by these contributors based on the events they produced. To do so, we first needed to come up with a classification of high-level activities and their mapping to lower-level GitHub events. The three authors of this article carefully went over the documentation of GitHub's Events API and the various event types to identify the high-level activities that can be deduced from them. We also manually performed various activities through the GitHub UI to observe the events generated by them in order to map events and activities. Through an iterative process, we unanimously agreed on a final list of 24 high-level activities and their mapping to the generated events.

[1] https://www.gharchive.org/

[2] https://docs.github.com/en/apps/creating-github-apps/creating-github-apps/about-apps

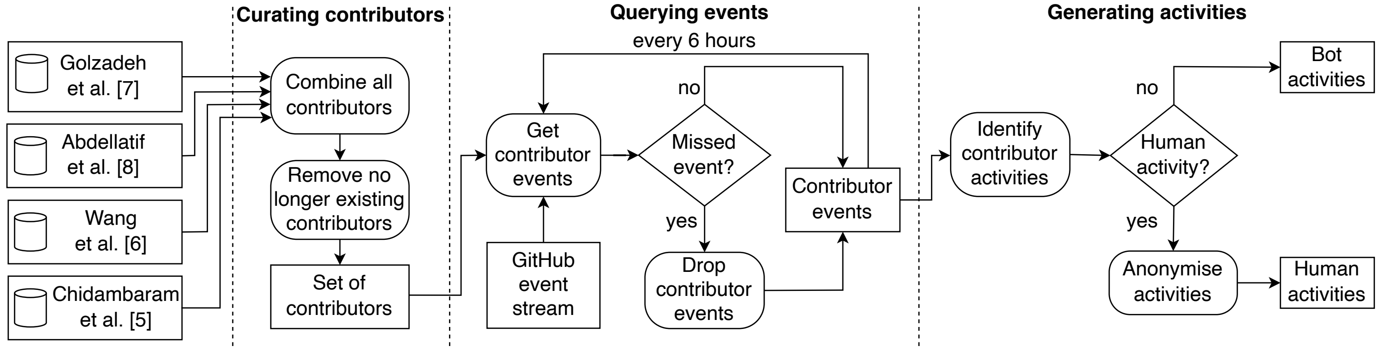

Fig. 1. Dataset construction process

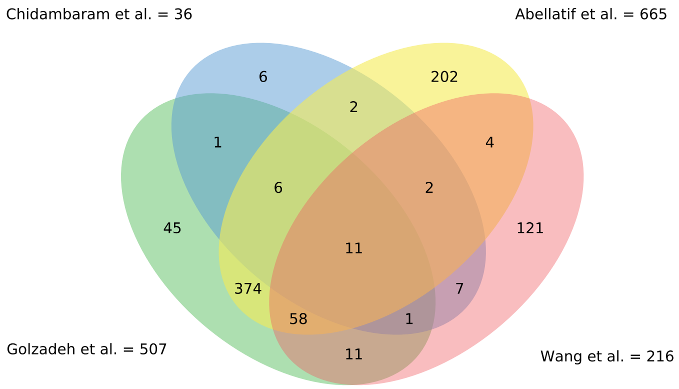

Fig. 2. Distribution of bots across the four initial datasets.

This mapping between activities and events is not one-to-one, since some activities are obtained from sequences of two events, and some event types may give rise to different activity types depending on the kind of data that is provided by the event. For example, activity type *Closing issue* (with a comment) is obtained from a combination of IssuesEvent and IssueCommentEvent event type. As another example, the CreateEvent event type can correspond to one of the activity types *Create repository*, *Create branch* or *Create tag*, depending on the value of its `ref_type` field. Table I lists the 24 activity types we identified as well as their mapping to event types. Optional events are preceded with a "?". A more detailed description of the mapping is provided alongside the shared dataset.

Using this mapping, we converted the 1M+ events from step B into 833,811 activities of which 649,755 are made by 408 bots and 184,056 activities by 655 human contributors. This already indicates that, on average, bots are considerably more active than humans. We provide the generated activities in two separate datasets: one for bots and one for human contributors. The latter dataset is anonymised to comply with GDPR regulations, by hashing the names of human contributors and the repositories they are active in, and by removing all unique identifiers that could be used to reveal their identities.

