Predicting the Impact of Using Bots in Collaborative Software Development*

Thesis Start: August 2021, Expected Defense: December 2024
Supervised by: Prof. Tom Mens (tom.mens@umons.ac.be) and Prof. Souhaib Ben Taieb (souhaib.bentaieb@umons.ac.be)

Natarajan Chidambaram[0000−0002−2295−8928]

University of Mons, Belgium
natarajan.chidambaram@umons.ac.be

Abstract. Contemporary social coding platforms such as GitHub promotes collaborative software development. These software projects tend to use bots to automate few tasks and reduce developer workload. However, this is not always true, bots tend to be noisy at times. This paper covers the thesis statement, briefly explains the research goals and questions to analyse and estimate bot behavior in software projects, past work and immediate future work along with the methodology.

1 Introduction

Open Source Software (OSS) development has become increasingly popular in the last decade especially among organizations [16, 15] and increased empirical research in the field of software evolution and software quality [24]. Social coding platforms like GitHub promotes collaborative software development and is used for developing most of the OSS projects. This brings collaboration at social (E.g., interactions) and technical level (E.g., source code) beyond small groups [17, 5]. The interactions might happen through comments under issues pull requests (PR) and so on. In the case of small projects, such interactions and contributions are scalable and the project maintainers are able to handle these along with their technical dependencies [25]. But in large projects, it is difficult to keep up with the pace of maintaining high-quality software releases along with their social and technical dependencies. So, to automate effort-intensive and repetitive activities, automated tools (e.g., apps and actions - distributed on the GitHub marketplace) and bots (machine accounts) are used [8, 22, 20].
However, empirical software engineering researchers who are studying the socio-technical aspects of software development find the presence of bots in the OSS projects a bit challenging. For example, it is most likely to have biased results during the analysis of the project productivity, quality of handling the bugs, pull requests and so on. So, it is better to consider the presence of bots and

* Supported by "Service Public de Wallonie Recherche under grant n° 2010235 - ARIAC by DIGITALWALLONIA4.AI"

treat them differently than humans [13, 12]. There have been many techniques and tools to detect the presence of bots. BIMAN [6] combines three different approaches to recognise bots in commits: (i) the presence of the string "bot" at the end of the author name, (ii) repetitive commit messages, and (iii) features related to files changed in commits. BoDeGHa [12] analyses comments posted in issues and pull requests to detect bots, based on the assumption that bots tend to frequently use a limited set of comment patterns. BotHunter [1] identifies the bots using a combination of features used in BoDeGHa and BIMAN. It is also known that bots perform 12 different tasks in a repository such as verifying license agreement, code review, dependency checks, merging PR, etc. [22, 7].
Bots make some positive contributions towards the collaborative software development process. For example, they perform regular dependency checks, merge the code and so on [2, 9, 14, 22]. However, a few bots are known to create some negative impacts depending on their usage [19, 23] and the bot challenges faced by developers are studied by Wessel et al. [23]. For example, they provide non-comprehensive feedback that requires a human to clarify it, giving too much information to developers, overwhelming the maintainers with pull request notifications and so on. To reduce this noise, Wessel et.al [21] developed a meta-bot that summarises the information from many other used bots before presenting it to the developer (could mitigate the information overload). Moreover, Brown and Parnin [2] developed a tool-recommender bot that recommends software engineering tools such as *FindBugs* to developers on GitHub. They analysed the developers reaction by creating a pull request for this recommendation tool, but the results were not promising as the PR was accepted by only 2 projects.

2 Research Goal

Bot-based studies are being performed by various researchers [13, 12, 6, 1, 21]. However, developing a model/tool that predicts the advantages/disadvantages that a software project could have upon using a bot would help the developers/maintainers to adopt a bot accordingly (E.g., predicting the noise).
*The goal of this PhD is to analyse the behaviour of bots in collaborative software development and develop AI models to predict their impact in software development.*Here, behaviour could be the type of tasks, number of tasks and the response time for a bot. Whereas, impact can be either negative (E.g., frequently tagging the developer, posting too many comments, providing too much information) or positive (E.g., clear code review comments, low number of false PR rejections) depending on the type of task and their usage in a software project. I divided the main goal of this thesis into the following research questions:

1. *How can we identify bots more accurately than the current methods/tools?* Existing bot identification tools are known to have many false positives and false negatives, so having a better bot identification tool would help me to get more bots and analyse more repositories in which they are being used. I shall extend and improve the current tools for better bot identification.

