

Bot Detection in GitHub Repositories

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

Pooya Rostami Mazrae
pooya.rostamimazrae@umons.ac.be
Software Engineering Lab, University of Mons
Mons, Belgium

ABSTRACT

Contemporary social coding platforms like GitHub promote collaborative development. Many open-source software repositories hosted in these platforms use machine accounts (bots) to automate and facilitate a wide range of effort-intensive and repetitive activities. Determining if an account corresponds to a bot or a human contributor is important for socio-technical development analytics, for example, to understand how humans collaborate and interact in the presence of bots, to assess the positive and negative impact of using bots, to identify the top project contributors, to identify potential bus factors, and so on. Our project aims to include the trained machine learning (ML) classifier from the BoDeGHa bot detection tool as a plugin to the GrimoireLab software development analytics platform.

In this work, we present the procedure to form a pipeline for retrieving contribution and contributor data using Perceval, distinguishing bots from humans using BoDeGHa, and visualising the results using Kibana.

ACM Reference Format:

Natarajan Chidambaram and Pooya Rostami Mazrae. 2022. Bot Detection in GitHub Repositories. In *19th International Conference on Mining Software Repositories (MSR '22)*, May 23–24, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3524842.3528520>

1 INTRODUCTION

Social coding platforms like GitHub promote collaboration and interaction between developers [1]. Along with this opportunity for engagement, developers also face some workload in performing error-prone, time-intensive or repetitive tasks such as conducting quality checks, testing, code reviewing, merging, building, deploying, and so on [6]. Thus, the developer overload increases as the frequency of these tasks increases [7]. Bots (machine accounts that act with as minimal human intervention as possible) are therefore frequently used to face this ever-increasing complexity in software development [2].

There are many bot accounts that are built by software developers and are used only in a specific set of repositories. For instance, the *highfive* account is responsible for greeting and assigning issues to contributors in *servo/servo*, one of the largest packages distributed through the Cargo package manager. Similarly, the *bors-diem* bot

is managing the merging of pull requests (PR) in the *diem/diem* Cargo package. However, they are not evidently identified as bots as they neither have *bot* in their name nor their GitHub description/bio¹ conveys this explicitly. The challenge of identifying bot accounts makes it difficult to conduct socio-technical analysis that need to distinguish human from bot behaviour, while doing so would be valuable for researchers to better understand the positive and negative impact of bots in software development, as well as for practitioners and organizations that want to accredit human project contributors [3].

The GrimoireLab software development analytics toolkit² provides *data retrieval* capability with tools such as Perceval and Graal, *data enrichment* using tools such as HatStall and SortingHat (for merging duplicate contributor identities), and *data visualisation* using tools such as Kidash and KiBiter (based on Kibana dashboards for visualising Elasticsearch data). Adding a bot identification model as one of GrimoireLab's data enrichment components would enable GrimoireLab users to collect data from GitHub repositories, identify the bot accounts, and visualise/analyse socio-technical collaborative development activities using a single platform.

As part of the MSR 2021 Hackathon³, we propose the creation of such an end-to-end pipeline (illustrated in Figure 1) that integrates BoDeGHa [4], a tool to identify bots in GitHub repositories on the basis of their PR and issue commenting activities. BoDeGHa comes with a trained machine learning classifier that could be added as a part of the GrimoireLab pipeline.

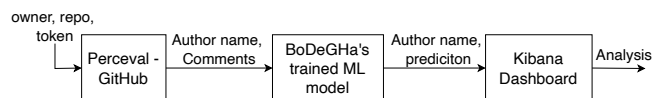


Figure 1: Integrating BoDeGHa in the GrimoireLab pipeline

2 APPROACH AND PRELIMINARY RESULTS

Since we aim to improve GrimoireLab's bot detection capability by integrating BoDeGHa's trained machine learning classifier (hereafter addressed as BoDeGHa), we need to execute the pipeline of Figure 1 with a set of GitHub repositories. This section presents a step-by-step process that needs to be followed to query, predict and visualise the number of contributors along with their type.

In the first step, we pass the GitHub repository and owner name along with the GitHub API token as an argument to Perceval for querying issues and pull requests (PRs) in the corresponding repository. The tool returns the comments present in each issue and PR along with user- and repository-specific information. Next, we

¹<https://github.com/bors-servo> and <https://github.com/libra-action>

²<https://chaoss.github.io/grimoirelab/>

³Team: NAP; repository: <https://github.com/pooya-rostami/Hackathon-21>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9303-4/22/05...\$15.00

<https://doi.org/10.1145/3524842.3528520>

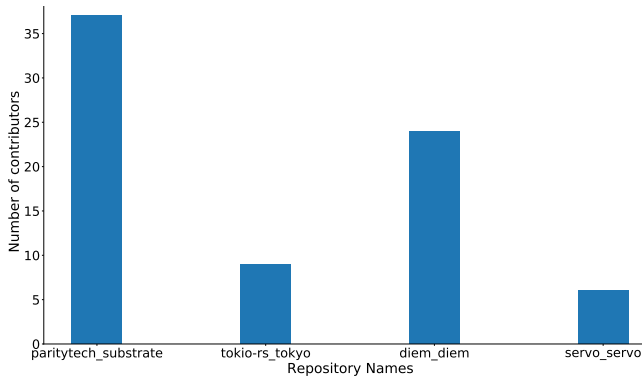


Figure 2: Number of contributors present in each repository

extract the fields *issue/PR number*, *comments*, *created at*, and *corresponding author names* and save them in a CSV file to execute BoDeGHa bot identification tool. The ML model predicts the type of contributor (*bot* or *human*) in the corresponding repository based on repetition and patterns in the comments made by the contributors in issues and PRs. The contributor names along with their predicted type (*bot* or *human*) are stored in a CSV file. In the last step, we use Kibana (an open user interface that is used for visualising Elasticsearch data) to present BoDeGHa’s prediction results.

