Studying the association between Gitcoin's issues and resolving outcomes ☆

Morakot Choetkiertikul [a], Arada Puengmongkolchaikit [a], Pandaree Chandra [a], Chaiyong Ragkhitwetsagul [a,*], Rungroj Maipradit [b], Hideaki Hata [c], Thanwadee Sunetnanta [a], Kenichi Matsumoto [b]

[a] Faculty of Information and Communication Technology, Mahidol University, Thailand
[b] Nara Institute of Science and Technology, Japan
[c] Shinshu University, Japan

ABSTRACT

The development of open-source software (OSS) projects usually have been driven through collaborations among contributors and strongly relies on volunteering. Thus, allocating software practitioners (e.g., contributors) to a particular task is non-trivial and draws attention away from the development. Therefore, a number of bug bounty platforms have emerged to address this problem through bounty rewards. Especially, Gitcoin, a new bounty platform, introduces a bounty reward mechanism that allows individual issue owners (backers) to define a reward value using cryptocurrencies rather than using crowdfunding mechanisms. Although a number of studies have investigated the phenomenon on bounty platforms, those rely on different bounty reward systems. Our study thus investigates the association between the Gitcoin bounties and their outcomes (i.e., success and non-success). We empirically study over 4,000 issues with Gitcoin bounties using statistical analysis and machine learning techniques. We also conducted a comparative study with the Bountysource platform to gain insights into the usage of both platforms. Our study highlights the importance of factors such as the length of the project, issue description, type of bounty issue, and the bounty value, which are found to be highly correlated with the outcome of bounty issues. These findings can provide useful guidance to practitioners.

Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.

1. Introduction

The development of Open-Source Software (OSS) projects is encouraged through collaborations and knowledge sharing among project contributors. It mostly relies on volunteer work. Recruiting an experienced developer for a challenging task becomes a non-trivial task (Choi et al., 2010). Thus, bounty platforms are used to support the development of OSS projects through the mechanisms of bounty and crowdsourcing (Zhou et al., 2021; Kanda et al., 2017; Zhou et al., 2020b; Hata et al., 2017). A bounty refers to a reward for developers who can accomplish tasks, such as bug fixing or new feature creation, declared on a bounty platform (Finifter et al., 2013). Those rewards could be in cash, cryptocurrency, or badges that can reflect the developer's reputation (Nakasai et al., 2019). The previous study has shown that monetary incentive is an essential motivation that can attract the developers to contribute to each project on the platforms (Hata et al., 2017).

Recently, there have been a number of bounty platforms allowing contributors to participate and contribute to OSS projects, such as Bountysource[1] and HackerOne.[2] Specifically, the crowdsourcing bounty platform allows backers to support any specific tasks of projects, such as implementing new features or fixing bugs by providing rewards on resolving tasks as bounties. Several empirical studies aim to investigate the impact on software development projects using this bounty mechanism. Kanda et al. (2017) report that bounties tend to attract developers to work on

---

1 https://www.bountysource.com.
2 https://www.hackerone.com.

the projects more than those without bounties. Several existing empirical studies on the Bountysource platform found that the amount of bounties is a factor that affects the outcome of the project (Kanda et al., 2017; Zhou et al., 2020b).

Gitcoin[3] is a new bounty platform established in 2017. It plays a role as a bounty-based collaboration community for funders (i.e., bounty issue owners) and developers (i.e., contributors) to work on the issues easily on GitHub.[4] Gitcoin purely uses the Ethereum blockchain and cryptocurrency, such as Bitcoin, for their reward system. Funders provide a bounty in cryptocurrency-based rewards. In particular, Gitcoin allows a funder to solely support a bounty task, such as a project development task and bug resolving task, which is different from traditional bounty platforms that rely on crowdfunding mechanisms that require a number of funders to support a bounty. Currently, Gitcoin is gaining more attention from funders and developers because of its unique reward system, as the number of bounties created on Gitcoin and the number of active developers are increasing. Over 6,000 bounty issues have been created on Gitcoin, and over 10,000 active developers participate in those bounties.[5] While previous studies have focused on traditional bounty platforms such as Bountysource, it is important to note that these platforms share some common characteristics while also exhibiting differences. There remains a gap in understanding the dynamics and characteristics specific to cryptocurrency-based bounty platforms. To address this gap, a study on cryptocurrency-based bounty platforms and a comparative study between the two platforms are required.

In this paper, we aim to investigate the factors that impact the success of the bounty issue resolution created in Gitcoin. We thus adopt the studying approach from Zhou et al. (2020b, 2021). In our study, we empirically study over 4,000 bounty issues (i.e., issue reports) in Gitcoin created between September 2017 to December 2020, which involve 1,096 software project repositories hosted in GitHub with a total bounty value of over 18,000 ETH and 11,000,000 USDT. Our study extracted 28 features characterizing the Gitcoin bounty issues (e.g., developer's experience level, bounty proposing time, and estimated development time). The study conducted by Zhou et al. (2020b, 2021) on Bountysource highlighted the significant relationship between the likelihood of issue resolution and factors such as the timing of proposing bounties, the bounty value, and the duration of the issue. Our study, conducted on a different bounty platform, confirms these findings as we observe similar factors that impact issues on Gitcoin. However, our study also uncovers additional insights. We found that the description of tasks, the experience level of contributors, and the type of contribution are crucial in determining the success rate of issues. Furthermore, our comparative analysis reveals differences in terms of project types and focused topics on these platforms, which can serve as a guide for platform selection and creating issues with a higher chance of success.

This paper is organized as follows. Section 2 provides the background of bounty platforms, explains the Gitcoin platform, and discusses related work on bounty platform studying. Section 3 discusses our studying approach, including data collection, preprocessing and labeling, and data analysis. We then discuss our extracted features of the Gitcoin bounties in Section 4. In Section 5, we discuss the results from our feature and correlation analysis. We then report and discuss our findings in identifying important features in Section 6. In Section 7, we discuss the threats to the validity of our study, and we conclude our study in Section 8.

----

3 http://gitcoin.co.
4 https://gitcoin.co/bounties/funder.
5 The data were collected on 20 January 2021.

2. Background and related work

2.1. Bounty platforms

Using bounties to accomplish tasks in software development projects has become a popular approach (Atiq and Tripathi, 2014). Since an OSS project allows developers to participate in the development of the projects, a bounty can thus enlarge the engagement of developers to work on the projects (Zhou et al., 2020a). A bounty could be in various forms, such as cash, cryptocurrency, and a badge. A badge is used to enhance the reputation of developers (Nakasai et al., 2019). Funding the project as a bounty is a popular way to drive OSS projects, which allows any developers to work on the projects freely. A funder can be either an individual or an organization that needs to accomplish a specific task (Coelho et al., 2018). For example, they are a project user and need to have a new feature implemented in the project.

Several studies were conducted to understand both intrinsic and extrinsic incentives of developer contributions. An intrinsic incentive refers to the enjoyment of contribution, the ability to work on the projects, or even the satisfaction of solutions (Krishnamurthy et al., 2014). In contrast, an extrinsic incentive can be motivated by the desire to get rewards. Studies reported that the developers are likely to complete tasks faster if an amount of bounty is provided (Kanda et al., 2017; Hata et al., 2017). The Bountysource platform is among the most popular and widely used (Zhou et al., 2020b). Bountysource allows several funders to fund a bounty issue. Then, the developers can work on the projects through an issue tracking system such as GitHub and Bugzilla. In contrast, a Gitcoin bounty platform allows only one funder to fund a bounty issue.

2.2. Gitcoin

Gitcoin is a bounty platform that allows developers to contribute to OSS projects with bounties. Gitcoin purely uses cryptocurrencies as a bounty reward for those who can resolve issues. This rewarding approach provides a high level of security on decentralized systems and no cost of transaction (Titov et al., 2021). Gitcoin supports several types of tokens in cryptocurrency networks, such as Ethereum (ETH) and Bitcoin (BTC) (Li et al., 2019). Gitcoin particularly supports ERC20 tokens in its ecosystem. ERC20 uses the Smart Contracts program to help organize and formalize the agreements on the networks to work properly among users (e.g., companies and entrepreneurs) (Chen et al., 2020). Therefore, advocating ERC20 on the Gitcoin platform could ensure the authenticity and credibility between funders and contributors. With this intention, the funders can use well-known cryptocurrencies to fund issues on Gitcoin safely in any granted types of tokens on the platform.

Fig. 1 shows the lifecycle of the Gitcoin bounty issue. Gitcoin uses GitHub's issue-tracking system to track issue-resolving progress. A GitHub issue corresponding to the Gitcoin bounty issue must exist in the GitHub issue tracking system. The GitHub issue is usually initiated by an issue owner (i.e., funder/backer) of a GitHub repository in Step (1). The issue owner then initiates the issue on the Gitcoin platform and fills in the issue's details, such as title, contribution type, required experience level, expected task duration, and bounty value, as well as the corresponding GitHub issue link (Step (2)). In Step (3), developers (i.e., contributors) can find Gitcoin issues that they would like to contribute from the Gitcoin platform. In addition, the issue owner can select whether developers must be approved before contributing or whether any developers can contribute to the issue (i.e., permissionless mode). In the former case, the prospect developer must send a request to contribute to the issue owner
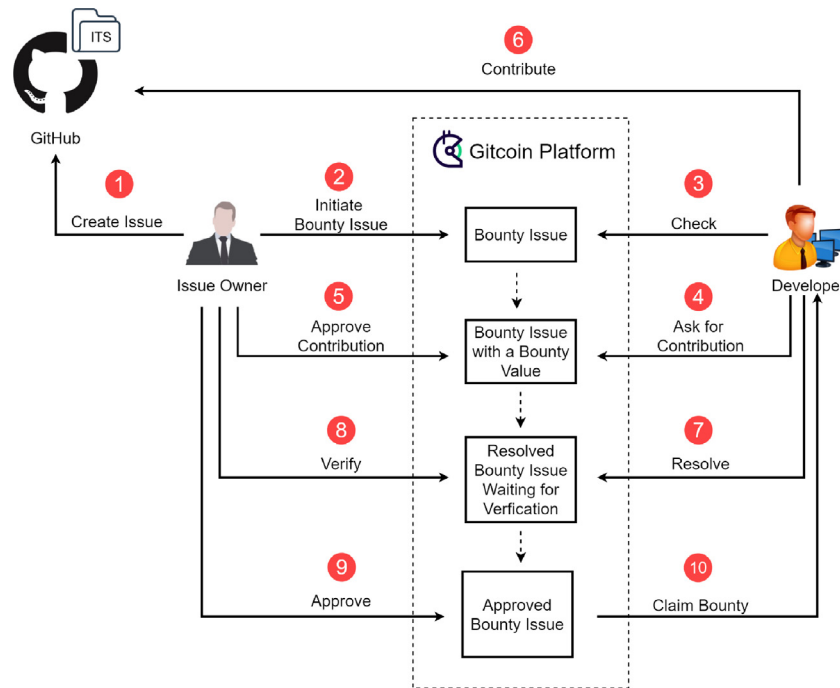
Fig. 1. Gitcoin issue's lifecycle.

and wait for the acceptance before starting the work (④). Once the developer has been approved to work on the issue (⑤), they can access to source code and related resources from the provided GitHub issue and can also communicate with the issue owner from there. The developer resolves the issue in Step (⑦), and the issue owner can review the issue and decide whether to accept the issue resolution in Step (⑧). The issue owner then confirms the completion of the task on the Gitcoin platform to close the Gitcoin issue (⑨). The developer can then claim the bounty reward in the last step (⑩). Fig. 2 shows an example of a Gitcoin issue.

