Noname manuscript No.
(will be inserted by the editor)

Studying Backers and Hunters in Bounty Issue Addressing Process of Open Source Projects

Jiayuan Zhou · Shaowei Wang · Haoxiang Zhang · Tse-Hsun (Peter) Chen · Ahmed E. Hassan

Received: date / Accepted: date

Abstract Issue addressing is a vital task in the evolution of software projects. However, in practice, not all issues can be addressed on time. To facilitate the issue addressing process, monetary incentives (e.g., bounties) are used to attract developers to address issues. There are two types of core roles who are involved in this process: *bounty backers*, who propose bounties for an issue report via bounty platforms (e.g., Bountysource), and *bounty hunters*, who address the bounty issues and win the bounties. We wish to study the process of bounty issue addressing from the angle of two important roles (i.e., backers and hunters) and their related behaviors. With a better understanding of how they address bounty issues, stakeholders (e.g., operators and developers) of open source projects may have a reasonable estimation of what they can expect from backers and hunters.

In this study, we investigate 2,955 bounty backers and 882 bounty hunters, and their associated 3,579 GitHub issue reports with 5,589 bounties that were proposed on Bountysource. We find that: 1) Overall, the value of a bounty is small (median bounty value of $20). Both individual and corporate backers

Jiayuan Zhou · Haoxiang Zhang
Centre for Software Excellence (CSE), Huawei, Canada
E-mail: jiayuan.zhou1, haoxiang.zhang@huawei.com This work is not related to their roles at Huawei.

Shaowei Wang (✉)
Department of Computer Science, University of Manitoba, Canada
E-mail: shaowei@cs.umanitoba.ca

Tse-Hsun (Peter) Chen
Software PErformance, Analysis, and Reliability (SPEAR) Lab, Concordia University, Montreal, Canada
E-mail: peterc@encs.concordia.ca

Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL), Queen's University, Kingston, ON, Canada
E-mail: ahmed@cs.queensu.ca

prefer to support implementing new features rather than fixing bugs. Corporate backers tend to propose larger bounties and propose bounties more frequently than individual backers. 2) 85.0% of the bounty hunters addressed less than 3 bounty issues. The income of 56.7% of the bounty hunters is no more than $100 and only 2.7% of the hunters have earned more than $2,000. In addition, most of the regular hunters and big hunters are developers that made at least one commit before addressing a bounty issue. 3) The value of a bounty issue is not a statistically significant factor that attracts developers that have never made any commit before to address an issue. Based on our findings, we provide several suggestions for stakeholders of open source projects and hunters.

Keywords bounty, open source projects, GitHub, issue addressing.

1 Introduction

Open source projects are widely used by many companies, government agencies, and individuals (Androutsellis-Theotokis *et al.*, 2011). Sustaining open source projects is challenging and requires a significant amount of effort from developers. A vital task for sustaining open source software projects is issue addressing. However, some issues may never be addressed, which in turn affects the users. Developers may avoid addressing issues that they consider of a low priority, or difficult to implement since such issues may require a massive amount of effort from developers (Roberts *et al.*, 2006).

Financial incentives are an important extrinsic motivator for developers to sustain open source projects (Atiq and Tripathi, 2016). Bounties are now being used to motivate developers to address issue reports, e.g., to fix bugs or to add new features. For instance, *Bountysource*[1] is a popular platform that allows users to propose bounties for open source projects that are hosted on multiple platforms (e.g., GitHub). Users on such bounty platforms (i.e., *bounty backers*) can propose bounties for an issue report and developers (i.e., *bounty hunters*) can address such bounty issues and win the bounties. These two types of contributors play a core role in the bounty issue addressing process. Several studies examined the impact of bounties on issue addressing (Zhou *et al.*, 2020b; Kanda *et al.*, 2017) and vulnerability discovery (Finifter *et al.*, 2013; Maillart *et al.*, 2017; Zhao *et al.*, 2017). The study by Zhou *et al.* (2020b), which is the most relevant to our study, focuses on studying the association between bounty-related factors and the likelihood of a bounty issue being addressed. It provides insights for backers on proposing bounties, e.g., backers should be cautious when proposing small bounties on long-standing issue reports since the risk of losing the bounty exists. However, the characteristics and behaviors of bounty backers and hunters, which are the two most important roles in the issue addressing process, and their bounty related behaviors have not been examined in depth. We wish to study the process of bounty issue addressing

[1] https://www.bountysource.com

from the angle of its main roles (i.e., backers and hunters). With a better understanding of the process, stakeholders (e.g., operators and developers) of open source projects may have a view of what they can expect from backers and hunters.

In this paper, we study 2,955 bounty backers and 882 bounty hunters, and their associated 3,579 issue reports. In total, there were 5,589 bounties (with a total bounty value of $412,478) that were proposed on Bountysource for 1,210 GitHub projects. We examine the characteristics of bounty backers and hunters, and their bounty-related behaviors. Our results highlight that:

– 95.2% of backers are individual backers and they supported 81.8% of all offered bounties (4,282 out of 5,243). Although corporate backers only represent a small portion of the population (4.8%), they contributed almost half (46.4%) of the total bounty amount.
– Overall, the value of bounties is small (median bounty value of $20). Corporate backers (median frequency of 2 and median bounty value of $25) tend to propose larger bounties and to propose bounties more frequently than individual backers (median bounty value of $15 and median frequency of 1). Both individual and corporate backers prefer to support implementing new features rather than fixing bugs.
– In general, 85.0% of the bounty hunters addressed less than 3 bounty issues. Only 2.1% of the hunters addressed more than 10 bounty issues. Most of the regular hunters and big hunters are the developers who has committed code to the projects before.
– The income of 56.7% of the bounty hunters is no more than $100 and only 2.7% of the hunters earned more than $2,000. 67.3% (502 out of 746) of the studied hunters who committed code to the projects before and they addressed 71.8% (1039 out of 1,448) of bounty issues.

Engaging and retaining new developers is important for promoting a sustainable community. Therefore, we further construct a logistic regression model to help stakeholders of open source projects understand the possible factors that may be associated with the likelihood of attracting developers that have never made any commit before addressing a bounty issue in a project to address bounty issue (i.e., new developers). We investigate 35 factors along five dimensions (e.g., project, bounty, and issue) and find that:

– Interestingly, the value of a bounty issue is not a statistically significant factor that attracts new developers of a project to address the issue.
– Bounty issues with a bounty label and more frequent usage of bounties are less likely to be addressed by new developers. Bounty issues of popular and aged projects are more likely to be addressed by new developers.

Based on our findings, we have several suggestions for stakeholders of open source projects and hunters. For instance, hunters may not expect to earn a large amount of money from addressing bounty issues on open source projects. Stakeholders of open source projects should not expect backers to support addressing bugs and should not expect a large amount of bounties to support issues unless their associated projects are very popular.

In summary, we highlight our contributions as follows: 1) We conduct an exploratory analysis on two important roles (i.e., backers and hunters) in the bounty issue addressing process of open source projects and provide a landscape of their characteristics and bounty-related behaviors. 2) We examine the relationship between various factors that are related to project, issue, bounty, backers, and the likelihood of a bounty issue being addressed by hunters who never commit code to the project before. Our observations provide insights to stakeholders of open source projects, hunters, and future research.

The paper is organized as follows. Section 2 introduces the background of Bountysource and GitHub's issue tracking system and related work. Section 3 presents our three research questions and design of our study. Section 4 presents findings of our five research questions. In Section 5, we discuss the implications of our study and future research directions. Section 6 discusses the threats to validity of our study. Section 7 makes conclusions for our study.

2 Background & Related Work

2.1 Bountysource

Bountysource is a platform on which users can offer a monetary incentive (i.e., a *bounty*) to address an issue report of an open source project. Users can have two roles on Bountysource: bounty backers and bounty hunters.

Bounty backers are users or developers who propose bounties for issue reports. An issue report can have multiple bounties from one or more backers, and a bounty can only be proposed for one issue report. 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. Bounty backers can be anonymous. Bountysource allows users to identify themselves as one of the two types of backers: *individual* and *corporate*. Corporate backers are referred as to corporations or organizations. Individual backers are referred as to the backer who is an individual person. Intuitively, these two types of backers have different characteristics and behaviors. We investigate their characteristics in Section 4.1. For the sake of simplicity, we refer to bounty backers as backers unless stated otherwise.

Bounty hunters are developers who address bounty issue reports (i.e., the issue reports that have bounties). If a hunter works on an issue report, the hunter can choose to attach certain information (i.e., the estimated time of addressing, the code URL, or some comments) on Bountysource to indicate the progress. However, we observe that in most of the cases, hunters did not leave any information. In other words, developers often work on bounty issues silently until they have addressed the issues. Once a developer claims to have addressed an issue report, its bounty backer(s) have to make a decision (accept or reject) on the claim within two weeks. If no backer explicitly rejects the claim, the bounties will be paid to the developer automatically after two weeks. One or more developers can choose to become bounty hunters to address the issue report but only one bounty hunter can get the bounty. However, hunters would not be aware of this if no one reports the progress information. For the sake of simplicity, we refer to bounty hunters as hunters unless stated otherwise. In this study, we investigate two types of hunters: the hunters who commit code to address a bounty issue for the first time (i.e., *new hunters*) and the hunters who have committed code to the project before. In this study, we use commit information to identify two types of hunters since bounties will only be paid to hunters who successfully address issues by committing code changes. We wish to investigate whether hunters with experience of making code contributions have different characteristics and behaviors from the ones without any such experience. This commit-based heuristic has also been used commonly in prior studies to identify different types of developers (Mockus *et al.*, 2002; Robles *et al.*, 2009; Coelho *et al.*, 2018). We investigate the characteristics of hunters and investigate the differences between these two types of hunters in Section 4.2. Hunters who commit code to an open source project for the first time are also considered as new developers to the open source project. Engaging and retaining new developers is important for promoting a sustainable community. Therefore, we also investigate the characteristics of bounty issues that are more likely to be addressed successfully by hunters who committed code to the project for the first time in Section 4.5.

Figure 1 shows the workflow of the bounty processes between backers and hunters, through Bountysource and GitHub. The process starts with a bounty backer offering a bounty on Bountysource for a GitHub issue report. Bountysource will link the bounty to the issue report on GitHub. Then bounty backers can choose to expose bounty information to the GitHub issue report, e.g., tagging the issue report on GitHub with a *bounty label* (see the example[2] 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 on GitHub. When a bounty hunter starts working on an issue, he/she can (although not required) update the 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. Note that the money can only be paid to one hunter according to Bountysource's current mechanism.

