Studying the Association between Bountysource Bounties and the Issue-addressing Likelihood of GitHub Issue Reports

Jiayuan Zhou, Shaowei Wang, Cor-Paul Bezemer, Ying Zou, *Member, IEEE,*
and Ahmed E. Hassan, *Fellow, IEEE*

Abstract—Due to the voluntary nature of open source software, it can be hard to find a developer to work on a particular task. For example, some issue reports may be too cumbersome and unexciting for someone to volunteer to do them, yet these issue reports may be of high priority to the success of a project. To provide an incentive for implementing such issue reports, one can propose a monetary reward, i.e., a bounty, to the developer who completes that particular task. In this paper, we study bounties in open source projects on GitHub to better understand how bounties can be leveraged to evolve such projects in terms of addressing issue reports. We investigated 5,445 bounties for GitHub projects. These bounties were proposed through the Bountysource platform with a total bounty value of $406,425. We find that 1) in general, the timing of proposing bounties is the most important factor that is associated with the likelihood of an issue being addressed. More specifically, issue reports are more likely to be addressed if they are for projects in which bounties are used more frequently and if they are proposed earlier. 2) The bounty value of an issue report is the most important factor that is associated with the issue-addressing likelihood in the projects in which no bounties were used before. 3) There is a risk of wasting money for backers who invest money on long-standing issue reports.

Index Terms—Software evolution, open source software, Bountysource, bounties, GitHub

1 Introduction

Open source software projects often use issue tracking systems (such as BugZilla or GitHub Issues) to store and manage issue reports. For example, developers or users can submit issue reports to report bugs or request new features, and wait for these issues to be addressed. However, some issue reports may never be addressed. For example, developers may avoid addressing issues that they consider too low priority, or difficult to implement. To encourage developers (or *bounty hunters*) to address such issue reports, one or more *backers* can propose a *bounty*.

A bounty is a monetary reward that is often used in the area of software vulnerabilities. Prior studies examined the impact of bounties on vulnerability discovery [7], [11], [32]. Finifter et al. [7] suggested that using bounties as an incentive to motivate developers to find security flaws is more cost-effective than hiring full-time security researchers.

Bounties are now being used to motivate developers to address issue reports, e.g., to fix bugs or to add features. $Bountysource$[1] is a platform for proposing bounties for open source projects across multiple platforms (e.g., GitHub) which currently has more than 46,000 registered developers.[2] Bounty backers can propose several bounties for the same issue report via Bountysource. Although bounties are used in the issue-addressing process, the role that bounties play in this process is not yet understood. For example, it is unclear whether a bounty is associated with improving the issue-addressing likelihood in projects. By understanding this association, we could provide insights on how to better leverage bounties to evolve open source projects, and on how to improve the usability and effectivity of bounty platforms.

In this paper, we study 3,509 issue reports with 5,445 bounties that were proposed on Bountysource from 1,203 GitHub projects, with a total bounty value of $406,425. We used a logistic regression model to study the association between 26 factors (including the timing of proposing a bounty and the bounty-usage frequency of a project) along 4 dimensions (i.e., the project, issue, bounty, and backer dimensions) and the issue-addressing likelihood. We found that:

1) The timing of proposing bounties is the most important factor that is associated with the issue-addressing likelihood.
2) Bounty issue reports are more likely to be addressed in projects which are using bounties more frequently.
3) Issue reports are more likely to be addressed if bounties are proposed earlier. Additionally, there

- *J. Zhou and A. E. Hassan are with the Software Analysis and Intelligence Lab (SAIL), Queen's University, Canada. E-mail: {jzhou, ahmed}@cs.queensu.ca*
- *S. Wang (corresponding author) is with the Department of Computer Science and Engineering, Mississippi State University, United States. E-mail: wang@cse.msstate.edu*
- *C. Bezemer heads the Analytics of Software, Games and Repository Data (ASGAARD) lab, University of Alberta, Canada. E-mail: bezemer@ualberta.ca*
- *Y. Zou is with the Department of Electrical and Computer Engineering, Queen's University, Canada. E-mail: ying.zou@queensu.ca*

1. https://www.bountysource.com

2. https://blog.canya.com/2017/12/20/canya-acquires-majority-stake-in-bountysource-adds-over-46000-users/

is a risk of wasting money for backers who invest money on long-standing issue reports.

4) The total bounty value of an issue report is the most important factor that is associated with the issue-addressing likelihood in the first-timer projects.

We also manually identified the reasons why developers ignored bounties (i.e., the cases in which bounty issue reports were addressed while the bounty remained unclaimed) that are worth more than $100. We found that some developers addressed an issue cooperatively, making it difficult to choose a single developer that would be awarded the bounty. In addition, some developers are not driven by money to address issues.

Based on our findings, we have several suggestions for backers and the Bountysource platform. For example, backers should be cautious when proposing small (i.e., $< \$100$) bounties on long-standing issue reports since the risk of losing the bounty exists. Bounty platforms should consider allowing for splittable multi-hunter bounties.

The rest of the paper is organized as follows. In Section 2, we present background information about GitHub and Bountysource. In Section 3, we discuss related work. In Section 4, we introduce our research questions and studied factors. In Section 5, we describe our data collection process. In Section 6, we study the characteristics of Bountysource bounties in GitHub. In Section 7, we investigate the association between the bounty-related factors and the issue-addressing likelihood. In Section 8, we study the closed-unpaid bounty issue reports and we discuss the implications of our study. In Section 9 we discuss the threats to validity of our study. We conclude our study in Section 10.

2 Background

In this section, we briefly introduce the issue tracking system on GitHub and the open source bounty platform Bountysource.

2.1 Issue tracking system on GitHub

The issue tracking system (i.e., ITS) on GitHub helps developers to manage the issue reports of their project. Users and developers can report bugs or request new features by posting an issue report on the issue tracking system. There are two statuses of an issue report: "open" and "closed". "Open" indicates that the issue report is still active and is waiting to be addressed. "Closed" indicates that the issue report has been closed. The most common reason for closing an issue report is that the issue has been addressed, but it could also have other reasons (e.g., duplicated issue reports). Users can attach free-text labels to issue reports to indicate the category of an issue report. An issue report contains a title to summarize the issue and a detailed description of the issue. Developers can discuss an issue report by leaving comments, which can include code snippets, links or images to improve the description.

2.2 Bountysource

Bountysource is a platform on which users can pledge a monetary incentive (a *bounty*) to address an issue report of an open source project. There exist two roles on Bountysource: the bounty backer and the bounty hunter roles.

Bounty backers, which may be anonymous, are users or developers who propose bounties for issue reports. A backer can set an expiration period for their bounty that has a value of more than $100. When the bounty expires, the money is refunded to the backer; otherwise, the bounty stays with the issue report until someone claims it. Note that bounties that are smaller than $100 are not refunded if they remain unclaimed. An issue report can have multiple bounties from one or more backers and a bounty can only be proposed for one issue report.

Bounty hunters are developers who address issue reports that have bounties. If a hunter works on an issue report, the hunter can attach certain information (i.e., the estimated time of addressing, the code URL, or some comments) on Bountysource to indicate the progress. However, a bounty hunter could also work on the issue report without notifying Bountysource. Once a developer claims to have addressed an issue report, its bounty backer(s) can choose to accept (no response will be taken as an acceptance) or reject the claim. In this situation, backers have two weeks to make the decision (accept or reject). If no backer explicitly rejects the claim, the bounties will be paid to the developer automatically. Multiple bounty hunters can work on an issue report at the same time, but the bounties of an issue can only be rewarded to one bounty hunter. In particular, this is the bounty hunter who first claims the bounties while no backer explicitly rejects the claim.

When an issue report is submitted by an issue reporter, one or more bounty backers can propose bounty(ies) on the issue report. One or more developers of the issue report can choose to become bounty hunters to address the issue report but only one bounty hunter can get the bounty(ies).

Developers and users from more than 12 platforms (e.g., GitHub) propose bounties for issue reports through Bountysource. In this study, we focus on GitHub issue reports, since the majority of the bounties (see Section 5 for more details) that are proposed on Bountysource are for GitHub issue reports. Figure 1 shows the workflow of the bounty processes between GitHub and Bountysource. The workflow of a bounty starts with a bounty backer offering a bounty on Bountysource for a GitHub issue report. The bounty backers pledge money to Bountysource (the money is held by Bountysource) and they can choose to add bounty information to the GitHub issue report. For example, tagging the issue report on GitHub with a bounty label (see the example[3] for details) to "advertise" the bounty, appending the bounty value to the title of the issue report or mentioning the bounty in the discussion of the issue report in GitHub. When a bounty hunter starts working on an issue, they can update their working status on Bountysource. After the issue report is addressed, the bounty hunter can submit a claim for the bounty on Bountysource and the backer will be notified by Bountysource. Once the bounty backer accepts the solution, the bounty hunter receives the money from Bountysource.