2.3. Studies of bounty platforms

Several bounty programs were launched in the past. For example, proprietary software like Google or Mozilla Firefox allowed internal employees to work on the vulnerability rewards programs (Luna et al., 2019; Finifter et al., 2013). The empirical study from Finifter et al. (2013) described two vulnerability rewards programs (VRPs) between Chrome and Mozilla Firefox. They indicated that the monetary incentive could also help prevent the researchers from selling the research results of the system's security in a gray market. In addition, Zhao et al. (2014) studied platforms that act as the middlemen between the contributors and the software vendors, such as BugCrowd,[6] Synack,[7] and CrowdCurity (as know as Cobalt at present).[8] In particular, bounties in these platforms focus on security and penetration testing.

Apart from the vulnerability rewards programs, the empirical studies of bounty platforms for open-source software projects are

also ubiquitous in the software engineering research community. A number of projects cannot be accomplished without collaboration among the contributors in a community. According to the study from Zhou et al. (2020a), a question-and-answer forum such as Stack Overflow[9] is one type of bounty mechanism that allows contributors (i.e., answerers) to get the forum's reputation points. Moreover, Wang et al. (2018) described that the incentive systems like gamification were developed to encourage users to provide answers to questions. Thus, it can help motivate users to engage with the questions.

As the number of bounty hunters (i.e., the ones who mainly work on resolving issues with a bounty reward) is increasing in software development communities, a crowdsourcing bounty platform allows multiple funders to fund a bounty. The empirical study of Bountysource by Kanda et al. (2017) demonstrated that issues with bounties are more likely to be solved than those projects without bounties. Further empirical studies from Zhou et al. (2020b, 2021) indicate that, on the Bountysource bounty platform, funders are likely to offer the bounties in higher value and more frequently than the individual backers (Zhou et al., 2021). They also reported that the bounty value is not the most important factor that attracts contributors to address the issues. They indicated that some contributors are not motivated by only rewards or monetization. Instead, they could be driven by their own interests or desires to commit to the work (Zhou et al., 2020b). We thus adopt their studying approach to our study to understand the factors deriving from the Gitcoin bounty platform.

3. Methodology

In this section, we describe the approach of our study, including data collection, preprocessing, and labeling. The overview of our study is shown in Fig. 3.

---

[6] https://www.bugcrowd.com/.
[7] https://www.synack.com/.
[8] https://cobalt.io/.

[9] https://stackoverflow.com/.

OCT Token Timelock Contract Bug Bounty Program

0.10 ETH   345.35 USD   </> Back End

OPEN

Time Left	In 3 Days	Opened	4 Days Ago
Project Type ❶	Cooperative	Time Commitment	Hours
Permissions ❶	Permissionless	Accepted	3 Days Ago

Start Work   ⊙ View On Github   ⮜ Share

Description

This bug bounty program is hosted on Octopus Network / oct-token-eth. All bug reports must be submitted t
created by Octopus Network.

Program Overview

Octopus Network is a sidechain network to host Web3.0 application specific blockchains, aka appchains, aro
out-of-box interoperability, complete infrastructure and ready to be engaged community, the Octopus Netwo
appchain by 100X.
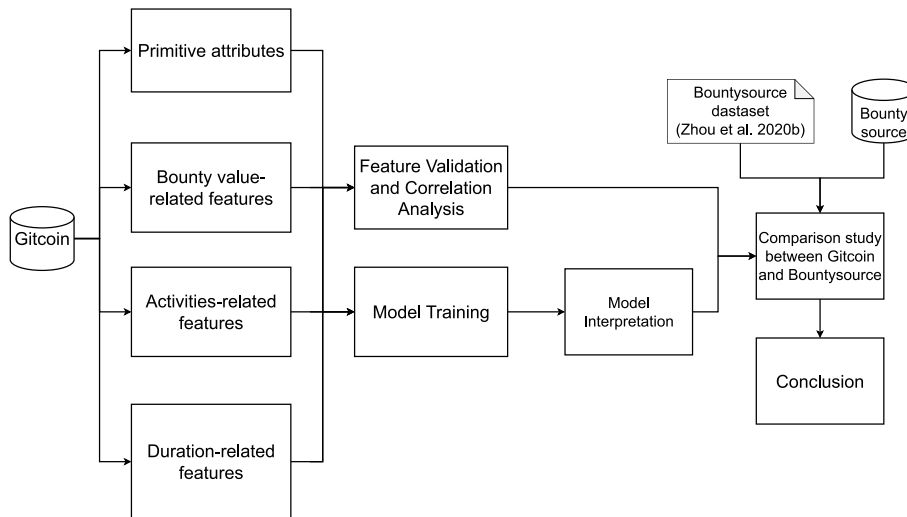
Fig. 2. An example of the Gitcoin issue.[10]

Fig. 3. Overview of our study.

3.1. Overview

We first collected bounty issues from the Gitcoin platform. We performed data preprocessing and labeling to identify issues that were successfully resolved, and the bounty rewards were claimed. We then perform feature extraction to characterize the Gitcoin issues. We extracted four groups of features from the issues, which are 1. primitive attributes, 2. bounty value-related features, 3. activity-related features, and 4. duration-related features. We then analyze the extracted features to investigate the factors that correlate and impact the outcome of Gitcoin's issues. In the first approach, we apply correlation analysis techniques, using the Spearman Rank Correlation Coefficient (Liu,

2010), to explore the correlation among each feature. To control for the risk of false positive results due to multiple comparisons, we use the Bonferroni correction in this analysis (Arcuri and Briand, 2014) (see Section 5). We then employed machine learning techniques, including Random forests (Lan, 2019) and logistic regression (Harrell, 2015), to construct classification models to study the relationship between the extracted features and the outcome of the issues (see Section 6). Specifically, we aimed to identify the features that have a strong correlation with the outcomes of the issues. We then applied the Point Biserial Correlation Coefficient (Bonett, 2020) with Bonferroni adjustment (Arcuri and Briand, 2014) to observe the correlation between important features and the outcomes. These approaches complement each other and help to conclude our study. The data used in our study, including raw data of collected Gitcoin issues and extracted features and the source code used in the study, are made publicly available at https://doi.org/10.5281/zenodo.8313155. Furthermore, to gain a comprehensive understanding of the differences and similarities in the issues, we conducted a

---

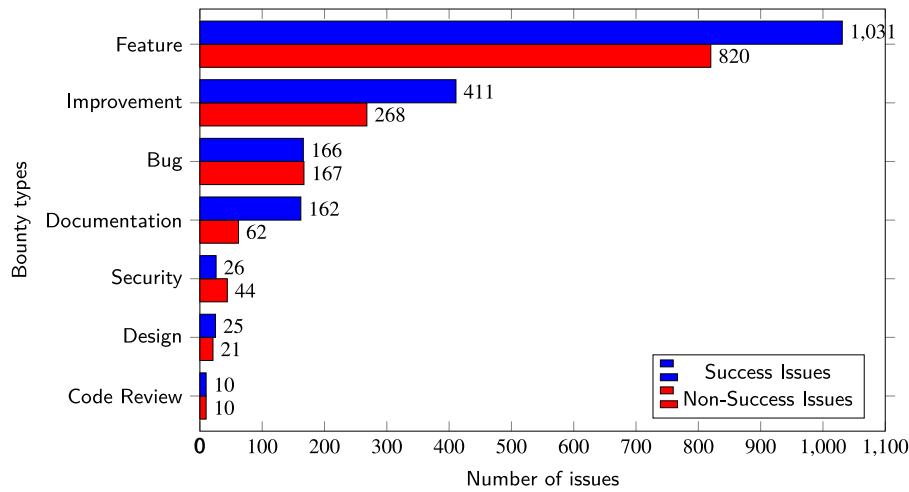10 https://gitcoin.co/issue/octopus-network/oct-token-eth/1/100026469.

Fig. 4. Bounty types.

comparison study between Gitcoin and Bountysource. This study is described in detail in Section 6.3. To perform the study, we used the Bountysource dataset provided by Zhou et al. (2020b) as well as additional data that we collected.

3.2. Data collection and preprocessing

We collected bounty issues that were created from September 2017 to December 2020 on the Gitcoin bounty platform via Gitcoin's Application Program Interface (API).[11] Specifically, we used the *bounties* API to retrieve a list of bounty issues. The API responds with files in JSON format that contains all the information we need for our analysis. To prepare the data for our study, we apply various techniques such as data preprocessing, filtering, and feature extraction to the raw JSON files. Table 1 shows the number of issues in our dataset. We collected a total of 6,638 bounty issues from Gitcoin. In Gitcoin, an issue can be created on two Ethereum networks: Mainnet and Rinkeby. The latter is used by developers for testing the platform, while the former is used for the actual transactions. Therefore, our study only uses the issues created on the Mainnet network. In total, our dataset contains 4,584 issues in Mainnet (69% of the collected issues) as shown in Table 1.

In addition, our study focuses on the Gitcoin issue-resolving outcomes. The collected issues were thus classified into two classes which are *success issues* and *non-success issues*. A success bounty refers to an issue that is successfully solved, and a bounty reward has been paid out. On the other hand, a non-success bounty implies an unsuccessfully resolved and unpaid issue. These labels were determined based on the status of the issues and the bounty-paid status of the Gitcoin issues. Among those collected issues, 2,662 issues (58.1%) were marked as success issues.

4. Features

Table 2 shows the list of extracted features. In this section, we discuss each feature in detail.

4.1. Primitive attribute of the Gitcoin issue

The primitive attributes indicate the basic information of an issue, e.g., bounty types, contribution types, and length of projects.

Table 1
Dataset description.

Total numbers of	Numbers	Percentage (Mainnet)
Issues in Gitcoin	6,638	
Issues in Mainnet network	4,584	
Closed issues	3,744	81.7%
Opened issues	840	18.3%
Success issues	2,662	58.1%
Non-success issues	1,922	41.9%

4.1.1. Bounty types

The bounty type feature (*bounty_type*) explains the types of bounty provided in Gitcoin. This informs contributors what the bounty type of an issue is. There are 9 bounty types of an issue as depicted in Table 3. The first type is *Feature* type. We found 1,851 issues (40.4%) from Mainnet which were assigned to this type. This is the largest type which reflects that an issue requires the contributors to develop or create new features of a system. The second most popular bounty type is *Improvement* which requires improvements to existing features, functions, or the system that the issue owners own. We found 679 issues (14.8%) assigned to an improvement type. The third type is *Bug* (333 issues — 7.3%). The *Bug* type indicates a bug-fixing task. For example, the issue `Not Able To Login Into WordPress From Metamask Mobile Browser`[12] is a bug type requiring someone to fix an error on login into the WordPress from the Metamask mobile browser. The next type is the *Documentation*. This type has only 224 issues (4.9% of the total number of issues). The *Documentation* type could be related to information documentation such as programming guides, tutorials, descriptions of technologies, and translations.