2. *How can we estimate the impact made by a bot in a software project?* Impact made by certain task-specific bots were studied in previous work. However, a measure to quantify the impact is yet to be defined. Also, a behaviour is noisy for a particular task but not for another task (e.g., comments - posting comment about the issue is good, but posting comment about assigning an issue to a developer is not good). I aim to define metrics to quantify the impact made by bots. Features for the metric would be obtained through qualitative study (surveys in the developer's community). Through the survey, I would know the characteristics that can be considered as a behaviour of a bot, and if a behaviour causes positive or negative impact for the considered task. Finally, estimation can be obtained by developing AI model (binary classification and probabilistic model).
3. *To which extent can we develop a bot recommendation model?* Cumulating all the results, I could recommend if a software project should use/remove/change a bot or its configuration to use the bot more efficiently.

My research could help developers to adopt a suitable bot for their required task knowing its impact. This understanding would eventually let the developer evaluate the advantages (E.g., pace of development, efficiency, quality etc.) of using that particular bot. Also, this research could highlight the lack of bots for performing certain tasks and might identify room for improving their performance.

3 Past Work

As the first task to answer RQ1 (question 1 in Research Goals), I collected a dataset of OSS project repositories that correspond to the development of Rust programming language and are distributed through the Cargo package manager. The reason for choosing Cargo package manger is that a large majority of its packages are hosted on GitHub, have packages that are developed in collaborative manner and has high probability to find the presence of development bots. I obtained relevant metadata of the repositories such as number of issues, PRs etc. After the filtering process such as - has documentation or homepage and number of issues and PRs greater than 100 (as bigger projects are probable to use bots), I ended up having 1039 GitHub repositories. Then I executed BoDeGHa [10] on these repositories and observed 239 repositories to use bot for their development. Among the 8,532 active contributors, 2,861 (33.5%) were active in multiple repositories, in which 229 contributors were classified as bot at least in one repository among which 123 contributors were classified as *bot* in one repository and *human* or *unknown* in another repository. So, I hypothesized that leveraging the predictions across multiple GitHub repositories would improve the prediction accuracy. The ground truth for the type of contributor was formed through an inter-rater agreement between the co-authors. I applied the wisdom of the crowd principle to leverage these predictions (WoC-P) and found this technique to improve the accuracy by 6.9% when used on top of BoDeGHa. By extending this finding, we found presence of bots in more repositories. This

gave us more data regarding bots to study the tasks and the ways in which they are being utilized in OSS projects. This led to a publication in BotSE'22 (Bots in Software Engineering, May 2022), a workshop in conjunction with ICSE'22 (International Conference on Software Engineering) [3].
Also, I published a short paper at the Hackathon track of MSR'22 (Mining Software Repositories) [4]. The GrimoireLab software analytics toolkit provides data retrieval capability, data enrichment and data visualisation using various tools. However, we noticed that a bot detection tool was not present in Griemoire-Lab and hypothesized that its presence would be useful for practitioners and researchers. As BoDeGHa comes with a trained machine learning classifier, we proposed that this model could be integrated into GrimoireLab's pipeline. The notebook to support its functionality was published via GitHub [1].
I was also a co-author in another paper that was submitted to BotSE'22 [11]. In this research paper, we performed an exploratory study on the accuracy of 5 bot detection techniques on a set of 540 accounts from 27 GitHub repositories. We created an ensemble model by combining the 5 bot detection techniques and found that it works better than any of the considered bot detection techniques with an average recall of 0.9 and average precision of 0.865 for bots.