TABLE I
ACTIVITY TYPES AND CORRESPONDING EVENT TYPES

Activity type	Event type(s)
Creating repository/tag/branch	CreateEvent
Deleting branch/tag	DeleteEvent
Making repository public	PublicEvent
Adding collaborator to repository	MemberEvent
Forking repository	ForkEvent
Starring repository	WatchEvent
Editing wiki page	GollumEvent
Publishing a release	ReleaseEvent and ?CreateEvent
Opening/Transferring issue	IssuesEvent
Closing/Reopening Issue	IssuesEvent and ?IssueCommentEvent
Commenting issue	IssueCommentEvent
Opening pull request	PullRequestEvent
Closing/Reopening pull request	PullRequestEvent and ?IssueCommentEvent
Commenting pull request	IssueCommentEvent
Commenting pull request changes	PullRequestReviewCommentEvent and ?PullRequestReviewEvent
Reviewing code	PullRequestReviewEvent
Pushing commits	PushEvent
Commenting commits	CommitCommentEvent

IV. Data Schema

The activity datasets are provided as JSON files accompanied by a corresponding JSON schema. Listings 1 and 2 provide excerpts showing two activities made by a bot contributor. Along with the date and the activity type, each activity mentions the name of the contributor and the repository in which the activity took place. Depending on the activity type, additional fields are provided, the details of which can be found alongside the shared dataset. For example, activity type *Commenting issue* in Listing 2 provides additional details about the `comment` (lines 6–9), the `issue` (lines 10–18) and the `conversation` (lines 19-21) involved in the activity. Whenever available, we provide the `GH_node` (lines 8 and 17) of the corresponding objects, a globally unique identifier to find related objects (e.g., comments, issues or pull requests) on GitHub.

V. Limitations

A first limitation of the datasets stems in the range of activity types contained in them. We relied on the Events API

Listing 1. Example of a *Publishing a release* activity.

```
{
 "date": "2023-01-03T16:45:52+00:00",
 "activity": "Publishing a release",
 "contributor": "kubevirt-bot",
 "repository": "kubevirt/kubevirt",
 "release": {
   "name": "v0.59.0-alpha.2",
   "description_length": 9834,
   "created_at": "2023-01-03T15:59:12+00:00",
   "prerelease": true,
   "new_tag": false,
   "GH_node": "RE_kwDOBJIk984FO7NC"
 },
 "gitref": {
   "type": "tag",
   "name": "v0.59.0-alpha.2",
   "description_length": 0
 }
}
```

Listing 2. Example of a *Commenting issue* activity.

```
{
 "date": "2022-11-26T14:13:19+00:00",
 "activity": "Commenting issue",
 "contributor": "kubevirt-bot",
 "repository": "kubevirt/kubevirt",
 "comment": {
   "length": 255,
   "GH_node": "IC_kwDOBJIk985PKH4s"
 },
 "issue": {
   "id": 8294,
   "title": "SRIOV VF interface not found in VM",
   "created_at": "2022-08-13T11:10:06+00:00",
   "status": "open",
   "closed_at": null,
   "resolved": false,
   "GH_node": "I_kwDOBJIk985Pvz5k"
 }
 "conversation": {
   "comments": 9
 }
}
```

to identify the activities performed by contributors. However, not all activities on GitHub generate public events provided by this API (e.g., opening or participating in a Discussion, publishing a package). As a consequence, the datasets do not correspond to the *complete* set of activity types that can be performed through GitHub.

A second limitation is a consequence of the fact that the Events API returns at most 300 events and that we queried this API every 6 hours. Since our goal was to provide a complete list of activities made by contributors, we had to ensure that no event was missed between consecutive calls (see Section III-B). As a consequence, we had to drop all contributors that generated at least once more than 300 events in less than 6 hours. While this affected only 28 human contributors, bots are usually more active, and we had to exclude 69 of them. For example, the *github-actions* bot frequently takes less than a minute to generate 300 events. Therefore, and to a limited extent, our datasets are slightly biased towards contributors that are not "overly active".

A last limitation relates to the lack of reliability of some data provided by GitHub. For example, a PushEvent reports on the number of commits pushed through the `size` and `distinct_size` fields. However, we found that the values indicated in these fields do not always correspond to the actual number of commits that were pushed, likely because of rebasing and commit squashing. Another example is the `merge` status reported in a PullRequestEvent that sometimes indicates that a pull request is merged when it is not (and vice-versa). These perils are well-known by the MSR community when mining git [13] and GitHub [14] data, and there is little we can do to address them.