3. https://github.com/austinpray/asset-builder/issues?q=label%3Abounty

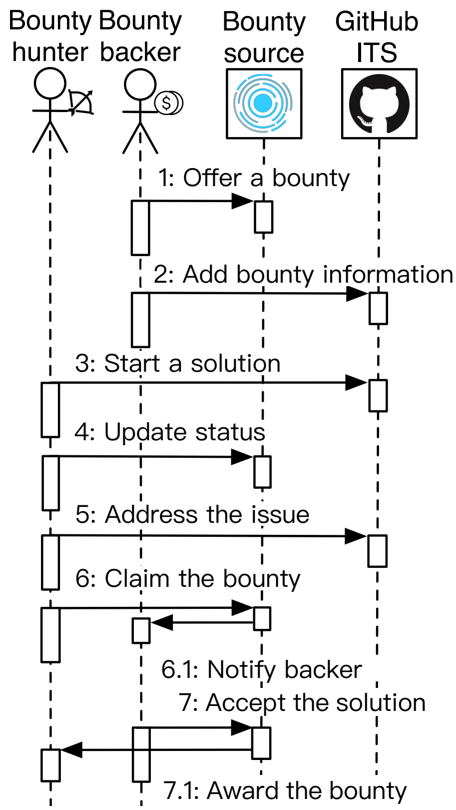

Fig. 1: The workflow of the bounty between GitHub and Bountysource.

Based on the status of an issue report and whether a bounty is paid out, a bounty issue report has the following three statuses:

Closed-paid: the issue report is closed and the bounty has been successfully rewarded to a bounty hunter. We defined such issue reports as *successful* bounty issue reports.

Open-unpaid: the issue report is open and the bounty is active. We defined such issue reports as *failed* bounty issue reports.

Closed-unpaid: the issue report is closed but the bounty remains unclaimed. We defined such issue reports as *ignored* bounty issue reports.

3 Related work

In this section, we discuss related work along two dimensions: the bounty in software engineering and the improvement of the issue-addressing process.

Bounties in software development: Bounties are used to attract developers and motivate them to complete tasks. Prior work has studied the impact of bounties on software development. Krishnamurthy and Tripathi [17] gave an overview of bounties in Free/Libre/Open Source Software (FLOSS). They observed that bounty hunters' responses are related to the workload, the probability of winning the bounty, the value of the bounty and the recognition that they might receive by winning the bounty. Different from their study, we focused on using bounties to improve the issue addressing process. Zhou et al. studied how bounties impact the question answering process on Stack Overflow [35]. They found that bounties increase the likelihood of a question getting answers and increase the traffic for questions.

Several studies focused on the usage of bounties to motivate developers to detect software security vulnerabilities. Finifter et al. [7] analyzed vulnerability rewards programs for Chrome and Firefox. They found that the rewards programs for both projects are economically effective, compared to the cost of hiring full-time security researchers. Zhao et al. [31] investigated the characteristics of hunters in bug-bounty programs and found that the diversity of hunters improved the productivity of the vulnerability discovery process. Hata et al. [11] found that most hunters are not very active (i.e., they have only a few contributions). Zhao et al. [32] and Maillart et al. [18] analyzed the effect of different policies of bug-bounty programs. By studying bug-bounties from several perspectives, they provided insights on how to improve the bug-bounty programs. For example, Maillart et al. [18] suggested project managers to dynamically adjust the value of rewards according to the market situation (e.g., increase rewards when releasing a new version).

There is not much research to study the effectiveness of bounties to improve the issue-addressing process. The work of Kanda et al. [16] is closest to ours. They studied GitHub and Bountysource data but studied only 31 projects (compared to 1,203 in our study). They compared the closed-rate and closing-time between bounty issue and non-bounty issue reports. Their results showed that the closing-rate of bounty issue reports is lower than that of non-bounty issue reports, and it takes longer for the bounty issue reports to get closed than non-bounty issue reports. Our study performs a deeper analysis of bounties at the project level. Besides, we further study the relationship between the issue-addressing likelihood and the bounty-related factors (e.g., the total bounty value of a bounty issue report) while controlling for the factors that are related to the issue report and project (e.g., the number of comments before the first bounty is proposed).

Improving the issue-addressing process: Issue addressing is an essential activity in the life cycle of software development and maintenance. Therefore, a large amount of research was done to improve the issue-addressing process. One group of studies focused on providing insights into improving the issue-addressing process in aspects of the quality of issue reports, the effectiveness of developers and automated bug localization and fixing. For example, Bettenburg et al. [2], [13] analyzed the quality of bug reports (i.e., a type of issue report) and provided some guidelines for users to generate high-quality reports so that developers can address issues more efficiently. Ortu et al. [20] analyzed the relation between sentiment, emotions, and politeness of developers in comments with the needed time to address an issue. They found that the happier developers are, the shorter the issue-addressing time is likely to be. Zhong et al. [33] performed an empirical study on real-world bug fixes to provide insights and guidelines for improving the state-of-the-art of automated program repair. Soto et al. [25] performed a large-scale study of bug-fixing commits in Java projects and provided insights for high-quality automated software repair to target Java code. A number of studies helped developers locate the buggy code in projects using information retrieval techniques [23], [27]–[30], [36].

Different from prior studies, we perform an empirical study to understand the relationship between bounties and the issue-addressing process. We provide insights into how to better use bounties to improve the efficacy of the issue-addressing process.

4 RESEARCH QUESTIONS & STUDIED FACTORS

In this section, we describe our research questions and their motivation, as well as the factors that we study to answer our research questions.

4.1 Research Questions

Prior studies showed that bounty-related factors (e.g., the value of bounties) have an association with various software development tasks, such as developing new features [17] and addressing security issues [18]. However, little is known about how these factors are related to the issue-addressing likelihood of bounty issue reports. In addition, factors that are related to a bounty issue report itself and its backers may have associations with the issue-addressing likelihood of the bounty issue report. For example, an issue report that attracts more attention (e.g., comments and participants) from the community may have a higher likelihood of being addressed. Therefore, in this study, when examining the association between bounty-related factors and the issue-addressing likelihood of bounty issue reports, we also take the factors that are related to issue reports and backers into consideration. More specifically, we control the factors of the issue report basic, project bounty, and backer experience dimensions when building our logistic regression models to examine the relationships between bounty-related factors and the issue-addressing likelihood of bounty issue reports, since such factors are not changeable when backers propose bounties (see Section 7 for more details). In particular, we address the following research questions:

- *RQ1: Are the studied factors associated with the issue-addressing likelihood of bounty issue reports in GitHub projects?*
- *RQ2: How does the association between the studied factors and the issue-addressing likelihood change in projects with different bounty usage frequencies?*

In RQ1, we investigate which studied factors are associated with the issue-addressing likelihood. In addition, prior work shows that the impact of bounties on the addressing of software security issues varies across projects [18]. Similarly, in RQ2, we investigate how the association between the studied factors and the issue-addressing likelihood changes in projects with a different bounty usage frequency. By understanding this association, we can provide insights for the backers into how to best leverage bounties to improve the issue addressing process. We can also provide suggestions for the Bountysource platform to improve its system.

4.2 Studied Factors

To answer our RQs, we extracted 26 factors from the bounty issue report and project information, through the process that is described in Section 5. In this section, we introduce these 26 factors along the following 4 dimensions:

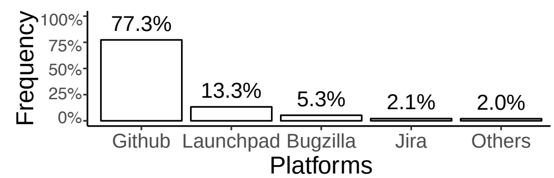

Fig. 2: The distribution of Bountysource bounties across the supported ITSs.

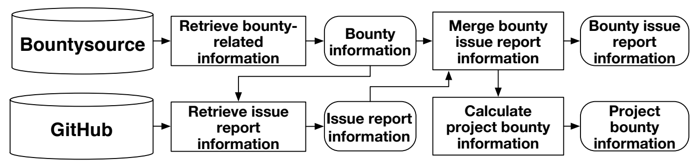

Fig. 3: An overview of our data collection process.

1) **Issue report basic:** Eight factors which estimate the length and the popularity of an issue report.
2) **Issue report bounty:** Four factors which describe the bounty usage within a bounty issue report.
3) **Project bounty:** Six factors which reflect the bounty usage within a project.
4) **Backer experience:** Eight factors which capture the bounty experience of the backers of a bounty issue report.

Table 1 summarizes the descriptions of and rationales behind the studied factors. The factors which are marked with '*' are time-dependent factors which are calculated at the time when the bounty is proposed. For example, the *I_content_len** factor is the length (in characters) of an issue report and its comments when the first bounty of the issue report was proposed.