To illustrate this process of integrating BoDeGHa into the GrimoireLab pipeline, we use four large GitHub repositories of Cargo packages, namely, *diem/diem*, *servo/servo*, *SergioBenitez/Rocket*, and *paritytech/substrate*. Due to their size and popularity, these repositories are likely to use bots in their collaborative software development process.

Figure 2 shows, without using the integrated bot prediction tool, the number of contributors that were actively posting comments in each repository between December 2021 and January 2022. In contrast, Figure 3 provides a fine-grained view by taking into account the type of contributor (*human* or *bot*) in each repository. We observe a proportionally high number of bots in three of the considered repositories: 6 out of 37 contributors are bots in *paritytech/substrate*, 8 out of 24 in *diem/diem* and 2 out of 6 in *servo/servo*. Also, it can be observed that all the issue and PR comments in *tokio-rs/tokio* were made by human contributors.

3 GOING FURTHER

Knowing the types of contributors within a repository could be leveraged for many other types of visualisation through the Kibana dashboard. For example, it would be useful to come up with visualisations that reveal whether bots are among the most active contributors within a given software repository. Similarly, one could provide visualisations involving data from multiple repositories, in order to understand how widespread specific bots are being used.

Given that BoDeGHa can occasionally produce incorrect classifications, it would also be useful to provide an interface in the BoDeGHa plugin to allow an operator to rectify misclassified contributor types (in a CSV file that has the predictions) before visualising the results.

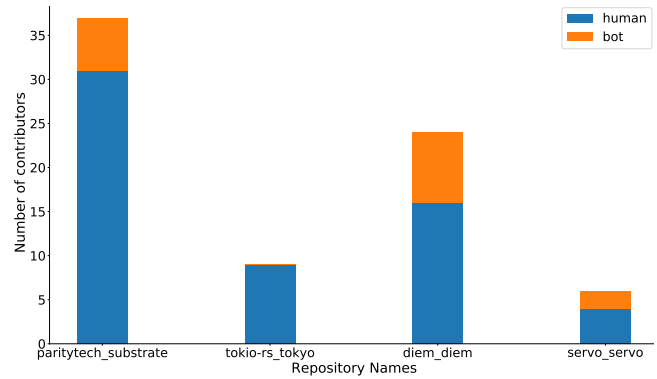


Figure 3: Number and type of contributors present in each repository

In a similar way, the BoDeGHa plugin could be integrated with SortingHat, GrimoireLab’s component in charge of managing contributors’ data (including the flag to mark them as bot) and of adding contributors’ details in the data shown in Kibiter [5].

4 CONCLUSION

Bots are being prevalently used in collaborative software development on GitHub to face ever-increasing complexity, by automating effort-intensive and repetitive activities. Identifying these bots is valuable for researchers performing socio-technical analysis in software development, as well as for practitioners and organizations that want to accredit human contributors. BoDeGHa is one such tool that identifies bot accounts based on their commenting activity in issues and pull requests. As part of the MSR 2022 Hackathon, we integrated BoDeGHa’s trained machine learning classifier into the GrimoireLab pipeline, by extracting repository data with Perceval, identifying bot contributors with BoDeGHa, and visualising the results with Kibana dashboard.

ACKNOWLEDGMENTS

This work is supported by DigitalWallonia4.AI research project ARIAC (grant number 2010235), as well as by the ARC-21/25 UMONS3 Action de Recherche Concertée financée par le Ministère de la Communauté française – Direction générale de l’Enseignement non obligatoire et de la Recherche scientifique

REFERENCES

- [1] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In *International Conference on Computer Supported Cooperative Work*. 1277–1286. <https://doi.org/10.1145/2145204.2145396>
- [2] Linda Erlenhov, Francisco Gomes de Oliveira Neto, Riccardo Scandariato, and Philipp Leitner. 2019. Current and Future Bots in Software Development. In *International Workshop on Bots in Software Engineering (BotSE)*. IEEE, 7–11. <https://doi.org/10.1109/BotSE.2019.00009>
- [3] Mehdi Golzadeh, Alexandre Decan, and Natarajan Chidambaram. 2022. On the accuracy of bot detection techniques. In *International Workshop on Bots in Software Engineering (BotSE)*. IEEE.
- [4] Mehdi Golzadeh, Alexandre Decan, Damien Legay, and T. Mens. 2021. A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments. *Journal of Systems and Software* 175 (2021). <https://doi.org/10.1016/j.jss.2021.110911>
- [5] David Moreno, Santiago Dueñas, Valerio Cosentino, Miguel Angel Fernandez, Ahmed Zerouali, Gregorio Robles, and Jesus M. Gonzalez-Barahona. 2019. SortingHat: Wizardry on Software Project Members. In *International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 51–54. <https://doi.org/10.1109/ICSE-Companion.2019.00036>
- [6] Mairieli Wessel, Bruno Mendes De Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The power of bots: Understanding bots in OSS projects. *The ACM International Conference on Human-Computer Interaction* (2018). <https://doi.org/10.1145/3274451>
- [7] Jean-Gabriel Young, Amanda Casari, Katie McLaughlin, Milo Z. Trujillo, Laurent Hébert-Dufresne, and James P. Bagrow. 2021. Which contributions count? Analysis of attribution in open source. *International Conference on Mining Software Repositories (MSR)* (2021), 242–253. <https://doi.org/10.1109/MSR52588.2021.00036>