The Landscape of Toxicity: An Empirical Investigation of Toxicity on GitHub

JAYDEB SARKER, University of Nebraska at Omaha, USA
ASIF KAMAL TURZO, Wayne State University, USA
AMIANGSHU BOSU, Wayne State University, USA

Toxicity on GitHub can severely impact Open Source Software (OSS) development communities. To mitigate such behavior, a better understanding of its nature and how various measurable characteristics of project contexts and participants are associated with its prevalence is necessary. To achieve this goal, we conducted a large-scale mixed-method empirical study of 2,828 GitHub-based OSS projects randomly selected based on a stratified sampling strategy. Using ToxiCR, an SE domain-specific toxicity detector, we automatically classified each comment as toxic or non-toxic. Additionally, we manually analyzed a random sample of 600 comments to validate ToxiCR's performance and gain insights into the nature of toxicity within our dataset. The results of our study suggest that profanity is the most frequent toxicity on GitHub, followed by trolling and insults. While a project's popularity is positively associated with the prevalence of toxicity, its issue resolution rate has the opposite association. Corporate-sponsored projects are less toxic, but gaming projects are seven times more toxic than non-gaming ones. OSS contributors who have authored toxic comments in the past are significantly more likely to repeat such behavior. Moreover, such individuals are more likely to become targets of toxic texts.

CCS Concepts: • **Software and its engineering → Open source model**; **Programming teams**.

Additional Key Words and Phrases: toxicity, developers' interactions, empirical analysis, pull request

ACM Reference Format:
Jaydeb Sarker, Asif Kamal Turzo, and Amiangshu Bosu. 2025. The Landscape of Toxicity: An Empirical Investigation of Toxicity on GitHub. *Proc. ACM Softw. Eng.* 2, FSE, Article FSE029 (July 2025), 24 pages. https://doi.org/10.1145/3715744

Warning: This paper contains examples of language that some people may find offensive or upsetting.

1 Introduction

In 2023, GitHub, the most popular Open Source Software (OSS) project hosting platform, hosted more than 284 million public repositories [Daigle 2023]. As OSS communities continue to grow, so do the interactions among contributors during various software development activities, such as issue discussions and pull request reviews. While these interactions are crucial to facilitating collaborations among the contributors, a few may take negative turns and cause harm [GitHub, Inc. et al. 2024]. Recent studies have investigated the umbrella of antisocial interactions among OSS developers using various lenses, which include 'toxicity,' [Miller et al. 2022; Raman et al. 2020; Sarker et al. 2023a, 2020, 2023b], 'incivility' [Ferreira et al. 2021], 'destructive criticism' [Gunawardena et al. 2022], and 'sexism and misogyny' [Sultana et al. 2021]. Although these lenses differ, they all share a common attribute: the potential to cause severe repercussions among the participants.

Authors' Contact Information: Jaydeb Sarker, University of Nebraska at Omaha, Omaha, USA, jsarker@unomaha.edu; Asif Kamal Turzo, Wayne State University, Detroit, USA, asifkamal@wayne.edu; Amiangshu Bosu, Wayne State University, Detroit, USA, amiangshu.bosu@wayne.edu.

ACM 2994-970X/2025/7-ARTFSE029
https://doi.org/10.1145/3715744

Consequences of antisocial interactions include stress and burnout [Raman et al. 2020], negative feelings [Egelman et al. 2020; Ferreira et al. 2021; Gunawardena et al. 2022], pushbacks [Murphy-Hill et al. 2022], turnovers of long-term contributors [Author 2014; Diggs 2021; Lawson 2017; X 2021], adding barriers to newcomers' onboarding [Raman et al. 2020], and hurting diversity, equity, and inclusion (DEI) by disproportionately affecting women and other underrepresented minorities [Albusays et al. 2021; Gunawardena et al. 2022; Murphy-Hill et al. 2022; Nafus 2012]. Moreover, the prevalence of antisocial behaviors present substantial challenges to the growth and sustainability of an OSS project.

Therefore, recent Software Engineering (SE) research has focused on characterizing toxicity and other antisocial behaviors and their consequences through surveys, interviews, and qualitative analyses [Egelman et al. 2020; Ferreira et al. 2022, 2021; Gunawardena et al. 2022; Miller et al. 2022]. In a sample of 100 GitHub issue comments, Miller et al. [2022] found entitlement, arrogance, insult, and trolling as the most common forms of toxicity. Destructive criticism is another anti-social behavior found in code reviews [Gunawardena et al. 2022]. Although destructive criticisms are rare, they may have severe repercussions, which include conflicts, demotivation, and even hindering the participation of minorities [Egelman et al. 2020; Gunawardena et al. 2022]. Ferreira et al. [2021]'s study of rejected patches in Linux kernel mailing lists reported frustration, name-calling, and impatience as the most prevalent forms of incivility. Another recent workplace investigation reported inappropriate communication style as the primary cause of incivility [Rahman et al. 2024]. However, many of these studies suffer from limitations such as small sample sizes or narrow focuses on specific projects [Ferreira et al. 2022; Miller et al. 2022], organizations [Egelman et al. 2020; Qiu et al. 2022], or a small developer group [Rahman et al. 2024], which raises questions regarding the external validity of these findings at different contexts. We also lack a quantitative empirical investigation of how various measurable characteristics of project contexts and participants are associated with the prevalence of antisocial behavior, such as toxicity. Such an investigation is necessary to formulate context-aware toxicity mitigation strategies for the broader OSS ecosystem.

In response to this need, we have conducted a large-scale empirical investigation of toxicity during Pull Requests (PRs). We selected PR since it is a crucial mechanism to attract contributions from non-members and facilitate newcomers' onboarding among OSS projects [Gousios et al. 2014]. PRs allow contributors to propose changes, which other community members then review PRs. Due to the interpersonal nature of PR interactions and the potential for dissatisfaction due to unfavorable decisions, PR interactions may raise conflicts and anti-social behaviors. We select the 'toxicity' lens since it has been most widely investigated [Miller et al. 2022; Raman et al. 2020; Sarker et al. 2023a, 2020] and has a reliable automated identification tool [Sarker et al. 2023b], which is a prerequisite for a large-scale empirical investigation. Following the toxicity investigation framework proposed by Miller et al. [2022], we investigate four research questions to characterize: i) the nature of toxicity, ii) projects that had higher toxic communication than others, iii) contextual factors that are more likely to be associated with toxicity, and iv) the participants of toxic interactions, respectively. We briefly motivate each of the research questions as follows.

RQ1: [Nature] *What are the common forms of toxicity observed during GitHub Pull Requests?*

Motivation: Understanding the nature of toxicity may help project maintainers improve guidelines and interventions to foster respectful and constructive interactions. Prior studies have proposed various categorizations of SE domain-specific toxicity [Miller et al. 2022; Sarker et al. 2023b] and incivility [Ferreira et al. 2021]. However, due to sampling criteria used in those studies (e.g., only locked issues, small samples, or a specific project), two key insights remain under-explored: i) whether these studies missed additional forms of toxicity, and ii) how frequently various toxicity categories occur on GitHub. RQ1 aims to fill in these knowledge gaps.

RQ2: [Projects] *What are the characteristics of the project that are more likely to encounter toxicity?*

Motivation: Does toxicity vary across project sponsorship, age, popularity, quality, domain, or community size? The identification of these factors will help project management undertake context-specific mitigation strategies. Determining which projects are more likely to suffer from toxicity may allow a project's management to allocate resources and define mitigation strategies. Moreover, this insight will help a prospective joiner select projects.

RQ3: [PR Context] *Which pull requests are more likely to be associated with toxicity on GitHub?*

Motivation: Does toxicity occur during poor-quality changes, unfavorable decisions, large changes, or delayed decisions? Understanding contextual factors is crucial to educating developers to avoid creating specific scenarios.

RQ4: [Participants] *What are the characteristics of participants associated with toxic comments?*

Motivation: Prior studies [Ferreira et al. 2021; Miller et al. 2022] suggested that some participants are likelier to author toxic comments due to their communication style or cultural background. On the other hand, another study suggests that participants representing underrepresented groups or newcomers are more likely to be targets [Murphy-Hill et al. 2022]. RQ4 aims to identify personal characteristics associated with being authors or victims of toxicity. This insight will help project management prepare community guidelines to combat toxicity and protect vulnerable participants.

Research method: We conducted a large-scale mixed-method empirical study of 2,828 GitHub-based OSS projects randomly selected based on a stratified sampling strategy. Our sample includes 16 million PRs and 101.5 million PR comments. Using ToxiCR [Sarker et al. 2023b], a state-of-the-art SE domain-specific toxicity detector, we automatically classified each comment as toxic or non-toxic. Additionally, we manually analyzed a random sample of 600 comments to validate ToxiCR's performance and gain insights into the nature of toxicity within our dataset. With ToxiCR demonstrating a reliable performance, we trained multivariate regression models to explore the associations between toxicity and various attributes of projects, PR contexts, and participants.

Key findings: We found 11 forms of toxic comments among GitHub PR reviews, with object-directed toxicity being a new form unreported in prior studies. The results of our study suggest that profanity is the most frequent toxicity on GitHub, followed by trolling and insults. While a project's popularity is positively associated with the prevalence of toxicity, its issue resolution rate has the opposite association. Corporate-sponsored projects are less toxic, but gaming projects are seven times more toxic than non-gaming ones. OSS contributors who have authored toxic comments in the past are significantly more likely to repeat such behavior. Moreover, such individuals are more likely to become targets of toxic texts.