The *Security* type has been found with 70 issues (1.5%). The *Security* type refers to security-related implementation tasks such as vulnerability protection, software penetration, and system audit. The next bounty type is the *Design* type, which has 46 issues (1%). The issue owners could require any design for their work in terms of flow design, wireframes, and architecture design, such as logo, schwag product, and POAP badges used for NFTs events. The last bounty issue type is the *Code Review* type. We found only 20 issues (0.4%). This bounty type requires contributors to review source code, such as defect identification, bug localization, and code quality analysis. Note that two more issue types do not have specific tasks. They are *NA* type, which has 1,015 issues (22.1%), and *Other* type, which has 346 issues (7.6%).

Table 2
Feature descriptions.

Group	Feature names	Descriptions	Rationale
1. Primitive attributes	bounty_type	Type of bounty, e.g., Bug, Feature, and Security	The primitive attributes provide essential information about the bounty and serve as foundational details that can significantly correlate with the outcome of the bounty by providing primitive information to contributors.
	project_length	Relative length of the project, e.g., hours, days, weeks, months	
	experience_level	Recommended experience level	
	contribution_type	Type of contribution, e.g., Traditional, Contest, Cooperative	
	github_comments[a]	The number of comments in an issue	
	description_length[a]	Length of issue description	
2. Bounty value-related	token_name	Type of token, e.g., ETH, GIT	The bounty value-related features focus on the value of the bounty and its changes. These features reflect the incentives and rewards associated with the bounty, which can impact contributors' motivation and willingness to engage.
	value_in_eth[a]	Value of the bounty in Ethereum	
	value_in_usdt[a]	Approximation of value in US Dollars at bounty web3_created timestamp	
	value_in_usdt_now[a]	Approximation of current value in US Dollars	
	token_value_in_usd	The actual value of the token associated with the bounty in USD at the time of issue creation	
	value_in_token[a]	Amount of tokens rewarded for bounty	
	increased_bounty_times	The number of times that the bounty value is increased	
	changed_bounty_value	The value that the bounty has been increased from created to the latest increasing bounty	
3. Activity-related	number_of_fulfillments[a]	The number of participants that submitted work to an issue	The activity-related features are the values that relate to the activities that happened to the bounty by other developers besides the creator of the bounty. These features reflect the interest and effort of contributors, indicating a level of collaboration and involvement that can contribute to the success of an issue.
	number_of_interests[a]	The number of participants that are interested in an issue	
	number_of_activities	The number of activities that occur in an issue	
	number_of_user_in_activities	The number of usernames that shows in the issue's activities, including a funder.	
	firstAct_activity_type	The activity type of the first activity occurs	
	lastAct_activity_type	The activity type of the last activity happens	
4. Duration-related	duration_create_to_expire	The number of days between the creation of an issue and issue expiration date	The duration-related features contain various time-related values associated with issues and their bounties. These features are aimed at providing an understanding of the temporal aspects of the bounty issues. They may offer insights into critical factors such as the time sensitivity of a bounty, the ability to set realistic deadlines, and the assessment of timeliness.
	duration_create_to_new_bounty[a]	The number of days between the creation of an issue and its first bounty	
	duration_create_to_worker_applied	The number of days between the creation of an issue and when participant request	
	duration_create_to_start	The number of days between the creation of an issue and when the participant starts working	
	duration_create_to_stop	The number of days between the creation of an issue and when the participant stops working	
	duration_create_to_done	The number of days between the creation of an issue and when the work is done	
	duration_create_to_submitted	The number of days between the creation of an issue and when the work is submitted	
	duration_create_to_killed	The number of days between the creation of an issue and when the bounty is killed	

[a] Adopted from Zhou et al. (2020b, 2021).

Table 3
Bounty types.

Bounty type	Description	Issues
Feature	Creating new features of a system.	1,851
Improvement	Improving existing features, functions, or the system.	679
Bug	Creating a bug fix.	333
Documentation	Creating documentation for a system.	224
Security	Performing security-related activities, e.g., penetration test, system audit.	70
Design	Creating system or artistic design.	46
Code review	Reviewing code	20

Table 4
Project lengths.

Length	Issues
Hours	2,249
Days	1,164
Weeks	276
Months	22
Unknown	536
NA	337

Fig. 4 shows the success issues of each bounty type. We found that most of the bounty types have a higher number of success issues compared to non-success issues, which are *Feature* (1,031 success issues — 55.7%), *NA* (633 success issues — 62.4%), *Improvement* (411 success issues — 60.5%), *Other* (198 success issues — 57.2%), *Documentation* (162 success issues — 72.3%), and Design (25 success issues — 54.4%). In contrast, there are two bounty types (*Bug* and *Security*) that have a higher number of non-success issues than success issues. Note that we handled the minority of the issues (7.6%) that were assigned to undefinable types such as *0*, *Andere*, and *Funkcja* by grouping them into the *Other* type.
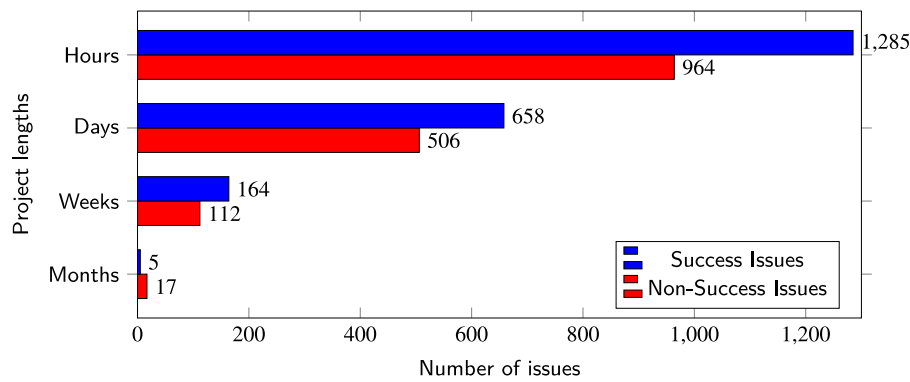
Fig. 5. Project lengths.

Table 5
Experience levels.

Exp. level	Issues
Beginner	2415
Intermediate	859
Advanced	515

4.1.2. Project lengths

The project length feature (*project_length*) describes the relative time duration that an issue owner expects an issue to be completed. The issue owner can select four types of project length which are *Hours*, *Days*, *Weeks*, and *Months* to provide an estimation of the task duration for contributors before committing to an issue. As shown in Table 4, the top two project lengths are *Hours* — 2,249 issues (49.1%) and *Days* — 1,164 issues (25.4%). However, we found that 536 issues (11.7%) were assigned to *Unknown* and 337 issues (7.4%) were assigned to *NA*. According to Fig. 5, most issues in the project lengths of Hours, Days, and Weeks are success issues. However, the project lengths of Months have 70% of issues that are non-success.

4.1.3. Experience levels

The required experience level feature (*experience_level*) is a feature that declares the required experience level of contributors who potentially participate in an issue. An issue owner can specify the experience level to ensure that a contributor can resolve an issue. In Gitcoin, we found three experience level values, including *Beginner*, *Intermediate*, and *Advanced*. We can see from Table 5 and Fig. 6 that the top required experience level in Gitcoin is the *Intermediate* level (2,415 issues — 52.7%).[13] It also shows that the *Intermediate* level has a large number of success issues (58.2%) followed by the *Beginner* level (18.7%). Interestingly, 2 experience levels have a higher non-success rate than the success ones: *Advanced* and the *Other* levels.

4.1.4. Contribution types

The contribution type feature (*contribution_type*) refers to the types of contribution provided in Gitcoin: *Tradition*, *Contest*, and *Cooperative*. The *Tradition* type means that there is only one contributor who can be approved to contribute and get a bounty reward. This is the most popular contribution type containing 3,307 issues (72.1%). In contrast, the *Contest* type allows a number of contributors to work on an issue, but only one can be paid.

There are 862 issues (18.8%) of the *Contest* type found. The least popular one is the *Cooperative* type, which contains 415 issues (9.1%). It allows several contributors to work on the issue, and the issue owners can decide to pay the bounty to more than one contributor.

Fig. 7 shows that the *Tradition* type has the highest number of success issues (1,921 success issues — 58.1%) and also has a higher number of success issues than non-success issues. Similarly, the *Contest* type has a higher number of success issues than non-success issues (571 success issues — 66.2%). However, the Cooperative type is the only one with a higher number of non-success issues than its counterparts (245 non-success issues — 59%).

4.1.5. GitHub comments

The GitHub comment feature (*github_comments*) indicates the number of comments from a corresponding GitHub issue. The GitHub issue tracking system is a platform used for tracking the contributors' work and addressing an issue they are working on. It can also be used as an intermediary channel of communication between issue owners and contributors. Comments can be the issue description, work discussion, or additional information on an issue. Fig. 8 shows the number of comments posted on the GitHub issues. It shows that the maximum number of comments reaches over 25 comments, and the minimum is 0. However, we found that the issues in any number of comments have a higher success rate than the non-success rate with 6 comments on average.

4.1.6. Length of description

The length of the description feature (*description_length*) describes the relative length of the issue description (i.e., number of characters). This feature is extracted to observe whether the description length correlates with the outcome of an issue. Fig. 9 shows the length of the issue description from the collected issues. We have noticed that the numbers of success issues and non-success issues are almost equal in those issues that have a longer description rather than those shorter description issues which have a higher number of success issues.

4.2. Bounty related features

The bounty value-related features reflect the value of the bounty reward of an issue, e.g., value in ETH, increasing bounty value, and token names.

---

[13] Note that we also found the experience level named in German (*Mittlere*) and Polish (*Pośredni*) which means 'Intermediate' in English. Thus, we then grouped this data into an Intermediate level as well.
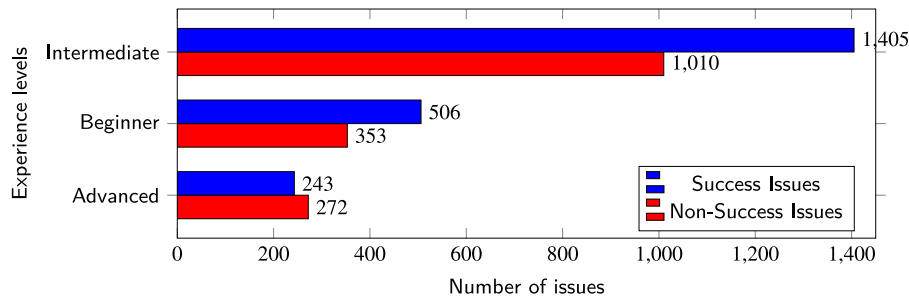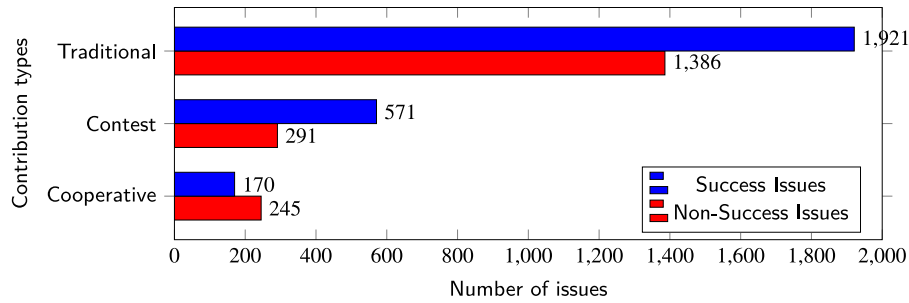
Fig. 6. Experience levels.
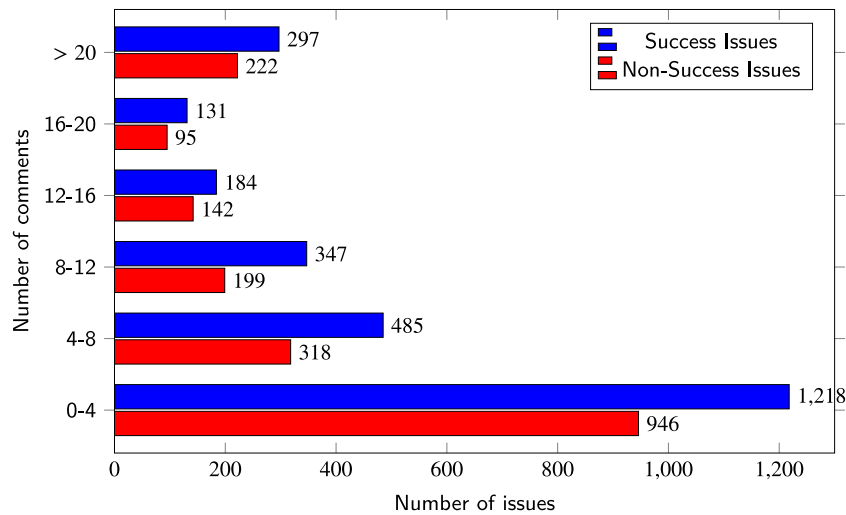

