A Dataset of Contributor Activities in the NumFocus Open-Source Community

Youness Hourri, Alexandre Decan*, Tom Mens
Software Engineering Lab, University of Mons, Belgium
Emails: {youness.hourri, alexandre.decan, tom.mens}@umons.ac.be
*F.R.S.-FNRS Research Associate

***Abstract*—Large open-source software (OSS) communities are composed of multiple interrelated projects, hosting numerous repositories involving thousands of interacting contributors. Socio-technical studies about a community's collaboration dynamics can benefit from historical data logs of the detailed activities performed by the projects' contributors.**

This paper provides an automated mapping of raw public events in GitHub repositories to structured activities that more accurately capture the intent of contributors. It also contributes a large dataset containing three years of activities of the 180K+ contributors of NUMFOCUS, a large OSS community supporting scientific research and data science. The dataset covers 58 projects, including 2.2M+ activities across 2,851 GitHub repositories. This dataset allows advanced studies of the NUMFOCUS community collaboration dynamics, and the activity mapping process enables the possibility to create and use similar datasets for other OSS communities.

***Index Terms*—open source, software community, collaborative development, contributor activities, repository mining**

I. INTRODUCTION

GitHub revolutionised community-driven software development by integrating the power of git, a decentralised version control system, into a social coding platform supporting a wide range of collaborative mechanisms such as issue tracking, pull-based development, and collaborative code reviewing. This enabled socio-technical empirical research on the interaction and collaboration dynamics of distributed development teams and OSS communities, and how specific mechanisms and tools affect factors such as productivity, quality, health, and sustainability [1], [2]. Many of these studies rely on the historical activities carried out by contributors in the GitHub repositories under study [3], [4], [5], [6], [7].

Despite the immense popularity of GitHub, and the promises of mining this very rich data source, there are many perils in doing so [8]. An important challenge that limits the ability to understand and capture the real intent of contributors, is the low-level nature in which GitHub encodes public events of repositories and user accounts.

Expanding the work of Chidambaram et al. [9], a first main contribution of this paper is therefore **an automated mapping of the raw GitHub event data into high-level activities** that capture the intent of GitHub contributors more faithfully, and its accompanying Python tooling to apply this mapping. Such a mapping enables recording meaningful activity sequences of contributors, allowing to understand the roles played by them in the projects they are involved in, and how these roles evolve over time and across projects.

The second main contribution is the use of this mapping to create **a historical dataset of activity sequences for the contributors of a large OSS community on GitHub**, called NUMFOCUS [10]. This extensive and diverse community supports major projects in scientific computing and data science, including Pandas, NumPy, scikit-learn, Matplotlib, and many others. This makes it an ideal starting point for studying complex socio-technical collaboration dynamics. The proposed dataset contains 2.2M+ activities, spanning three years (from January 2022 to December 2024) of activities for 180,935 contributors across 2,851 GitHub repositories pertaining to 58 projects.

II. RELATED WORK

GitHub allows to access the most recent public events performed by repository contributors through its REST and GraphQL APIs. Both APIs, however, impose significant practical limitations (rate limit, data retention, etc.) To overcome these limitations, alternative data sources have been proposed. While *GHTorrent* [11] provides a structured source of GitHub activities, combining raw events with metadata to enable long-term analysis of contributor and repository dynamics, it has been deprecated in 2019. Since 2011, *GH Archive* [12] provides an archive of the public timeline of millions of GitHub repositories, making them accessible for large-scale studies.

These data sources have been instrumental for many empirical studies related to OSS collaboration. Bai et al. [6] proposed a recommendation system based on event data for identifying like-minded contributors. Liao et al. [5] visualised issue-related behaviours, highlighting the centrality of issues in project management. Onoue et al. [3] categorised developers based on their activity patterns, examining preferences for coding versus communication. Lima et al. [4] derived collaboration, contribution, and social networks from GitHub events, revealing the low reciprocity of social ties, power-law distributions of collaboration and stargazing, and the influence of geographic proximity on interaction.