Contributions The primary contributions of this study include:

- An empirical investigation of various categories of toxic communication among GitHub PRs.
- A large-scale empirical investigation of factors associated with toxicity on GitHub.
- Actionable recommendations to mitigate toxicity among OSS projects.
- We publicly make our dataset and scripts available at: 10.5281/zenodo.14802294 [Sarker et al. 2025].

Organization: The remainder of the paper is organized as follows. Section 2 briefly overview closely related works. Section 3 details our research methodology. Section 4 presents the results of our empirical investigation. Section 5 and Section 6 discuss implications and threats to the validity of our findings, respectively. Finally, Section 7 concludes the paper.

2 Related Works

Antisocial Behaviors in OSS: While internet-based communication mediums help people across multiple geographical regions to collaborate with ease, these interactions can turn negative due to various anti-social behaviors such as toxicity [Bhat et al. 2021], harassment [Lindsay et al. 2016], cyberbullying [Kowalski et al. 2014], trolling, and hate speech [Del Vigna et al. 2017; Gagliardone et al. 2015]. Although these behaviors are less frequent among professional work-focused communities such as OSS projects than social mediums [Miller et al. 2022], they may have severe repercussions on the productivity and even the sustainability of an OSS project [Lawson 2017; Raman et al. 2020; Salter 2021]. Among the various lenses studied by SE researchers, 'toxicity,' which is 'behaviors that are likely to make someone leave,' has been investigated most frequently [Qiu et al. 2022; Raman et al. 2020; Sarker et al. 2020, 2023b]. Toxicity among OSS projects on GitHub differs from other social communication platforms such as Reddit, Wikipedia, Twitter, and Stack Overflow [Miller et al. 2022]. 'Incivility,' defined as a broader superset including toxicity, is a text with an unnecessary disrespectful tone [Ferreira et al. 2021]. 'Destructive criticism' is another lens characterized by negative feedback during code reviews [Gunawardena et al. 2022]. Interactions during code reviews may also cause interpersonal conflicts among the parties, which can be termed as 'pushback' [Egelman et al. 2020; Murphy-Hill et al. 2022].

Automated identification of toxicity and other anti-social behaviors: To identify and mitigate online toxic interactions, researchers from the Natural Language Processing (NLP) domain have published datasets and classifiers, where several come from Kaggle challenges [Bhat et al. 2021; Kumar et al. 2021; Zaheri et al. 2020; Zhao et al. 2021]. For the SE domain, Raman et al. [2020] proposed the first customized toxicity detector. However, this classifier performed poorly during subsequent benchmarks [Miller et al. 2022; Qiu et al. 2022; Sarker et al. 2020]. Qiu et al. [2022]'s classifier aims to detect interpersonal conflicts by combining pushback [Egelman et al. 2020] and toxicity [Raman et al. 2020] datasets. Cheriyan et al. [2021] proposed an offensive language detector that considers a subset of toxicity, such as swearing or cursing. Sarker et al. [2023b] developed a rubric for toxicity in the SE domain and developed a toxicity classifier (ToxiCR) with their manually labeled 19,651 code review texts, which achieved 88.9% F1-score for the toxic class. Two recent studies have proposed classifiers to identify uncivil comments, where Ferreira et al. [2024] used their dataset of locked issue comments and Rahman et al. [2024] augmented ToxiCR dataset with ChatGPT generated instances to improve identification of mockery and flirtation.

Contexts and consequences of anti-social behaviors among OSS: Prior SE studies investigated contexts and consequences of anti-social behaviors using surveys and qualitative analyses. These studies suggest toxic interactions among OSS developers as a 'poison' that not only degrades their mental health but [Carillo et al. 2016] also can cause stress and burnout [Raman et al. 2020]. The threat of an OSS community disintegrating rises with the levels of toxicity due to developers' turnover [Carillo et al. 2016]. Miller et al. [2022]'s investigation found toxicity originated from both newcomers and experienced contributors due to various causes, which include technological disagreements, frustrations with a system, and past interactions with the target. Project sponsorship and domain may influence toxicity as corporate projects are less toxic than non-corporate projects. On the other hand, gaming projects are more toxic than non-gaming ones [Raman et al. 2020]. A project's toxicity may also decrease with age [Raman et al. 2020]. While uncivil discussions may arise in various locked issue contexts, they are more common among versioning and licensing discussions [Ferreira et al. 2022]. On the Linux kernel mailing list, inappropriate feedback from maintainers and violation of community conventions are the top causes of incivility [Ferreira et al. 2021]. On the other hand, among industrial developers, excessive workloads and poor-quality code are top factors [Rahman et al. 2024]. Two lenses of antisocial behaviors, destructive criticism,

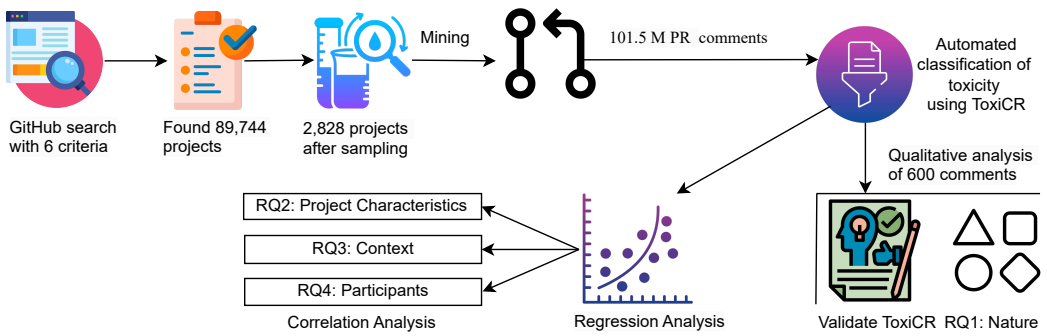

Fig. 1. An overview of our research method

and pushback, are specific to code reviews. They occur due to unnecessary harsh critiques of code and interpersonal conflicts caused by disagreements over development directions [Egelman et al. 2020; Gunawardena et al. 2022; Murphy-Hill et al. 2022]. Both pushback and destructive criticisms not only decrease productivity and degrade interpersonal relationships [Egelman et al. 2020; Murphy-Hill et al. 2022], they disproportionately harm underrepresented minorities and cause barriers to promoting DEI [Gunawardena et al. 2022]. Besides these academic works, several gray literature have also documented burnout and turnover of long-term OSS contributors due to toxicity [Author 2014; Barnes 2020; Diggs 2021; Lawson 2017; Salter 2021; Vaughan-Nichols 2018; X 2021].

Novelty: This study differs from prior empirical investigations of antisocial behaviors in three ways. First, prior studies focused on communication from specific contexts such as locked issues [Miller et al. 2022; Raman et al. 2020] or rejected patches [Ferreira et al. 2021]. Therefore, characteristics of toxicity outside these known negative contexts are missing. Second, these investigations are qualitative. While these investigations are crucial to forming hypotheses, whether these hypotheses apply to a broader spectrum of OSS projects remains unanswered. Finally, these studies explored a limited set of factors, whether other plausible factors, such as community size, project popularity, code complexity, and unresolved defects associated with toxicity, remain unanswered. For example, we investigated the association between toxicity and 32 different factors, where 21 are unique to our study.

3 Research Method

Figure 1 provides an overview of our research methodology, detailed in the following subsections.

3.1 Project Selection

We leverage the GitHub search tool developed by Dabic et al. [2021], which enables filtering based on various criteria such as the number of contributors, programming language, forks, commits, and stars to select candidate projects. Following recommendations by Kalliamvakou et al. [2016], we searched for projects satisfying the following six criteria: i) uses one of the top ten programming languages on GitHub: Java, C, C++, Python, JavaScript, C#, Go, PHP, Typescript, and Ruby; ii) has at least 20 contributors, iii) is publicly available with an OSS license; iv) at least two years old, v) has at least 20 PRs; vi) and has at least 10 stars. The first five criteria ensure the selection of OSS projects with adequate analyzable interpersonal communication among the contributors. The last

criterion reduces the search space; without this filter, the number of projects grows exponentially but adds only trivial OSS projects.

Our search conducted in September 2022 found 89,744 projects. We exported the results as a CSV file and categorized the projects into the following three groups based on project activity, which we measured using monthly Pull Request Frequency (PRF) on GitHub.

- *Low PRF (PRF-L)*: A project belongs to this group if it has less than 8 PRs per month (i.e., less than 2 PRs/week). Total 66,791(74.5%) projects belong to this group.
- *Medium PRF (PRF-M)*: A project belongs to this group if it has between 8 -31 PRs per month (i.e., 2-8 PRs/week). Total 14,694 (16.3%) projects belong to this group.
- *High PRF (PRF-H)*: A project belongs to this group if it has more than 32 PRs per month (i.e., >8 PRs/week). Total 8,259 (9.2%) projects belong to this group.