Fig. 7. Contribution types.

Fig. 8. GitHub comments.

4.2.1. Token names

The token names feature (*token_name*) indicates the types of cryptocurrency tokens provided as a bounty reward given to the contributors who can resolve bounty issues in Gitcoin. Nevertheless, over one hundred token types are found on the platform, and we then picked the 5 most popular token types (Fig. 10). The *Ethereum (ETH)* is the most popular token type that has been used in Gitcoin projects with 2416 issues, accounting for 52.7% of all the issues. The *Dai* token (DAI) is the second most popular token, with 1,208 issues (26.4%). The second token is the *Sai* token (SAI) found 435 issues (9.5%), followed by *USD* Coin token (USDC) which consists of 66 issues (1.4%). The last one is the *MyBit* token (MYB), which has 25 issues or only 0.6% of the total issues. According to the token types, most of them are operated under the Ethereum blockchain, as explained in Section 2.2. We found that the USDC token has 56.1% for non-success issues in the platform.

4.2.2. Bounty value

For the features that are related to the bounty value in Gitcoin, we particularly focus on the bounty value in Ether or ETH (*value_in_eth*) since it is a token worked under the Ethereum blockchain and is mostly used as a bounty reward in the Gitcoin platform. Fig. 11 shows the success of the issues with ETH tokens. The maximum value of the ETH that has been proposed is 1.14 (0.2 on average). Additionally, we found that the issues are mostly successful in the range of bounty values between 0.1 to 0.4 ETH. On the other hand, the upper values have a higher non-success rate than the success rate. Since the ETH tokens are in digital currency, we also observe those values in U.S. Dollars (USD). Therefore, we extracted the Tether or USDT value (*value_in_usdt*) feature from Gitcoin issues to refer to the bounty value in USD since the USDT can imply a stablecoin of the U.S. Dollars (Grobys et al., 2021). Fig. 12 shows the success of the issues with USDT token, in which we found that those values corresponded to the values in ETH. Moreover, Fig. 13 shows the average token value
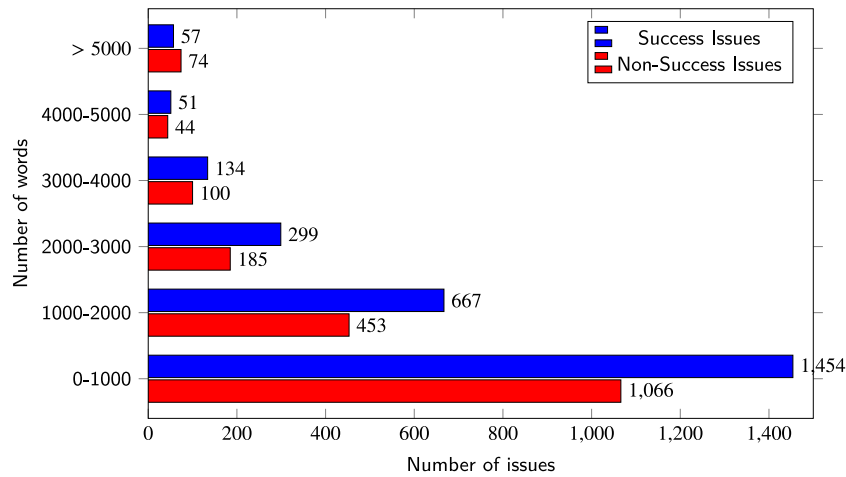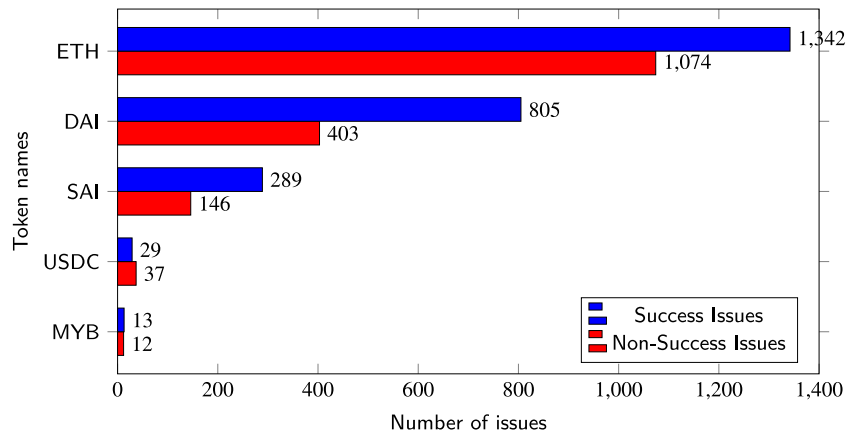
Fig. 9. Length of description.
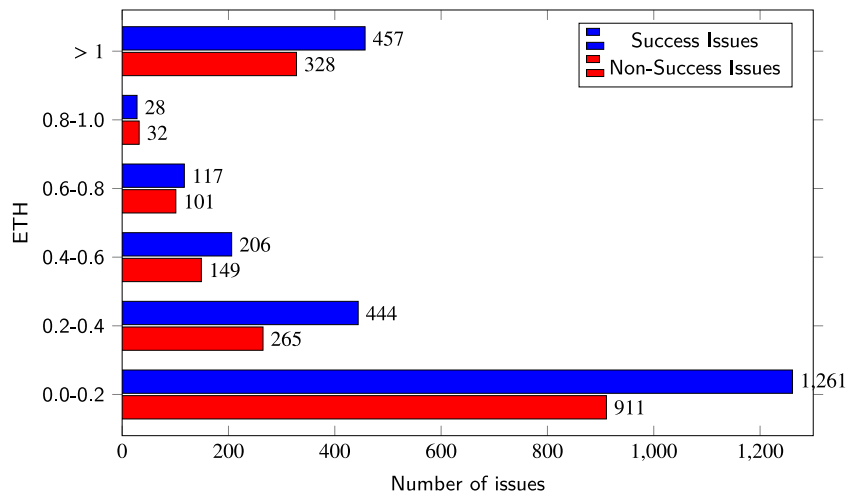

Fig. 10. Token names.

Fig. 11. ETH bounty value.

in USD (*token_value_in_usd*) over time across all token names. It reflects the market values of various cryptocurrencies. It is observed that the majority of tokens experienced a decline in value during the year 2019. This downturn was followed by a notable recovery and upward trend in token values during the year 2021.

4.2.3. The changes of bounty values

The issue owners can determine whether to increase or decrease a bounty value for their issues, such as attracting contributors to work on issues. We extract two features related to the changes in bounty values: the number of times that bounty has been increased (*increased_bounty_times*) and the total bounty
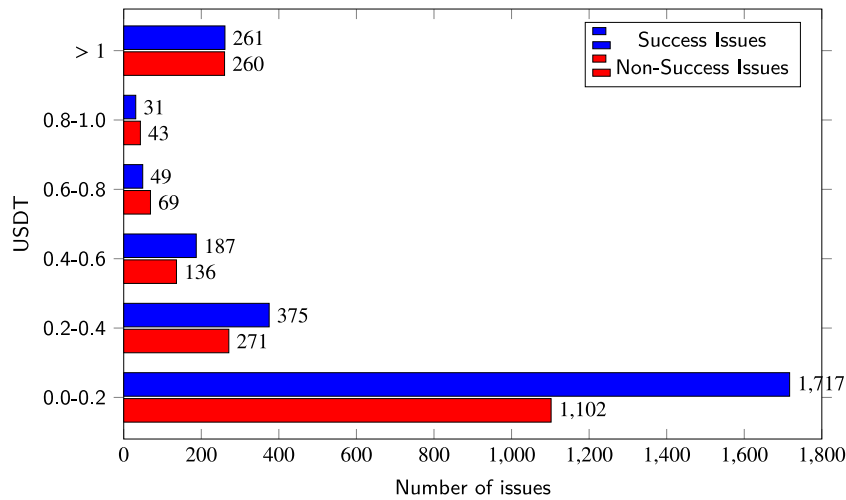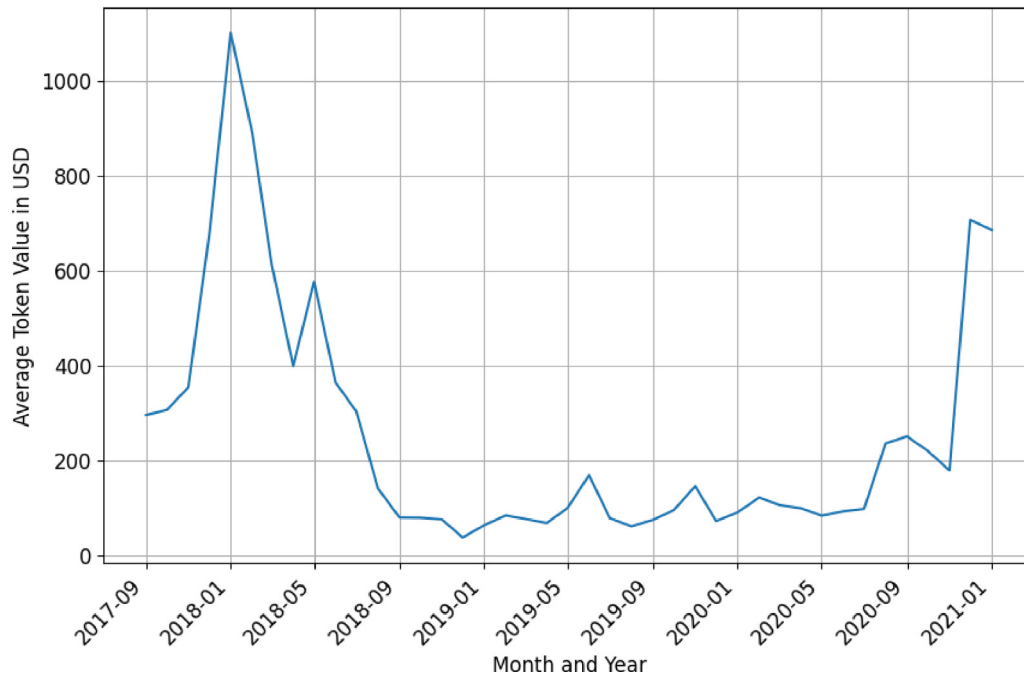
Fig. 12. USDT bounty value.