Note that the factors in the project bounty, issue report basic, and backer experience dimensions cannot be changed by a backer who wants to propose a bounty and we consider these factors as the confounding factors for which we want to control. The bounty backers can control the factors in the issue report bounty dimension. For example, a bounty backer can choose the timing of proposing a bounty on an issue report (i.e., *I_B_days_before_bounty*), the bounty value (i.e., *I_B_total_value*), and whether to add a bounty label to the issue report (i.e., *I_B_has_label*).

5 DATA COLLECTION

In our study, we focus on the bounties that are proposed through the Bountysource platform since it is one of the most popular platforms for open source projects. As explained in Section 2, Bountysource supports issue reports from several ITSs (e.g., GitHub and Bugzilla). Figure 2 shows the distribution of Bountysource bounties across its supported ITSs. The majority of the issue reports come from GitHub (77.3%), hence we focus our study on the bounties that were proposed for GitHub issue reports.

All information about the bounties is stored on Bountysource and all details about issue reports and their corresponding projects are stored on GitHub. Hence, we collected data for our study along three dimensions: the bounty, the issue report, and the project.

TABLE 1: The description and rationale for the factors in the *Issue report basic*, the *Issue report bounty*, *Project bounty* and the *Backer experience* dimensions. The factors which are marked with '*' are time-dependent factors which are calculated at the time when the bounty is proposed

Factor name	Description	Rationale
Issue report basic		
I_content_len*	The length of an issue report and its comments (in characters).	These factors reflect the amount of supportive information that an issue report has. Issue reports with more supportive information may help developers to address them.
I_code_len*	The total length of the code snippets in an issue report and its comments (in characters).	
I_code_proportion*	The proportion of code in an issue report and comments (i.e., $\frac{I\_code\_len}{I\_content\_len}$).	
I_link_cnt*	The number of links in an issue report and its comments.	The discussion activities reflect the popularity of an issue report, which may have a relationship with the issue-addressing likelihood.
I_img_cnt*	The number of images in an issue report and its comments.	
I_cmnt_cnt	The number of comments that an issue report received.	
I_participant_cnt*	The number of participants in the discussion of an issue.	
I_cmnt_per_day_mean*	The mean number of comments per day for an issue report.	
Issue report bounty		
I_B_days_before_bounty*	The number of days between the creation of an issue report and its first bounty.	The timing of proposing bounties may have a relationship with the issue-addressing likelihood.
I_B_total_value	The total bounty value of the issue report.	A higher bounty may attract more developers.
I_B_cnt	The number of bounties that a bounty issue report has.	A higher number indicates that more backers are interested in getting this issue addressed.
I_B_has_label	Whether a bounty issue report is tagged with a bounty label.	A bounty label could help draw attention from the community (i.e., because the label acts as an advertisement), which may have an association with the issue-addressing likelihood.
Project bounty		
P_B_I_cnt*	The total number of issue reports with at least one bounty of a project.	These five factors reflect the bounty activity of the project. A different level of activity may have a different association with the issue-addressing likelihood in the project.
P_B_paid_cnt*	The total number of paid bounty issue reports of a project.	
P_B_open_cnt*	The number of open bounty issue reports of a project.	
P_B_paid_proportion*	The proportion of paid bounty issue reports of a project.	
P_B_total_value*	The total value of the bounties of a project.	
P_B_usage_group	The group of projects.	Different groups of projects may have different issue-addressing likelihoods (see Section 6).
Backer experience		
Backer_exp_B_median/sum/max_value*	The median/sum/max value of bounties which the backers of this bounty have ever proposed in the past.	Bounties from a backer who has proposed bounties often, or proposed high-value bounties in the past may attract more attention from developers.
Backer_exp_B_median/sum/max_cnt*	The median/sum/max number of bounties which the backers of this bounty have ever proposed in the past.	
Backer_role_any_insider*	Whether any of the backers has ever contributed to the project.	A backer who has ever interacted with the project before may help the bounty attract more attention from the community.
Backer_role_have_reporter*	Whether the issue reporter is one of the backers for that issue report.	

Figure 3 presents an overview of our data collection process, which is broken down as follows:

Step 1: We retrieved the bounty and issue information from Bountysource using its official web API automatically.[4] The bounty information includes the backers who proposed the bounty, the proposed bounty value and the hunter who addressed the issue report. In addition, we collected basic information about the GitHub issue reports such as their id and URL.

Step 2: We retrieved the details of the issue reports, which are linked to Bountysource bounties by using the URL and id that we retrieved in step 1, from GitHub using its official web API automatically.[5] For example, we collected the description of the issue report, the creation date of the issue report, the comments that developers left under the report, and the labels of the issue report.

Step 3: We calculated the corresponding project's bounty information for each collected bounty issue report, such as the number of total bounty issue reports of a project.

TABLE 2: Dataset description.

Total number of bounties	5,445
Total number of claimed bounties	2,402
Total bounty value	$406,425
Total number of bounty hunters	882
Total number of bounty backers	2,534
Total number of issue reports	3,509
Total number of issue reports with multiple bounties	795
Total number of projects	1,203

4. https://bountysource.github.io/
5. https://developer.github.com/v3/

In total, we collected 5,445 bounties with a total value of $406,425, together with their corresponding issue reports which were reported between Oct 19, 2012, and Oct 5, 2017. Since some bounty issue reports were just created when we collected the data, we updated the status of the collected bounty issue reports after 200 days (i.e., Apr. 22, 2018) to have a more reliable status for these issue reports. We published our dataset online.[6] Table 2 describes our dataset.

We observed that **62.7% of the bounty issue reports are closed, while the bounties in almost one-third of these closed issue reports remain unpaid with a value of $41,856 in total.** Figure 4 shows the distribution of bounty issue reports across the three statuses. 37.3% of the bounty issue reports are failed (i.e., open-unpaid). Although 62.7% of the bounty issue reports were closed, almost one-third of their bounties were ignored (i.e., closed-unpaid). The total value of the ignored bounties ($41,856) is "frozen" in the Bountysource platform unless someone claims the bounty.

In the rest of the paper, when we discuss the issue-addressing likelihood, we only refer to the bounty issue reports that are successful (i.e., closed and paid out) or failed (i.e., still open). We do not take the issue reports which were ignored into consideration because the hunters might not be driven by the bounty in such issue reports. We conducted a qualitative study of these closed-unpaid bounty issue reports to better understand them in Section 8.1. When a bounty issue report is closed and the bounty is paid out, we define this bounty issue report as addressed.

6 Characteristics of Bountysource Bounties in GitHub

As mentioned in Section 4, we aim to understand the association between the issue-addressing likelihood of an issue report and the factors that are related to the bounties of the issue report (e.g., the total value of bounties being proposed for an issue report) in different projects. Therefore, in this section, we present a basic view of such bounty-related factors. We first present the following basic descriptive statistics: (1) the distribution of the number of days between the reporting of an issue and its first bounty being proposed (*I_B_days_before_bounty*); (2) the distribution of the total bounty value of an issue report (*I_B_total_value*); (3) the distribution of the number of bounties that a bounty issue report has (*I_B_cnt*); (4) the distribution of bounty issue reports that have a bounty label (*I_B_has_label*). We also investigate how bounties are used across projects. We present the distribution of the total number of bounties used in projects (the bounty-usage frequency). From these statistics, we can get a basic view of the characteristics of bounties, and of how bounties are used across projects.

35% of the bounties were proposed within 7 days from the creation of an issue report, while 30% of the bounties were proposed after more than 180 days. Figure 5 shows the empirical cumulative distribution function of *I_B_days_before_bounty*. We observe that in 35% of the issue reports their first bounty was proposed within seven days after their creation. Only 11% of the bounties were proposed

6. https://github.com/SAILResearch/wip-18-jiayuan-bountysource-SupportMaterials

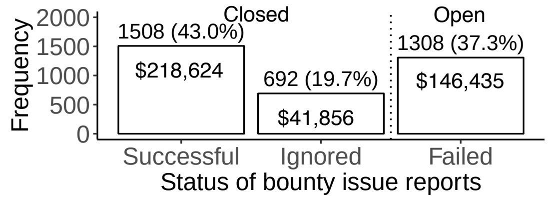

Fig. 4: The distribution of the possible statuses of bounty issue reports and their corresponding cumulative bounty value.

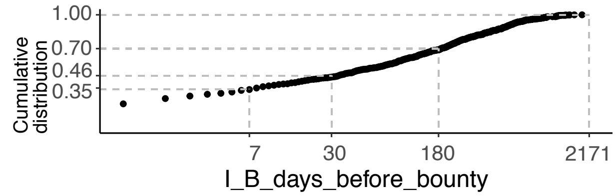

Fig. 5: The empirical cumulative distribution of *I_B_days_before_bounty*.

between 7 and 30 days after the creation of an issue report. 24% of the bounties were proposed between 30 and 180 days and the remaining 30% of the bounties were proposed after 180 days. The frequency with which bounties are proposed is lower in the first seven days than later on. One possible explanation is that bounty backers may wait and see if issues are addressed without a bounty. After waiting for a period of time without getting their issue addressed, bounty backers start to propose bounties.