These two chosen grouping thresholds, 8 and 32, represent approximately 75 and 90 percentiles based on PRF. We chose PRF instead of commit frequency because we noticed that many popular projects use GitHub as a mirror (e.g., the Linux kernel) while development activities predominantly use other platforms. Next, we randomly selected 800 projects from each of the three groups. This stratified sampling is necessary since three-fourths of the projects found based on our search criteria belong to the PRF-L group. Hence, a random selection will have been dominated by PRF-L projects and will fail to adequately capture the characteristics of the remaining two groups. Our sample size of 800 is adequate to obtain results within a 3% margin of error with a 95% confidence interval [Cochran 1977].

Since prior studies have suggested higher occurrences of toxicity among gaming projects [Miller et al. 2022], we also added projects with the topic 'game' and at least 10 stars. The search result found 6.1k projects. However, most projects did not satisfy criteria such as a minimum number of participants or PRs. Hence, the gaming group includes 439 projects, adequate for 95% confidence interval (CI) and a 5% Margin of Error [Cochran 1977]. We include the gaming projects in our RQ2 analysis only to investigate project characteristics associated with toxicity. However, some projects were no longer available for mining (e.g., deleted or moved). Therefore, our final dataset contains 2,828 projects.

3.2 Dataset Preparation

We wrote a Python Script using the PyGithub library [Jacques 2024] to mine all the PR details, metadata, PR labels, user information, inline review comments, and PR discussions from the selected projects and store them in a MySQL database. Our data mining started in October 2022 and completed in January 2023. Our dataset requires approximately 172 GB of storage. Our dataset includes approximately 16.1 million (M) PRs, 101.5 M comments, and 1.3 M unique users. We also mined all publicly available user information, including profile photos, emails, full names, and user types. We exclude all the interactions from Bot accounts (i.e., userType='Bot'). However, we also noticed many bot accounts using incorrect flags (i.e., userType ='User'). Therefore, we filtered accounts with bot-specific keywords (bot, robot, auto, Jenkins, static, etc.) in user names and full names. We manually inspected the filtered accounts to make a final determination.

3.3 Toxicity Classification Scheme

Multiple recent studies have investigated toxicity and other antisocial behaviors within OSS communities [Egelman et al. 2020; Ferreira et al. 2021; Gunawardena et al. 2022; Miller et al. 2022; Rahman et al. 2024; Raman et al. 2020; Sarker et al. 2023b; Sultana 2022b]. These studies enumerate several prevalent forms of such behaviors and conclude that toxicity in the OSS context are different from a non-SE domain such as social media [Miller et al. 2022]. To answer RQ1, we focused on

aggregating various categories of anti-social behaviors to prepare our manual labeling scheme. We started with consolidating categories derived from studies on 'toxicity' [Miller et al. 2022; Qiu et al. 2022; Raman et al. 2020; Sarker et al. 2023b]. During this process, we identified overlapping concepts based on definitions included in those papers and merged those into a single category. We noticed a conflict as 'self-deprecation' was marked as non-toxic by Sarker et al. [2023b], while Miller et al. [2022] marks 'self-directed pejorative', a similar concept as toxic. We follow Sarker et al. [2023b]'s definition, marking such texts as toxic only if they involve profanity since we use their tool. Additionally, Ferreira et al. [2021]'s incivility lens, which encompasses a broader spectrum of anti-social behaviors, including toxicity, also includes frustration, impatience, irony, mocking, name-calling, threat, and vulgarity. Based on their definitions, 'name-calling, threat, mocking, and vulgarity' overlap with existing categories identified by Miller et al. [2022] and Sarker et al. [2023b]. As such, these were considered toxic. Although 'irony,' 'frustration,' and 'impatience' are not a part of these toxicity schemes, they may fit existing categories, such as trolling, arrogance, and insult, depending on context. It's worth noting that existing SE studies have used different terminologies to study these subjective social constructs, and an agreed-upon standard scheme or definition is currently missing and perhaps challenging to establish. At the end of this step, we prepared our manual labeling scheme of 10 categories (Table 3) with a broader definition for each group.

3.4 Automated identification of toxic comments

We select ToxiCR [Sarker et al. 2023b] for automated classification since it is i) trained on large-scale training data, ii) developed as a reusable standalone tool, iii) is publicly available on GitHub, iv) provides a well-defined interface to conduct a large-scale classification required by this study, v) trained on Code review data which is similar to pull requests that this study aims to analyze, and vi) reports the best performance according to its evaluation with 95.8% accuracy, 90.7% precision, 87.4% recall, and an 88.9% F1-score. ToxiCR provides the toxicity probability of a text from 0 to 1, and its authors recommend using a threshold >= 0.5 to consider a text as toxic. Using ToxiCR's best-performing configuration [Sarker et al. 2023b], we classified all the PR comments, totaling 101.5 million. ToxiCR found approximately 756K toxic comments (0.74%) from our dataset.

Evaluation of ToxiCR: Prior research on SE domain-specific NLP tools [Novielli et al. 2021, 2018] recommends independent assessments before application on new settings. Therefore, we conducted an empirical evaluation to assess ToxiCR's reliability on our dataset. To achieve this goal, we randomly selected 600 PR comments marked as toxic by ToxiCR. This sample adequately provides results within a 2.6% margin of error and 95% confidence interval [Cochran 1977]. We also use this sample to investigate the frequencies of various forms of toxicity on GitHub (Section 3.5).

Precision: Two raters independently labeled those 600 samples as toxic or non-toxic and resolved the conflicts after a discussion. To mitigate the bias of the labeling process, two labelers follow the toxicity rubric from Sarker et al. [2023b]. The agreement between the two labelers is 95.8%, and Cohen's kappa [Cohen 1960] value is $\kappa = 0.80$, which is 'substantial'. After conflict resolution, 532 comments were labeled toxic, suggesting 88.8% precision. This result is within the margin of error (i.e., 2.6%) of ToxiCR's claimed precision (90.7%) [Sarker et al. 2023b].

Recall: Evaluation of recall is also essential to ensure that ToxiCR does not miss many positive instances. On this goal, we focused on finding existing labeled toxicity datasets curated from GitHub issue requests. We did not use Raman et al. [2020]'s dataset since it has only 106 toxic instances. We chose Ferreira et al. [2022]'s dataset of locked GitHub issues, which includes 896 uncivil sentences out of 1,364. According to Ferreira et al. [2021], incivility is a super-set of toxicity. While all toxic comments are uncivil, some uncivil comments (e.g., irony and impatience) may not fit the toxicity lens. To encounter this challenge, two authors relabeled the 896 uncivil comments based on Sarker et al. [2023b]'s toxicity labeling rubric. The raters achieved an inter-rater agreement of $\kappa = 0.76$

and resolved the conflicts through a mutual discussion. On this dataset, ToxiCR achieved 87% recall, which is also within the sampling margin of error of ToxiCR's reported recall (i.e., 87.4%).

3.5 Manual Categorization of Toxic Comments

Using the 10 class classification scheme described in Section 3.3, two of the authors independently placed the 532 toxic comments identified during ToxiCR's evaluation (Section 3.4) into one or more groups. We also included the 'Others' category to label toxic comments that do not fit the existing ten categories. We measured the inter-rater reliability of this multiclass labeling using Krippendorff's alpha, which was 0.35, indicating a 'Fair' agreement. We noticed higher ratios of disagreements since, theoretically, the number of possible labeling for a single instance is 2^{11}. Conflicting labels were resolved through mutual discussions. After conflict resolution, the raters reviewed the 34 instances from the 'Others' group to identify missing categories. The new category identified is 'Object-Directed Toxicity', which includes anger, frustration, or profanity directed toward software, products, or artifacts. For example, "also the mask sprite is beyond horrid, I might have something that could do better.." represents this form. With this category, they went through the labeled instances again to identify other cases that may also fall under this category since a text can fall under multiple categories. We found a total of 49 instances belonging to this new category.

3.6 Attribute Selection

Table 1 lists attributes selected to answer RQ2, RQ3, and RQ4 introduced in Section 1. In addition to each attribute's definition, Table 1 hypothesizes why an attribute may be associated with toxicity.

RQ2: Project We select eleven project characteristics attributes based on prior studies on toxicity and incivility [Ferreira et al. 2022, 2021; Miller et al. 2022; Raman et al. 2020]. These 11 attributes characterize a project's activity, popularity, domain, governance, and age.

RQ3: PR Context We select nine contextual attributes based on prior studies [Egelman et al. 2020; Miller et al. 2022; Rahman et al. 2024; Raman et al. 2020; Sultana et al. 2023; Thongtanunam et al. 2017]. These attributes characterize the type of change, outcome, complexity, required review /resolution efforts, completion time, and number of identified issues in a PR.

RQ4: Participant We select six participant attributes based on prior studies [Cohen 2021; Ferreira et al. 2022; Miller et al. 2022; Murphy-Hill et al. 2022; Rahman et al. 2024; Sultana 2022a]. These attributes represent a participant's GitHub tenure, project experience, gender, and communication history. We compute each attribute for both the author and the target of a comment; therefore, we have 12 attributes from this category.