4 Going Further

In the coming months, I aim to expand the observations in my previous work by answering the following questions:

1. *How prevalent are bots in GitHub repositories?* Execute the bot identification tools in top-star rated GitHub repositories to identify number of bot accounts
2. *What are the various tasks performed by bots in these popular repositories?* I will manually analyse the account details and form a data-set with the tasks that each bot account does in software projects
3. *How to extract all the possible configuration details of a bot?* Through automated scripts, I would find the .yml file for the bot accounts hosted through GitHub and extract the configuration details from all the repositories
4. *What are the features that represents the behaviour of bot (different features for different tasks) in GitHub repositories? and what is their impact?* I would contact the developers and maintainers of OSS projects to get their opinion through surveys. Survey questionnaire would include which characteristics do they consider to be behaviour of a bot and its impact in projects
5. *Does similar repositories use bots for the same tasks? Do they use same/similar bots? Do they use those bots with same/similar configuration?* I would define metrics for measuring repository similarity and conduct empirical analysis to answer this question
6. *Can we predict the behaviour of a bot by combining all the observations obtained in previous questions (bot's configuration, repository's metric and its similarity with other repositories that use the same or similar bot)?* I would

[1] https://github.com/Natty07/MSR-Hackathon-21

use approaches such as Naive Bayes probabilistic model, deep neural networks and predictive models (e.g., Random Forest) to answer this question.

For the data set, I would depend on GitHub repositories and Cargo package manager. William et al. [18] has collected 7 years of data related to Cargo package manager and classified the type of all the accounts. Also, we would depend on the ground-truth data set formed by Mehdi et al. [10]. If required, I will collect more data using GitHub's GraphQl API and form the ground-truth for the new accounts before proceeding with the analysis. For the intermediate and the final results, I would like to publish my work in ICSE'23 or SANER'23 for which the submission deadline is Sept'22 and Oct'22 respectively and another question at MSR'23 for which the deadline is expected to be in Feb'23.

References

1. Abdellatif, A., Wessel, M., Steinmacher, I., Gerosa, M.A., Shihab, E.: BotHunter: An Approach to Detect Software Bots in GitHub. In: Mining Software Repositories conference. Association for Computing Machinery (5 2022)
2. Brown, C., Parnin, C.: Sorry to Bother You: Designing Bots for Effective Recommendations. In: International Workshop on Bots in Software Engineering (BotSE). IEEE (5 2019). https://doi.org/10.1109/BotSE.2019.00021
3. Chidambaram, N., Decan, A., Golzadeh, M.: Leveraging Prediciotns from Multiple Repositories to Improve Bot Detection. In: International Workshop on Bots in Software Engineering (BotSE). Association for Computing Machinery (5 2022)
4. Chidambaram, N., Rostami Mazrae, P.: Bot Detection in GitHub Repositories. In: International Conference on Mining Software Repositories (MSR). Association for Computing Machinery (5 2022). https://doi.org/10.1145/3524842.3528520
5. Constantinou, E., Mens, T.: An empirical comparison of developer retention in the RubyGems and npm software ecosystems. Innovations in Systems and Software Engineering **13**(2) (9 2017). https://doi.org/10.1007/s11334-017-0303-4
6. Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., Mockus, A.: Detecting and Characterizing Bots That Commit Code. In: International Conference on Mining Software Repositories (MSR). Association for Computing Machinery (9 2020). https://doi.org/10.1145/3379597.3387478
7. Dey, T., Vasilescu, B., Mockus, A.: An Exploratory Study of Bot Commits. International Workshop on Bots in Software Engineering (BotSE) (5) (6 2020). https://doi.org/10.1145/3387940.3391502
8. Erlenhov, L., Neto, F.G.d.O., Leitner, P.: An Empirical Study of Bots in Software Development: Characteristics and Challenges from a Practitioner's Perspective. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ESEC/FSE 2020, Association for Computing Machinery (11 2020). https://doi.org/10.1145/3368089.3409680
9. Erlenhov, L., Gomes de Oliveira Neto, F., Scandariato, R., Leitner, P.: Current and Future Bots in Software Development. In: International Workshop on Bots in Software Engineering (BotSE). pp. 7–11. Association for Computing Machinery (5 2019). https://doi.org/10.1109/BotSE.2019.00009