The distribution of *I_B_total_value* is skewed and the correlation between *I_B_total_value* and the issue-addressing likelihood is weak. We observe that the skewness and kurtosis values of the distribution of *I_B_total_value* are 13 and 236, respectively. The first, second, and third quartile values are $15, $30 and $100. Figure 6 presents the issue-addressing likelihood of an issue report against the bounty value of the issue report. We do not observe an obvious pattern between them. The correlation between the bounty value and the issue-addressing likelihood is surprisingly weak (0.14).

90% (i.e., 2,541) of the studied bounty issue reports only have one or two bounties. We observe that 75% of the bounty issue reports only have one bounty and 15% of the bounty issue reports have two bounties.

We also observe that 56% (i.e., 1,568) of the bounty issue reports are explicitly labeled as such.

More than half of the projects only used a bounty once, while two projects used bounties very frequently (more than 100 times). Figure 7 shows the empirical cumulative distribution of the bounty-usage frequency across projects. As shown in Figure 7, the distribution is skewed (with a variance of 57.02). 612 (66%) projects used a bounty only once. 52 (6%) projects used bounties at least 10 times and only 9 projects used a bounty more than 50 times. In order to better study the research questions, we propose a bootstrap-derived data preprocessing method to reduce bias caused by different bounty-usage frequency across projects in Section 7.

7 Results of our Research Questions

In this section, we present the results of our research questions. We discuss each research question along three

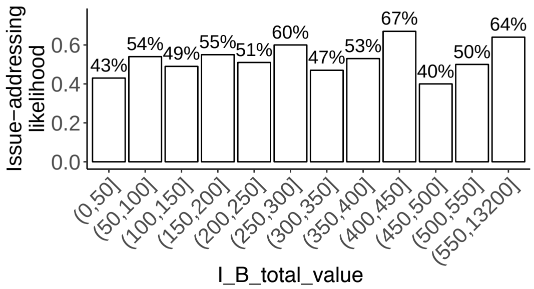

Fig. 6: The issue-addressing likelihood of the proposed bounty value ranges.

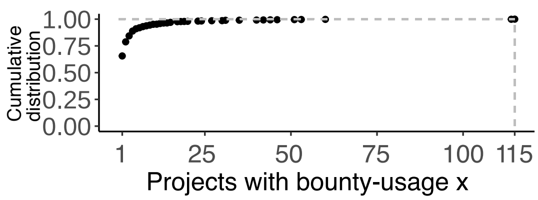

Fig. 7: The empirical cumulative distribution of the bounty-usage frequency of projects. The bounty-usage frequency is the total number of used bounties in a project.

parts: used approach, experimental findings, and a detailed discussion of our findings.

7.1 RQ1: Are the studied factors associated with the issue-addressing likelihood of bounty issue reports in GitHub projects?

7.1.1 Approach

We construct logistic regression models to study the relationship between the studied factors (see Section 4.2) and the issue-addressing likelihood. Note that our goal of constructing models is not for prediction but for interpretation. The logistic regression model is a robust and highly interpretable technique, which has been applied successfully in software engineering studies, e.g., to predict the closure rate of GitHub issues [14], predict bugs [19], [21], and classify the information that is discussed in GitHub issues [1].

Figure 8 shows the flow of our approach. Below, we elaborate on the processes of the data preprocessing, the model construction, and the analysis of our models.

Data preprocessing Figure 9 gives an overview of our data preprocessing approach. We elaborate on each step below.

Project categorization: Given the variance of the bounty-usage frequency across different projects, it is not advisable to study all the issue reports as one group when we study the bounties at the issue report level. Therefore we categorize the projects into the following three groups:

1) **First-timer project:** Projects which have only one bounty issue report.
2) **Moderate project:** Projects which have 2 to 50 bounty issue reports.
3) **Frequent project:** Projects which have more than 50 bounty issue reports.

It is important to study the bounties in the first-timer projects, since users of such projects may not have former bounty experience. We grouped the projects that have more than 50 bounty issue reports as well since we assume that in such projects the community is more familiar with the use of bounties. Note that we set the threshold 50 for moderate and frequent projects empirically. We performed a sensitivity analysis on different thresholds (i.e., 40 and 60) and the results show that our findings still hold (see Appendix C in our supplementary material [34] for more details).

After grouping the projects into the above mentioned three groups, we have 550 (59%) first-timer projects with 550 bounty issue reports, 374 (40%) moderate projects with 1,717 bounty issue reports, and 9 (1%) frequent projects with 549 bounty issue reports.

Bootstrap sampling: After grouping the projects into the three groups, we used a bootstrap sampling approach to sample issue reports across projects in order to balance the data. We used bootstrap sampling to reduce the bias that is caused by the unbalanced number of projects across the three groups. We first randomly sampled 1,000 projects from each group with replacement. Then we randomly sampled one bounty issue report from each sampled project, to avoid a bias towards projects with more issue reports than other projects in the same group. Hence, we sampled 1,000 bounty issue reports from each of the 3 project groups. To make our results more reliable, we repeated the sampling process 100 times with different random seeds. We ended up with 100 samples with 3,000 issue reports each (1,000 issue reports for each group). On average, 54.3% of the bounty issue reports were sampled during one iteration of the bootstrap sampling process.

Model construction Figure 8 shows an overview of our model construction approach. The presence of correlated and redundant features greatly impacts the interpretability of the generated models (i.e., multicollinearity) [6]. Hence, we first removed correlated and redundant factors using the Spearman rank correlation test and through redundancy analysis to avoid multicollinearity similar to prior studies [9], [12], [15], [22], [26]. We performed correlation and redundancy analysis instead of other common and state-of-the-art dimensionality reduction techniques such as PCA, since such techniques combine and transform the original features into principal components, which are no longer directly interpretable. First, we use the Spearman rank correlation test to measure the correlation between factors and remove highly-correlated factors (using a cut-off value of 0.7 [4], [15], [24]). For each of the highly-correlated factors, we keep one factor in our model. We performed a redundancy analysis to remove redundant factors (see Appendix A in our supplementary material [34] for more details and the factors that were included in the models). We ended up with three factors in the project bounty dimension, six factors in the issue report basic dimension, four factors in the issue report bounty dimension, and three factors in the backer experience dimension. We added non-linear terms in the model to capture more complex relationships in the data by employing restricted cubic splines [10]. Finally, we built logistic regression models based on 100 samples (3,000 issue reports with 1,000 issue reports for each group) and ended up with 100 models. We refer to these 100 models which are constructed to understand the global relationship as the **global model**. See our supplementary material [34] for more details about our model construction.

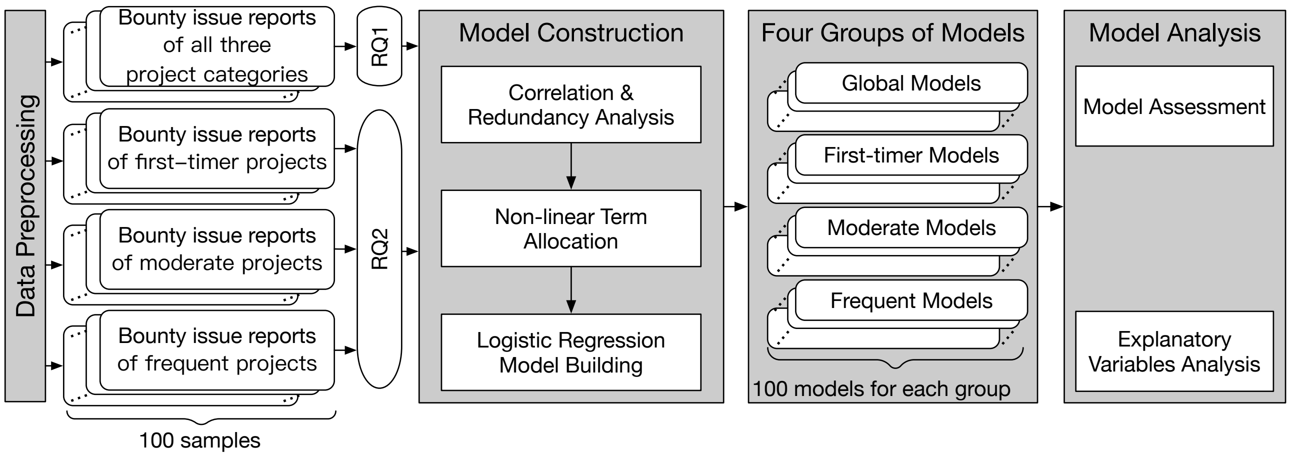

Fig. 8: An overview of the data preprocessing, model construction, and analysis steps of our approach.