3.7 Attribute Calculation

To investigate RQ3 and RQ4, it is necessary to compute attributes at pull request (PR) and comment levels, respectively. Given that our dataset comprises 16 M PRs and 101.5 M comments, calculating the PR and comment-level attributes listed in Table 1 for the entire dataset would be exceedingly time-consuming and resource-intensive. Therefore, we reduced the sample size for RQ3 and RQ4 by randomly selecting 385 projects from the three project groups (i.e., 'PRF-L', 'PRF-M', and 'PRF-H'). We choose this sample size to satisfy a 5% error margin and 95% confidence interval [Cochran 1977]. This sample of 1,155 projects includes 6.3 M PRs, 30 M comments, and 416 K users. We exclude gaming projects from this analysis since they have a higher prevalence of toxicity, and many such projects do not consider profanities offensive. Therefore, contexts and participants of toxicity among gaming projects are not representative of non-gaming ones. We wrote Python scripts and MySQL queries to compute the 32 attributes listed in Table 1 based on their definitions. While most attributes are straightforward to calculate, five require additional heuristics, as defined in the following.

Table 1. The list of attributes selected to investigate their association with project characteristics (RQ2), Pull request context (RQ3), and participants' characteristics (RQ4). We selected this set of attributes since prior studies on code reviews and anti-social behaviors suggest the likelihood of association with toxicity or conflict-instigating contexts. * -indicates attributes that were investigated in prior studies.

Variable	Definition	Rationale
RQ2: Project Characteristics		
PR /month*	Average number of PRs per month.	Indicates the volume of development activity. Active projects may have a higher probability of toxic interactions [Miller et al. 2022].
issues /month	Average number of issues per month.	A higher number of bugs indicates the lack of quality, which may cause frustration among users and developers [Miller et al. 2022; Raman et al. 2020].
commits /month	Average number of commits per month.	Commit is another indication of the volume of development activity. High activity may cause burnouts [Raman et al. 2020], and lack may cause frustration among users [Miller et al. 2022].
release /month	Average number of releases per month.	Frequent releases may satisfy the customers to decrease toxicity, and vice versa [Costa et al. 2018].
issue resolution rate	Percentage of issues resolved.	Users may become frustrated due to issues affecting them not being resolved [Miller et al. 2022].
isCorporate*	Whether the project is sponsored by a corporation.	Corporate projects may have less toxicity than non-corporate ones due to the consequences of HR policy violations [Raman et al. 2020].
project age*	Total months since a project's creation.	Older projects showed more toxicity [Raman et al. 2020].
member count	Number of users with write access.	Toxicity increases with community size due to diverse views and higher potential conflicts [Basirati et al. 2020].
isGame*	Whether the project is gaming or not.	Prior studies have found prevalence of toxicity among gaming communities [Belskie et al. 2023; Beres et al. 2021; Miller et al. 2022; Paul 2018].
stars	Number of stars on GitHub project.	Popularity shows users' interests. Scrutiny and expectations increase with popularity and therefore stress on developers [Raman et al. 2020].
forks	Number of forks on GitHub project.	Fork is another measure of project popularity [Zhou et al. 2020].
RQ3: PR Context		
commit count	Number of commits in a PR.	A large number of commits increases review effort [Thongtanunam et al. 2017], which may frustrate reviewers, cause delays, and frustrate the author.
number of changed files	The number of files changes in each PR.	A higher number of file changes requires a longer review time [Kononenko et al. 2015] and comprehension difficulty.
code churn (log)*	The total number of rewritten or deleted code.	A high number of changed lines increases the probability of defects in the code [Nagappan and Ball 2005, 2007] and may link to more toxic comments.
isAccepted*	Whether the code review is accepted or rejected.	Developers used more toxic comments in rejected codes/patches [Ferreira et al. 2021].
isBugFix	Whether the code review is for fixing a bug or not.	Issue discussions may instigate toxicity when the resolution is not liked by affected parties [Ferreira et al. 2022; Miller et al. 2022].
change entropy (log)	A measure of change complexity, which estimates how much dispersed a changeset is among multiple files [Thongtanunam et al. 2017].	Complexity of code change affects review time and participation [Thongtanunam et al. 2017]. Moreover, unnecessary complexity may be a sign of a poor quality change, which may receive harsh critique [Rahman et al. 2024].
review interval	Time difference from the start of the code review to the end.	Delayed code reviews are more likely to cause frustration for developers [Egelman et al. 2020; Turzo and Bosu 2024].
number of iterations (num iter)	Total number of iterations (i.e., number of times changes requested) in a PR.	Higher number of iterations frustrates both developers and reviewers due to additional time [Turzo and Bosu 2024]. Higher iteration also indicates the lack of common understandings [Ebert et al. 2019] and potential disagreements [Murphy-Hill et al. 2022].
review comments*	The total number of review comments from reviewers in a PR.	A higher number of review comments indicate significant concerns from the reviewers over its quality, which often causes toxicity [Rahman et al. 2024].
RQ4: Participants		
isWoman	Whether the person is a woman	Prior studies have found women and marginalized minorities as frequent victims of toxicity [Gunawardena et al. 2022; Raman et al. 2020].
isMember*	Whether the person is a project member or not.	Project members are the authors of many toxic comments in replying to the outside members' query [Cohen 2021; Miller et al. 2022].
isNewComer*	Whether the person is a newcomer to the current project.	Newcomers may get frustrated due to delays [Steinmacher et al. 2013] and unfavorable decisions [Ferreira et al. 2021].
GitHub tenure*	Age of GitHub account, in terms of the total months at the time of an event.	[Miller et al. 2022] reported toxic comments from accounts with no prior activity on GitHub.
project tenure	Tenure with the current project in terms of the total months.	Although long-term members of a project are more committed to maintaining a professional environment in a community, Miller *et* al. found toxic comments from them [Miller et al. 2022]. Moreover, they may be targets if their decisions are not liked by issue reporters [Ferreira et al. 2022].
toxicity / month*	The total number of toxic comments a user posts per month.	[Miller et al. 2022] found many repeat offenders, as many OSS developers have toxic communication styles [Author 2014; Miller et al. 2022].

Gender: We adopted a similar protocol to Sultana et al. [2023] to automatically predict users' genders. We have used genderComputer [Vasilescu et al. 2014] and Wiki -Gendersort [Bérubé et al. 2020] tools to resolve the gender from a user's name, preferred pronoun, and location, if available. We have also downloaded a user's GitHub avatar and applied an automated human face detection model [Goyal et al. 2017]. Further, we used a pre-trained photo to gender-resolution model [Eidinger et al. 2014] to predict the user's gender. Conflicts between the two approaches were resolved by manually investigating users' profiles. Finally, we successfully resolved 75.4% of the total users (92% with full names). We only include gender-resolved users for RQ4.

Project Member: Following the recommendation of Gousios and Spinellis [2012], we consider a user a project member if that user has write access (i.e., merged at least one PR or created an intra-branch PR) to the repository.

GitHub Tenure and Project Tenure: We compute a user's GitHub tenure at an event as the months between their account creation and the event's timestamp. Similarly, we calculate a developer's project tenure during each project interaction (e.g., commit, pull request, or comment).

Newcomer: Following the definitions of prior studies [Steinmacher et al. 2013; Subramanian et al. 2020], we consider a user as a newcomer to an OSS project until they have got their first PR accepted to this project.

3.8 Regression Modeling

Regression analysis offers a robust statistical method for examining the influence of one or multiple independent variables on a dependent variable [Foley 2018]. Two categories of regression models are used: (i) *Predictive analysis*: involves creating a formula to forecast the value of a dependent variable based on the values of one or more independent variables; and (ii) *Inferential analysis*: seeks to establish whether a specific independent variable affects the dependent variable and to quantify that impact if present [Allison 2014]. An inferential analysis differs from a predictive analysis in two key aspects. First, *multicollinearity*: when two or more independent variables are highly correlated, incorporating all correlated variables simultaneously in inferential analysis can lead to an overfitting problem. However, in predictive analysis, multicollinearity is not a concern. Second, the importance of R^2 – the goodness of fit of a regression model [Helland 1987]. While a higher R^2 is desirable, it holds greater importance in predictive analysis. In inferential analysis, even with a low R^2, the regression model can provide valuable insights into the relationships between the independent and dependent variables [Allison 2014]. We train multivariate inferential regression models to analyze associations between the toxicity and the 32 attributes listed in Table 1. The following subsections detail the regression models to answer RQ2, RQ3, and RQ4.

3.8.1 Multinomial Logistic Regression for RQ2. We found training a regression model for RQ2 challenging since computing various project characteristics variables at the creation timestamp of a comment requires the entire event log for a project (e.g., when a new star was added), which is resource-intensive to mine due to the enormous size of our dataset. While Google's BigQuery hosts a dataset of GitHub events, it would be expensive to query this service. Therefore, we used aggregated attributes over the lifetime of a project. We calculated toxicity per hundred comments (*percent_toxic*) for each project over its lifetime and used it as the dependent variable for RQ2. However, if the dependent variable is a ratio, a model can identify spurious associations [Kronmal 1993]. Following the recommendation of Long and Freese [2006], we transform the *percent_toxic* variable into a three-level categorical variable named *toxicity_group*. We selected the number of categories and thresholds for this grouping based on the inflection points[1] in the cumulative distribution curve. The 'Low toxic' group includes 324 projects with *percent_toxic* $< 0.02\%$. The

[1]points of a curve at which a change in the direction of curvature occurs

Table 2. For the bootstrapped logistic regression models, model fit measured using Vealll-Zimmermann Psuedo R^2. A 95% confidence interval is also reported for R^2 values. All models are significantly better than null models ($p < 0.001$).