Fig. 13. Average token value in USD over time across all token names.

value that has been changed (*changed_bounty_value*). The former explains how frequently an issue has been increased by an issue owner. The latter is calculated based on the changed bounty value based on the difference between the most recent value and the original value, i.e., the value at the first time that bounty has been proposed. This feature is to observe how much issue owners have to increase their bounty values to attract contributors. We found that 63% of those issues having their bounty value increased were success issues. However, the bounty values have never been changed in most of the issues.

4.3. Activity related features

The activity-related features indicate activities that occur with an issue, e.g., the number of interests and the number of activities.

4.3.1. Number of fulfillments
The number of fulfillments (*number_of_fulfillments*) indicates the number of participants who submitted the work to issue

owners. We found that 67.8% of total issues contain at least one fulfillment, and the highest number is 156 participants. Moreover, the number of issues with a large number of fulfillments also has higher success issues.

4.3.2. Number of interests
The number of interests (*number_of_interests*) indicates the number of participants who are interested in working on issues. From our investigation, only 19.3% of issues have zero interests, while 3,689 issues (80.7%) have at least one interest from contributors.

4.3.3. Number of activities
The number of activities (*number_of_activities*) indicates the total number of activities that occurred in an issue. We extracted the activities from Gitcoin's issue change log. Those activities include proposing bounties, increasing bounty values, approving candidates, and making submissions.
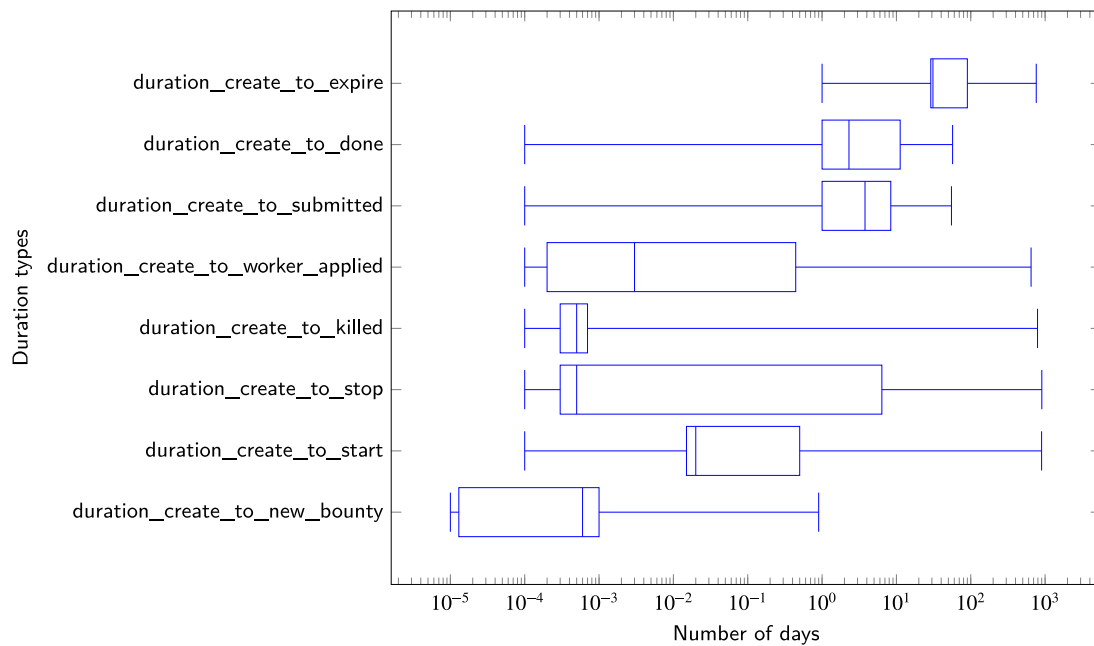
Fig. 14. The duration from issue creation to each stage.

4.3.4. Number of users who interact with an issue

This feature (*number_of_user_in_activities*) is the total number of users who perform actions on an issue. On average, there are 5 users who interact with an issue. The highest number of users that interact with one issue is 170 users.

4.3.5. The first activity type occurred on an issue

This feature *firstAct_activity_type* indicates the activity type that occurred in an issue after issue creation. We found that proposing bounty, start working, and worker applied are the top three types that appeared as the first activity.

4.3.6. The last activity type occurred on an issue

This feature *lastAct_activity_type* captures the last activity type that occurred on an issue. Although we found that the majority type is the work submission activity (*work_submitted*) which indicates the resolving of issues, we have noted that the bounty may not be paid, which causes a non-success issue. We found that over 300 issues have been marked as submitted, but the bounties were not paid.

4.4. Duration-related features

We determine the duration from the issue creation time to each stage of issues (i.e., issue status) to investigate the relationships between durations and the issue-addressing outcome. We extract eight duration-related features.

Fig. 14 shows the extracted duration-related features. We found that issue owners usually add bounty rewards 10 min after the issue creation (*duration_create_to_new_bounty*). In addition, contributors mostly apply to work on an issue on day six after issues were created to apply to issues (*duration_create_to_worker_applied*) and spent, on average, eight days to resolve issues.

5. Feature and correlation analysis

Fig. 15 shows the number of issues in each experience level and each bounty type. We found that the number of issues that are reequired the experience level of *Intermediate* is the majority in several bounty types such as Improvement, Bug, and Feature.

Therefore, this experience level is appropriate for various contributors. In addition, we found that the intermediate-level issues mostly relate to language translation, data migration/integration, and developing features in open-source projects. For the *Advanced* level, the issues are mostly related to complex development tasks which require contributors that are proficient in applying technical knowledge since they need to highly understand the complication of work to implement the work successfully. In addition, we investigate the relationship between the different experience levels and their duration. We found that, on average, the *Intermediate* level issues took five days to find a contributor while the other levels took six days (Fig. 16). The resolution duration of the majority of issues at all levels is between six to eight days.

We then apply the Spearman Rank Correlation technique (Liu, 2010) to determine the correlation among the features. To account for multiple comparisons, we adjust the significance level using the Bonferroni correction. In this study, our focus is to investigate the correlation between each pair of features. The null hypothesis to be rejected is that there is no correlation between the two features being compared. If the adjusted p-value is less than the significance level, we reject the null hypothesis and conclude that there is a statistically significant correlation between the two features. The pairs of features with a strong correlation and a statistically significant p-value ($p < 0.05$/number of comparisons) are reported as follows.

Positive correlation between Project length and Experience level

The correlation between *Project length* and *Experience level* is a moderately positive correlation ($p < 0.05$/number of comparisons). From our investigation, we found the majority of issues that requires the beginner and intermediate experience levels are expected to be resolved within hours and days, while the issues that require the advanced level are mostly expected to be resolved in the length of months.

Positive correlation between the number of times that bounty has been increased and the total bounty value that has been changed The number of times that bounty has been increased and the total bounty value that has been changed is positively correlated. This indicates that the higher number of times that issue owner increases bounty, the higher the bounty value added.
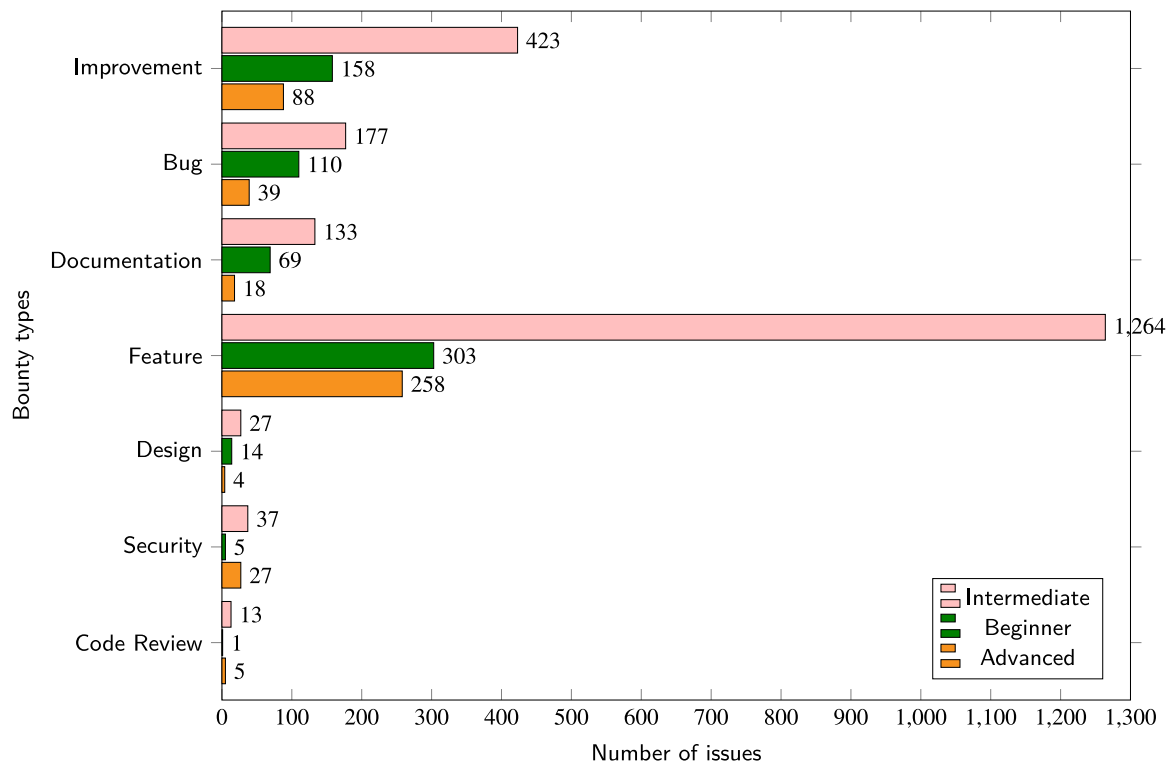
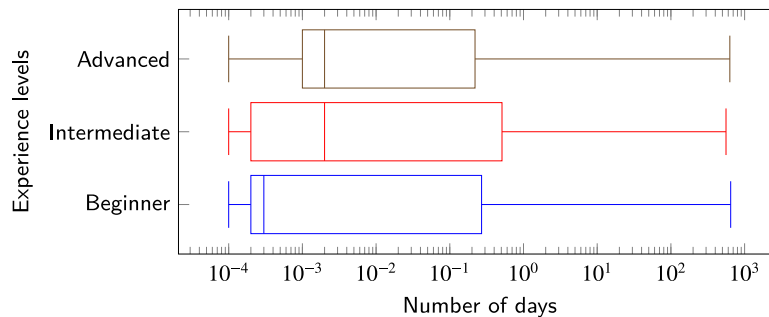**Fig. 15.** The number of issues in each experience level and each bounty type.