2.2 Related work and motivation

Funding is important for the creation and maintenance of open source projects and there are various types of approaches to fund open source projects (Eghbal,

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

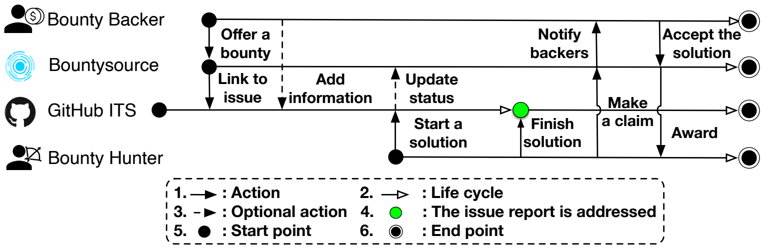

Fig. 1: The workflow of the bounty between bounty backer, bounty hunter, GitHub ITS, and Bountysource.

2019), such as providing financial support (e.g., donations and bounties) and hiring developers to work on the projects. Funding is usually provided by individual and corporate backers.

As financial incentives (e.g., money) have become more widespread, a body of research focuses on investigating the impact of financial incentives on participants both in positive and negative ways. On one hand, financial incentives do engage participants to make contributions to open source projects, e.g., fixing bugs and adding new features (Zhou *et al.*, 2020b; Krishnamurthy and Tripathi, 2006). On other hand, financial incentives also have a negative impact on open source projects. Zhou *et al.* (2016) observed that commercial involvement can increase the inflow of paid developers in an open source project, but may potentially decrease the retention of key developers. Studies also show that external financial incentives can undermine the intrinsic motivation for participants, change their mindset from volunteers to unpaid employees, and work passively (Frey and Goette, 1999). Zhou *et al.* (2020a) observe that some developers refuse to accept bounties even after addressing bounty issues since they thought accepting money would bring wrong messages for the open source community. Nakasai *et al.* (2018) observe that developers respond faster to bug reports that are submitted by users that have donor badges, which are used to acknowledge users for their contribution in donation, than users that do not have any donor badges.

Various studies have been done to investigate why individual developers and corporations provide support to open source projects. Some developers are driven by intrinsic incentives (e.g., enjoyment and sense of obligation and volunteer) (Zhou *et al.*, 2020b; Lakhani and Wolf, 2003; Shah, 2006; Von Krogh *et al.*, 2012; Coelho *et al.*, 2018). For example, Zhou *et al.* (2020b) observed that some developers fixed issues with bounties and were not willing to receive bounties. On the one hand, some developers make contributions because of extrinsic (e.g., getting paid or reputation) reasons (Coelho *et al.*, 2018; Lakhani and Wolf, 2003; Roberts *et al.*, 2006; Shah, 2006; Von Krogh *et al.*, 2012; Krishnamurthy *et al.*, 2014). Economic motivation has been shown as a factor to attract individual developers to contribute to open source projects. For example, over 70% of changes to the Linux kernel and over 80% of commits to the Eclipse platform have been made by developers who are paid by companies to contribute to those projects (Weiss, 2011). Corporations support open source projects for promoting the important movement of the open source projects for their business (Izquierdo and Cabot, 2018). Corporations use open source to reduce their development and maintenance cost and to shorten their time to market (Weiss, 2011). In addition, developers of open source projects create innovation in a way that has a significant advantage over the manufacturer-centric innovation development systems (Von Hippel, 2007). Therefore, corporations are motivated to participate in open source projects if innovations enhance profits (Harhoff *et al.*, 2003).

Bounties, as one of the funding models, are used to attract developers and motivate them to complete various software engineering tasks. One of the most common software engineering tasks is attracting hunters to disclose software security vulnerabilities. For instance, Google rewards bounties to the communities (e.g., hackers and researchers) to promote the disclosure of security bugs for Chrome[3]. A number of prior studies investigate the impact and the effectiveness of such bounty programs on the vulnerability disclosure process (Finifter *et al.*, 2013; Zhao *et al.*, 2017; Maillart *et al.*, 2017). For instance, Finifter *et al.* (2013) 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.* (2017) and Maillart *et al.* (2017) analyzed the effect of different policies of security bug bounty programs and they provided insights on how to improve such programs, e.g., project managers should dynamically adjust the value of rewards according to the market situation (e.g., increase rewards when releasing a new version).

Vulnerability bounty programs are usually supported by corporations and the amount of bounties is usually big. Recent reports show that Firefox paid out an average of $2,775 for their security bug bounties between 2017 and 2019 (Tom Ritter, 2020). HackerOne platform reports that corporations have awarded hunters over $31 million from 2012 to June 2018 and the average bounty paid for critical vulnerabilities across all industries on the HackerOne platform is $2,041 in 2017 (HackerOne, 2018). Apple provides a reward up to $1 million for a specific iPhone hack as part of its expanded bug-bounty program (Apple Inc, 2020). Different from such vulnerability bounty programs, bounties that are proposed for issues on GitHub projects are not specific for security bugs and are not always supported by corporations. Many of such bounties are proposed by individual backers. However, little is known about the bounties that are proposed by such individual backers in the issue addressing context on open source projects compared with corporate backers. In addition, Eghbal (2016) reported the risks and challenges that are associated with maintaining open source projects, and argued that open source projects still lack a reliable and sustainable source of funds. Therefore, it is interesting to investigate the backers in such a context.

There are two specific types of backers: individual backers and corporate backers as we discussed in Section 2. Intuitively, these two types of backers

[3] https://www.google.com/about/appsecurity/chrome-rewards/

are different from each other due to their nature and motivation as discussed above. For example, corporate backers may offer a larger bounty for addressing an issue compared to individual backers and support in a more continuous fashion instead of one shot. Therefore, to help stakeholders of an open source project obtain a view on what types of backers they can expect, how much bounties they would typically receive from backers, first, we investigate the following RQ:

> *RQ1: How do individual and corporate backers propose bounties in terms of bounty amount and proposing frequency?*

Prior studies mainly focus on the bounty program for security bugs, little is known about feature requests and the preference between bug reports vs feature requests. Individual and corporate backers may have a different preference in offering bounties to bug reports and feature requests. Therefore, we investigate the following RQ:

> *RQ2: What are the preferences of individual and corporate backers in supporting feature requests and bug reports.*

Zhao *et al.* (2014) investigated hunters in security bug bounty programs and found that the diversity of hunters improved the productivity of the vulnerability discovery process. Hata *et al.* (2017) found that most hunters are not very active (i.e., they have only a few contributions). They also observed that most hunters are not project-specific and that bounty program managers should strive to attract non-project-specific security specialists with reasonable bounties. Several reports show that the investment from corporations for such bounty programs is huge and certain hunters have made a significant income from hunting such security-related bounties. HackerOne's annual report also reveals that seven hackers have now earned more than $1 million in bug bounties so far in their career, with another 13 surpassing $500,000 in lifetime earnings (Robert Lemos, 2019). Similarly, little is known about the hunters in the issue addressing context in open source projects. For example, are bounty hunters continuously active in hunting bounties? How much money does a hunter make through addressing bounty issues? Therefore, we investigate the following research question:

> *RQ3: How much do hunters earn from hunting bounty issues?*

For a project, there are also two types of hunters, i.e., hunters who commit code to address a bounty issue for the first time (i.e., new hunters) and hunters who committed code before. Due to their different characteristics (e.g., levels of experience for the project), we suspect they behave differently. For example, hunters who committed code to the project before are probably more likely to hunt bounty issues more frequently than new hunters. Therefore, it is interesting to investigate how these two types of hunters behave. Lee *et al.* (2017) studied barriers that are experienced by the one-time developers and found various barriers that hinder such one-time developers to become long-term developers, e.g., entry difficulties and lack of time. To help new developers, Canfora *et al.* (2012) proposed an approach to identify and recommend mentors in software projects by mining data from mailing lists and version control systems. Bounty could be an external incentive to attract developers and improve the sustainability of open source projects. It is also interesting to investigate what barriers hunters occur when hunting bounty issues? We investigate the following research question:

> *RQ4: How do different hunters hunt bounty issues in terms of bounty amount and proposing frequency?*

Sustaining open source projects is challenging and usually relies on a very few developers. For instance, Avelino *et al.* (2016) found that the majority (65%) of their studied projects rely on one or two developers to survive. Therefore, attracting new developers is essential to maintain the sustainability of an open source project. Ye and Kishida (2003) investigated the motivation of developers to participate in open source projects and found that the desire to learn is one of the major motivations. Economic motivation has been shown as a factor to attract individual developers to contribute to open source projects (Weiss, 2011). Therefore, we are interested in investigating what characteristics of a bounty issue are more attractive to hunters and more likely to be addressed by hunters who committed code to the project for the first time (i.e., new hunters), such as the value of bounties or the popularity of projects. We investigate the following research question:

> *RQ5: What are the characteristics of the bounty issues that are more likely to be addressed by hunters who committed code to the project for the first time (i.e., new hunters)?*

The following two studies are the most related work to our paper. Zhou *et al.* (2020b) studied the relationship between the issue-addressing likelihood and the bounty-related factors (e.g., the total bounty value of a bounty issue report) and provided insights on the usage of bounty, e.g., 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. Kanda *et al.* (2017) performed an explorative study on bounty issues on Bountysource and 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. These two studies focusing on studying the relationship between bounties (e.g., bounty value) and the issue-addressing process. However, little is known about the human aspects (i.e., hunters and backers) in the bounty issue addressing process,

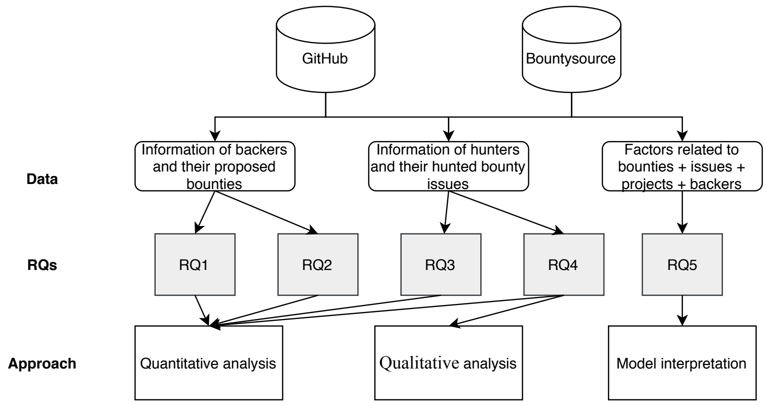

Fig. 2: The overview of our research design.

especially in the open source projects. Therefore we focus on investigating backers and hunters in this study.

3 Study Overview

In this section, we describe the design of our study to answer these research questions. We also discuss the limitations of our design.

3.1 Research Design

Figure 2 presents an overview of our research design. To answer our RQs, we apply an exploratory mixed-method research design to explore phenomena and seek explanations. Our study is exploratory in nature and can be the first step in providing a landscape on two important roles (i.e., backers and hunters) in the bounty issue processing. We perform quantitative analysis in RQ1, RQ2, and RQ3. In RQ4, we perform both quantitative and qualitative analysis. In RQ5, we construct classification models to study the relationship between the potential factors and the likelihood of an issue to attract hunters who committed code to the project for the first time. We elaborate on our design below.