Model	PRF-L	PRF-M	PRF-H
RQ3	0.16 [0.15, 0.17]	0.19 [0.19, 0.20]	0.18 [0.18, 0.19]
RQ4 (author)	0.01 [0.01 ,0.01]	0.02 [0.02, 0.02]	0.09 [0.09, 0.09]
RQ4 (target)	0.01 [0.01 ,0.01]	0.01 [0.01, 0.01]	0.11 [0.11, 0.11]

2,082 projects from the 'Medium toxic' group have $0.02 \leq percent\_toxic < 1\%$. The remaining 421 projects belong to the 'High toxic' group with $percent\_toxic \geq 1\%$. Since $toxicity\_group$ has three levels, we use a Multinomial Logistic Regression (MLR) model, where $toxicity\_group$ is the dependent variable and 11 project characteristics attributes are independents.

3.8.2 Bootstrapped Logistic Regression for RQ3 and RQ4. For RQ3, the dependent variable is *HasToxicComment*, set to 1 if a PR has at least one toxic comment and 0 otherwise. For RQ4, we use participant attributes computed at the comment level as independents. We use *isToxic* as the dependent, 1 if the comment is toxic, and 0 otherwise. We train two models for RQ4, one with the author's attributes as the independents and the other with the target's attributes. Since the dependents are binary for RQ3 and RQ4, we use Logistic Regression models. As the dataset of RQ3 and RQ4 consist of a rare binary outcome variable (i.e., *HasToxicComment*, *isToxic*), we use a bootstrapped regression modeling technique [Xu et al. 2020]. In this technique, we choose a desired ratio between the minority and the majority. We randomly downsample the majority until the desired ratio is reached. We fit a logistic regression model with each bootstrapped sample, measure its fit, and compute regression coefficients. This process is repeated 100 times, and we record the results of each iteration in a dataset. We report median and 95% confidence interval for model fit and regression coefficients. We also explored various ratios between the minority and the majority and found that the model's goodness of fit (i.e., R^2) reduces with the increment of the majority's share. We chose a ratio of 1:10 since increasing the majority's share beyond that produced unreliable models in a few cases, according to the Log-likelihood test (*lrtest*).

3.8.3 Correlation and Redundancy Analysis. For an inferential regression model, multicollinearity poses a threat to validity. We used the variable clustering approach suggested by Sarle [1990] to identify multicollinearity. With this approach, we create a hierarchical cluster representation of independents using Spearman's rank-order correlation test [Statistics 2013]. As recommended by Hinkle et al. [1998], we set the cutoff value at $|\rho| = 0.7$ for the correlation coefficient. Only the explanatory variable with the strongest correlation with the dependent was chosen from a cluster of variables with $|\rho| \geq 0.7$.

3.8.4 Model analysis. We also use the Log-likelihood test (*lrtest*) to assess whether a model significantly differs (Chi-Square, $p < 0.05$) from a null model and can be reliably used for inference. We evaluate each model's goodness-of-fit using Veall-Zimmermann Psuedo-R^2 [Veall and Zimmermann 1994] since prior research [Smith and McKenna 2013] found this measure having closer correspondence to ordinary least square R^2. A higher R^2 value indicates a better fit. We use the Odds ratio (OR) to quantify the association between the dependent and independents and estimate effect size. For a binary independent (e.g., *isGame*), OR indicates the odds of an outcome if the independent variable changes from 0 to 1, while all other factors remain constant. For a continuous variable (e.g., project age), OR indicates an increase or decrease in odds for the dependent with one unit change in the factor. In simple terms, OR >1 indicates a positive association and vice versa. We use the p-value of the regression coefficient to assess the significance of an association, with $p < 0.05$ indicating a statistical significance. Table 2 shows goodness-of-fit measured with

Pseudo-R^2 for the regression models trained for RQ3 and RQ4. Since we bootstrapped each model 100 times, we report median and 95% confidence intervals for each model. The results of *lrtest* indicate that all models are significantly better than a null model ($p < 0.001$) and are reliable to infer insights to answer our RQs. However, although models for RQ4 are significant, they have low R^2 values, which we further explain in Section 4.4.

4 Results

The following subsections detail the results of the four RQs.

4.1 RQ1: Nature of toxicity

Table 3 shows the distributions of 11 forms of toxicities among our manually labeled dataset of 532 PR review comments. Similar to Miller et al. [2022]'s investigation, we found profanity (i.e., severe language, swearing, cursing) as the most common form, with more than half of the sample (≈ 58%) belonging to it. Toxic texts authored by OSS contributors frequently include profane words such as 'shit', 'fuck', 'ass', 'crap', 'suck', and 'damn'. We found trolling to be the second most common form, with 18%, followed by insult, self-deprecation, and object-directed toxicity. Identity attacks, insults, and threats, which are regarded as severe toxicities [Goyal et al. 2022], were found among ≈22% of the samples. We also noticed over two-thirds of our samples ≈72% belonging to multiple forms. For example, "*:laughing: Holy shit, you are fucking stupid. It is an extremely simple proc with a switch. Seriously, did you even look at the code? Next, you'll tell me all switches are copypaste.*" represents both profanity and insult. While toxicity on social media has higher occurrences of flirtation and obscenity [Goyal et al. 2022; Gunasekara and Nejadgholi 2018], we found ≈2% such cases in our sample.

> **Key finding 1:** *Profanity is the dominant form of toxicity in GitHub PRs. Severe toxicities, such as insults, identity attacks, and threats, represent ≈ 22% cases. Unlike other online mediums, flirtation and obscenity were less common in our sample.*

4.2 RQ2: Project characteristics

During RQ2's model training two factors (i.e., forks and pulls per month) were dropped due to multicollinearity and were not included in our MLR. We estimated the fit of our MLR with NagelKerke R^2 =0.224. Our Log likelihood test results suggest that this model significantly differs (χ^2 =545.16, $p < 0.001$) from a null model. In addition to modeling the probability of a specific result based on a group of independent variables, MLR also enables the assessment of the probability of transitioning to a different dependent category from the current one when a specific independent variable changes [Bayaga 2010]. Hence, we set the 'Low toxic' projects as the reference group in MLR and compute the odds of a project moving to the 'Medium toxic' or 'High toxic' group if one of the independents changes by a unit. Table 4 shows the result of our MLR model, with OR values for each factor. Our results suggest that projects with corporate sponsorship (*isCorporate*) are significantly less likely to belong to the 'Medium toxic' or 'High toxic' groups than the 'Low toxic' group. HR rules, professional codes of conduct, and the potential for job loss may be the reasons. We also noticed a significantly higher level of toxicity among the popular projects (i.e., stars). We found that the prevalence of toxicity significantly increased with project age. Moreover, our analysis found no significant association between toxicity and development activities (i.e., commits/month and releases/month) and project quality (i.e., issues/month). On the other hand, issue resolution rates (i.e., percentage of resolved issues) significantly reduce toxicity, as projects with higher rates are more likely to belong to the 'Low toxic' group than the 'Medium toxic' or 'High toxic' group. Supporting observations from prior studies [Miller et al. 2022], we also noticed

Table 3. The most common forms of toxicities with definitions within our sample of manually labeled 532 PR comments. We also showed the mapping from existing works.

Type	Mapping	Definition	Example	Count‡	Ratio
Profanity	Profanity [Sarker et al. 2023b], Expletives [Miller et al. 2022], Vulgarity [Ferreira et al. 2021]	A comment that includes profanity.	"You know, at some point, github fucked me over. In Visual Studio, this was just fine, wtf...."	311	58.45%
Trolling	Trolling [Ferreira et al. 2021; Miller et al. 2022]	Using trolling with destructive discussions and those are more severe and provoke arguments.	"@clusterfuck There's a difference between being in cryogenics and not in cryogenics you big nerd"	96	18.04%
Insult	Insult [Miller et al. 2022; Sarker et al. 2023b]	Disrespectful expression towards another person.	"Acknowledge that the vote wasn't entirely singulo shitposters > ARE YOU SCHIZOPHRENIC?? Jesus dude why are you even here still"	92	17.3%
Self-deprecation	Self-deprecation [Miller et al. 2022]	If a demeaning word towards him/herself consists of severe language, it would be marked as self-deprecation.	"@ComicIronic Okay, I'll fix it when I fix my shitty code, which will have to happen tomorrow"	67	12.6%
Object Directed Toxicity	New	Anger, frustration, or profanity directed toward software, products, or artifacts.	"the PR has fallen into conflict hell, I'll be closing this and re-opening some of its changes shornestly"	49	9.21%
Entitled	Entitled [Miller et al. 2022]	When people demand due to the expectation related to contractual relationship or payment.	"Again I didn't break it Are you fucking stupid lol > merge your update into PR > buckling doesn't work"	17	3.2%
Identity attack	Identity attack [Sarker et al. 2023b], Inappropriate jokes about an employee [Egelman et al. 2020]	Attacking the person's identity.	"Fuck those argentinians. Did you test it?"	17	3.2%
Threats	Threats [Ferreira et al. 2021; Sarker et al. 2023b]	A behavior that is aggressive or threatening someone.	"Done - I can always revoke your access if you mess things up ;"	12	2.25%
Obscenity	Reference to sexual activities [Sarker et al. 2023b]	An extremely offensive comment that demeans women or LGBTQ+ people.	"dude you need to spend less time on programming and more time with women"	6	1.1%
Arrogance	Arrogance [Miller et al. 2022]	Imposing the own view on others due to superiority.	"Araneus is shit and generic as hell I think steely is an acceptable name"	5	< 1%
Flirtation	Flirtation [Sarker et al. 2023b]	A comment that represents flirting.	"Frigging love you Niki. Seriously"	5	< 1%

‡ *-since a text can belong to multiple categories, the sum of the categories is greater than our sample size.*

the significantly higher prevalence of toxicity among gaming projects, as a gaming project is seven times more likely to belong to the 'High toxic' group than the 'Low toxic' or 'Medium toxic' group.