Fig. 16. The duration to find a contributor.

6. Identifying features correlated with the outcome of the Gitcoin bounty issue

To identify a set of features that strongly correlate with the outcome of the Gitcoin bounty issue, we adopt two machine learning techniques to train classifiers to determine whether an issue will be a success or a non-success issue. Firstly, we adopt Random Forests (RF) (Breiman, 2001) to train a classifier. By using RF, we can also identify feature importance which can determine the strength of association between features and the target variable (i.e., successful issue resolution). RF uses the *Gini* impurity measure to calculate the feature importance to explain to determine the significance of each feature in determining the output. Since RF provides an interpretable classification in terms of the impact of each feature on the outcome, this method has been applied in various empirical studies, e.g., Laaber et al. (2021) and Tagra et al. (2022). In addition to Random Forests, this study employs Logistic Regression Modeling (LG) and removing multi-collinearity (Harrell, 2015), as seen in previous empirical studies in software engineering, e.g., Thongtanunam et al. (2017). In LG, a model derives a set of coefficients that represent the relationship between each feature and the binary outcomes (i.e., success or

non-success) The coefficients can be interpreted as the change in the log-odds of the outcome as in a change of the feature variable. In terms of interpretability, we then apply the Wald test statistic to measure the strength and direction of the relationship between each feature and the outcomes.

We split the dataset into two parts: 70% for the training set and 30% for the test set. We also applied the stratified sampling technique (Ye et al., 2013) to preserve the proportion of the target variable in the training and test data set and applied the bootstrap sampling technique to overcome the over-fitting problem. Table 6 shows the number of issues in our dataset. In our study, we conduct two different experiments that aim to identify the importance of features in two scenarios by varying a set of features that are used in model training.

In the first experiment (Setting 1), we used all extracted features (Table 2) in model training to identify important features among all features that characterized the bounty issues. Since we also aim to provide suggestions to the issue owners when they create a bounty issue. In the second experiment (Setting 2), we thus used only the features that can be manipulated by the issue owners at their issue creation time. We can then identify a set of important features at the issue creation time.

Table 6
Experimental setting.

	Success issue	Non-success issue
Training set	1,861	1,348
Test set	797	578

Table 7
Evaluation results of each experimental setting using Random Forests (RF), Logistic Regression (LG), and Baseline.

	Precision	Recall	F1	Acc.
RF:Setting 1	0.99	0.98	0.99	0.99
RF:Setting 2	0.85	0.91	0.88	0.86
LG:Setting 1	0.77	0.77	0.77	0.77
LG:Setting 2	0.70	0.70	0.70	0.70
Baseline	0.51	0.50	0.50	0.58

We then discuss the top important features. Note that we replace the values of *project_length* and *experience_level* with the ordinal values to reflect their meaning. For example, the *experience_level* was mapped as follows: 1 for *Beginner*, 2 for *Intermediate*, and 3 for *Advanced*.

Table 7 shows the performance evaluation in terms of precision, recall, F1-score, and accuracy of the models trained from the two settings using Random Forest (RF) and Logistic Regression (LG). The results show that, in both settings from the two models, it can accurately classify the issue outcomes by achieving over 70% in all measurements. We however acknowledge the lower performance of logistic regression (LG) compared to Random Forests (RF). However, it is important to consider that LG and RF offer different advantages in terms of model interpretability and feature importance identification. While the importance of features in a Random Forest model helps identify which features have the most predictive power, LG models provide greater interpretability due to their linear nature. The coefficients in LG allow us to analyze the impact of each feature on the output, providing valuable insights into the relationship between features and the target variable. By combining insights from both models, we can achieve a more comprehensive understanding of the data and uncover underlying patterns. Moreover, both RF and LG also identify a common set of important features such as project length and experience level. In addition, to address concerns about the models' acceptance, we have included a baseline method using Zero-Rule as a sanity check. Zero-Rule predicts the most frequent label in the training set, serving as a simplistic benchmark. Both LG and RF outperform this baseline method, indicating their superiority in predictive performance. Therefore, it is reasonable to use these models for our interpretation and analysis. By considering the interpretability strengths of LG and the feature importance of RF, and validating their performance against a baseline, we ensure a robust evaluation of the models' effectiveness and reliability for our study.

6.1. Analysis of the feature importance from Random forests

Table 8 shows the list of the top ten important features from the first and second settings, respectively. The top three highest important features of the first setting are *duration_create_to_done*, *duration_create_to_submitted*, and *lastAct_activity_type_work_done*. As can be seen that those three most important features reflect a common scenario in software development that a proper time duration spent on an issue is critical to the success of issue resolution. However, it is worth noting that these features can only be gathered when the issues are closed.

Table 8
Feature importance from the two experimental settings using Random Forests.

Features	Importance values
Setting 1:	
duration_create_to_done	0.229
duration_create_to_submitted	0.076
lastAct_activity_type_work_done	0.062
number_of_fulfillments	0.048
duration_create_to_killed	0.044
token_value_in_usdt	0.042
number_of_activities	0.035
lastAct_activity_type_killed_bounty	0.034
number_of_interests	0.034
Setting 2:	
description_length	0.143
duration_create_to_new_bounty	0.139
value_in_usdt	0.126
value_in_usdt_now	0.102
duration_create_to_expire	0.097
value_in_eth	0.089
value_in_token	0.052
experience_level_code	0.037
project_length_code	0.037
bounty_type_Feature	0.016

Table 9
The correlation of the features with the issue outcomes using the Point Biserial Correlation Coefficient.

Features	Correlation	p-value
token_name_DAI	0.110 ↑	<0.001
bounty_type_Documentation	0.097 ↑	<0.001
contribution_type_contest	0.062 ↑	<0.001
token_name_SAI	0.055 ↑	<0.001
bounty_type_NA	0.049 ↑	0.001
bounty_type_Security	−0.070 ↓	<0.001
experience_level_code	−0.070 ↓	<0.001
token_name_ETC	−0.075 ↓	<0.001
token_name_DOT	−0.089 ↓	<0.001
contribution_type_cooperative	−0.110 ↓	<0.001

In the second setting, we thus use the features that can be manipulated or controlled by the issue owner, *description_length*, *duration_create_to_new_bounty*, and *value_in_usdt* are the top three most important features which must be taken into account when creating issues. In particular, we found that the number of days from issue creation until the first proposed bounty is the second most important feature that potentially determines the issue outcome. This finding corresponds with the result reported in Zhou et al. (2020b), which also found that the earlier bounty proposed, the higher likelihood of being addressed. We then further investigate the correlation between these features.

We apply the Point Biserial Correlation Coefficient (Bonett, 2020) with Bonferroni correction to measure the correlation between the numerical features and the binary outcomes of issues (i.e., success and non-success issues). Table 9 lists the features used in Setting 2 along with their correlation coefficient and *p*-value. Significant correlations at a statistically strong level ($p < 0.05$/number of features) are indicated in the table. We used the features in Setting 2 for our analysis because they are manipulable and more useful to practitioners.

The studying result suggests that the features "token_name _DAI" and "bounty_type_Documentation" have the strongest positive correlation with issue outcomes, while the features "contribution_type_cooperative" and "token_name_DOT" have the strongest negative correlation with issue outcomes. The results are statistically significant, with all p-values falling below the significance threshold of 0.05 after the Bonferroni correction. It shows that the type of token used and the type of bounty offered could be strongly correlated with issue success. The results suggest that

Table 10

Coefficient and *p*-value estimates from logistic regression analysis of issue success.

Feature	Coefficient	p-value
token_name_DAI	0.619 ↑	<0.001
token_name_ETH	0.130 ↑	0.015
project_length_code	−0.008 ↓	0.015
bounty_type_Feature	−0.039 ↓	0.589
bounty_type_NA	−0.068 ↓	0.582
contribution_type_traditional	−0.132 ↓	0.009
experience_level_code	−0.159 ↓	<0.001

the token "DAI" is positively correlated with issue success, potentially due to the stability of the coin. Regarding the contribution types, "Documentation" bounties are positively correlated with issue success, while "Security" bounties are negatively correlated. This may reflect the fact that documentation bounties are easier to define and assess, while security bounties require more expertise to evaluate and address.

6.2. Analysis of the feature's coefficients from logistics regression

Table 10 presents the results of a logistic regression analysis of issue success. The coefficients of each feature indicate the direction of the correlation with issue success, while the p-values provide information on the statistical significance of each correlation. The analysis shows that issues associated with the tokens "DAI" and "ETH" are more likely to be successful. The strong correlation between "token_name_DAI" and issue success is further supported by the results of the point biserial correlation analysis. Additionally, shorter project length, traditional contribution type, and lower experience level are all negatively correlated with issue success. This suggests that shorter projects may increase the chance of issue success, and the issues that require less experienced contributors may be more successful. Furthermore, the contest contribution type is preferred over cooperative and traditional.

6.3. The comparison study between Gitcoin and Bountysource

This study compares some characteristics of bounty issues from Gitcoin and Bountysource, with the aim of providing useful recommendations to bounty issue funders and contributors on selecting the most appropriate platform to meet their needs. Specifically, we examine three aspects of these platforms: the programming languages used, the topics of issues posted, and the value of bounties offered. To conduct this study, we utilized the Bountysource dataset provided by Zhou et al. (2020b), which includes valuable information on bounty issues in the Bountysource platform. In addition, we collected supplementary data, such as the topics of each issue corresponding to the issue key provided in the dataset.

Our first comparison is to investigate the programming languages used in bounty issues across both platforms. To identify the programming languages used in bounty issues across both platforms, we collected language-related tags from the GitHub repositories associated with each bounty issue in both datasets. Table 11 shows the percentage distribution of programming languages used in bounty issues on both Gitcoin and Bountysource platforms. The results show the differences in the distribution of programming languages between the two platforms. JavaScript is the most commonly used language on both platforms, with a distribution of 43.90% on Gitcoin and 17.50% on Bountysource. This could be attributed to the popularity of JavaScript in web development, which may be a primary focus of the bounty issues on both platforms. In addition, Bountysource has a higher percentage

Table 11

Percentage distribution of programming languages in Bountysource and Gitcoin.

Bountysource	%	Gitcoin	%
JavaScript	17.50	JavaScript	43.90
C++	14.20	TypeScript	12.50
Python	12.40	Go	9.50
PHP	9.10	Python	7.70
C#	7.60	Clojure	4.00
C	6.60	Rust	3.30
Java	6.10	Solidity	3.10
Ruby	5.40	C++	2.70
TypeScript	3.80	HTML	2.00
Go	2.40	CSS	1.60

Table 12

Top 30 topics on Gitcoin and Bountysource platforms.

Bountysource's top 30 topics
Hacktoberfest, JavaScript, Python, Game, C, Linux, PHP, Game engine, C++, Cross-platform, RTS, Real-time strategy, OpenRA, Command and Conquer, Strategy game engine, Backup, Encryption, C#, .NET, Python 3, SSH, Engine, Deduplication, Dedupe, Cython, Compression, BorgBackup, Tiberian Dawn, Red Alert, Java.