3.1.1 Data Collection

Bountysource allows users to propose bounties on issue reports on various platforms (e.g., GitHub and Bugzilla). In this study, we focus on bounties are proposed for GitHub issue reports, since the majority (77.3%) of the bounties that are proposed on Bountysource are for GitHub issue reports and GitHub allows us to get various types of metadata for issue reports which are not available in other platforms. The rest of the bounties are for issue reports on other platforms, e.g., Launchpad (13.3%) and Bugzilla (5.3%).

To conduct our study, we collect bounties and their associated issues of GitHub projects from Bountysource. We also collect details of the contributors of these collected bounties (including backers and hunters if applicable) and their bounty-related activities from Bountysource. For example, we collect the type of backers (i.e., individual and corporate) from Bountysource. Note that users on Bountysource usually provide their GitHub account in their profile so that we can map the users on Bountysource back to GitHub. All the information about the bounties is stored on Bountysource, and all the details about issue reports and their corresponding projects are stored on GitHub. We retrieved the bounty and issue information from Bountysource automatically using its official web API[4]. The bounty information includes the backer(s) who proposed the bounty, the value of the proposed bounty, and the hunter who addressed the issue report if applicable (some bounty issues have not yet been addressed). Note that we focus on issues of GitHub projects in this study. Therefore all hunters are GitHub users and we collected their GitHub account information from their profile. In addition, we collected basic information about the issue reports such as their ID and URL that are associated with the collected bounties. Once we collect the bounty information from Bountysource, we also need to collect detailed information of the issues and their associated projects which are not available on Bountysource. We retrieve the details of the issue reports on GitHub using the collected ID and URL of the issue report from Bountysource. To collect the details of issue reports, we used the official web API provided by GitHub[5]. We collected the description, the creation date of the issue report, the comments that developers posted under the report, and the labels of the issue report. In addition, we collected the bounty information of the corresponding project, such as the total number of bounty issue reports of a project. We also stored the snapshot of the metadata (e.g., forks, watchers, and issues) of all the associated projects. Finally, we retrieved the details of the hunters' activities in the GitHub project for which the bounty is proposed by using their GitHub user ID collected from Bountysource, e.g., commits that were pulled by each hunter in the associated project. The information of backers and hunters are used to perform analysis for RQ1 to RQ4. We also calculate various metrics to construct models for RQ5 (see more details in Section 3.2.3).

Table 1 shows an overview of the data that we collected. In total, we collected 5,589 bounties that were offered by 2,995 backers, with a total value of \$412,478 across 1,210 GitHub projects. There are 2,265 claimed bounties (with a median of \$22.0 and mean of \$90.8) and 3,324 unclaimed bounties (with a median of \$15.0 and mean of \$62.3), respectively. The amount of claimed bounties is statistically significantly larger than that of unclaimed bounties (p-value < 0.05). We also collected the corresponding issue reports which were created between Oct 19, 2012 and Oct. 22, 2018. There are 882 hunters who successfully claimed 1,448 bounty issues. The median and mean

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

Table 1: An overview of the collected data.

Total number of bounties	5,589
Total number of claimed bounty issues	1,448
Total number of claimed bounties	2,265
Total number of unclaimed bounties	3,324
Total bounty value	$412,478
Total number of bounty hunters	882
Total number of bounty backers	2,955
Total number of issue reports	3,579
Total number of issue reports with multiple bounties	817
Total number of projects	1,210
Median/Mean bounty proposing time (days)	188/ 336.3
Median/Mean bounty resolving time (days)	97.4/246.7

time for hunters to resolve bounty issues since the creation of bounties are 97.4 and 246.7 days, respectively. The median and mean time for backers to propose bounties since the creation of issue reports are 188 and 336.3 days, respectively.

Our dataset is made publicly available online[6].

3.1.2 Approach for RQ1

In RQ1, we wish to investigate the differences between individual and corporate backers by conducting a quantitative analysis. First, we compare these types of backers in two dimensions: the amount and frequency of the bounties that are offered by individual and corporate backers. Note that 346 bounties were proposed by anonymous backers that are not allowed us to link back to GitHub, so we exclude them. We visualize the results in beanplots. We also apply the Wilcoxon rank-sum test to test whether the differences between these two types of backers are statistically significant or not in terms of the above-mentioned dimensions. We choose the Wilcoxon rank-sum test since it is a non-parametric test that does not have an assumption on the distribution of the data. We compute the Cliff's delta d to measure the magnitude of the differences between these two types of backers. The magnitude is assessed using the thresholds provided by Romano et al. (Romano *et al.*, 2006) (i.e., $|d|$ <0.147 "negligible", $|d|$ <0.33 "small", $|d|$ <0.474 "medium", otherwise "large").

3.2 Approach for RQ2

Next, we investigate the differences between the two types of backers in terms of their preferences of bounty proposal on bug reports versus feature requests. More specifically, for each type of backer, we calculate the amount and the frequency of bounties that are for feature requests and bug reports, respectively.

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

To identify these two types of issue reports, we first examine the labels of each issue report if applicable. We consider an issue report that is tagged with "bug" as a bug report and an issue report that is tagged with "feature" or "enhancement" as a feature request. By using the above-mentioned label-based approach, we identify 344 bug reports and 1,222 feature requests, respectively. For the rest of the bounty issue reports, we apply a keyword-based heuristic approach on the title and body of each issue report to identify issue types. We have the following three heuristic rules:

- **H1:** If the title of an issue report contains the keywords: "request", "feature", "expect", "propose" "wish" "add", "support", "implement", "need", "improve", "optimize", "able", "allow"; then, we consider the issue report as a feature request.
- **H2:** If the title of an issues report contains keywords: "miss", "do not", "error", "fail", "bug", "fix", "don't","unavailable", "error", "issue", "can't", "cant", "exception", "crash", "could not", "cannot", "warn", "should not", "should be"; then, we consider the issue report as a bug report.
- **H3:** If the title of an issue report does not contain any keyword listed in H1 and H2, we consider it as a feature request.

We empirically select the keywords used in **H1** and **H2** by manually examining 100 issue reports. We also empirically set the priority of our heuristics as **H1** > **H2** > **H3**. In other words, if the title of an issue report meets both H1 and H2, we consider it as a feature request. We end up with 470 bug reports and 3,108 feature requests. To verify the accuracy of our heuristics, the first two authors (A1 and A2) manually verify a statistically representative sample (100) with a 95% confidence level and a 10% confidence interval. A1 and A2 independently label the sampled issue reports and take notes for any uncertainty. After finishing the independent labeling, A1 and A2 discuss the labeling results to resolve any disagreements until a consensus is reached. We find that our heuristics have a high accuracy of 90%.

3.2.1 Approach for RQ3

As mentioned in Section 2, we can identify a hunter if he/she claimed that he/she started on a bounty issue or he/she addressed a bounty issue and claimed for the bounty. In this RQ, we first focus on the bounty hunters who have completed at least one bounty issue and filter out the rest (i.e., the ones who did not have any income). We end up with 746 (out of 821) bounty hunters. We calculate two metrics for hunters: 1) the number of bounty issues that were completed by a hunter; and 2) the income that each made from his/her completed bounty issues.

3.2.2 Approach for RQ4

We categorize the hunters into two groups based on their roles in a project and investigate their characteristics. We identify the role of a hunter in a project by checking the commit history of the hunter in the project. Specifically, we divide hunters into two groups: hunters who committed code to the project for the first time (i.e., new hunters) and hunters who committed code to the project before. Note that a user could be a hunter who never commits code to one project but a hunter who commits code to another project before. For each project, we first investigate the proportion of two types of hunters and their distribution. Furthermore, we investigate the behaviors between two types of hunters. For example, who are more likely to be regular hunters and are more likely to address high-value bounty issues (i.e., big bounty hunters). To do so, we examine the proportion of hunters who committed code to the project before over the entire hunters (including both types of hunters) against the total and the mean value of bounty issues that were addressed by these hunters. For example, if the proportion of hunters who committed code to the project before increases as the number of their addressed bounty issues increases, it may suggest that hunters who committed code to the project before are more likely to be regular hunters.

We observe 75 cases in which hunters started to work on a bounty issue but eventually stopped. To understand the reason behind such cases, we perform a qualitative analysis. Hunters sometimes indicate the reasons why they stop in the comments. The first and second authors manually checked the comments that were left by hunters. Any disagreement is discussed until consensus is reached.

3.2.3 Approach for RQ5

Table 2: The description and rationale for the studied factors. The factors that are calculated at the time when the bounty is proposed are marked with '*'.

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 attract hunters who never commit code to the project 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 likelihood of attracting hunters who never commit to the project to address it.
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.	

I_type*	Binary type: bug report or issue request.	Hunters who never commit to the project may be more interested in addressing one type of issue than another.
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 likelihood of attracting hunters who never commit to the project to address it.
I_B_total_value	The total bounty value of the issue report.	An issue report with higher bounty value and more bounties may attract more hunters who never commit to the project.
I_B_cnt	The number of bounties that a bounty issue report has.	
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 likelihood of attracting hunters who never commit to the project to address the issue report.
Project basic		
P_branch_cnt	The total number of branches of a project.	These nine factors reflect the popularity and maturity of the project. A different level of popularity and maturity may have a different association with the likelihood of attracting hunters who never commit to the project to address their issues.
P_issue_cnt	The total number of issues of a project.	
P_pull_request_cnt	The total number of pull requests of a project.	
P_commit_cnt	The total number of commits of a project.	
P_contributor_cnt*	The total number of contributors of a project.	
P_fork_cnt	The total number of forks of a project.	
P_watcher_cnt	The total number of watchers of a project.	
P_star_cnt	The total number of stars of a project.	
P_age	The age of a project (in days).	
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 likelihood of attracting hunters who never commit to the project. For example, a project with more bounty issues may be more likely to attract hunters who never commit to the project to address them.
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.	
Backer experience		
Backer_exp_B_median/sum/max_value*	The median/sum/max value of the bounties that 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 hunters who never commit to the project.
Backer_exp_B_median/sum/max_cnt*	The median/sum/max number of bounties that the backers of this bounty have ever proposed in the past.	
Backer_has_no-commit*	Whether an issue has at least one backer that has never made any commit to a project before proposing a bounty.	An issue report that has a corporate or a backer that has never made any commit is probably more likely to attract hunters who never commit to the project to address it.

Backer_has_corp	Whether an issue has at least one corporate backer.