10. Golzadeh, M.: Analysing Socio-Technical Congruence in the Package Dependency Network of Cargo. In: European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Association for Computing Machinery (8 2019). https://doi.org/10.1145/3338906.3342497
11. Golzadeh, M., Decan, A., Chidambaram, N.: On the accuracy of bot detection techniques. In: International Workshop on Bots in Software Engineering (BotSE). IEEE, Association for Computing Machinery (5 2022)
12. Golzadeh, M., Decan, A., Legay, D., Mens, T.: A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments. Journal of Systems and Software **175** (5 2021). https://doi.org/10.1016/j.jss.2021.110911
13. Golzadeh, M., Legay, D., Decan, A., Mens, T.: Bot or Not? Detecting Bots in GitHub Pull Request Activity Based on Comment Similarity. In: International Workshop on Bots in Software Engineering (BotSE). Association for Computing Machinery (6 2020). https://doi.org/10.1145/3387940.3391503
14. Lebeuf, C., Storey, M.A., Zagalsky, A.: Software Bots. IEEE Software **35** (12 2017)
15. McClean, K., Greer, D., Jurek-Loughrey, A.: Social network analysis of open source software: A review and categorisation. Information and Software Technology **130** (2 2021). https://doi.org/https://doi.org/10.1016/j.infsof.2020.106442
16. Nagle, F., Wissel, J., Zaggl, M., Lifshitz-Assaf, H., Shaikh, M., Tambe, P., Wang, S.: Open Source Software Development and Organizations. Academy of Management Proceedings (08 2019). https://doi.org/10.5465/AMBPP.2019.18072symposium
17. Palyart, M., Murphy, G.C., Masrani, V.: A Study of Social Interactions in Open Source Component Use. IEEE Trans. Softw. Eng. **44** (12 2018). https://doi.org/10.1109/TSE.2017.2756043
18. Schueller, W., Wachs, J., Servedio, V.D., Thurner, S., Loreto, V.: Evolving Collaboration, Dependencies, and Use in the Rust Open Source Software Ecosystem. arXiv (5 2022)
19. Storey, M.A., Zagalsky, A.: Disrupting developer productivity one Bot at a Time. ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE) (11 2016). https://doi.org/10.1145/2950290.2983989
20. Urli, S., Yu, Z., Seinturier, L., Monperrus, M.: How to design a program repair bot: Insights from the repairnator project. International Conference on Software Engineering (ICSE) (5 2018). https://doi.org/10.1145/3183519.3183540
21. Wessel, M., Abdellatif, A., Wiese, I., Conte, T., Shihab, E., Gerosa, M., Steinmacher, I.: Bots for Pull Requests: The Good, the Bad, and the Promising. International Conference on Software Engineering (ICSE) (5 2022)
22. Wessel, M., De Souza, B.M., Steinmacher, I., Wiese, I.S., Polato, I., Chaves, A.P., Gerosa, M.A.: The power of bots: Understanding bots in OSS projects. International Conference on Human-Computer Interaction **2** (11 2018)
23. Wessel, M., Wiese, I., Steinmacher, I., Gerosa, M.A.: Don't Disturb Me: Challenges of Interacting with Software Bots on Open Source Software Projects. Proc. ACM Hum.-Comput. Interact. **5** (10 2021). https://doi.org/10.1145/3476042
24. Xie, G., Chen, J., Neamtiu, I.: Towards a better understanding of software evolution: An empirical study on open source software. In: International Conference on Software Maintenance. IEEE (9 2009). https://doi.org/10.1109/ICSM.2009.5306356
25. Young, J.G., Casari, A., McLaughlin, K., Trujillo, M.Z., Hébert-Dufresne, L., Bagrow, J.P.: Which contributions count? Analysis of attribution in open source. In: International Conference on Mining Software Repositories (MSR). Association for Computing Machinery (5 2021). https://doi.org/10.1109/MSR52588.2021.00036