Key finding 2: *While popularity and staleness are positively associated with the prevalence of toxicity, issue resolution rate has the opposite association. While corporate-sponsored projects are likelier to be 'low toxic,' gaming projects are likelier to belong to the opposite spectrum.*

Table 4. Results of our MLR model to identify associations of project characteristics with toxicity. We set the 'Low toxic' group as the reference to compute odds ratios. Hence, $OR > 1$ indicates a higher likelihood of a project transitioning to the 'Medium' or 'High' toxic group with an increment of that factor and vice versa.

Attribute	Medium toxic		High toxic	
	OR	p	OR	p
isCorporate	0.888	**0.000*****	0.47	**0.000*****
member count	0.999	0.821	1.0001	0.754
stars	1.001	**0.000*****	1.001	**0.000*****
issues/month	1.001	0.234	0.998	0.061
project age	1.004	**0.000*****	1.009	**0.000*****
commits/month	1.0001	0.316	1.0001	0.447
release/month	1.003	0.310	0.998	0.813
bug resolution	0.204	**0.000*****	0.985	**0.000*****
isGame	0.486	**0.000** ***	7.259	**0.000*****

***, **, and * represent statistical significance at $p < 0.001$, $p < 0.01$, and $p < 0.05$ respectively.

Table 5. Associations between pull request contexts and toxicity. Values represent the median odds ratio for each factor with 95% confidence intervals inside brackets.

Attribute	PRF-L	PRF-M	PRF-H
isBugFix	1.01[0.99, 1.05]	1.02 [1.01, 1.03]	1.06*** [1.06, 1.07]
commit count	1.00 [1.00, 1.01]	0.99* [0.99, 0.99]	0.99*** [0.99, 1]
code churn (log)	1.11 [1.10, 1.12]	1.13*** [1.13, 1.14]	1.11*** [1.11, 1.11]
review interval	1.11*** [1.11, 1.12]	1.17*** [1.17, 1.17]	1.20***[1.20, 1.20]
review comments	1.06*** [1.05, 1.07]	1.06*** [1.05, 1.06]	1.04*** [1.04, 1.04]
isAccepted	0.77*** [0.75, 0.80]	0.71*** [0.70, 0.73]	0.93*** [0.93, 0.94]
num iter	1.01 [0.99, 1.03]	1.01*** [1.01, 1.02]	1.01*** [1.01, 1.01]
change entropy (log)	1.57 [1.51, 1.67]	1.35*** [1.32, 1.37]	1.26*** [1.25, 1.27]

***, **, and * represent statistical significance at $p < 0.001$, $p < 0.01$, and $p < 0.05$ respectively.

4.3 RQ3: Pull request context

One of the nine pull request context factors (i.e., the number of changed files) was dropped due to multicollinearity. Table 5 shows median odds ratios with 95% confidence intervals for the remaining eight factors based on our bootstrapped logistic regression models repeated over 100 times. All eight factors show significant associations for the PRF-H group (i.e., PR/month > 32). For this group, bug fix PRs, code churn, review interval, the number of review comments, the number of review iterations, and change entropy are positively associated with toxicity. On the other hand, the number of commits and acceptance decisions are negatively associated. Similarly, we noticed almost identical associations for the PRF-M group (i.e., 8 < PR/month < 32), except *isBugFix* does not have a statistically significant association. For the PRF-L group (i.e., PR/month < 8), only three factors have statistically significant associations with toxicity, where the review interval and the number of review comments have positive ones. In contrast, *isAccepted* has a negative one.

A positive correlation between toxicity and isBugFix for the PRF-H group indicates that discussions on approach to fix pending issues might become heated for highly active projects, which support prior studies on locked issues [Ferreira et al. 2022; Miller et al. 2022]. However, such a trend is not seen among projects belonging to PRF-M and PRF-L. While the number of commits

Table 6. Associations between characteristics of authors toxicity. Values represent the median odds ratio for each factor with 95% confidence intervals inside brackets.

Attribute	PRF-L	PRF-M	PRF-H
isWoman	0.80** [0.71, 0.793]	1 [0.96, 1.01]	0.90*** [0.86, 0.87]
isNewComer	1.09 [1.05, 1.14]	0.88*** [0.86, 0.89]	0.69*** [0.68, 0.69]
isMember	0.89** [0.87, 0.92]	0.77*** [0.77, 0.88]	0.64*** [0.64, 0.64]
GitHub tenure	0.99*** [0.99, 0.99]	0.99*** [0.99, 0.99]	0.99*** [0.99, 0.99]
project tenure	1.01** [1.01, 1.01]	1.01 [0.99, 1.01]	1.01*** [1.01, 1.01]
toxicity/month	1.06*** [1.05, 1.07]	1.02*** [1.02, 1.02]	1.01*** [1.01, 1.01]

*** , **, and * represent statistical significance at $p < 0.001$, $p < 0.01$, and $p < 0.05$ respectively.

Table 7. Associations between characteristics of targets and toxicity. Values represent the median odds ratio for each factor with 95% confidence intervals inside brackets.

Attribute	PRF-L	PRF-M	PRF-H
isWoman	0.48*** [0.46, 0.50]	0.95 [0.93, 0.98]	0.86*** [0.85, 0.87]
isNewComer	0.91 [0.88, 0.95]	0.90***[0.89, 0.92]	0.74*** [0.74, 0.75]
isMember	1.12* [1.09, 1.15]	1.04 [1.02, 1.05]	0.84*** [0.84, 0.84]
GitHub tenure	1.01 [0.99, 1.01]	1.01*** [1.01, 1.01]	1.01 [1.01, 1.01]
project tenure	0.99* [0.99, 0.99]	1.01***[1.01, 1.01]	1.01*** [1.01, 1.01]
toxicity/month	1.99*** [1.81, 2.19]	1.46*** [1.36, 1.52]	1.26*** [1.25, 1.26]

*** , **, and * represent statistical significance at $p < 0.001$, $p < 0.01$, and $p < 0.05$ respectively.

included in a PR is negatively associated with toxicity for both PRF-M and PRF-H groups, our results suggest contradicting associations between toxicity and commit size measured using code churn. Since we code churn and the number of commits included in a PR are positively correlated, we were surprised by this finding. Our in-depth investigation suggests that PRs with large numbers of commits are often due to inter-branch clean-up or feature imports. Hence, such PRs are less likely to have discussions [Thongtanunam et al. 2017] and, therefore, are less likely to be toxic. Positive correlations between toxicity and review interval across all project groups suggest that delayed decisions frustrate the participants and, hence, are more likely to instigate toxicity. Similarly, the number of review comments for a PR, an indicator of issues identified by reviewers, is positively associated with toxicity. This result indicates that PRs with poor-quality code are more likely to be associated with toxicity. Unsurprisingly, we found a negative correlation between accepted PRs and toxicity among all groups, which indicates that rejected PRs are significantly more likely to be associated with toxicity than accepted ones. The number of iterations, which indicates the number of times an author must add additional commits based on reviewers' suggestions, is positively associated with toxicity for projects belonging to the PRF-M and PRF-H groups. Finally, change entropy, a proxy for change complexity, is positively associated with toxicity among projects belonging to PRF-M and PRF-H groups.

Key finding 3: *Accepted PRs are less likely to encounter toxicity. On the contrary, code churn, review intervals, the number of review comments, change entropy, and the number of review iterations are positively associated with toxicity on GitHub.*

4.4 RQ4: Participants

We train two types of regression models, one to investigate the characteristics of persons authoring toxic comments and the other with the targets. Similar to the RQ3, we train bootstrapped logistic

regression models repeated over 100 times. Tables 6 and 7 show the odds ratios of authors and targets for each factor with 95% confidence intervals for the three PRF-based project groups. The results of Log-likelihood ratio tests (*lrtest*) suggest that these models are significantly better than Null models and, therefore, are suitable to provide inferential insights. However, these models have low R^2 (i.e., low explainability power). This result indicates that the characteristics of participants, i.e., the attributes used to fit these regression models, have a very low explanatory power for toxic occurrences. Regardless, our models indicate several participant characteristics having significant associations, which we detail in the following.

Key finding 4: *Although occurrences of toxic comments are significantly associated with several participant characteristics, these have low explanatory power for toxicity in OSS PR contexts.*

Characteristics of authors of toxic comments: Our results suggest significantly lower odds of women authoring toxic comments among PRF-L and PRF-H groups. Similarly, newcomers have lower authoring odds among PRF-M and PRF-H groups. Among all three groups, project members are significantly less likely to author toxic comments, and the likelihood of being such an author significantly decreases with GitHub tenure. The likelihood of authoring toxic comments significantly increased with project tenure among PRF-L and PRF-H groups. Our results support Miller et al. [2022]'s observation that there are many repeat offenders since the likelihood of authoring toxic comments significantly increases with the prior frequency of such occurrences.