Despite these contributions, relying on public GitHub events suffers from several limitations: events are often encoded at a too low-level and lack a consistent structuring, making it

2574-3864/25/$31.00 ©2025 IEEE
DOI 10.1109/MSR66628.2025.00035

effort-intensive to reason about the contributors' intents. Chidambaram et al. [9] took a first step to address these issues by converting low-level GitHub event types into 24 higher-level activity types. Their dataset covered around 1,000 contributors during a five-month observation period. They leveraged this dataset to create a model and tool to distinguish bots from humans based on differences in their activity sequences [7], [13]. By scrutinising their activity generation process, we observed that it could not capture complex activities composed of more than two events, or for which the events were generated over more than 2 seconds.

In the current article, we overcome these limitations by providing a mapping and encoding of higher-level activity types than those proposed and used in [7], [9], [13]. Based on this mapping we propose a more recent dataset spanning a considerably longer activity period containing the activity sequences of a much larger collection of contributors. A key aspect of this new dataset is that it contains all the activities of a large open source community, encompassing 58 interrelated projects. This opens up the potential of enabling ecosystemic socio-technical studies of collaborative development, by analysing how contributors behave, interact and communicate over extended periods of time and across different GitHub organisations and repositories.

III. Mapping and Tooling

GitHub generates low-level *events* to log publicly visible user operations. There are in total 17 types of events, such as *PullRequestEvent*, *CreateEvent* and *PushEvent*.

Unfortunately, such low-level events often lack consistent structuring and contextual richness, making it difficult and effort-intensive to infer the intent of a contributor's tasks that generated these events. The same event type may reflect distinct higher-level actions. For example a *PullRequestEvent* corresponds to either the opening, closing, reopening or merging of a pull request; and a *CreateEvent* is generated when either a repository, branch or tag is created. Determining the exact intent of these events requires analysing their payload. As another example, some contributor tasks may generate multiple events, obscuring the contributor's intent and potentially leading to misinterpretations. For example, the activity of publishing a new release involves a *ReleaseEvent* and possibly a *CreateEvent* (to create a tag associated to the release). Similarly, closing an issue involves an *IssuesEvent* and an optional *IssueCommentEvent* (to leave a comment while closing the issue). Merging a pull request that is linked to some issue could even lead to a succession of four different event types *PullRequestEvent*, *PushEvent*, *DeleteEvent* and *IssuesEvent*, the latter being repeated for each linked issue.

To overcome these limitations, we propose a two-step mapping process that transforms raw, noisy event data into structured and more interpretable activities. Figure 1 illustrates this process, showcasing the mapping from raw GitHub **events**, over higher-level **actions** to structured **activities**. Two events *IssuesEvent* and *IssueCommentEvent* are first mapped onto two corresponding higher-level actions *ReopenIssue* and *CreateIssueComment*. Subsequently, both actions are grouped together into a single structured *ReopenIssue* activity. The full details of this mapping, and its implementation as an open-source Python-based tool can be found online.[1]

The first step of the process is a one-to-one mapping to convert GitHub **events** into higher-level **actions** corresponding to granular, more meaningful operations taking into account the event's `payload` metadata. This payload is parsed to derive the exact action that should be generated. For example, a *CreateBranch* action is generated from a *CreateEvent* whose payload contains `"ref_type":"branch"`. A *CloseIssue* action is generated from an *IssuesEvent* whose payload contains `"action":"closed"`.

The complete mapping for this step is stored as a static JSON file that lists all actions and, for each action, specifies the corresponding event, the conditions that must be met for the event to be mapped to the action, and which fields should be extracted from the event payload. The metadata of each action contains both *common* and *action-specific* fields. The common fields include the action type, a unique identifier for the event, the date of the action, the name and unique identifier of the actor, of the repository and of its organisation, if any. The action-specific fields are extracted from the event payloads and depend on the type of action being mapped. These fields are stored within the `details` field of the action. For example, actions related to issues will store fields such as the issue number, its title, its state and its author.

By using a static JSON file, the mapping process is flexible and easily modifiable, allowing for the addition of new actions and events as needed. This structure also facilitates the extraction of additional information from events in the future, or the extraction of events from other collaborative development platforms. The tool designed for this process automatically extracts relevant event data, applies the mapping rules defined in the JSON file, and generates the corresponding actions.

The second step of the mapping groups related **actions** into even higher-level structured **activities** that capture the contributor's intent more faithfully. For example, a *MergePullRequest* activity corresponds to the intent of merging a pull request and could potentially be composed of four action types *MergePullRequest*, *PushCommits*, *DeleteBranch* and *CloseIssue*.