To understand the characteristics of issues that are more likely to attract hunters who committed code to the project to address a bounty issue for the first time, we construct logistic regression models to study the relationship between the potential factors and the likelihood of an issue to attract new hunters. Zhou *et al.* (2020b) studied the association between various factors and the likelihood of a bounty issue being addressed. We reuse these factors since they are also potentially related to the likelihood of an issue report being addressed by hunters. We include additional factors that are related to the popularity and maturity of a project, issue type, and backer role (e.g. individual or corporation backer). In short, we study 35 factors from the bounty issue reports and their corresponding project, along the following five dimensions:

1. **Issue report:** Nine factors that estimate the basic information of an issue report, such as the supportive information of the issue report (e.g., text, code, link, and image) and the popularity (e.g., number of comments and participants).
2. **Issue report bounty:** Four factors that describe the bounty usage within a bounty issue report, such as the total value of bounties on an issue report and the time of the creation of bounties.
3. **Project:** Nine factors that reflect the basic information of a project, such as the popularity (e.g., number of watches and stars) and the activity level of a project (e.g., the number of commits, forks, and pull requests).
4. **Project bounty:** Five factors that reflect the bounty usage within a project, such as the value of total proposed bounties and number of bounties in the project.
5. **Backer experience:** Eight factors that capture the bounty experience of the backers of a bounty issue report, such as whether the backer is individual or corporate backer.

Table 2 summarizes the descriptions and rationales for the studied factors. For each bounty issue, we will calculate the value of each studied factor. We collect the information of each issue report, its associated bounties, backers, and projects from Bountysource and GitHub as discussed in Section 3.1.1. For instance, to calculate the factors of Project dimension, we will need the information of the associated project, such as the number of comments, stars, and forks, etc. Note that the factors which are marked with '*' are time-dependent factors that are calculated at the time when the bounty is proposed. For example, P_B_paid_cnt* is the total number of paid bounty issue reports of a project when the first bounty of the issue report was proposed.

After collecting the studied factors, we construct a classification model to understand the association between the studied factors and the likelihood of a bounty issue being addressed by new hunters. Note that once a new hunter has addressed his/her issue, he/she becomes a hunter who commits code to the project before. We ended up with 1,115 and 300 issues that were addressed by hunters who committed code to the project before and hunters who never commit code to the project, respectively. Below, we elaborate on each step of the model construction, model validation, and variable importance analysis.

Correlation & redundant analysis. Before constructing the model, we first follow prior studies (Wang *et al.*, 2018; Rajbahadur *et al.*, 2019) to remove correlated and redundant factors since highly correlated factors can cause multicollinearity problems in our model. 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). For each of the highly-correlated factors, we keep one factor in our model. We then performed a redundancy analysis to remove redundant factors (Wang *et al.*, 2018). We ended up with one factor in the project bounty dimension, three factors in project basic, six factors in the issue report basic dimension, four factors in the issue report bounty dimension, and four factors in the backer experience dimension (shown in Table 6).

Model construction. We built a logistic regression model which enables us to examine the effect of one or more variables on a response variable when controlling for other variables. We use the R package **rms**[7] as the implementation of our logistic regression model. Similar to previous work (Wang *et al.*, 2018; Zhou *et al.*, 2020b), we also add non-linear terms in the model to capture more complex relationships in the data by employing restricted cubic splines. Previous study shows that hyperparameter tuning impacts the interpretation of a classifier (Tantithamthavorn *et al.*, 2018). Therefore, we tune the hyperparameter alpha (regularization strength) for our constructed classifier using random search (Bergstra and Bengio, 2012) with the *caret* R package (Kuhn *et al.*, 2008).

Model validation. We use AUC and bootstrapping to assess the explanatory power of the built logistic regression model by following prior studies (Wang *et al.*, 2018; Zhou *et al.*, 2020b). AUC is the area under the Receiver Operating Characteristic Curve (i.e., *ROC*), which is often used as a measure for the quality of classification models. A random classifier has an AUC of 0.5, while the AUC for a perfect classifier is equal to 1. In practice, most of the classification models have an AUC between 0.5 and 1. In general, an AUC of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding (Akobeng, 2007; Mandrekar, 2010). To ensure our models are not overfitted, we calculate their optimism values using a bootstrap-derived approach by following previous studies (Wang *et al.*, 2018; Zhou *et al.*, 2020b), since prior study shows that bootstrap validation yields the best balance between the bias and variance over cross-validation (Tantithamthavorn *et al.*, 2016). 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.

[7] https://cran.r-project.org/web/packages/rms/rms.pdf

Variable importance analysis. 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. We consider factors of which the χ^2-test has a p-value of less than 0.05 as statistically significant.

3.3 Limitation of our design

The first limitation is that our findings only show the correlation between the studied factors and the likelihood of a bounty issue being addressed by a hunter who commits code to the project for the first time, but not causation. A reasonable way to study the causation is performing user interviews and surveys. In this study, we did not conduct surveys and interviews with developers. We made this decision due to the limitation of the public data that is available and ethical considerations. Money and bounty is a sensitive topic that is often framed in the context of larger discussions on fairness, stress, or even burnout, and unequal distribution of bounties on few participants or projects (Matt Asay, 2020; Robert Lemos, 2019). Rather than adding stress to participants, we analyze the public artifacts, e.g., the messages left by hunters and backers, and discussion between developers on forums and blogs.

4 Results of the Research Questions

4.1 Results of RQ1

Table 3: Basic descriptive summary of individual and corporate backers and their bounties.

	Individual	**Corporate**	**Total**
Total amount of bounties	208,811 (50.6%)	191,300 (46.4%)	412,478
Total count of bounties	4,282 (81.8%)	961 (17.2%)	5,589
Number of backers	2,484 (95.2%)	125 (4.8%)	2,609

95.2% of the backers are individual backers and they supported 81.8% of all offered bounties (4,282 out of 5,243). Although corporate backers only represent a small portion of the population (4.8%), they contributed almost half (46.4%) of the total bounty amount. Table 3 summarizes the basic descriptive statistics for individual and corporate backers. Almost all backers (95.2%) are individual backers and they proposed 81.8% of the studied bounties across 85.7% (1,037/1,210) of the

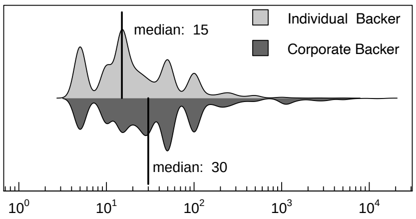

Fig. 3: Distributions of the bounty amount of a single bounty that is proposed by individual and corporate backers.

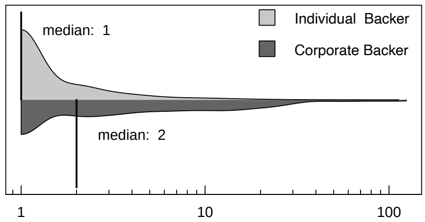

Fig. 4: Distributions of the number of bounties proposed by individual and corporate backers.

analyzed projects. This observation highlights the importance of individual backers in terms of their broad impact. We observe that although only 4.8% of backers are corporate backers, they offered almost half (i.e., 46.4%) of the total bounty amount. For instance, IBM is the biggest backer on Bountysource and it proposed bounties with a total value of $112,250, which is 27.2% of the total amount of the studied bounties.

Overall, the value of bounties that are proposed by backers is small (median bounty value is $20). Corporate backers tend to propose larger bounties (median bounty value of $30) with a higher frequency (median frequency of 2) than individual backers (median bounty value of 15 and median frequency of 1). In terms of the amount of a single bounty that is proposed by individual and corporate backers, as Figure 3 shows, corporate backers offer more money in a single bounty ($30) compared to individual backers ($15). The Wilcoxon rank sum-test shows that the difference between these two distributions is statistically significant (p-value < 0.05) with a small effect size ($d = 0.25$). The median total amount of the bounties that are proposed by individual backers is $25, while the median total amount of the bounties that are proposed by corporate backers is $120. The statistical test shows that the differences are statistically significant with a small effect size ($d = 0.25$). Figure 4 shows the bounty frequency of individual and corporate backers. The median bounty frequency of corporate backers is 2, which is twice as the median value of individual backers (i.e., 1). The statistical test shows that the difference between the two distributions is statistically significant (p-value < 0.05) with a medium effect size ($d = 0.46$), suggesting that corporate backers propose bounties more frequently than individual backers. This may also explain that, although only 4.8% of the backers are corporate backers, they offered almost half (i.e., 46.4%) of the total amount of the bounties. One typical instance is IBM. IBM proposed 51 bounties on the issues of 20 GitHub projects with a median value of $1,150, and a maximum value of $13,200. Compared with bounties provided by corporations in security bug programs, the average value of bounties that are proposed for issues of GitHub projects is much smaller. As HackerOne reports, the average value of bounties paid for critical vulnerabilities across all industries on the HackerOne platform is $2,041 in 2017 (HackerOne, 2018).

We also examine the correlation between the total amount of bounties that an issue report received and the popularity of its associated project. We use the number of issues and watchers of a project as the proxies of the popularity of a project. The spearman correlation between the total amount of bounties that an issue report received and the number of issues and watchers of its associated project are 0.57 and 0.46, respectively, indicating **a moderate positive correlation (Moore and Kirkland, 2007) between the amount of bounties of an issue report and the popularity of its associated project.**

We observe that the time for corporate backers to propose bounties (median/mean time is 25.4 days/138.8days) is shorter than that of individual backers (median/mean time is 112.3 days/267.3 days). The statistical test shows that the difference is statistically significant (p-value < 0.05) with a small effect size ($d = 0.25$).

95.2% of backers are individual backers and they supported 81.8% of the offered bounties (4,282 out of 5,243). Although corporate backers only represent a small portion of the population (4.8%), they contributed almost half (46.4%) of the total bounty amount. Overall, the value of bounties that are proposed by backers is small (median bounty value is $20). Corporate backers tend to propose bounties more frequently (median frequency of 2) and with a larger amount (median bounty value of $25) than that of individual backers (median frequency of 1 and median bounty value of $15).

4.2 Results of RQ2

Table 4: The comparison between individual (Ind) and corporate (Corp) backers in terms their bounties posted on feature requests and bug reports.

	#Issues		**#Bounties**		**Total $Bounties**	
	Ind	**Corp**	**Ind**	**Corp**	**Ind**	**Corp**
Feature Request	2,454 (87.0%)	654 (86.4%)	4,046 (88.1%)	875 (87.7%)	$206,690 (91.8%)	$178,754 (95.4%)
Bug Report	367 (13.0%)	103 (13.6%)	544 (11.9%)	123 (12.3%)	$18,436 (8.2%)	$8,548 (4.6%)
Total Issue	2,821	757	4,590	998	$225,126	$187,302

Both individual and corporate backers tend to propose bounties on addressing feature requests rather than bug reports. Table 4 shows the number of issues, number of bounties, and their total bounty amount contributed by individual and corporate backers. Individual and corporate backers proposed 91.8% and 95.4% of their total bounty amount on addressing feature requests, respectively, which suggests that backers are more likely to offer their money in implementing new features rather than fixing bugs. For instance, IBM spent 99% ($110,850 out of $112,250) of its total bounty amount on feature requests. One possible reason is that such corporate backers are using or relying on such projects and they wish to have new features to benefit their business. For example, IBM proposed bounties on 14 feature requests of a single project "libvpx: VP8/VP9 Codec SDK"[8], which is an implementation of the VP8/VP9 video format that is widely used (e.g., YouTube). IBM sponsored new features of libvpx since IBM's new released hardware with the new Power9 processor benefits from the implementation of such new features[9].