Gitcoin's top 30 topics
Ethereum, Blockchain, Hacktoberfest, Gitcoin, Bounties, ETH, Open source, Web3, Solidity, Wallet, Smart contracts, Dapp, Cryptocurrency, React Native, DeFi, Oracle, Android, Mobile, Bitcoin, iOS, React Native app, Messenger, ClojureScript, Clojure, Reagent, ReactNative, Re-frame, React, TypeScript, Blockchain technology.

of C++ and Python, which is also widely used in general software development. However, the difference in language distribution may also reflect the types of projects and bounties available on each platform. Gitcoin, for example, has a focus on blockchain applications, which may explain the higher percentage of TypeScript and Solidity, which are both used in the development of smart contracts and other blockchain-related applications.

In the second comparison aspect, we focus on issue topics across the two platforms. Specifically, we analyzed the tags associated with bounty issues to identify the most commonly used topics on each platform. Table 12 shows a comparison of the top 30 topics identified from the bounty issues from Gitcoin and Bountysource platforms. The results show that Gitcoin focuses on blockchain-related topics such as Ethereum, Blockchain, ETH, Solidity, and DeFi, which are not represented in Bountysource. On the other hand, Bountysource shows more diversity in terms of topics, with programming languages, game-related topics, and backup-related topics appearing more frequently. This difference in focus may reflect the different goals and priorities of the two platforms and suggests that issue funders and contributors may need to consider these differences when selecting a platform.

The last comparison focuses on the bounty value of issues on the two platforms. We analyze the distribution of bounty value for the successful issues of both platforms. Fig. 17 shows the cumulative distribution of USDT value for successful issues on Gitcoin and Bountysource platforms. By comparing the curves, it shows the differences in the distribution of USDT value for successful bounty issues on each platform. We can notice that the cumulative distribution for Gitcoin is generally lower than the cumulative distribution for Bountysource. This suggests that successful bounty issues on Gitcoin tend to have lower USDT values compared to successful bounty issues on Bountysource.

Moreover, since the dataset from Gitcoin (from 2017 to 2020) is newer than the Bountysource dataset from Zhou et al. (2020b) (from 2012 to 2017), there might be differences in terms of the topics, bounty values, and languages used in the bounties. We have performed an additional study to compare the topics of
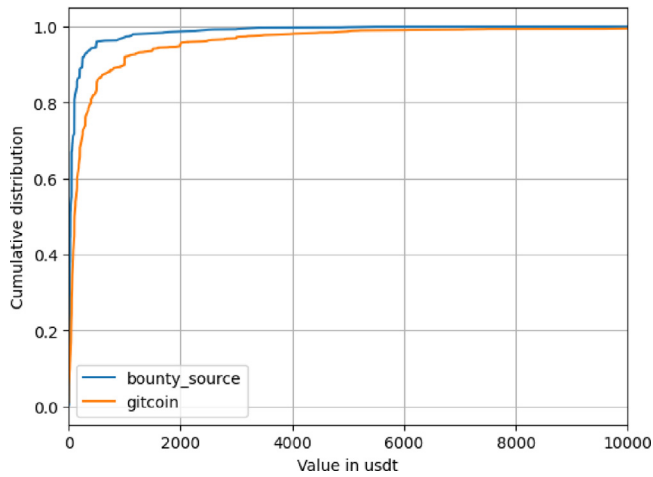
Fig. 17. Comparison of cumulative distribution of USDT value for Gitcoin and Bountysource bounty issues.

Fig. 18. The distribution of description length grouped by the success and non-success issues.

Table 13
The bounty value (in USD) of success issues in each feature.

Group		Value in USD		
		Min	Mean	Max
Project types	Traditional	0.00	148.96	736.48
	Cooperative	0.01	170.97	579.20
	Contest	0.00	118.45	737.85
Experience levels	Beginner	0.00	79.64	500.00
	Intermediate	0.00	155.50	736.14
	Advanced	0.10	200.51	736.48
Bounty types	Feature	0.00	146.03	700.00
	Improvement	0.00	146.14	691.35
	Bug	0.00	94.73	688.72

the latest issues of Gitcoin and BountySource. Since the API of Bountysource is no longer available,[14] we performed a manual comparison by going through the 20 latest issues on Gitcoin and Bountysource. To get the topic of the issues, we relied on multiple approaches. First, we checked the topics or labels assigned to Gitcoin or Bountysource bounties. Second, we checked the labels given to the bounties' associated GitHub issues. Third, we also checked the topics given to the GitHub project containing the issue. After analyzing the collected topics, we observed that 10 Gitcoin bounties contain blockchain-related issues compared to 3 bounties in Bountysource.

6.4. Discussion

According to our feature analysis, correlation analysis, and feature importance analysis, we can discuss them as follows.

- The duration in each state of an issue potentially determines the issue outcome. We found that the duration-related features, especially the duration between issues creation until the work is done, are highly correlated with the success of issues. On average, those success issues usually take fourteen days to be resolved, while it is only one day for those non-success issues. In addition, the correlation analysis with the issue outcome using Point Biserial shows a strong positive correlation which indicates that the longer time spent working on issues can potentially lead to the issue being resolved and successfully have been paid. This suggests that issue funders should carefully determine the appropriate project length when creating an issue. This should reflect the estimated duration required for issue resolution. By providing an accurate estimate, funders enable contributors to assess their ability to deliver as expected.
- The length of the issue description appears to have a significant impact on the success of an issue. Fig. 18 shows the number of characters of issue descriptions. As can be seen, on average, the description of the success issues is slightly longer than those of non-success. In addition, issues that require advanced-level experience usually have longer issue descriptions. Therefore, practitioners may benefit from spending the time and effort to write clear and detailed issue descriptions that clearly communicate the expectations and requirements for issue resolution.
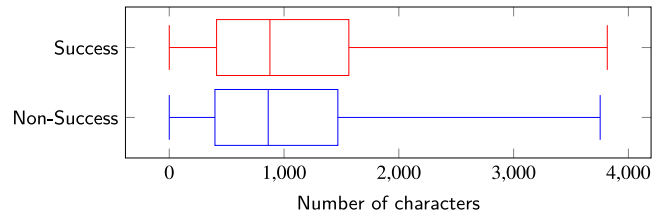
- We found that contribution type, experience level, and bounty type strongly correlate with the outcome of issues. We then provide the bounty value of success issues in each group in Table 13, which can help issue owners initiate the bounty value of their issues. In addition, our analysis suggests that certain token types, such as DAI and ETH, are more commonly used in successful bounty issues, indicating that these tokens may be more attractive to contributors.
- We found that there is a correlation between the experience level of contributors and the outcome of issues. Issues that require beginner and intermediate experience levels have a higher percentage of success compared to non-success issues. However, those issues with an advanced experience level have a higher number of non-successful outcomes. The correlation between experience level and issue outcome is negative, suggesting that the scope of work for an issue should be clear and not too complicated for the contributor to complete.
- Our study found that although the actual token value was identified as an important feature in predicting the outcomes of bounty issues in RF, it did not show a strong statistical correlation with the success issues. It is important to note that the cryptocurrency market is known for its high volatility, and the value of cryptocurrencies can fluctuate significantly. To gain further insights, we specifically focused on Ethereum (ETH), as it was the most commonly assigned token to the bounty issues in our dataset. Fig. 19 shows the actual value of ETH and the cumulative count of success issues, grouped by month and year according to the timeline. The figure shows the fluctuation in ETH value over time. Notably, during 2019, the value of ETH experienced a decline. We observed that the ETH value was particularly high in early 2018, coinciding with a high number of successful issues. However, as time progressed, both the value of ETH and the count of success issues exhibited a declining trend. These findings suggest a potential relationship between the value of cryptocurrency and the success of bounty issues.
- Our comparative study confirms that Gitcoin's topics primarily revolve around blockchain-related issues. Consequently, the variations in programming languages can

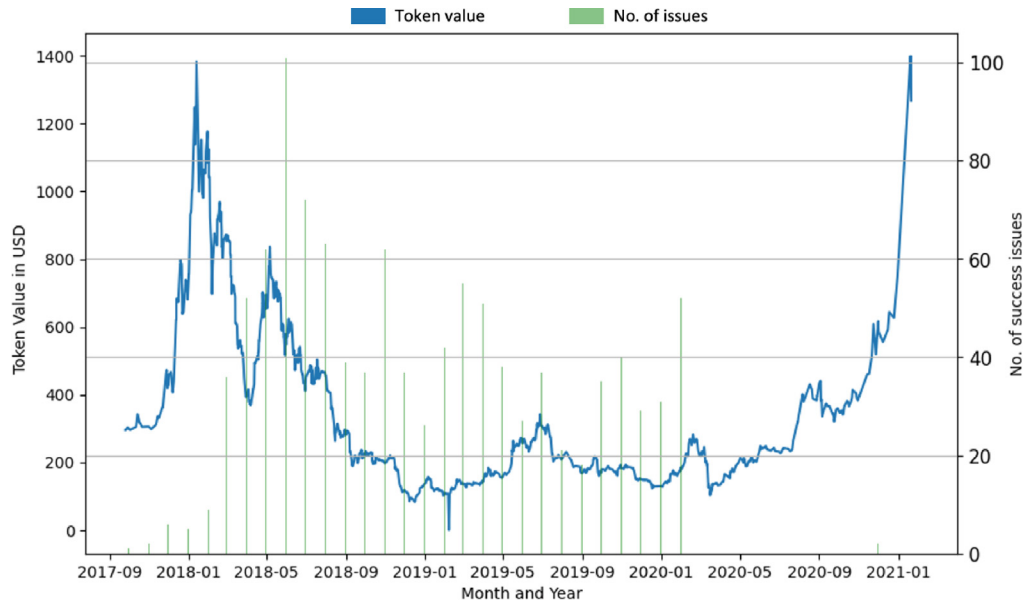---

14 We checked on 20th June 2023.

Fig. 19. The actual value of ETH and the cumulative count of success issues.

be considered a natural consequence of this focus, while the value of the bounty is comparatively lower than that of Bountysource. This could suggest issue owner to select a platform that fit their requirements. In addition, this observation suggests that a bounty platform dedicated to a specific type of project may effectively attract attention from contributors.

7. Threats to validity

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

7.1. External validity

Regarding the generalizability of the work, our study focuses on only bounties issues from the Gitcoin platform. The results from our study might not be generalizable to other platforms. Therefore, our future work relates to studying issue reports from other platforms and different project settings that their bounty rewards are related to cryptocurrencies.

7.2. Internal validity

We consider closed issues and examine only those that have been created and subsequently resolved. The data used in this analysis is sourced from the issues in Mainnet. These may have threats to the internal validity of the study. Nonetheless, we mitigate the threats by ensuring the validity of all extracted features through a validation process (e.g., manual validation process). The dataset from Gitcoin and the Bountysource dataset from Zhou et al. (2020b) overlap only in 2017. Thus, there might be a threat to validity when comparing them. We mitigate this threat by performing an additional manual comparison of the 20 latest issues in Gitcoin and Bountysource.

7.3. Construct validity

In order to mitigate threats to construct validity, we apply the Bonferroni correction (i.e., p-value adjustment) to our correlation analysis among the extracted features and between the features and outcome of the issues. This technique helps to eliminate bias and improve the reliability of our results. We thus used statistical testing to determine the significance of any correlations found, further mitigating any potential threats to the construct validity of our experiment. In addition, to ensure the robustness of our experiments, we use a combination of techniques to analyze the correlation between the extracted features and the outcome of the issues. Specifically, we apply the Point Biserial Correlation Coefficient, which is a statistical method specifically designed for the correlation analysis of binary variables.