To define these activities, the three authors iteratively and independently analysed the actions obtained in the previous step and observed the GitHub UI to come up with a list of activity types and to determine the actions corresponding to each activity type. They merged and discussed their observations, until a complete mapping between actions and activities was obtained. The actions are grouped into related activities taking into account multiple factors: the *type of action* (e.g., actions related to closing an issue, managing tags, or wiki pages belong to separate groups); the *time window* (related actions performed within a reasonable time window are grouped together, while actions outside this time window are considered separate activities); the *actor* (i.e., we

[1] https://github.com/uhourri/ghmap

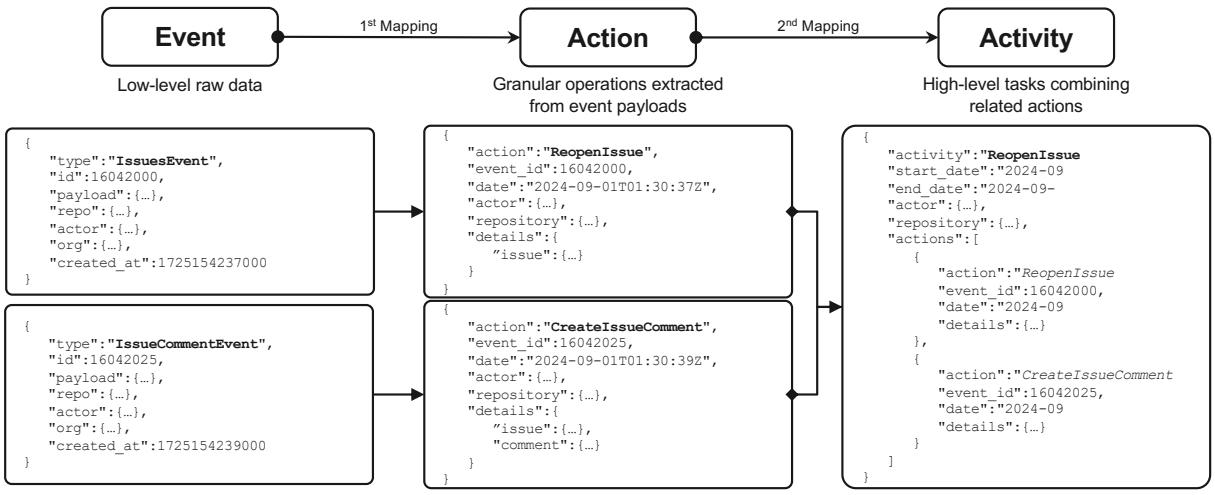

Fig. 1. Two-step mapping process to transform GitHub Events into actions and contributor activities.

do not group actions performed by distinct contributors); the *repository* (actions performed in distinct repositories are not grouped); and the *involved objects* (e.g., we do not group issue-related actions that concern different issues). We provide these relationships in a static JSON file that specifies the activities that can be performed, the actions to be grouped together, and the conditions under which they should be grouped. This structure allows for easy modification and extension if new actions or activities arise in the future.

Based on this mapping between actions and activities, we can generate sequences of activities. Each activity is represented as a JSON object containing multiple fields (see Figure 1 for an example). These fields include the type of the activity, the start and end dates of the activity, its actor, repository and organisation (as for actions), and the complete list of actions (and their details) that led to the activity.

IV. Dataset Creation and Description

The creation of a dataset capturing the activities of all contributors involved in a large OSS community is motivated by the need to better understand the collaboration dynamics within such an ecosystem. Several requirements guide the creation of our dataset. First, we want to target a community that involves different interrelated OSS projects. Second, we want each of these projects to cover multiple GitHub repositories. These two requirements aim to increase the likelihood of having partially overlapping sub-communities and evolving contributors' roles both over time and across projects and repositories. We also want the community to be composed of a huge number of contributors of different nature, ranging from very active "core" developers, over occasional contributors, to "peripheral" ones [14], [15]. Finally, we want the dataset to span a sufficiently long time period, in order to be able to infer change patterns in the collaboration dynamics over time.