Our findings suggest that Stakeholders of open source projects may not expect a large amount of bounties to support unresolved issues since the value of bounties is usually small. Especially for bug reports, the likelihood of receiving bounties is low, both individual and corporate backers are mostly only interested in proposing bounties to support new features.

> *Both individual and corporate backers prefer to support implementing new features rather than fixing bugs.*

4.3 Results of RQ3

85.0% of the bounty hunters addressed less than 3 bounty issues. Only 4.8% of the hunters addressed 5 or more bounty issues. Figure 5 presents the distribution of the hunters that have addressed different numbers of bounty issues. The distribution is long-tailed and left-skewed. 85.0% of the studied hunters have only addressed one or two bounty issues, suggesting that most of the bounty hunters are not sustainable in hunting bounty issues. Interestingly, we observe that 4.8% of the bounty hunters addressed more than 5 bounty issues. One possible explanation is that such hunters are the developers of a project and they need to maintain the project, e.g., addressing issues. For instance, a user addressed 23 bounties and earned $2,457 in total. All his addressed bounty issues are from the *OpenRA* project and the user is one of the developers of this project[10]. Another possible explanation is that some hunters are professional hunters and they address issues for money. For instance, a user addressed 9 bounty issues that are across 9 different projects and earned $1,240 in total. However, in general, most of the bounty hunters are one-time hunters. Our finding is aligned with the finding of a prior study

[8] https://github.com/webmproject/libvpx/
[9] https://medium.com/@luc.trudeau/video-compression-bounty-hunters-c8edf43d440
[10] https://github.com/OpenRA

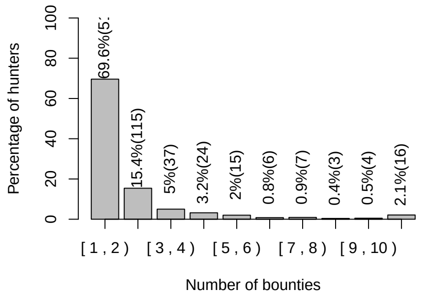

Fig. 5: The distribution of hunters that have addressed different numbers of bounty issues.

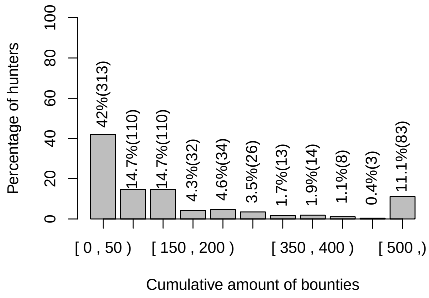

Fig. 6: The distribution of hunters that earn different total amount of income from addressing bounty issues.

by Hata *et al.* (2017) that most hunters are not very active (i.e., they have only a few contributions).

Most bounty hunters (56.7%) make less than $100, and only 2.7% of the hunters make more than $2,000. Figure 6 presents the distribution of the cumulative income for each bounty hunter. We observe that the distribution is long-tailed. 56.7% (423 out of 746) of the hunters have a cumulative income of less than $100. In particular, the income of 42% of the hunters is less than $50. In other words, most hunters only earn a small amount of income from addressing bounty issues. One possible explanation is that the value of a single bounty issue is small, with a median value of $30 and an average value of $142.2. Only 5.5% of the bounties have a value higher than $1,000. We observe that only 2.7% (20 out of 746) of the hunters have earned more than $2,000 from addressing bounty issues. **The maximum cumulative income among the studied hunters by addressing bounty issues is $17,300.** The user addressed four bounty issues (the value of the bounties are $3,000, $3,300, $5,500, and $5,500) that were proposed by IBM for the *OpenBLAS* project spanning over 10 months. Our finding suggests that hunters may not expect to earn a large amount of money from addressing bounty issues. This finding aligns with an online report for security bug bounty hunters, which shows that only very few hunters (top hunters) can earn a good amount of money by hunting security bugs (Matt Asay, 2020; Robert Lemos, 2019). Only seven hackers have now earned more than $1 million in bug bounties so far in their career, with another 13 surpassing $500,000 in lifetime earnings on HackerOne over millions of hackers.

Table 5: The comparison between hunters who committed code to the project for the first time and hunters who committed code to the project before in terms of their total number of bounties, total value of bounties, and median/mean value of each bounty.

	Hunter who commits code to the project for the first time	Hunter who commits code to the project before
Total number of hunters	502 (67.3%)	267 (32.7%)
Total number of bounties	1,039 (71.8%)	409 (28.2%)
Total value of bounties	$161,838.66 (78.6%)	$44,063.25 (21.4%)
Median/mean bounty value	30/137.8	33/160.8

In general, most bounty hunters do not address a large number of bounty issues over time. 85.0% of the bounty hunters addressed less than 3 bounty issues. Only 4.8% of the hunters addressed 5 or more bounty issues. Most bounty hunters make less than $100, and only 2.7% of the hunters is larger than $2,000. Our findings suggest that hunters may not expect to earn a large amount of money from addressing bounty issues.

4.4 Results of RQ4

67.3% (502 out of 746) of the studied hunters are hunters who committed code to the project before, who addressed 71.8% (1039 out of 1,448) of bounty issues. Table 5 compares hunters who committed code to the project for the first time (i.e., new hunters) and hunters who committed code to the project before in terms of the total number of bounties, the total value of their addressed bounty, and the median and mean bounty value of the bounty issues across all studied projects. We observe that most of the hunters committed code to the project before (67.3%), and such hunters addressed the majority of the bounty issues. The finding may not be surprising since it can be explained by the fact that hunters who committed code to the project before are usually more familiar with the project than new hunters. Interestingly, we also observe a significant portion of new hunters (32.7%) addressed 28.2% (409 out of 1,448) of the bounty issues. In addition, we observe that 46.7% (215 out of 481) of the projects had such hunters addressed at least one bounty issue. For instance, Bountysource proposed bounties on 70 issues for various purposes (e.g., fixing search box bug and improving UI) as of Oct. 22, 2018. Among the fixed ones, 91.6% (22 out of 24) of them were addressed by hunters who committed code to the project for the first time. Our finding shows that new hunters are still important contributors in bounty hunting and help sustain open source projects.

Hunters who committed code to the project before are more likely to be regular hunters for a project compared to new hunters. Figure 7 presents the proportion of hunters who committed code to the project

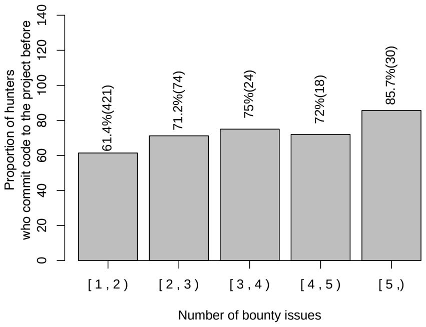

Fig. 7: The proportion (number) of hunters who committed code to the project before against the number of their addressed bounty issues.

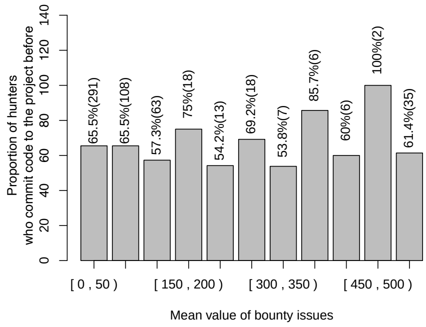

Fig. 8: The proportion (number) of hunters who committed code to the project before against the mean value of their addressed bounty issues.

before against the number of bounty issues that were addressed by all hunters. When looking at all the hunters who have addressed two or more bounty issues, 74.3% of the hunters are hunters who committed code to the project before, which is much higher than the proportion of the hunters who have only addressed one bounty issue (61.4%). The proportion of hunters who committed code to the project before is even higher (85.7%) when considering the hunters who addressed more than 5 bounty issues in a project. In other words, hunters who committed code to the project before are more likely to be regular hunters for a project compared to hunters who committed code to the project for the first time. This is intuitive since hunters who committed code to the project before are more likely to be familiar with the projects and have less overhead to start working on a bounty issue compared to new hunters. Such hunters who committed code to the project before probably have a stronger motivation to maintain the projects even without considering bounties. Figure 8 shows the proportion of hunters who committed code to the project before against the mean value of bounty issues that were addressed by these hunters. We do not observe a clear trend of the proportion of hunters who committed code to the project before against the mean value of bounty issues. One possible explanation is that some issues are difficult even for hunters who committed code to the project before and may require domain-specific expertise for resolving them. Furthermore, we also investigate the experience of hunters who committed code to the project before in the project in terms of 1) the number of commits that are made by these hunters before their first hunting and 2) the interval time between these hunters' first commit and first hunting. Hunters who committed code to the project before made a median/mean of

29/350.7 commits in the associated project before their first hunting and the median/mean age before their first hunting is 190/431.5 days.

We find 7 cases in which hunters left messages either on Bountysource or GitHub that explicitly mentioned the reason for stopping working on the issue reports. **Long learning curve or lack of time (6 cases).** We observe 6 cases in which the unfamiliarity of a project (e.g., design and architecture) and its development environment are barriers for hunters to address bounty issues successfully. For example, in a bounty issue of LuaJIT[11], one hunter mentioned "I have started looking into this late last year but after some research I don't feel qualified enough to solve this at the moment. I don't have the time to go through the learning curve to get it done. Apologies." We observe 2 cases in which the hunters eventually stopped working on the issue due to lack of time. For example, one hunter mentioned "My time is limited so no gaurentee I can make it happen." Our observation is similar to a prior study by Lee et al. (Lee *et al.*, 2017), which studied barriers that hinder open source developers from becoming long-time developers from one-time developers. Lee et al. found that entry difficulties and lack of time are the major barriers. We also found similar challenges in security bug bounty programs from online blogs and news. For instance, "First make sure you know what you are doing, as hacking has a very very steep learning curve and it is overwhelming in the beginning. Before making the switch to a full-time bug hunting job, it's important to have at least half a year or a year of experience as a part-time bug bounty hunter." said by a professional bounty hunter (Mirko Zorz, 2020).