7.4. Conclusion validity

We mitigate threats to conclusion validity by taking a rigorous and cautious approach when drawing conclusions based on the extracted features from the studied platform. This includes carefully limiting our conclusions to only those observations and insights that can be directly supported by the data extracted from the platform.

8. Conclusion and future work

The development of open source software (OSS) projects is heavily progressed by contributions from volunteering developers. Bounty rewards in different schemes have been used to motivate participation in the OSS development. Gitcoin proposes a platform that allows issue owners to create a bounty reward using cryptocurrencies solely. The understanding of the phenomenon related to the use of bounty rewards in OSS projects promotes benefits yielded from this reward mechanism. We thus perform a study on over 4,000 Gitcoin issues by categorizing those issues into four main aspects, including primitive attributes, bounty value-related, activity-related, and duration-related features. Using statistical and machine learning techniques, we identify factors that influence the outcome of the issues. It could

be served as a guide for issue owners to increase the efficiency of their bounty rewards. We acknowledge the high volatility of the cryptocurrency market, and it is possible that fluctuations in cryptocurrency value may influence the success or failure of bounty issues. Our additional investigation into the actual value of cryptocurrency and its correlation with bounty success demonstrates the potential for a relationship. However, further study is required as part of our future work to better understand this relationship.

CRediT authorship contribution statement

Morakot Choetkiertikul: Conceptualization, Methodology, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing Supervision, Project administration, Funding acquisition. **Arada Puengmongkolchaikit:** Software, Investigation, Data curation, Writing – original draft, Visualization. **Pandaree Chandra:** Software, Investigation, Data curation, Writing – original draft, Visualization. **Chaiyong Ragkhitwetsagul:** Investigation, Writing – original draft, Writing – review & editing, Visualization. **Rungroj Maipradit:** Conceptualization, Data curation, Investigation, Visualization. **Hideaki Hata:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Thanwadee Sunetnanta:** Writing – review & editing, Project administration. **Kenichi Matsumoto:** Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Morakot Choetkiertikul reports financial support was provided by Office of the Permanent Secretary, Ministry of Higher Education, Science, Research and Innovation (OPS MHESI), Thailand.

Acknowledgments

References

Arcuri, A., Briand, L., 2014. A Hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. Softw. Test. Verif. Reliab. 24 (3), 219–250. http://dx.doi.org/10.1002/stvr.1486.

Atiq, A., Tripathi, A., 2014. Monetary rewards for open source software developers. In: Proceedings of the 35th International Conference on Information Systems 'Building a Better World Through Information Systems' (ICIS 2014). pp. 1–10.

Bonett, D.G., 2020. Point-biserial correlation: Interval estimation, hypothesis testing, meta-analysis, and sample size determination. Br. J. Math. Stat. Psychol. 73 (S1), 113–144. http://dx.doi.org/10.1111/bmsp.12189.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Chen, W., Zhang, T., Chen, Z., Zheng, Z., Lu, Y., 2020. Traveling the token world: A graph analysis of Ethereum ERC20 token ecosystem. In: Proceedings of 2020 International World Wide Web Conference (WWW 20). pp. 1411–1421. http://dx.doi.org/10.1145/3366423.3380215.

Choi, N., Chengalur-Smith, I., Whitmore, A., 2010. Managing first impressions of new open source software projects. IEEE Softw. 27 (6), 73–77. http://dx.doi.org/10.1109/MS.2010.26.

Coelho, J., Valente, M.T., Silva, L.L., Hora, A., 2018. Why we engage in FLOSS: Answers from core developers. In: Proceedings of 2018 ACM/IEEE 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). pp. 114–121. http://dx.doi.org/10.1145/3195836.3195848.

Finifter, M., Akhawe, D., Wagner, D., 2013. An empirical study of vulnerability rewards programs. In: Proceedings of the 22nd USENIX Security Symposium (USENIX Security 13). pp. 273–288.

Grobys, K., Junttila, J.-P., Kolari, J.W., Sapkota, N., 2021. On the stability of stablecoins. J. Empir. Financ. 64 (2021), 207–223. http://dx.doi.org/10.2139/ssrn.3764457.

Harrell, F.E., 2015. Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. In: Springer Series in Statistics, Springer International Publishing, URL: https://books.google.co.th/books?id=sQ90rgEACAAJ.

Hata, H., Guo, M., Babar, M.A., 2017. Understanding the heterogeneity of contributors in bug bounty programs. In: Proceedings of 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). pp. 223–228. http://dx.doi.org/10.1109/ESEM.2017.34.

Kanda, T., Guo, M., Hata, H., Matsumoto, K., 2017. Towards understanding an open-source bounty: Analysis of bountysource. In: Proceedings of 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, pp. 577–578. http://dx.doi.org/10.1109/SANER.2017.7884685.

Krishnamurthy, S., Ou, S., Tripathi, A.K., 2014. Acceptance of monetary rewards in open source software development. Res. Policy 43 (4), 632–644. http://dx.doi.org/10.1016/j.respol.2013.10.007.

Laaber, C., Basmaci, M., Salza, P., 2021. Predicting unstable software benchmarks using static source code features. Empir. Softw. Eng. 26 (6), http://dx.doi.org/10.1007/s10664-021-09996-y.

Lan, H., 2019. A crowdsourcing quality prediction model based on random forests. In: Proceedings of 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS). IEEE, pp. 315–319.

Li, Z., Lu, Q., Chen, S., Liu, Y., Xu, X., 2019. A landscape of cryptocurrencies. In: Proceedings of 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE, pp. 165–166. http://dx.doi.org/10.1109/BLOC.2019.8751469.

Liu, D., 2010. A spearman correlation coefficient ranking for matching-score fusion on speaker recognition. In: Proceedings of IEEE Region 10 International Conference (TENCON 2010). IEEE, pp. 736–741.

Luna, D., Allodi, L., Cremonini, M., 2019. Productivity and patterns of activity in bug bounty programs: Analysis of HackerOne and google vulnerability research. In: Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019). pp. 1–10. http://dx.doi.org/10.1145/3339252.3341495.

Nakasai, K., Hata, H., Matsumoto, K., 2019. Are donation badges appealing?: A case study of developer responses to eclipse bug reports. IEEE Softw. 36 (3), 22–27. http://dx.doi.org/10.1109/MS.2018.2874568, arXiv:1803.04129.

Tagra, A., Zhang, H., Rajbahadur, G.K., Hassan, A.E., 2022. Revisiting reopened bugs in open source software systems. Empir. Softw. Eng. 27 (4), http://dx.doi.org/10.1007/s10664-022-10133-6, arXiv:2202.08701.

Thongtanunam, P., McIntosh, S., Hassan, A.E., Iida, H., 2017. Review participation in modern code review. Empir. Softw. Eng. 22 (2), 768–817. http://dx.doi.org/10.1007/s10664-016-9452-6.

Titov, V., Uandykova, M., Litvishko, O., Kalmykova, T., Prosekov, S., Senjyu, T., 2021. Cryptocurrency Open Innovation Payment System: Comparative Analysis of Existing Cryptocurrencies. J. Open Innov.: Technol. Mark. Complex. 7 (1), http://dx.doi.org/10.3390/JOITMC7010102.

Wang, S., Chen, T.H., Hassan, A.E., 2018. Understanding the factors for fast answers in technical Q&A websites: An empirical study of four stack exchange websites. In: Proceedings of 2018 ACM/IEEE 40th International Conference on Software Engineering Understanding (ICSE 18). pp. 1552–1593. http://dx.doi.org/10.1007/s10664-017-9558-5.

Ye, Y., Wu, Q., Zhexue Huang, J., Ng, M.K., Li, X., 2013. Stratified sampling for feature subspace selection in random forests for high dimensional data. Pattern Recognit. 46 (3), 769–787. http://dx.doi.org/10.1016/j.patcog.2012.09.005.

Zhao, M., Grossklags, J., Chen, K., 2014. An exploratory study of white hat behaviors in a web vulnerability disclosure program. In: Proceedings of 2014 ACM Workshop on Security Information Workers (SIW 14). pp. 51–58. http://dx.doi.org/10.1145/2663887.2663906.

Zhou, J., Wang, S., Bezemer, C.P., Hassan, A.E., 2020a. Bounties on technical Q&A sites: a case study of stack overflow bounties. Empir. Softw. Eng. 25 (1), 139–177. http://dx.doi.org/10.1007/s10664-019-09744-3.

Zhou, J., Wang, S., Bezemer, C.P., Zou, Y., Hassan, A.E., 2020b. Studying the association between bountysource bounties and the issue-addressing likelihood of GitHub issue reports. IEEE Trans. Softw. Eng. 47 (12), 2919–2933. http://dx.doi.org/10.1109/TSE.2020.2974469.

Zhou, J., Wang, S., Zhang, H., Chen, T.H.P., Hassan, A.E., 2021. Studying backers and hunters in bounty issue addressing process of open source projects. Empir. Softw. Eng. 26 (4), 81. http://dx.doi.org/10.1007/s10664-021-09979-z.

Morakot Choetkiertikul is a lecturer at the Faculty of Information and Communication Technology (ICT), Mahidol University, Thailand where he co-founded the Software Engineering Research Unit (SERU). He graduated Ph.D. in computer science from the University of Wollongong (UOW), Australia. His research interests include applying AI solutions to improve software quality and software processes.

Arada Puengmongkolchaikit received the B.S. in Information and Communication Technology from the Faculty of Information and Communication Technology (ICT), Mahidol University.

Pandaree Chandra received the B.S. in Information and Communication Technology from the Faculty of Information and Communication Technology (ICT), Mahidol University.

Chaiyong Ragkhitwetsagul is a lecturer at the Faculty of Information and Communication Technology (ICT), Mahidol University, Thailand. He received the Ph.D. degree in Computer Science at University College London. His research interests include software dependencies, software visualization, code search, code clone detection, and mining software repositories. More info at https://cragkhit.github.io.

Rungroj Maipradit received his PhDand master degrees from the Department of Information Science at the Nara Institute of Science and Technology, Japan. His research interests include machine learning. Maipradit received a B.Eng. in computer engineering from Kasetsart University, Thailand, in 2016.

Hideaki Hata received the Ph.D. degree in information science from Osaka University, Suita, Japan. He is an associate professor with Shinshu University. His research interests include software ecosystems, human capital in software engineering, and software economics. More about him and his work is available online at https://hideakihata.github.io.

Thanwadee Sunetnanta is an assistant professor at the Faculty of Information and Communication Technology (ICT), Mahidol University, Thailand. She received a Ph.D. in distributed software engineering from Department of Computing, Imperial College, London. Her research interests include requirements engineering, software process improvement, qualitative approaches to software quality and secure software engineering.

Kenichi Matsumoto is a professor in the Graduate School of Science and Technology at the Nara Institute of Science and Technology. His research interests include software measurement and software processes. He received a Ph.D. in information and computer sciences from Osaka University. He is a Senior Member of the IEEE and a member of the IEICE and the IPSJ. Contact him at matumoto@is.naist.jp.