Characteristics of targets of toxic comments: Contrary to our expectations, formed based on results [Gunawardena et al. 2022; Raman et al. 2020; Steinmacher et al. 2015], we did not find any significantly higher odds of women or newcomers being targets of toxicity on GitHub. We noticed the opposite among PRF-H and PRF-L. Similarly, newcomers have significantly lower odds of becoming targets among PRF-H and PRF-M. Being a project member significantly increases the odds of being a target among PRF-L and PRF-M but reduces among PRF-H. Project tenure increases the odds of being a target for PRF-M and PRF-H but reduces among PRF-L. The age of a GitHub account is positively associated with being a target only for PRF-M. Finally, these results suggest a 'quid pro quo,' i.e., prior frequent authoring of toxic comments significantly increases the odds of becoming a target.

Key finding 5: *Women and newcomers are less likely to be either authors or targets of toxic comments in GitHub PR comments. Developers who have authored toxic comments frequently in the past are significantly more likely to repeat and more likely to become toxicity targets.*

5 Discussion

The following subsections compare our findings against prior works and suggest recommendations.

5.1 Potential Explanations of Several Key Findings

The results of our RQ2 (Section 4.2) suggest that project popularity, measured in terms of the 'number of stars,' is associated with increased toxicity. A project's popularity may put pressure on contributors to deliver new features and maintain quality. However, a rapid development pace can cause stress, burnout, and toxicity. We also found toxicity increasing with project age. Our manual investigation of sample projects suggests staleness (i.e., lack of response to issues or PRs) may be a potential reason.

The results of RQ3 (Section 4.3) suggest that review duration and the number of required iterations are positively associated with toxicity, with stronger associations seen among higher PRF groups. These results suggest that frustrations may grow between authors and reviewers due to multiple review iterations, particularly among projects with higher activity levels. We also found a positive

Table 8. Comparison against prior empirical studies investigating anti-social behaviors among OSS project

Study	Method	Sample Size	Sampling Criteria	# factors	Comparison with ours
Raman *et* al. [Raman et al. 2020]	Quantitative	872k issues from 30 popular projects.	i) Training dataset from 'too heated locked issues; ii) Poor performance of their classifier with 47% F1-score.	3	One overlapping factor, which concurs with our finding.
Ferreira *et* al. [Ferreira et al. 2021]	Qualitative	1,545 email threads from Linux.	i) Only rejected patches, ii) Linux kernel maintainers are known to be harsh.	9	Two overlapping factors, where finding for one contradicts, and the other one concurs.
Miller *et* al. [Miller et al. 2022]	Qualitative	100 issue discussions.	i) Small sample, ii) Only locked issue threads.	10	Six overlapping factors, where findings for two contradict, and the remaining four concur.
Egelman *et* al. [Egelman et al. 2020]	Opinion survey	Surveyed 1,397 developers in Google.	i) One organization, ii) lack of quantitative validation	5	Three overlapping factors, where all concur with our finding.
Ours	Mixed, mostly quantitative	2,828 GitHub projects and over 100M comments.	Limitations of ToxiCR [Sarker et al. 2023b] applies.	32	-

association between code complexity and toxicity. This result indicates that complex changes, which are difficult to understand and review, may cause confusion [Ebert et al. 2019] and are more likely to be associated with toxicity.

5.2 Comparison with Prior SE Studies

Our large-scale empirical investigation includes 32 attributes from four categories. Out of those, 11 were investigated in other contexts in prior studies. Therefore, Table 8 compares sample size, projects, and the number of factors and overlaps with our studies against the others to illustrate the novelty and significance of this study. Similar to Miller et al. [2022], profanity is the prevalent toxicity in our randomly sampled dataset. They reported a high share (25%) of entitled issue comments, which are demands to project maintainers as if they had a contractual relationship or obligation [Miller et al. 2022]. However, we found only 3.2% such cases in our sample. Supporting their findings, our results indicate repeat offenders, toxicity increasing with project popularity, long-term project contributors being authors of toxicity, and gaming projects harboring more toxic cases [Miller et al. 2022]. They also reported toxic comments from new GitHub accounts [Miller et al. 2022]. Aligning with this finding, we noticed the likelihood of authoring toxic comments decreased with GitHub tenure. However, contrary to their findings, we notice a lower likelihood of project newcomers authoring toxic comments. Our results also concur with one finding by Raman et al. [2020], as we found a lower likelihood of toxicity among corporate-sponsored projects. During their manual investigation of the Linux kernel, Ferreira et al. [2021] found uncivil comments during reviews of rejected codes. Aligning with their findings, we noticed lower odds of toxicity among accepted PRs. They also reported incivility among project maintainers' feedback [Ferreira et al. 2021]. However, contrasting their findings, we noticed a lower likelihood of toxicity from project members. Egelman et al. [2020] reported a higher likelihood of pushback on large code changes. Our result aligns with this finding, as we found that the odds of toxicity increase with code churn. Raman et al. [2020] reported incivility due to poor-quality code changes. While we did not measure code quality directly, we may use the number of review comments as an indication of code quality

since each review comment indicates an issue identified by a reviewer. Aligning with their findings, our results indicate higher odds of toxicity with the number of review comments.

Potential reasons behind some of our findings contradicting prior studies: While we do not have a concrete answer to why some of our results contradict prior studies, we hypothesize that sampling differences may be a major factor. Prior studies picked samples from contexts where antisocial behaviors are more likely to occur (e.g., locked issues, rejected patches, heated discussions). However, these cases are not very frequent. For example, only 3.1% of PR-linked issues in our sample are locked. We noticed the most differences (i.e., two) against Miller et al. [2022], an exploratory study investigating a sample of 100 locked issues. Although valid for a specific context, their selected cases may not represent broader trends across GitHub. For example, arrogant contributors forcefully demanding acceptance of their pull request is a reasonable cause to lock issue threads. Hence, they obtained 25% entitlement, but such cases are significantly lower among non-locked issue threads. For the same reason, Miller et al. [2022] reported entitled type behavior from newcomers, but our analysis suggests that newcomers are less likelier to author toxic texts than long-term contributors. We also noted a discrepancy compared to Ferreira et al. [2021], who drew their sample from rejected patches of the Linux kernel mailing list and reported uncivil behavior from maintainers. Several Linux kernel maintainers have been known for their blunt communication style for years [Barnes 2020; Vaughan-Nichols 2018]. Our findings suggest that what was reported from LKML may not be a broader trend across the OSS spectrum. These contradictions also highlight the need for a large-scale study with diverse samples to understand the landscape of toxicity better.

5.3 Actionable Recommendations

Due to our study design, we cannot claim causal relationships for the associations identified in this study. However, some of the following recommendations apply only if such relationships exist.

I. Project Maintainers: Our results from RQ3 (i.e., table 5) suggest that delays in fixing bugs or answering user queries may create unhappy users and toxic comments targeted toward maintainers. As a project's popularity grows, maintainers should focus on improving bug resolution since our results also show that a higher bug resolution rate is negatively associated with toxicity. Even if an issue is delayed, maintainers should respond politely and suggest workarounds, if possible, to avoid toxic interactions. From the RQ4 analysis, project tenure is positively associated with toxicity. Therefore, building a positive culture needs to start with project maintainers since they are likelier to be project members with the longest tenures [Vaughan-Nichols 2018]. Supporting prior studies [Belskie et al. 2023; Beres et al. 2021; Miller et al. 2022; Paul 2018], we also found a proliferation of toxicity among gaming projects in RQ2. Therefore, we recommend that maintainers of gaming projects adopt a Code of Conduct and its enforcement mechanism to build a diverse community.

II. Developers: We recommend developers avoid creating pull request contexts that are positively associated with toxicity. For example, our results from RQ3 indicate that delayed pull requests are associated with toxicity. Therefore, reviewers should provide on-time reviews to avoid frustrating authors. Similarly, large code changes are not only bug-prone [Bosu et al. 2015] and difficult to review [Thongtanunam et al. 2017] but also likely to encounter toxicity. Therefore, when possible, creating pull requests with smaller changes is recommended. According to the findings from RQ3, pull requests with a large number of issues indicate poor quality codes and are more likely to receive harsh critiques. Therefore, developers should not create pull requests with changes that do not yet meet the quality standards for a project. A higher number of review iterations also frustrates authors and may cause toxicity. Hence, if possible, reviewers should request all required

changes within a single cycle to avoid back and forth. Complex changes are hard to review and are more likely to receive toxicity. Hence, authors should annotate such changes and include helpful descriptions to avoid confusion [Ebert et al. 2019] as well as toxicity. Even when frustrated or angry, developers should not use toxic languages since developers who use such languages are more likely to become victims (i.e., findings from RQ4). Finally, while contrary to prior evidence, we find that women and newcomers are less likely to be targets of toxicity in RQ4, we still recommend long-term contributors avoid such language if such persons are present in a discussion since toxicity not only dissuades newcomers from becoming a part of the communities [Steinmacher et al. 2015] but also disproportionately hurts minorities [Gunawardena et al. 2022].