VI. Conclusion

We proposed a curated dataset of 833,811 activities performed from 25 November 2022 to 9 March 2023 by 408 bots (accounting for 649,755 activities) and 655 human contributors (accounting for 184,056 activities) involved in Github software repositories. The dataset was constructed by repetitively querying GitHub's Events API in order to obtain contiguous sequences of events for each contributor, to circumvent the API's limit of at most 300 events. We identified 24 activity types that could be extracted from 15 lower-level Github event types. These activity types relate to issues, pull requests, releases, branches, tags, commits, code reviews, and repository management.

Since the proposed dataset contains activities performed by bots and human contributors in software repositories, it can be used to answer relevant research questions such as:

- Can we observe significant differences in the number of repositories bots and human contributors are involved in?
- Which activity types are automated by bots, and which ones are mostly carried out by human contributors?
- What are the most frequently observed activity patterns and sequences?
- How can we identify contribution profiles based on observed activity patterns?
- Can we observe different contribution profiles for bots and human contributors?
- Which contributors are specialised towards specific subsets of activities?
- Can we forecast future contributor activities?
- How can we improve existing bot identification tools?

Acknowledgement

This work is supported by DigitalWallonia4.AI research project ARIAC (grant number 2010235), as well as by the Fonds de la Recherche Scientifique – FNRS under grant numbers F.4515.23, O.0157.18F-RG43 and T.0149.22.

References

[1] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, “Social coding in GitHub: Transparency and collaboration in an open software repository,” in *International Conference on Computer Supported Cooperative Work (CSCW)*. ACM, 2012, pp. 1277–1286.

[2] J. M. Costa, M. Cataldo, and C. R. de Souza, “The scale and evolution of coordination needs in large-scale distributed projects: implications for the future generation of collaborative tools,” in *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 3151–3160.

[3] M. Wessel, B. M. De Souza, I. Steinmacher, I. S. Wiese, I. Polato, A. P. Chaves, and M. A. Gerosa, “The power of bots: Understanding bots in OSS projects,” *International Conference on Human-Computer Interaction (CHI)*, 2018.

[4] M. Golzadeh, T. Mens, A. Decan, E. Constantinou, and N. Chidambaram, “Recognizing bot activity in collaborative software development,” *IEEE Software*, vol. 39, no. 5, pp. 56–61, 2022.

[5] N. Chidambaram, A. Decan, and M. Golzadeh, “Leveraging predictions from multiple repositories to improve bot detection,” in *International Workshop on Bots in Software Engineering (BotSE)*. IEEE, 2022.

[6] Z. Wang, Y. Wang, and D. Redmiles, “From specialized mechanics to project butlers: The usage of bots in open source software development,” *IEEE Software*, vol. 39, no. 5, pp. 38–43, 2022.

[7] M. Golzadeh, A. Decan, D. Legay, and T. Mens, “A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments,” *Journal of Systems and Software*, vol. 175, 2021.

[8] A. Abdellatif, M. Wessel, I. Steinmacher, M. A. Gerosa, and E. Shihab, “BotHunter: An approach to detect software bots in GitHub,” in *International Conference on Mining Software Repositories (MSR)*, 2022, pp. 6–17.

[9] J.-G. Young, A. Casari, K. McLaughlin, M. Z. Trujillo, L. Hébert-Dufresne, and J. P. Bagrow, “Which contributions count? Analysis of attribution in open source,” in *International Conference on Mining Software Repositories (MSR)*, 2021, pp. 242–253.

[10] S. Onoue, H. Hata, and K.-i. Matsumoto, “A study of the characteristics of developers’ activities in GitHub,” in *Asia-Pacific Software Engineering Conference (APSEC)*, vol. 2. IEEE, 2013, pp. 7–12.

[11] Y. Ma, C. Bogart, S. Amreen, R. Zaretzki, and A. Mockus, “World of code: an infrastructure for mining the universe of open source VCS data,” in *International Conference on Mining Software Repositories (MSR)*. IEEE, 2019, pp. 143–154.

[12] G. Gousios and D. Spinellis, “GHTorrent: Github’s data from a firehose,” in *Working Conference on Mining Software Repositories (MSR)*. IEEE, 2012, pp. 12–21.

[13] C. Bird, P. C. Rigby, E. T. Barr, D. J. Hamilton, D. M. German, and P. Devanbu, “The promises and perils of mining git,” in *Working Conference on Mining Software Repositories (MSR)*. IEEE, 2009, pp. 1–10.

[14] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. German, and D. Damian, “An in-depth study of the promises and perils of mining GitHub,” *Empirical Software Engineering*, vol. 21, no. 5, pp. 2035–2071, 2016.