Vague specification (1 cases). Hunters need to have a very clear specification for an issue before they can start to work on it. A vague specification could be a potential barrier that leads hunters to stop addressing bounty issues. For example, we observe one hunter mentioned: "Specifications are too vague. I need to know exactly what functionality to code and how the interface must accommodate this." when the hunter stopped the progress on a feature request (i.e., repeat every X days (monthlies)) of habitica[12]. In fact, the communities were still brainstorming on the feature request when the hunter asked the specification under the issue report[13].

Most of the regular hunters and big hunters of a project are hunters who committed code to the project before. 67.3% (502 out of 746) of the studied hunters are hunters who committed code to the project before, who addressed 71.8% (1,039 out of 1,448) of bounty issues.

Table 6: The results of the model analysis with a 0.1 of R^2. The **NL** indicates the non-linear term, the **D.F.** indicates the degree of freedom, the **Coef** indicates the coefficients, and **S.E.** indicates the standard error associated with the coefficients. Note that if the **D.F.** of a factor is larger than 1, the **Coef** and **S.E.** are presented from lowest order to highest order. The factors are ranked by their importance (i.e., Wald χ^2). P-value of the χ^2 test: '*' < 0.05.

Factor		Overall	NL	Coef	S.E.
I_B_has_label	D.F.	1		-0.53	0.14
	χ^2	29.5*			
P_B_I_cnt	D.F.	3	2	0.87	0.22
	χ^2	26.7*	3.5		
P_age	D.F.	4	3	0.0003, -0.0010,	0.0004, 0.0015,
	χ^2	17.5*	7.4	-0.0007, 0.0046	0.0024, 0.0044
P_branch_cnt	D.F.	4	3	-0.0149, 0.0078,	0.0071, 0.0178,
	χ^2	16.7*	10.7*	0.0100, 0.0202	0.0210, 0.0364
P_issue_cnt	D.F.	4	3	-0.0001, 0.0000,	0.0001, 0.0004,
	χ^2	11.7*	10.7*	0.0001, 0.0002	0.0005, 0.0009
I_cmnt_cnt	D.F.	3	2	-0.0733, 0.0747,	0.0326, 0.1791,
	χ^2	11.3*	11.3*	0.1114	0.2439
Backer_has_corporate	D.F.	1		-0.3442	0.2188
	χ^2	10.0*			
I_B_total_value	D.F.	4	3	0.0014, -0.0009,	0.0010, 0.0029,
	χ^2	5.8	3.9	-0.0011, -0.0015	0.0033, 0.0041
Backer_has_nocommit	D.F.	1		0.3058	0.1488
	χ^2	5.4*			
Backer_exp_B_max_value	D.F.	3	2	0.0000, 0.0001,	0.0001, 0.0005,
	χ^2	3.4	0.1	0.0001	0.0005
I_type	D.F.	1		0.2980	0.1765
	χ^2	2.4			
I_code_proportion	D.F.	4	3	0.3344, -0.6896,	1.0803, 10.9599,
	χ^2	1.9	0.3	-0.6839, -0.3250	11.2505, 15.9133
I_B_days_before_bounty	D.F.	4	2	-0.0004, 0.0006,	0.0013, 0.0110,
	χ^2	1.6	1.5	0.0006, 0.0006	0.0111, 0.0133
I_B_cnt	D.F.	1		-0.0161	0.0357
	χ^2	1.6			
I_content_len	D.F.	1		0.0000	0.0000
	χ^2	1.2			
I_link_cnt	D.F.	1		-0.0008	0.0194
	χ^2	0.1			
I_img_cnt	D.F.	1		0.0438	0.0800
	χ^2	0.06			

4.5 Results of RQ5

Our models capture the relationship between the explanatory variables and the response variable well, and have a reliable performance. Our models achieve a median AUC value of 0.7, which indicates that

[11] https://www.bountysource.com/issues/25924774-enable-implement-ppc64-le-linux-lj_gc64-interpreter-and-jit

[12] https://www.bountysource.com/issues/5413688-repeat-every-x-days-monthlies

[13] https://github.com/HabitRPG/habitica/issues/4173

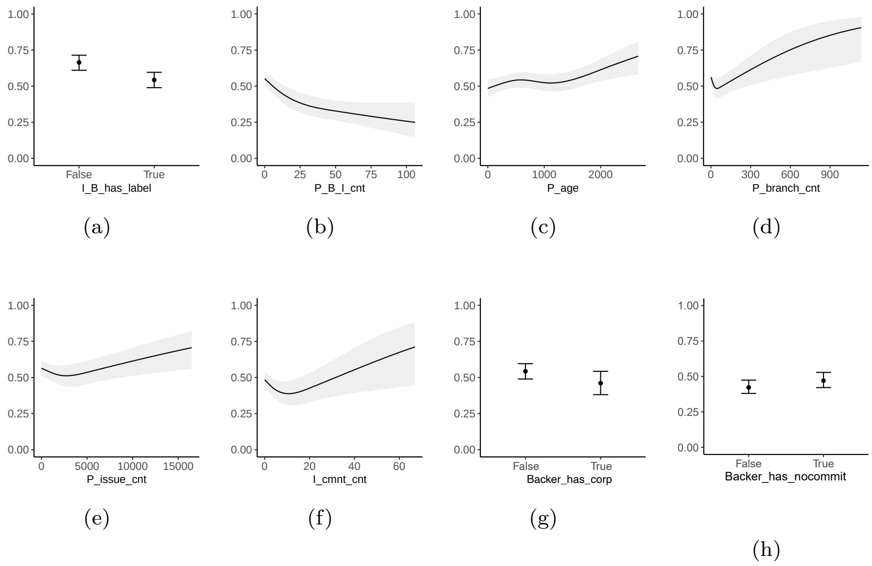

Fig. 9: The relationships between the likelihood of a bounty issue being addressed by a hunter who commits code to the project for the first time and the eight factors that are statistically significant important.

our models have a good capability to explain the dataset, and the low median optimism values (0.03) indicate that our models do not overfit the dataset.

The value of a bounty issue is not a statistically significant factor that impacts the likelihood of a bounty issue being addressed by hunters who committed code to the project for the first time to address the issue (i.e., new hunters). Table 6 presents the results of our model analysis. Eight factors (i.e., factors with '*') are statistically significant important in our constructed model. Interestingly, we observe that the value of a bounty issue does not have a statistically significant association with the likelihood of a bounty issue being addressed by a new hunter. One possible explanation is that the bounty amount is usually small therefore are not very attractive for new hunters. We observe that the median values of the bounties that were addressed by hunters who commits code to the project before and new hunters are the same (i.e., $35). Another possible explanation is that hunters are not driven by financial incentives to address bounty issues; instead, they may be driven by their interests (Zhou *et al.*, 2020b; Krishnamurthy *et al.*, 2014).

Bounty issues with a bounty label are less likely to be addressed by new hunters.. I_B_has_label is the most important factor in our constructed model. Figure 9 (a) shows the relationship between I_B_has_label and the likelihood of a bounty issue being addressed by a hunter who commits code to the project for the first time. Bounty issues with a bounty label are less likely to be addressed by a new hunter who never commits code to the project than those without a bounty label. One possible explanation is that bounty labels are only exposed within a project so that hunters who committed code to the project before are more likely to be aware of the existence of bounty issues and address them than new hunters.

Bounty issues of a project that have a lower frequency of bounty usage are more likely to be addressed by new hunters. The factor with the second-highest explanatory power is P_B_I_cnt, which indicates the number of bounty issues in a project before the creation of the new bounty issue. Figure 9 (b) presents the relationship between P_B_I_cnt and the likelihood of a bounty issue being addressed by new hunters. We observe a negative association between them, i.e., the more bounty issues that a project has, the lower the likelihood that a new bounty issue is addressed by new hunters. One possible explanation is similar to the above-mentioned reason, hunters who committed code to the project before are more likely to aware of the existence of bounty issues if bounties are used more frequently in a project and address them. Another possible explanation is that a higher frequency of bounty usage of a project may indicate more hunters are aware of the bounties in this project, which leads to stronger competition for bounty hunting. Due to the nature of hunters who committed code to the project before and new hunters, hunters who committed code to the project before have a higher chance to win such competitions and earn bounties. In other words, hunters who committed code to the project for the first time are more likely to address bounty issues in new projects regarding bounty usage. This observation is compliable with our finding in Section 4.1 that hunters who committed code to the project before are more likely to repeatedly hunt for bounties.

Bounty issues of a popular and aged project are more likely to be addressed by new hunters. From Table 6, we observe that P_age, P_branch_cnt, and P_issue_cnt are statistically significant important factors in our model. Figure 9 (c), (d), and (e) show the relationship of the likelihood of a bounty issue being addressed by new hunters against these three factors. Bounty issues of a project with more branches and issues are more likely to be addressed by new hunters. P_branch_cnt and P_issue_cnt reflect the popularity of a project. Therefore, the bounty issues of popular projects are more likely to be addressed by new hunters. Similarly, we observe a positive association between the age of the project that a bounty issue belongs to and the likelihood.

The type of the backer of an issue also has a statistically significant association with the likelihood of the issue being addressed by new hunters, as our constructed model suggests. Figure 9 (g) and (h) show their relationships. The figures indicate that a bounty issue that has backers who have never made any commit before proposing a bounty and individual backers is more likely to be addressed by new hunters. Such an observation probably indicates that new hunters tend to address bounty issues that are supported by individual backers rather than corporate backers. We cannot make any conclusion on this. One possible explanation is that issues that are supported by corporate backers tend to be more difficult than the ones that are supported by individuals, and require more expertise, which hinders new hunter to address them.

Interestingly, the value of a bounty issue is not a statistically significant factor that impacts the likelihood of a bounty issue being addressed by hunters who committed code to the project for the first time (i.e., new hunters). Bounty issues with a bounty label and projects that have more frequent usage of bounties are less likely to be addressed by new hunters. Bounty issues of a popular and aged project are more likely to be addressed by new hunters.

5 Discussion

In this section, we discuss the implications of our study and the direction for future research.

5.1 Implications of our findings

The stakeholders of open source projects should not expect a large amount of bounties from backers to support unresolved issues, especially for bug reports, unless their associated projects are very popular. Eghbal (2016) reported that open source projects still lack a reliable and sustainable source of funds when analysing the risks and challenges that are associated with maintaining open source projects. Our empirical results are compatible with their findings. In general, the value of the bounties is small (median bounty value of $20) and there is a positive correlation between the total amount of bounties and the popularity of its associated project. When looking at the issues that received more than $1,000 bounties, the median numbers of issues and watchers of their associated projects reach 1,672 and 219, respectively. However, the median numbers of issues and watchers for GitHub projects of different languages ranging from 3 to 25 and 1 to 5, respectively (Bissyandé *et al.*, 2013). Moreover, even the projects obtain bounties, the stakeholders may not expect backers to provide support on addressing bugs, since backers are almost only interested in offering bounties to implement new features and rarely support bug fixing.