We selected the NumFocus OSS community as our case study to meet these requirements. NumFocus is a non-profit organisation supporting widely used OSS projects for scientific computing and data science. It contains 60+ projects, including prominent ones such as NumPy, pandas, SciPy and Matplotlib, exhibiting multiple cross-project dependencies. For instance, pandas builds on NumPy for data manipulation, while Matplotlib is a primary library for visualising pandas dataframes. Next to these well-established projects, NumFocus also comprises emerging ones like DataTables, hence covering a broad spectrum of collaborative contributor dynamics. NumFocus is also a very vivid community, with thousands of people involved in, at different activity levels and focusing on different types of activity. This makes NumFocus an ideal choice as a starting point for our dataset.

To create the dataset we followed three steps:
(1) Identifying projects and organisations. We manually extracted, on 22 October 2024, the names of all NumFocus-sponsored projects listed on their website. We excluded three projects, conda-forge, bioconductor and Open Journals, since they correspond to large collections (or forges or registries) of packages or journals, and do not constitute real software development projects themselves. We mapped the remaining 58 projects to their corresponding GitHub organisations and obtained the list of related repositories.
(2) Collecting public GitHub events. Next, we extracted all public events in all repositories contained in these GitHub organisations. To circumvent the practical limitations of the GitHub API, we relied on Google BigQuery to consult the *GH Archive* service that archives public GitHub events since 2011. We obtained 2,716,910 events belonging to 2,851 GitHub repositories spread over 58 GitHub organisations and made between January 2022 and December 2024 (3 years).
(3) Mapping events to activities. We used the two-step mapping process of Section III to convert these events into actions and activities. First, we converted the events into 2,716,910 actions belonging to 24 action types. Second, we grouped these

TABLE I
DESCRIPTIVE STATISTICS OF DATASET

	mean	5%	25%	median	75%	95%
58 projects						
repositories	49	11	20	35	55	124
contributors	4,345	239	837	2,176	6,014	15,541
activities	39,281	3,422	12,467	28,180	60,726	123,198
activity types	19	18	20	20	21	21
2,851 repositories						
contributors	106	1	3	8	27	253
activities	799	1	6	41	240	2,328
activity types	8.36	1	3	8	13	17
180,935 contributors						
repositories	1.7	1	1	1	1	4
activities	12.6	1	1	1	2	9
activity types	1.5	1	1	1	1	4
5% most active contributors						
repositories	7	1	2	4	9	21
activities	199	9	11	17	41	579
activity types	5.6	1	3	5	7	13

actions into 2,278,299 activities belonging to 21 activity types.

At the end of this process, the dataset includes detailed activity records for 180,935 distinct contributors active in 2,851 repositories belonging to 58 projects. The dataset is publicly accessible[2] and stored in a user-friendly JSON Lines (`.jsonl`) format that is suitable for data analysis purposes. We provide two files: a first one containing JSON objects representing all *actions* (step 1 of the mapping), and another one containing JSON objects representing all *activities* (step 2 of the mapping). Records are provided chronologically, facilitating efficient exploration of activity patterns over time.

Table I provides some descriptive statistics of the dataset. One can observe rather skewed data distributions, reflected by a marked difference between mean and median values. Half of the projects (between the 25th and 75th percentile) contain between 20 and 55 active repositories, involving between 837 and 6,014 contributors, taking part in between 12,467 and 60,726 activities each. However, we also observe that the large majority of contributors have a limited number of activities. For instance, 95% of the contributors have no more than 9 activities and more than 75% of all contributors even have less than 2 activities. This observation is in line with empirical evidence that a small fraction of developers is responsible for performing the majority of work [14], [16]. The 5% most active contributors were active in 7 repositories on average, generating 199 activities belonging to 5.6 activity types. They were most frequently involved in pushing commits (18% of their activities), in commenting, reviewing, opening and merging pull requests (resp. 19%, 16%, 8% and 7%), and in commenting issues (14%). On the other hand, the 95% less active actors were mostly involved in starring repositories (54% of their activities), forking repositories (15%) and commenting or opening issues (resp. 14% and 9%).

V. USAGE AND LIMITATIONS

The NUMFOCUS activity dataset provides an essential resource for in-depth empirical studies of OSS community dynamics. It supports the analysis of contributor roles and their evolution. By examining how roles and activities are distributed among contributors, researchers can compare collaborative structures across projects, uncovering variations in teamwork dynamics. Furthermore, the dataset enables the identification of key players in the community, such as core maintainers or communication facilitators across projects.