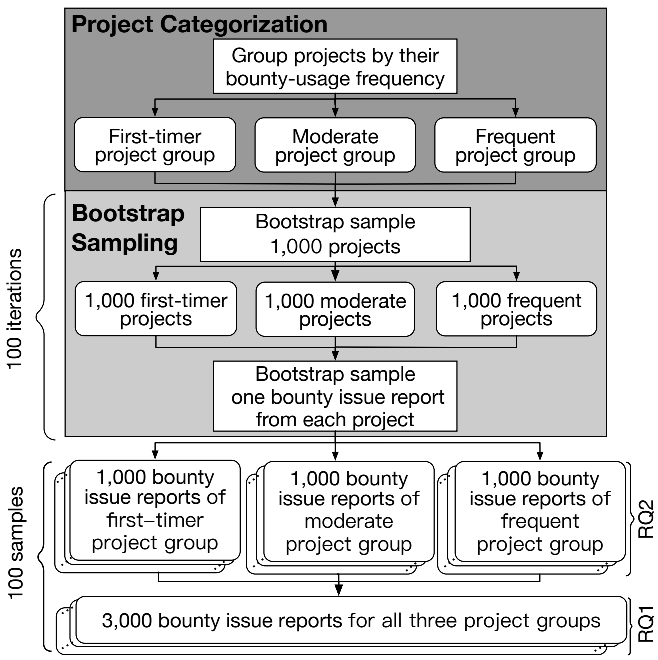

Fig. 9: An overview of our data preprocessing approach.

Model analysis For each logistic regression model, we used the Area Under the Receiver Operating Characteristic Curve (i.e., *AUC*) and a bootstrap-derived approach [5] to assess the explanatory power of the models following prior studies [15], [19], [26]. The AUC ranges from 0 to 1, with 0.5 being the performance of a random guessing model and a higher AUC meaning that the model has a higher ability to capture the relationships between the explanatory factors and the response factor. In practice, an AUC value is usually never smaller than 0.5, since we can switch the binary output labels of such a classifier to get a higher AUC ($newAUCvalue = 1 - currentAUCvalue$). To check whether the models are not overfitted, we calculate their *optimism* values using a bootstrap-derived approach. The *optimism* value ranges from 0 to 1. A small optimism value suggests that a model does not suffer from overfitting, while an optimism of 1 indicates that the model is 100% overfitting the dataset (see Appendix B in our supplementary material [34] for the calculation of the optimism value).

TABLE 3: The 5-number summary of AUC and optimism values of our models.

Model Types		**Quantile**				
		Min	1^{st}	Median	3^{rd}	Max
Global	AUC:	0.72	0.73	0.74	0.74	0.75
	optimism:	0.00	0.00	0.01	0.01	0.02
First-timer	AUC:	0.70	0.73	0.74	0.76	0.80
	optimism:	0.00	0.00	0.01	0.02	0.04
Moderate	AUC:	0.66	0.68	0.70	0.71	0.74
	optimism:	0.00	0.01	0.01	0.02	0.05
Frequent	AUC:	0.79	0.81	0.82	0.83	0.86
	optimism:	0.00	0.01	0.01	0.02	0.04

To measure the explanatory power of each factor in the constructed model, we computed its Wald χ^2 value. A larger Wald χ^2 value indicates a higher explanatory power of the factor in the constructed model. To test whether a factor contributes a statistically significant amount of explanatory power to the model, we further applied a χ^2-test to the calculated Wald χ^2 values. In this study, we consider factors of which the χ^2-test has a p-value of less than 0.001 as significantly important.

In addition, to further understand how a factor influences the value of the response variables, we plotted the estimated issue-addressing likelihood against a factor. Since all models across 100 samples showed similar patterns of influence for the factors, we randomly selected a sample as an example to build models and visualize the results (see Figures 10 and 12). The analysis allows us to further understand how a factor affects the issue-addressing likelihood. We used the `R rms` package during the construction and analysis of our models.

7.1.2 Results

Our models capture the relationship between the explanatory variables and the response variable well, and have a reliable performance. The median AUC of our global models is 0.74 (see Table 3), which indicates that our models have a good ability to capture the relationship between the explanatory variables and the response variable, and the low median optimism values (0.01) indicate that our models do not overfit the dataset.

TABLE 4: The results of the model analysis for four groups of models. The **NL** indicates the non-linear term and the **D.F.** indicates the degree of freedom.

		Global Model		**First-timer Model**		**Moderate Model**		**Frequent Model**	
Factors		Overall	NL	Overall	NL	Overall	NL	Overall	NL
I_B_days_before_bounty	D.F.	4	3	4	3	4	3	4	3
	χ^2	172.30***	3 5.71***	26.83***	4.79	43.59***	7.44	51.67***	10.47
P_B_usage_group	D.F.	2		-		-		-	
	χ^2	35.08***		-		-		-	
I_B_total_value	D.F.	2	1	2	1	2	1	2	1
	χ^2	31.19***	29.80***	34.00***	0.28	3.07	2.97	13.67	9.99
I_code_proportion	D.F.	1		1		1		1	
	χ^2	11.62		0.74		14.84***		0.21	
I_B_has_label	D.F.	1		1		1		1	
	χ^2	33.08***		4.16		6.99		0.022	
Backer_exp_B_max_value	D.F.	3	2	3	2	3	2	3	2
	χ^2	16.21	16.14	1.95	0.80	1.36	1.32	22.86***	21.87***
P_B_paid_proportion	D.F.	3	2	-		2	3	2	2
	χ^2	14.56	2.51	-		7.77	0.01	29.39***	29.39***
P_B_total_value	D.F.	2	1	-		2	1	2	1
	χ^2	15.75***	12.18***	-		1.44	0.34	13.86	10.99
I_img_cnt	D.F.	1		1		1		1	
	χ^2	1.58		0.68		1.55		0.69	
I_link_cnt	D.F.	1		1		1		1	
	χ^2	0.09		3.09		0.02		2.60	
I_content_len	D.F.	1		1		1		1	
	χ^2	4.32		0.02		2.8		1.72	
I_cmnt_perday_mean	D.F.	2	1	2	1	2	1	2	1
	χ^2	4.37	0.76	3.83	0.01	0.73	0.34	0.56	0.07
I_B_cnt	D.F.	1		1		1		1	
	χ^2	0.433		2.54		1.48		8.39	
I_cmnt_cnt	D.F.	1		1		1		1	
	χ^2	0.49		0.00		1.55		7.34	
Backer_role_any_insider	D.F.	1		1		1		1	
	χ^2	1.96		7.92		1.71		7.60	
Backer_role_have_reporter	D.F.	1		1		1		1	
	χ^2	0.34		7.92		1.78		0.03	

P-value of the χ^2 test: '***' < 0.001

In the global view, the timing of proposing the bounties is the most important factor that has a significant relation with the issue-addressing likelihood. Table 4 shows that the timing of proposing the bounties, the bounty-usage frequency of projects, the bounty label of issue reports, the total value of the bounties of a project, and the total bounty value of the issue report contribute a significant amount of explanatory power to our models. The timing of proposing the bounties contributes the most explanatory power by far, based on the Wald χ^2 value.

Projects that use bounties more frequently have a higher bounty issue-addressing likelihood. We observe a positive association between the issue-addressing likelihood and *P_B_usage_group* in the global models. One possible explanation is that projects with a higher bounty-usage frequency are more likely to maintain documents to introduce how bounties work in such projects, so that backers can gain more experience and background about proposing bounties (e.g., at the proper time with a proper value) and the hunters react to bounties more actively than in projects with a lower bounty-usage frequency. For example, the `eslint` project maintains a document on how bounties work.[7] The `eslint` project has 43 successful (i.e., closed-paid) and only one failed (i.e., open-unpaid) bounty issue report.

To further test our assumption, we performed a qualitative study to investigate whether projects that use bounties more frequently are more likely to have a bounty document. We calculated the representative sample sizes [3] and randomly sampled 80 first-timer projects and 77 moderate projects as statistically representative samples with a 95% confidence level and a 10% confidence interval. We selected all nine frequent projects. The first two authors manually examined the GitHub pages of each sampled project and checked whether the project has a document that explains the bounty process. The Cohen's Kappa is 0.83, which indicates a high level of agreement. The proportions of

7. https://eslint.org/docs/developer-guide/contributing/working-on-issues

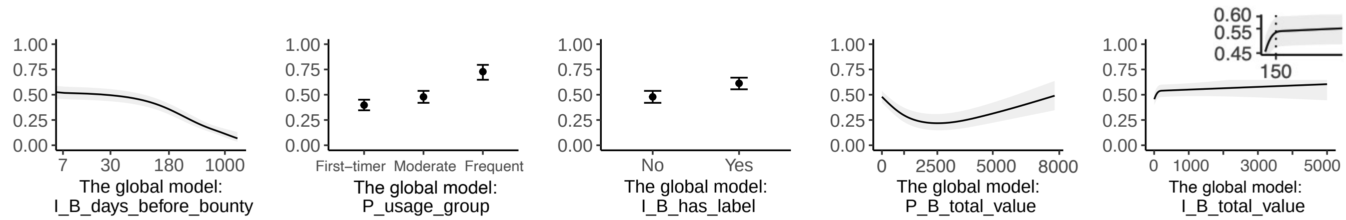

Fig. 10: The plots show the relationship between the studied factors and the issue-addressing likelihood for the global models. For each plot, we adjusted all factors except the studied factor to their median value in the model and recomputed the issue-addressing likelihood. The grey area represents the 95% confidence interval.