In general, hunters should not expect to earn a large amount of money from addressing bounty issues. Prior reports show that the income inequality among hunters is remarkable (Matt Asay, 2020; Robert Lemos, 2019). Only very few hunters (top hunters) can earn a good amount of money by hunting security bugs (e.g., 20 surpassed $500,000 in lifetime earnings on HackerOne over millions of hackers), while the majority of hunters do not earn much money from security bounty programs. Similar to the observations in security bounty programs, in Section 4.2, we observe that the income of hunters is usually low, i.e., the income of 56.7% of the bounty hunters is no more than

$100, and only 2.7% of the hunters earned more than $2,000. Several possible reasons for such low income for hunters are: 1) bounties proposed by backers are usually small (see Section 4.1); 2) most of the hunters only addressed one issue (see Section 4.2) and it may be difficult for hunters to address bounty issues due to the barriers that new hunters face (see Section 4.2). Therefore, relying on bounty hunting to earn money may not be very cost effective for most hunters.

The stakeholders of open source projects should not expect much help from new hunters. As we discussed in Section 2.2, a number of studies found that a significant portion of projects rely on one or two developers to survive and it is challenging to attract new developers to make contributions in open source projects (Lee *et al.*, 2017; Canfora *et al.*, 2012). We suspect that bounty could be an extrinsic incentive to attract developers and improve the sustainability of open source projects. However, our findings indicate that bounties seem not to be very attractive for hunters who never commit code to the project (i.e., new hunters). The majority of the hunters are hunters who committed code to the project before and they addressed most of the bounty issues, as we found in Section 4.2. In Section 4.3, we also observe that bounty issues that have a bounty label or are in projects that have frequent usage of bounties are less likely to be addressed by new hunters. One possible justification may be that it is easier for hunters who committed code to the project before to work on and address a bounty issue. Therefore, when hunters who committed code to the project before see an issue with a bounty label, they may be more likely to prioritize fixing the issue. Our findings also show that the stakeholders of open source projects may not expect many new hunters to help address issues unless the projects are popular and aged. Prior studies found that some developers are driven by intrinsic incentives (e.g., enjoyment of volunteer and desire to learn) (Zhou *et al.*, 2020b; Lakhani and Wolf, 2003; Shah, 2006; Von Krogh *et al.*, 2012; Coelho *et al.*, 2018). One possible assumption is that popular and aged projects are likely to provide more opportunities for developers to gain such intrinsic achievement. We encourage future research to investigate this.

The stakeholders should consider providing certain support to shorten the learning curve for hunters who never commit code to the project to get familiar with new projects. In Section 4.2, we found that one barrier that hinders hunters to address bounty issues successfully is the long learning curve. Some hunters stated that they eventually stopped working on the issue due to the difficulty in getting familiar with the new projects (e.g., design and architecture) or the working environment within a reasonable time. Prior studies also reported similar barriers that new developers face when moving into new projects (Dagenais *et al.*, 2010; Coelho *et al.*, 2018; Steinmacher *et al.*, 2014; Lee *et al.*, 2017) and propose approaches to alleviate such barriers, e.g., recommending mentors (Ye and Kishida, 2003). Therefore, the stakeholders should consider providing certain support to shorten the learning curve for hunters who committed code to the project for the first time, e.g., providing well-maintained documentation for the project including information, such as the design and architecture of the project, and instructions of how to get it started quickly, and developing techniques to shorten the learning curve for hunters who committed code to the project for the first time such as developing chatting room for new developers to ask questions.

5.2 Future research directions

Our study is exploratory in nature and can be a first step in understanding the characteristics of two important roles (i.e., backers and hunters) in the bounty issue addressing process. Of course, there are many aspects that we do not explore in this study, as large parts of data are not available at the current stage. Still, our work reveals insights that can be starting points for many interesting future research directions.

First, future research is encouraged to explore **what kind of projects are more likely to attract corporate backers**. In Section 4.1, we observe that corporation backers tend to propose larger bounties than individual backers and there is a median correlation between the popularity of a project and its received bounty amount. It is not clear what kind of projects are more attractive to corporation backers. For instance, one direction is that developers of open source projects create innovation in a way that has a significant advantage over the manufacturer-centric innovation development systems (Von Hippel, 2007) so that corporations are motivated to participate in open source projects if innovations enhance profits (Harhoff *et al.*, 2003). Future research can take such factors (e.g., the innovation of a project) into consideration during the investigation.

Second, it is interesting to explore **the preference of backers on more fine-grained categories of issues and the reason behind this**. In Section 4.1, we explore backers' preference in a coarse-grained categorization (i.e., bug vs feature), and observe that both individual and corporate backers tend to propose bounties on addressing feature requests rather than bug reports. Prior studies show that issue reports could be classified into more fine-grained categories (e.g., bug, documentation, performance improvement, and build new systems) (Kochhar *et al.*, 2014). Furthermore, our observation shows that the majority of bounties are proposed on features. Different types of features may have different priorities (Dinnie Muslihat, 2019). Future research is encouraged to study what type of features (e.g., UI features vs back-end features) are more likely to attract bounties. In addition, prior studies report that companies use the silent update mechanism to boost security (Duebendorfer and Frei, 2009; Frei *et al.*, 2008). How do such factors impact the preference of backers on bugs vs feature? Future research is also encouraged to study the reason for backers' preference.

Third, an interesting direction is to perform a large-scale study to understand **what barriers that hinder hunters from successfully addressing a bounty issue**. Prior research studied barriers that are experienced by the one-time developers through user survey from 184 responses, and found various barriers that hinder such one-time developers to become long-term developers, e.g., entry difficulties and lack of time (Lee *et al.*, 2017). In Section 4.2, we perform a small-scale qualitative study to understand the reason why hunters started working on a bounty issue but eventually stopped. However, due to the limited size of the samples we have, our findings might not be generalized and reliable. We encourage future research to perform a large-scale study, e.g., performing a survey on hunters.

Fourth, another interesting direction is to explore **the negative effects of the bounty program on the open source community and how to mitigate them**. Financial incentives could have a negative impact on open source projects. Zhou *et al.* (2016) observed that commercial involvement can increase the inflow of paid developers in an open source project, but may potentially decrease the retention of key developers. Frey and Goette (1999) showed that external financial incentives can undermine the intrinsic motivation for participants, change their mindset from volunteers to unpaid employees, and work passively. Nakasai *et al.* (2018) observed that developers respond faster to bug reports that are submitted by users that have donor badges, which are used to acknowledge users for their contribution in donation, than users that do not have any donor badges. Such a negative impact may also apply to the bounty program. For example, is there any negative impact of bounties on the quality of issue addressing? Would issues without bounties delay due to other bounty issues? We encourage future research to study in this direction.

Fifth, it is interesting to explore **the reason why bounty issues are more likely to be addressed by hunters who committed code to the project before**. In Section 4.3, we study the correlation between various factors and the likelihood of being addressed by hunters who committed code to the project for the first time. Future research is encouraged to study the reasons behind this. For example, future research is encouraged to test the following hypotheses: 1) Is that bounty labels are only exposed within a project so that hunters who committed code to the project before are more likely to be aware of the existence of bounty issues and address them than hunters who committed code to the project for the first time? 2) Is a project with a higher frequency of bounty usage more likely to have stronger competition between hunters who committed code to the project before and hunters who committed code to the project for the first time?

Finally, to sustain open source projects, it is important to engage new developers. Our findings in Section 4.3 indicate that the value of a bounty issue is not a statistically significant factor that impacts the likelihood of a bounty issue being addressed by hunters who committed code to the project for the first time, which to some extent echoes prior studies that not all developers are motivated by money but by intrinsic (Zhou *et al.*, 2020b; Lakhani and Wolf, 2003; Shah, 2006; Von Krogh *et al.*, 2012; Coelho *et al.*, 2018). Future research is encouraged to explore **how to combine extrinsic and intrinsic incentives to improve the sustainability of open source projects**.

6 Threats to validity

Threats to **external validity** are related to the generalizability of our findings. We studied only hunters and backers and their associated bounty activities on GitHub and Bountysource. Our observations might be not generalizable to other bounty platforms and projects. For example, some open source systems, such as Firefox, focus on bounties related to security bugs. 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), bounty platforms (e.g., platform for private bounties), and projects. In Section 4.3 we construct models to study the association between the studied factors and the likelihood of attracting hunters who committed code to the project for the first time to address bounty issues. Although we follow a prior work (Zhou *et al.*, 2020b) by considering 35 factors along five dimensions, there might be additional factors. Future studies should investigate more factors. In this study, we study 7 cases in which hunters explicitly indicate the reason why they stopped working on the bounty issues. The number of cases is limited and might not be generalized. To reduce the bias, we also search online blogs and literature to validate our findings. Nevertheless, we consider it as a threat to validity and encourage future research to conduct study on more cases.

Threats to **internal validity** relate to the experimenter's bias and errors. One threat to internal validity is that we identify feature requests and bug reports by using the heuristics we design. It may introduce bias to our study (i.e., mislabeling) and the findings. To mitigate the threat, we manually verify our approach by examining 100 reports and find its accuracy is high (i.e., 90%). Another internal threat is that our findings only show the correlation between studied factors and the likelihood of a bounty issue being addressed by a hunter who commits code to the project for the first time , but not causation. Future research is encouraged to study the causation. Another threat to internal validity is that we include all bounties (both unclaimed and claimed bounties) in our analysis for RQ1. There might be some unclaimed bounties (e.g., the ones that were proposed for testing) are invalid, which may bias our findings. In this study, we do not consider concept drift when constructing models to understand the characteristics of bounty issues that potentially impact the likelihood of an issue being addressed by hunters who committed code to the project for the first time . We consider it as a threat to internal validity and we encourage future research to investigate it. In this study, we use the studied factors as the proxy to estimate the popularity and maturity of a project. For instance, we use age as a proxy of the maturity of a project. Although age is shown to be associated with the maturity of a project (Comino *et al.*, 2007), there may be other factors that are associated with popularity and maturity. We encourage future studies to investigate more factors to estimate the popularity and maturity of a project.