Despite its utility, the current dataset has some limitations that could be addressed in future work. The three-year observation period does not allow to detect long-term change trends in the dynamics of contributions but can be easily extended. The dataset is exclusively based on activities obtained from GitHub event sequences. It could be extended to include activities occurring outside GitHub such as mailing lists, communication channels, developer fora, issue trackers, or even other collaborative platforms such as GitLab or BitBucket. Including such sources would provide a more accurate ecosystemic view of contributor activity.

Finally, datasets similar to NUMFOCUS could be extracted in order to perform comparative studies of collaboration dynamics across diverse ecosystems, providing insights about which factors in community collaboration lead to improved productivity, sustainability, community cohesion, and so on. Such insights could inform better tooling and strategies to enhance community engagement, improve project governance, and promote long-term community sustainability and health.

VI. CONCLUSION

We presented **an automated mapping of low-level public GitHub events to extract structured activity sequences** for project contributors. Such activities offer a higher-level representation of contributor intents, enabling the analysis of the roles played by contributors in the projects of which they are part, as well as how they interact with their collaborators within and across different projects.

We provided the necessary tooling to apply this mapping to any collection of GitHub events generated by contributors, repositories or organisations, provided that the public event data is available for them, for example by using the GitHub API or the *GH Archive* service.

We used this mapping to create **a large historical dataset of activities for all contributors to the NUMFOCUS OSS scientific community** hosted on GitHub. The dataset spans three years of activity data and contains 2,278,299 activities, categorised into 21 unique activity types, performed by 180,935 distinct contributors across 2,851 GitHub repositories and 58 projects. The mapping and dataset can be easily extended to other collaborative development platforms in order to come to a generic and platform-agnostic representation of contributor activity sequences.

ACKNOWLEDGEMENTS

This research is supported by F.R.S.-FNRS research projects F.4515.23 and J.0147.24.

[2]https://doi.org/10.5281/zenodo.14230406

REFERENCES

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

[2] G. Gousios, M.-A. Storey, and A. Bacchelli, "Work practices and challenges in pull-based development: The contributor's perspective," in *International Conference on Software Engineering (ICSE)*, 2016, pp. 285–296.

[3] 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.

[4] A. Lima, L. Rossi, and M. Musolesi, "Coding together at scale: GitHub as a collaborative social network," in *International AAAI conference on weblogs and social media*, vol. 8, no. 1, 2014, pp. 295–304.

[5] Z. Liao, D. He, Z. Chen, X. Fan, Y. Zhang, and S. Liu, "Exploring the characteristics of issue-related behaviors in GitHub using visualization techniques," *IEEE Access*, vol. 6, pp. 24 003–24 015, 2018.

[6] S. Bai, L. Liu, H. Liu, M. Zhang, C. Meng, and P. Zhang, "Find potential partners: A GitHub user recommendation method based on event data," *Information and Software Technology*, vol. 150, p. 106961, 2022.

[7] N. Chidambaram, T. Mens, and A. Decan, "RABBIT: A tool for identifying bot accounts based on their recent GitHub event history," in *International Conference on Mining Software Repositories (MSR)*. IEEE, 2024, pp. 687–691.

[8] 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*, vol. 21, pp. 2035–2071, 2016.

[9] N. Chidambaram, A. Decan, and T. Mens, "A dataset of bot and human activities in GitHub," in *International Conference on Mining Software Repositories (MSR)*. IEEE, 2023, pp. 465–469.

[10] NumFocus, "Numfocus: Open code, better science," https://numfocus.org, accessed: 2024-10-22.

[11] G. Gousios, "The GHTorrent dataset and tool suite," in *Working Conference on Mining Software Repositories (MSR)*. IEEE, 2013, pp. 233–236.

[12] GH Archive, "GH Archive: Open source GitHub data for research and analysis," https://www.gharchive.org.

[13] N. Chidambaram, T. Mens, and A. Decan, "A bot identification model and tool based on GitHub activity sequences," *Journal of Systems and Software*, vol. 221, March 2025.

[14] K. Crowston, K. Wei, Q. Li, and J. Howison, "Core and periphery in free/libre and open source software team communications," in *Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2006.

[15] C. Jergensen, A. Sarma, and P. Wagstrom, "The onion patch: migration in open source ecosystems," in *European Conference on Foundations of Software Engineering (FSE)*. ACM, 2011, pp. 70–80.

[16] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002.