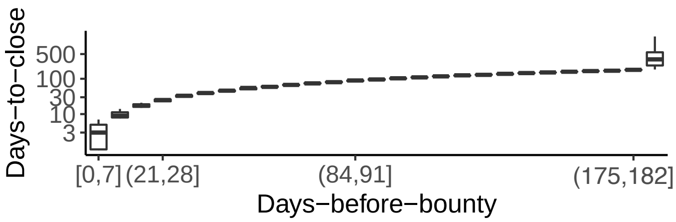

Fig. 11: The distribution of the number of days to close issue reports since bounties were proposed across different time ranges.

projects that have bounty documents are 5% (4/80), 31% (24/77), and 89% (8 out of 9) in the first-timer, moderate, and frequent projects, which suggests that projects that use bounties more frequently are more likely to have a bounty document.

In general, issue reports for which bounties were proposed earlier have a higher likelihood of being addressed. We observe a negative trend of the issue-addressing likelihood as the time to propose a bounty increases, especially for the issue reports in which bounties were proposed after 180 days. One possible explanation is that as time progresses, the risk of a report becoming obsolete exists, leaving the issue report unaddressed even after a bounty is proposed. For example, an issue report[8] that was created on Feb 4, 2016 in the `uappexplorer` project requested a new feature for an Ubuntu Phone Application. The owner of the application and another developer both showed great interest in this issue. Because of the lack of time, the feature was never added. A bounty of $5 was proposed[9] after almost one year, on Jan 12, 2017. However, the issue report was closed because Ubuntu Phone was no longer used making the issue report obsolete. In addition, **backers carry the risk of wasting their money by proposing small bounties on such long-standing issue reports as such small amounts are not refunded to the backer in case the bounty fails.**

Another assumption for the lower issue-addressing likelihood of the issue reports for which bounties were proposed later is that such issue reports are difficult to address. To test our assumption, we studied the relationship between the issue-addressing speed and *I_B_days_before_bounty*. Figure 11 shows the boxplot of the number of days that were taken to close issues (i.e., *days-to-close*) against different *days-before-bounty*. We observe that the issue reports in which bounties were proposed later took longer to be addressed.

8. https://github.com/bhdouglass/uappexplorer/issues/69
9. http://bit.ly/2Q3BIns

Issue reports with a bounty label have a higher likelihood of being addressed than bounty issues without a bounty label. Whether a bounty issue has a bounty label (i.e., *I_B_has_label*) is the third most important factor in the global model. Figure 10 shows that bounty issue reports with a bounty label have a higher likelihood of being addressed. It is intuitive that a better exposure of the bounty can help attract more attention from the community. Tagging an issue report with a bounty label is the most direct way of advertising a bounty because the label will be shown in the ITS. In addition, developers can search for bounty issue reports easily using the bounty label.

Finally, *I_B_total_value* contributes significant explanatory power to the global model and we suggest one to propose bounties with a value of $150. Figure 10 shows that the issue-addressing likelihood increases from 0.45 to 0.54 as the bounty value increases from $5 to $150 and stays almost stable after $150. In other words, the bounty value does not improve the issue-addressing likelihood further once the bounty value is equal to $150. The *P_B_total_value* is a significantly important factor in the global model, which indicates that the total amount of bounties that a project has is also of significance. Figure 10 shows that the issue-addressing likelihood and *P_B_total_value* has a negative relationship when the *P_B_total_value* is no more than $2,500. After $2,500, the higher *P_B_total_value*, the higher the issue-addressing likelihood.

The timing of proposing bounties is the most important factor that has a significant relation with the issue-addressing likelihood. Issue reports are more likely to be addressed if they are for projects in which bounties are used more frequently and if they are proposed earlier. In addition, it is important to advertise bounties for bounty issue reports by tagging bounty labels. The total value of bounties of a project and an issue also have a significant relation with the issue-addressing likelihood.

7.2 RQ2: How does the association between the studied factors and the issue-addressing likelihood change in projects with different bounty usage frequencies?

7.2.1 Approach

To understand how the association between bounties and the issue-addressing likelihood changes in projects with a different frequency of using bounties, we follow the same model construction and analysis approach as introduced in Section 7.1. Instead of building models on the entire set of issue reports, we build logistic regression models on

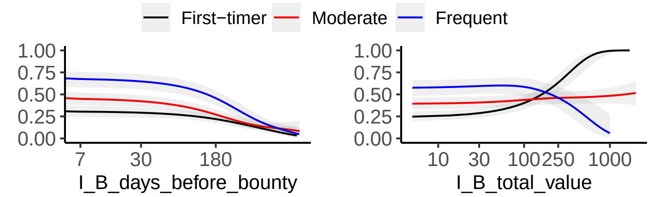

Fig. 12: The plots show the relationship between the studied factors and the issue-addressing likelihood for the first-timer models, the moderate models and the frequent models in the selected sample. For each plot, we adjusted all factors except the studied factor to their median value in the model and recomputed the issue-addressing likelihood. The grey area represents the 95% confidence interval.

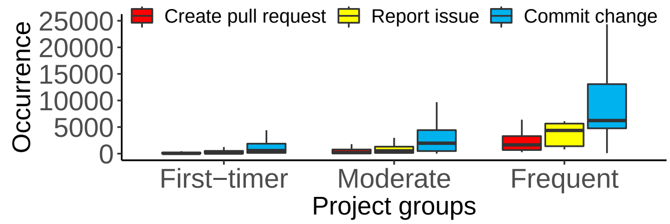

Fig. 13: The distributions of the occurrences of three activities (i.e., the create pull request, the report issue and the commit change) in each project group.

the bounty issue reports of each project group separately (i.e., the first-timer projects, the moderate projects, and the frequent projects). To condense our writing, we refer to the models for the first-timer, moderate, and frequent projects as the **first-timer**, **moderate**, and **frequent models**, respectively.

7.2.2 Results

Our models capture the relationship between the explanatory variables and the response variable well, and have a reliable performance. The median AUCs for the first-timer, moderate, and frequent models are 0.74, 0.70, and 0.82, respectively (see Table 3), which indicates that our models have a good ability to capture the relationship between the explanatory variables and the response variable. The low median optimism values (i.e., 0.01 for all models) indicate that our models do not overfit the dataset.

The timing of proposing bounties still plays a significantly important role in all three categories of projects. Table 4 shows that *I_B_days_before_bounty* is the most important factor (i.e., it contributes the highest explanatory power) in the moderate and the frequent models. In the first-timer model, *I_B_days_before_bounty* is the second important factor. Figure 12 presents the relationship between the issue-addressing likelihood and *I_B_days_before_bounty* for the first-timer, moderate, and frequent models. We observe that *I_B_days_before_bounty* has the same negative relationship with the issue-addressing likelihood in all three models. We also observe that the frequent model has the highest issue-addressing likelihood compared with the first-timer model and the moderate model when receiving bounties in the same number of *days-before-bounty*, which indicates that proposing bounties earlier will achieve the highest issue-addressing likelihood in projects which use bounties frequently.

The total bounty value of an issue report is the most important factor that has an association with the issue-addressing likelihood in the first-timer projects, while it is less important in the projects where bounties are used more frequently. From Table 4, we can see that *I_B_total_value* (i.e., the total bounty value of a bounty issue report) is the most important factor in the first-timer model with a positive association with the issue-addressing likelihood, while it is not a significantly important factor (the p-value is larger than 0.001) in the moderate and frequent models. When comparing the ratio of the bounty value between successful and failed issue reports among the first-timer, moderate, and frequent projects, we can see that the first-timer projects have a larger ratio (2.5) than the moderate (2) and frequent projects (1.4). This explains why the value of bounty is more important in the first-timer projects than that in the moderate and frequent projects. The highest ratio in the first-timer projects also indicates that developers may expect a better payout when addressing issues in first-timer projects than in other projects.

Why do the first-timer projects have a larger ratio than moderate and frequent projects? One possible assumption is that the first-timer projects may not be as active as moderate and frequent projects, therefore backers would be required to propose bounties with higher values to attract enough attention from the community for addressing issues. To investigate this assumption, we examined the frequency of various activities of the projects, in terms of the number of pull requests, issue reports, and commits. Figure 13 shows the distributions of the occurrences of these three activities in each project group. **Projects with fewer bounty issue reports are usually less active (in terms of the number of pull requests, issue reports, and commits) than projects with more bounty issue reports.** Another possible explanation is that backers in first-timer projects have no experience in proposing bounties and sometimes overestimate the value of addressing an issue report. In this situation, the overestimated bounty issue reports may be more likely to attract more attention from the community and get addressed.