One threat to **construct validity** is that it is possible that a hunter may have multiple GitHub accounts and can be both types of hunters at the same time, which may bias our findings. We do not consider such cases in our study since GitHub does not provide an email address for a user anymore, which makes it difficult to identify accounts that belong to the same user. However, a prior study shows that more than 90% of GitHub users only have one account (Vasilescu *et al.*, 2015). Future studies should consider such cases when the data is available. In this study, we categorize hunters into two groups (i.e., hunters who committed code to the project before and hunters who committed code to the project for the first time) based on whether they have made any commit to a project before addressing a bounty issue of the project. There might be other ways to categorize the hunters, e.g., categorizing them based on their roles in GitHub projects and the extent of contribution. We use commit information to identify developers as hunters who committed code to the project before or hunters who committed code to the project for the first time since the mechanism to judge if a bounty issue is successfully addressed is to check whether a commit is made to address the associated issue. In addition, this commit-based heuristic has been used commonly in prior studies to identify different types of developers (Mockus *et al.*, 2002; Robles *et al.*, 2009; Coelho *et al.*, 2018). Nevertheless, we consider it as a construct threat to validity and future research is encouraged to study the characteristics and behaviors between other different groups.

7 Conclusions

Sustaining open source projects is challenging and requires community effort to help address issue reports (i.e., either fixing bugs or implementing new features). Due to the large number of issues that open source projects receive, developers may prioritize the effort on addressing issues that are easier to implement or have a higher priority. However, the other issues may still be blockers for some users. To facilitate addressing issues, some platforms offer bounties (i.e., financial incentives) to developers who address issues. In this paper, we study the characteristics of bounty backers and hunters, and their bounty related behaviors that have not been examined in depth. We find that: 1) Overall, the value of bounties that were proposed by both individual (median bounty value of $15) and corporate backers (median bounty value of $25) is small. Both individual and corporate backers are more interested in supporting implementing new features rather than fixing bugs. 2) In general, the income of 56.7% of the bounty hunters is no more than $100 and that of only 2.7% of the hunters is larger than $2,000. Most of the studied hunters are hunters who committed code to the project before and they addressed the majority of bounty issues. 3) The value of a bounty issue is not a statistically significant factor that attracts hunters who never commit code to the project to address the issue. Based on our findings, we have several suggestions for stakeholders of open source projects and hunters. For instance, stakeholders of open source projects should not expect backers to support on addressing bugs and should not expect a large amount of bounties to support issues unless their associated projects are very popular. Hunters should not expect to earn a large amount of money from addressing bounty issues.

References

Akobeng, A. K. (2007). Understanding diagnostic tests 3: receiver operating characteristic curves. *Acta paediatrica*, **96**(5), 644–647.

Androutsellis-Theotokis, S., Spinellis, D., Kechagia, M., Gousios, G., *et al.* (2011). Open source software: A survey from 10,000 feet. *Foundations and Trends in Technology, Information and Operations Management*, **4**(3–4), 187–347.

Apple Inc (2020). Apple Security Bounty). https://developer.apple.com/security-bounty/. (last visited: Dec 12, 2020).

Atiq, A. and Tripathi, A. (2016). Impact of financial benefits on open source software sustainability. In *International Conference on Information Systems (ICIS)*, pages 1–10.

Avelino, G., Passos, L., Hora, A., and Valente, M. T. (2016). A novel approach for estimating truck factors. In *IEEE 24th International Conference on Program Comprehension (ICPC)*, pages 1–10.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, **13**(1), 281–305.

Bissyandé, T. F., Thung, F., Lo, D., Jiang, L., and Réveillère, L. (2013). Popularity, interoperability, and impact of programming languages in 100,000 open source projects. In *IEEE 37th annual computer software and applications conference*, pages 303–312. IEEE.

Canfora, G., Di Penta, M., Oliveto, R., and Panichella, S. (2012). Who is going to mentor newcomers in open source projects? In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (FSE)*, pages 1–11.

Coelho, J., Valente, M. T., Silva, L. L., and Hora, A. (2018). Why we engage in floss: Answers from core developers. In *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 114–121.

Comino, S., Manenti, F. M., and Parisi, M. L. (2007). From planning to mature: On the success of open source projects. *Research Policy*, **36**(10), 1575–1586.

Dagenais, B., Ossher, H., Bellamy, R. K. E., Robillard, M. P., and de Vries, J. P. (2010). Moving into a new software project landscape. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ICSE '10, page 275–284.

Dinnie Muslihat (2019). How to Prioritize Feature Requests for Software Development. https://zenkit.com/en/blog/how-to-prioritize-feature-requests-for-software-development. (last visited: November 8, 2019).

Duebendorfer, T. and Frei, S. (2009). Why silent updates boost security. *TIK, ETH Zurich, Tech. Rep*, **302**.

Eghbal, N. (2016). *Roads and bridges: The unseen labor behind our digital infrastructure*. Ford Foundation.

Eghbal, N. (2019). A handy guide to financial support for open source.

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

Frei, S., Duebendorfer, T., and Plattner, B. (2008). Firefox (in) security update dynamics exposed. *ACM SIGCOMM Computer Communication Review*, **39**(1), 16–22.

Frey, B. S. and Goette, L. (1999). Does pay motivate volunteers? *Working paper/Institute for Empirical Research in Economics*, **7**.

HackerOne (2018). 118 FASCINATING FACTS FROM HACKERONE'S HACKER-POWERED SECURITY REPORT 2018. https://www.hackerone.com/blog/118-Fascinating-Facts-HackerOnes-Hacker-Powered-Security-Report-2018. (last visited: August 27, 2018).

Harhoff, D., Henkel, J., and Von Hippel, E. (2003). Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research policy*, **32**(10), 1753–1769.

Hata, H., Guo, M., and Babar, M. A. (2017). 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.

Izquierdo, J. L. C. and Cabot, J. (2018). The role of foundations in open source projects. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society*, pages 3–12.

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

Kochhar, P. S., Thung, F., and Lo, D. (2014). Automatic fine-grained issue report reclassification. In *2014 19th International Conference on Engineering of Complex Computer Systems*, pages 126–135. IEEE.

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

Krishnamurthy, S., Ou, S., and Tripathi, A. K. (2014). Acceptance of monetary rewards in open source software development. *Research Policy*, **43**(4), 632–644.

Kuhn, M. *et al.* (2008). Building predictive models in r using the caret package. *Journal of statistical software*, **28**(5), 1–26.

Lakhani, K. R. and Wolf, R. G. (2003). Why hackers do what they do: Understanding motivation and effort in free/open source software projects.

Lee, A., Carver, J. C., and Bosu, A. (2017). Understanding the impressions, motivations, and barriers of one time code contributors to floss projects: A survey. In *IEEE/ACM 39th International Conference on Software Engi-*

neering (ICSE), pages 187–197.

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

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, **5**(9), 1315–1316.

Matt Asay (2020). Bug bounties won't make you rich (but you should participate anyway). https://www.techrepublic.com/article/bug-bounties-wont-make-you-rich-but-you-should-participate-anyway/. (last visited: January 21, 2020).

Mirko Zorz (2020). Full-time bug hunting: Pros and cons of an emerging career. https://www.helpnetsecurity.com/2020/04/07/bug-hunting-career/. (April 7, 2020).

Mockus, A., Fielding, R. T., and Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, **11**(3), 309–346.

Moore, D. S. and Kirkland, S. (2007). *The basic practice of statistics*, volume 2. WH Freeman New York.

Nakasai, K., Hata, H., and Matsumoto, K. (2018). Are donation badges appealing?: A case study of developer responses to eclipse bug reports. *IEEE Software*, **36**(3), 22–27.

Rajbahadur, G. K., Wang, S., Kamei, Y., and Hassan, A. E. (2019). Impact of discretization noise of the dependent variable on machine learning classifiers in software engineering. *IEEE Transactions on Software Engineering*.

Robert Lemos (2019). Bug Bounties Continue to Rise, but Market Has Its Own 1% Problem). https://www.darkreading.com/vulnerabilities---threats/vulnerability-management/bug-bounties-continue-to-rise-but-market-has-its-own-1--problem/d/d-id/1335689.

Roberts, J. A., Hann, I.-H., and Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects. *Management science*, **52**(7), 984–999.

Robles, G., Gonzalez-Barahona, J. M., and Herraiz, I. (2009). Evolution of the core team of developers in libre software projects. In *2009 6th IEEE international working conference on mining software repositories*, pages 167–170. IEEE.

Romano, J., Kromrey, J. D., Coraggio, J., and Skowronek, J. (2006). Appropriate statistics for ordinal level data: Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys. In *annual meeting of the Florida Association of Institutional Research*, pages 1–33.

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management science*, **52**(7), 1000–1014.

Steinmacher, I., Silva, M. A. G., and Gerosa, M. A. (2014). Barriers faced by newcomers to open source projects: a systematic review. In *IFIP International Conference on Open Source Systems*, pages 153–163. Springer.

Tantithamthavorn, C., McIntosh, S., Hassan, A. E., and Matsumoto, K. (2016). An empirical comparison of model validation techniques for defect prediction models. *IEEE Transactions on Software Engineering*, **43**(1), 1–18.

Tantithamthavorn, C., McIntosh, S., Hassan, A. E., and Matsumoto, K. (2018). The impact of automated parameter optimization on defect prediction models. *IEEE Transactions on Software Engineering*, **45**(7), 683–711.

Tom Ritter (2020). Firefox's Bug Bounty in 2019 and into the Future. https://blog.mozilla.org/security/2020/04/23/bug-bounty-2019-and-future/. (last visited: April 23, 2020).

Vasilescu, B., Posnett, D., Ray, B., van den Brand, M. G., Serebrenik, A., Devanbu, P., and Filkov, V. (2015). Gender and tenure diversity in github teams. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3789–3798.

Von Hippel, E. (2007). Horizontal innovation networks—by and for users. *Industrial and corporate change*, **16**(2), 293–315.

Von Krogh, G., Haefliger, S., Spaeth, S., and Wallin, M. W. (2012). Carrots and rainbows: Motivation and social practice in open source software development. *MIS quarterly*, pages 649–676.

Wang, S., Chen, T.-H., and Hassan, A. E. (2018). Understanding the factors for fast answers in technical Q&A websites. *Empirical Software Engineering*, **23**(3), 1552–1593.

Weiss, M. (2011). Control and diversity in company-led open source projects. *Open Source Business Resource*, (April 2011).

Ye, Y. and Kishida, K. (2003). Toward an understanding of the motivation open source software developers. In *Proceedings of the 25th International Conference on Software Engineering (ICSE)*, pages 419–429.

Zhao, M., Grossklags, J., and Chen, K. (2014). 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.

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

Zhou, J., Wang, S., Bezemer, C.-P., and Hassan, A. E. (2020a). Bounties on technical Q&A sites: a case study of stack overflow bounties. *Empirical Software Engineering*, **25**(1), 139–177.

Zhou, J., Wang, S., Bezemer, C.-P., Zou, Y., and Hassan, A. E. (2020b). Studying the association between bountysource bounties and the issue-addressing likelihood of github issue reports. *IEEE Transactions on Software Engineering*.

Zhou, M., Mockus, A., Ma, X., Zhang, L., and Mei, H. (2016). Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects. *ACM Transactions on Software Engineering and*

Methodology (TOSEM), **25**(2), 1–29.