For the frequent model, we observe a negative relationship between the issue-addressing likelihood and the total bounty value. One possible explanation is that in the frequent projects, where communities have more experience in using bounties, backers are more likely to propose bounties with a well-estimated value. Therefore, issue reports with bounties of higher value are more likely difficult to resolve and have a lower issue-addressing likelihood. For the moderate model, we observe a weak positive relationship.

Except for the bounty-related factors that we discussed above, we observed other factors from the project bounty and the backer experience dimensions which are also significantly important (i.e., the p-value of the χ^2-test is less than 0.001) in frequent models. In the backer experience dimension, the max value of bounties which the backers of this bounty have ever proposed in the past (i.e., *Backer_exp_B_max_value*) is significantly important in the frequent models, while it is not significantly important in the other two models. In other words, the experience of backers is more important in projects that use bounties frequently than in those that use bounties less frequently. We also observed that the proportion of paid bounty issue reports (i.e., *P_-B_paid_proportion*) plays a significant role in the frequent models, while its role is not significant in the other two models. In addition, it has a positive association with the issue-addressing likelihood of bounty issue reports. In short, the project bounty and the backer experience dimensions are more important in frequent models than in another two models.

In the issue report basic dimension, the proportion of code in an issue report (i.e., *I_code_proportion*) is important in moderate models.

In general, the timing of proposing bounties is the most important factor that has a relation with the issue-addressing likelihood in moderate and frequent projects. The total bounty value that an issue report has is the most important factor that has a relation with the issue-addressing likelihood in the first-timer projects, while it is not as important for projects in which bounties are more frequently used.

8 Discussion

In this section, we first study the ignored bounty issue reports. Then we highlight the implications of our findings.

8.1 Studying ignored bounty issue reports

In Section 6, we observed that in 19.7% of the bounty issue reports the bounties were ignored (i.e., closed-unpaid). In these cases, the issue reports were closed but the bounties remained unclaimed. It seems that money was not the driver that motivated developers to address these issues. To understand the possible reasons behind this phenomenon, we manually studied all 692 ignored bounty issue reports (with a total bounty value of $41,856). Because the "closed" status of an issue report does not necessarily mean that the issue was addressed (e.g., a report may have been a duplicate of another issue report), it is difficult to automatically identify whether an issue in the closed issue report was addressed. Therefore, we need to manually examine the closed-unpaid bounty issues reports to filter out the reports that were closed for another reason than the issue being addressed.

21.8% (479 out of 2,200) of the addressed bounty issue reports were not paid out. We identified that 479 out of the studied 692 bounty issue reports were closed because the issues were addressed. Such cases are interesting since the developers could have claimed the bounty but they did not. We manually examined the discussion for these 479 issue reports. We identified 19 cases in which developers gave an explanation for not claiming the bounty. We grouped the explanations as follows:

The developer is not driven by money. In 7 out of 19 cases a developer refused to claim the bounty because they were not motivated by money to address the issue. For example, one developer was against the bounty because they felt that the issue-addressing process should be driven by the interests of the community rather than money. A contributor of the `Brython` project, refused the bounty because he wanted to keep `Brython` free from monetary motivations: "*What is this 'bounty' thing? Needless to say, I refuse that anybody (me included, of course) gets paid for anything related to `Brython`.*"[10] In addition, he also asked backers to remove all bounties within the `Brython` project although he respected prior paid bounties. There were five bounty issue reports in the `Brython` project and four bounty issue reports that were addressed without claiming the bounty.

The developer is afraid of sending the wrong message. Krishnamurthy and Tripathi [17] pointed out that financial incentives may cause confusion in the community because the financial incentives may drive a project's own product development cycle away from what is in place. We observed that developers expressed similar concerns. A developer of the `Facebook/HHVM` project, explained that: "*That's very generous of you, but I can't accept a bounty for doing my job. :-P It would be a conflict of interest, and I worry it sends the wrong message about how we prioritize issues from the community.*"[11]

The issue report was addressed by more than one developer. We found nine cases where bounties ended up unclaimed because an issue report was addressed by multiple developers cooperatively and they felt inappropriate to claim the bounty by one developer. For example, the issue[12] was addressed by two developers and because a bounty cannot be split into two parts, no one claimed it.

8.2 The implications of our findings

Backers should consider proposing a bounty as early as possible and be cautious when proposing small bounties on long-standing issue reports. The timing of proposing a bounty is an important factor that is related to the issue-addressing likelihood. In Sections 7.1 and 7.2, we showed that issue reports for which bounties were proposed earlier are more likely to be addressed. Additionally, we observed that issue reports for which bounties were proposed earlier are more likely to be addressed faster. Backers benefit from the higher issue-addressing likelihood and faster issue-addressing speed by proposing bounties earlier.

In Section 7.1, we also noticed a drop (i.e., from 53.2% to 30.1%) of the issue-addressing likelihood when backers proposed bounties for long-standing (i.e., more than half a year) issue reports. This drop might be due to such issue reports having become obsolete or being hard to address. Since bounties with a value of less than $100 will not be refunded to the backers if the issue report remains unaddressed, we suggest that backers be cautious when proposing small bounties on long-standing issue reports.

Backers should consider proposing a bigger bounty in first-timer bounty-projects. Although the issue-addressing likelihood is only 37.4% for projects with no bounty-usage experience, the first-timer model in Section 7.2 shows that the bounty value of an issue report is the most important factor in the first-timer projects, as the issue-addressing likelihood is higher for higher bounty values. The high ratio (2.5) of the bounty value of successful bounty issue reports to the bounty value of failed bounty issue reports also supports this finding. We suggest that backers of projects with no bounty-usage experience propose higher bounty values for issue reports.

10. http://bit.ly/2OTYx0x
11. http://bit.ly/2OZw1uw
12. http://bit.ly/2PrMiHV

Bounty platforms should allow for splittable multi-hunter bounties. In addition to a voluntary nature, open source projects have a collaborative nature. Some issues are hard for a developer to address alone. Hence, we encourage developers to work together, especially for issue reports which have a high bounty value (as these issue reports are often harder to address). However, the current bounty workflow only allows **one** bounty hunter to claim the bounty, which goes against the collaborative nature of open source. It may also drive the developers, who want to collaboratively address the issue, away because not every participant will get a reward at the end. Therefore, bounty platforms should consider adding the ability for a bounty to be split across multiple hunters to encourage developers to work together on difficult bounty issues.

Bounties should be transferable. The total value of all addressed-unpaid bounties ($43,256) is “frozen” in Bountysource. In addition, the median number of days between the closing date of the issue report and the date of collecting our data is 372.5 (Figure 11), which means that more than half of the bounties from the ignored bounty issue reports were unclaimed for at least one year. By manually examining these 479 addressed-unpaid bounty issue reports, we found 31 cases in which someone reminded the bounty hunter to claim the bounty, however, the reminder was ignored. By reassigning these unclaimed bounties to other issue reports, a larger value could be created for these “stale” bounties. For example, Bountysource can suggest and enable backers to assign their long-standing unclaimed bounties to another unaddressed issue report, which has many comments (i.e., people care about it), to encourage developers to address the issue report. Interestingly, we also found suggestions from developers who did not want to receive the bounty but suggested the bounty backers transferring the bounty to other issue reports or to the project as a kind of funding.

9 Threats to Validity

In this section, we discuss the threats to the validity of our results.

Threats to **external validity** are related to the generalizability of our findings. We studied only bounty issue reports from GitHub and Bountysource. Future research should study issue reports from other bounty platforms, issue tracking systems and open source projects to determine whether our findings are generalizable to other types of issue reports (e.g., from commercial platforms), other bounty platforms and projects. Although our models have a high explanatory power, there might be additional factors that relate to the likelihood of an issue being addressed. Future studies should investigate more factors.

Threats to **internal validity** relate to the experimenter bias and errors. One threat is that we rely on manual analysis to identify the addressed-unpaid issues and to identify why developers did not claim a bounty in Section 8.1, which may introduce bias due to human factors. To mitigate the threat of bias during the manual analysis, two of the authors conducted the manual analysis and discussed conflicts until a consensus was reached. We used Cohen’s kappa [8] to measure the inter-rater agreement and the value is 0.86, which indicates a high level of agreement.

There are many additional factors which may have an association with our model, e.g., the type of a project. Since there is no clearly defined project type for a project in GitHub, we would need to manually identify the project type (which would introduce a bias as well). Future studies should consider this factor if the type of a project can be clearly defined.

Another threat is that we regarded all open issue reports as failed ones, which may introduce bias, since some issue reports could be worked on by one or more hunters at the time we collected our data. However, it is not possible to distinguish between bounties which are worked on or actual failed bounties, since it is not mandatory for a hunter to update their progress on an issue report. To alleviate this threat, we updated the status of our studied issue reports after 200 days since the first time of our data collection. In other words, only the issue reports that remain unsolved for more than 200 days are regarded as failed ones in this study.

Threats to **construct validity** concern the relation between theory and observation. One threat relates to the project categorization in Section 7, in which we used 50 bounty issue reports as a threshold to distinguish whether a project uses bounties moderately or frequently. To alleviate this threat, we redid the analysis of Section 7 with other thresholds for bounty-usage frequency (i.e., 40 and 60). The results show that our findings still hold (see Appendix C in our supplementary material [34] for more details).

Threats to **conclusion validity** concern the relation between the treatment and the outcome. One threat is caused by the statistical tests that we performed. To alleviate the threat, we used non-parametric tests that do not make an assumption about the underlying data distribution. Another threat is that there may exist confounding factors that bias our conclusion. To alleviate this threat, we constructed multi-factor models to control for confounding factors.

10 Conclusion

In this paper, we studied 5,445 bounties with a total value of $406,425 from Bountysource along with their associated 3,509 issue reports from GitHub to study the relationship between the bounty (e.g., timing of proposing a bounty, bounty value, and bounty-usage frequency) and the issue-addressing likelihood. We found that:

1) The timing of proposing bounties is the most important factor that is related to the issue-addressing likelihood. Issue reports are more likely to be addressed with a faster addressing-speed if bounties are proposed earlier.
2) In first-timer bounty-projects, the issue-addressing likelihood is higher for higher bounty values and in these projects, backers should consider proposing a relatively bigger bounty.
3) Backers should be cautious when proposing small bounties on long-standing issue reports as they risk losing money without getting their issue addressed.

Our findings suggest that backers should consider proposing a bounty early and be cautious when proposing small bounties on long-standing issue reports. Bounty platforms should allow dividing bounties between hunters, and transferring bounties to other issue reports.

References

[1] D. Arya, W. Wang, J. L. Guo, and J. Cheng. Analysis and detection of information types of open source software issue discussions. In *Proceedings of the 41st International Conference on Software Engineering*, pages 454–464. IEEE Press, 2019.

[2] N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann. What makes a good bug report? In *Proc. of the Int'l Symp. on Foundations of Software Engineering*, pages 308–318. ACM, 2008.

[3] W. G. Cochran. *Sampling techniques*. John Wiley & Sons, 2007.

[4] C. F. Dormann, J. Elith, S. Bacher, C. Buchmann, G. Carl, G. Carré, J. R. G. Marquéz, B. Gruber, B. Lafourcade, P. J. Leitão, et al. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1):27–46, 2013.

[5] B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.

[6] D. E. Farrar and R. R. Glauber. Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics*, pages 92–107, 1967.

[7] M. Finifter, D. Akhawe, and D. Wagner. An empirical study of vulnerability rewards programs. In *USENIX Security Symp.*, pages 273–288, 2013.

[8] K. Gwet et al. Inter-rater reliability: dependency on trait prevalence and marginal homogeneity. *Statistical Methods for Inter-Rater Reliability Assessment Series*, 2:1–9, 2002.

[9] T. Gyimothy, R. Ferenc, and I. Siket. Empirical validation of object-oriented metrics on open source software for fault prediction. *IEEE Transactions on Software engineering*, 31(10):897–910, 2005.

[10] F. E. Harrell, Jr. *Regression Modeling Strategies*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] H. Hata, M. Guo, and M. A. Babar. Understanding the heterogeneity of contributors in bug bounty programs. In *Proc. of the ACM/IEEE Int'l Symp. on Empirical Software Engineering and Measurement*, pages 223–228, 2017.

[12] P. He, B. Li, X. Liu, J. Chen, and Y. Ma. An empirical study on software defect prediction with a simplified metric set. *Information and Software Technology*, 59:170–190, 2015.

[13] P. Hooimeijer and W. Weimer. Modeling bug report quality. In *Proc. of the Int'l Conf. on Automated Software Engineering*, pages 34–43. ACM, 2007.

[14] O. Jarczyk, S. Jaroszewicz, A. Wierzbicki, K. Pawlak, and M. Jankowski-Lorek. Surgical teams on GitHub: Modeling performance of GitHub project development processes. *Information and Software Technology*, 100:32–46, 2018.

[15] S. Kabinna, C.-P. Bezemer, W. Shang, M. D. Syer, and A. E. Hassan. Examining the stability of logging statements. *Empirical Software Engineering*, 23(1):290–333, 2018.

[16] T. Kanda, M. Guo, H. Hata, and K. Matsumoto. Towards understanding an open-source bounty: Analysis of Bountysource. In *Int'l Conf. on Software Analysis, Evolution and Reengineering*, pages 577–578. IEEE, 2017.

[17] S. Krishnamurthy and A. K. Tripathi. Bounty programs in free/libre/open source software. In *The Economics of Open Source Software Development*, pages 165–183. Elsevier, 2006.

[18] T. Maillart, M. Zhao, J. Grossklags, and J. Chuang. Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity*, 3(2):81–90, 2017.

[19] S. McIntosh, Y. Kamei, B. Adams, and A. E. Hassan. An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering*, 21(5):2146–2189, 2016.

[20] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are bullies more productive?: Empirical study of affectiveness vs. issue fixing time. In *Proc. of the Working Conf. on Mining Software Repositories*, pages 303–313, 2015.

[21] F. Palomba, M. Zanoni, F. A. Fontana, A. De Lucia, and R. Oliveto. Toward a smell-aware bug prediction model. *IEEE Transactions on Software Engineering*, 2017.

[22] G. K. Rajbahadur, S. Wang, Y. Kamei, and A. E. Hassan. The impact of using regression models to build defect classifiers. In *Proc. of the Int'l Conf. on Mining Software Repositories*, pages 135–145, 2017.

[23] R. K. Saha, M. Lease, S. Khurshid, and D. E. Perry. Improving bug localization using structured information retrieval. In *Proc. of the IEEE/ACM Int'l Conf. on Automated Software Engineering*, pages 345–355, 2013.

[24] W. S. Sarle. The VARCLUS Procedure. In: SAS/STAT User's Guide, 4th edn. 1990.

[25] M. Soto, F. Thung, C.-P. Wong, C. Le Goues, and D. Lo. A deeper look into bug fixes: Patterns, replacements, deletions, and additions. In *Proc. of the Int'l Conf. on Mining Software Repositories*, pages 512–515, 2016.

[26] S. Wang, T.-H. Chen, and A. E. Hassan. Understanding the factors for fast answers in technical Q&A websites. *Empirical Software Engineering*, pages 1–42, 2017.

[27] S. Wang and D. Lo. Version history, similar report, and structure: Putting them together for improved bug localization. In *Proceedings of the 22nd International Conference on Program Comprehension*, pages 53–63. ACM, 2014.

[28] S. Wang and D. Lo. Amalgam+: Composing rich information sources for accurate bug localization. *Journal of Software: Evolution and Process*, 28(10):921–942, 2016.

[29] S. Wang, D. Lo, and J. Lawall. Compositional vector space models for improved bug localization. In *IEEE Int'l Conf. on Software Maintenance and Evolution*, pages 171–180, 2014.

[30] S. Wang, D. Lo, Z. Xing, and L. Jiang. Concern localization using information retrieval: An empirical study on Linux kernel. In *2011 18th Working Conference on Reverse Engineering*, pages 92–96. IEEE, 2011.

[31] M. Zhao, J. Grossklags, and K. Chen. An exploratory study of white hat behaviors in a web vulnerability disclosure program. In *Proc. of the Workshop on Security Information Workers*, pages 51–58. ACM, 2014.

[32] M. Zhao, A. Laszka, and J. Grossklags. Devising effective policies for bug-bounty platforms and security vulnerability discovery. *Journal of Information Policy*, 7:372–418, 2017.

[33] H. Zhong and Z. Su. An empirical study on real bug fixes. In *Proc. of the Int'l Conf. on Software Engineering*, pages 913–923, 2015.

[34] J. Zhou. Supplementary material for our paper. https://github.com/SAILResearch/wip-18-jiayuan-bountysource-SupportMaterials/blob/master/appendix.pdf, 2018.

[35] J. Zhou, S. Wang, C.-P. Bezemer, and A. E. Hassan. Bounties on technical Q&A sites: A case study of Stack Overflow bounties. *Empirical Software Engineering*, pages 1–41, 2019.

[36] J. Zhou, H. Zhang, and D. Lo. Where should the bugs be fixed? - more accurate information retrieval-based bug localization based on bug reports. In *Proceedings of the 34th International Conference on Software Engineering*, pages 14–24, 2012.