III. Prospective joiners: If a newcomer wants to avoid negative experiences associated with toxic cultures, we recommend they start with a corporate-sponsored OSS project that matches their expertise and interests. We also recommend such contributors avoid gaming or stale projects.

IV. Researchers: We found variations among terminologies used for almost identical concepts among SE studies investing in anti-social behaviors. We also noticed conflicting opinions about whether a particular category should be considered anti-social. Since existing schemes are primarily based on the decisions of the respective researchers, they may not reflect the broader OSS community. Therefore, existing identification tools based on these schemes may not align with OSS developers' needs and would fail to achieve broader adoption. Moreover, recent research suggests whether a text should be considered toxic depends on various demographic characteristics [Goyal et al. 2022]. Hence, understanding the opinions of the broader OSS community and how their demographics influence perspectives of toxicity is essential to developing a custom mitigation strategy.

6 Threats to Validity

Internal Validity Our selection of 2,828 GitHub projects based on our sampling method threatens internal validity. GitHub hosts over 284 million projects, and mining all of them is infeasible. We defined six filtering criteria to reduce this sample space to 89k without excluding projects with significant communication and collaboration. A lower threshold for the number of contributors or stars would increase the number of projects in this sample and may potentially change our results. We applied a stratified sampling strategy to categorize the projects according to PR activity to encounter this threat. Therefore, threats due to threshold selection are more likely to influence only the PRF(L) group since most of the projects with a lower number of contributors or stars would fall under this group. However, there is no evidence that changing these thresholds would significantly alter the results, even for the PRF(L). Our selection of the list of attributes represents another threat to internal validity. Prior studies have found various factors such as politics or ideology triggering toxicity [Miller et al. 2022]. However, we could not investigate those factors due to the unavailability of automated mechanisms to identify such scenarios at a large scale. This study only investigates automatically measurable factors that may be associated with toxicity.

Construct Validity Our *(first)* threat in this category is due to using ToxiCR [Sarker et al. 2023b] to identify toxic comments automatically. Our validation of ToxiCR found 88.88% precision, which is within the sampling error margin reported by ToxiCR's authors. ToxiCR has false positives in approximately one out of 10 cases. Similarly, ToxiCR has a false negative rate of between 10-14%. Hence, these false positives and negatives may have influenced our results if ToxiCR is biased for/against any particular attributes (e.g., review interval or woman) included in our study. However, we do not have any evidence of such biases. *(Second)*, our manual labeling scheme to identify the nature of toxicities to answer RQ1 is a threat. Although multiple SE studies have studied antisocial behaviors, no agreed-upon scheme exists. Moreover, researchers from NLP and SE domains have used different terminologies to characterize similarly subjective concepts. To mitigate this threat,

we have analyzed existing studies [Egelman et al. 2020; Ferreira et al. 2022, 2021; Miller et al. 2022; Sarker et al. 2023b] and aggregated their categories to build our scheme. We acknowledge the subjectivity bias, where another set of researchers disagree with our scheme and definitions. *(Third)*, our manual labeling process may have subjectivity biases. We prepared a scheme with category definitions and examples to mitigate this threat. The labelers had a discussion session before starting to build an agreed-upon understanding. We also measured inter-rater reliability to assess your labeling process. *(Finally)*, automated gender resolution is another threat. We followed a procedure as the ones in multiple recent empirical studies [Bosu and Sultana 2019; Santamaría and Mihaljević 2018; Sultana et al. 2023]. We used multiple gender resolution tools, considered users' location and profile photos, and searched LinkedIn to improve resolution accuracy. This resolution process may be subject to misclassification. We did not attempt to identify non-binary genders since we are unaware of any automated resolution of those without users' inputs.

External Validity The nature of toxicities in an OSS project may depend on factors such as project domain, governance, the number of contributors, and project age. We used a stratified random sampling strategy to select 2,828 projects representing diverse demographics, including the top OSS projects on GitHub, such as Kubernetes, Odoo, PyTorch, Rust, Ansible, pandas, rails, Django, numpy, angular, flutter, CPython, and node.js. Yet, our sample and its results may not adequately represent the entire OSS spectrum.

Conclusion Validity We assess the reliability of our models using goodness-of-fit metrics and log-likelihood tests. Hence, we do not anticipate any threats from the results obtained from our models. Although our models account for various confounding variables, these models identified associations between dependents and predictors, and no causal relationships can be implied.

7 Conclusion

We conducted a large-scale mixed-method empirical study of 2,828 GitHub-based OSS projects to understand the nature of toxicities on GitHub and how various measurable characteristics of a project, a pull request's context, and participants associate with their prevalence. We found profanity to be the dominant form of toxicity on GitHub, followed by trolling and insults. While a project's popularity is positively associated with the prevalence of toxicity, its issue resolution rate has the opposite association. Corporate-sponsored projects are less toxic, but gaming projects are seven times more likely than non-gaming ones to have a high volume of toxicities. OSS developers who have authored toxic comments in the past are significantly more likely to repeat them and become toxicity targets. Based on the results of this study and our experience conducting it, we provide recommendations to OSS contributors and researchers.

Acknowledgment

This research is partially supported by the US National Science Foundation under Grant No. 1850475 and 2340389 and funding from The University of Nebraska at Omaha, College of Information Science and Technology. The findings of this research do not necessarily provide the views of the National Science Foundation. Jaydeb Sarker was affiliated with the Wayne State University when this research was conducted.

Data Availability

We have made our dataset and source code publicly available at: 10.5281/zenodo.14802294.

References

Khaled Albusays, Pernille Bjorn, Laura Dabbish, Denae Ford, Emerson Murphy-Hill, Alexander Serebrenik, and Margaret-Anne Storey. 2021. The diversity crisis in software development. *IEEE Software* 38, 2 (2021), 19–25.

Paul Allison. 2014. Prediction vs. causation in regression analysis. *Statistical Horizons* 703 (2014).

Anonymous Author. 2014. Leaving Toxic Open Source Communities. https://modelviewculture.com/pieces/leaving-toxic-open-source-communities.

Hayden Barnes. 2020. Toxicity in Open Source. https://boxofcables.dev/toxicity-in-linux-and-open-source/.

Mohammad R Basirati, Marko Otasevic, Koushyar Rajavi, Markus Böhm, and Helmut Krcmar. 2020. Understanding the relationship of conflict and success in software development projects. *Information and Software Technology* 126 (2020).

Anass Bayaga. 2010. Multinomial Logistic Regression: Usage and Application in Risk Analysis. *Journal of applied quantitative methods* 5, 2 (2010).

Matthew Belskie, Hanlin Zhang, and Bradley M Hemminger. 2023. Measuring toxicity toward women in game-based communities. *Journal of Electronic Gaming and Esports* 1, 1 (2023).

Nicole A Beres, Julian Frommel, Elizabeth Reid, Regan L Mandryk, and Madison Klarkowski. 2021. Don't you know that you're toxic: Normalization of toxicity in online gaming. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–15.

Nicolas Bérubé, Gita Ghiasi, Maxime Sainte-Marie, et al. 2020. Wiki-Gendersort: Automatic gender detection using first names in Wikipedia. (2020).

Meghana Moorthy Bhat, Saghar Hosseini, Ahmed Hassan, Paul Bennett, and Weisheng Li. 2021. Say 'YES'to positivity: Detecting toxic language in workplace communications. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2017–2029.

Amiangshu Bosu, Michaela Greiler, and Christian Bird. 2015. Characteristics of useful code reviews: An empirical study at microsoft. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 146–156.

Amiangshu Bosu and Kazi Zakia Sultana. 2019. Diversity and inclusion in open source software (OSS) projects: Where do we stand?. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–11.

Kevin Daniel André Carillo, Josianne Marsan, and Bogdan Negoita. 2016. Towards Developing a Theory of Toxicity in the Context of Free/Open Source Software & Peer Production Communities. *SIGOPEN 2016* (2016).

Jithin Cheriyan, Bastin Tony Roy Savarimuthu, and Stephen Cranefield. 2021. Towards offensive language detection and reduction in four Software Engineering communities. In *Evaluation and Assessment in Software Engineering*. 254–259.

William G Cochran. 1977. Sampling techniques. *Johan Wiley & Sons Inc* (1977).

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

Sophie Cohen. 2021. Contextualizing toxicity in open source: a qualitative study. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1669–1671.

Daniel Alencar da Costa, Shane McIntosh, Christoph Treude, Uira Kulesza, and Ahmed E Hassan. 2018. The impact of rapid release cycles on the integration delay of fixed issues. *Empirical Software Engineering* 23 (2018), 835–904.

Ozren Dabic, Emad Aghajani, and Gabriele Bavota. 2021. Sampling Projects in GitHub for MSR Studies. In *18th IEEE/ACM International Conference on Mining Software Repositories, MSR 2021*. IEEE, 560–564.

Kyle Daigle. 2023. Octoverse: The state of open source and rise of AI in 2023. https://github.blog/2023-11-08-the-state-of-open-source-and-ai/.

Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*. 86–95.

Alan Diggs. 2021. Windows is sh*t:' Linux Users and The Technical Superiority Problem. https://medium.com/linuxforeveryone/windows-is-sh-t-linux-users-and-the-technical-superiority-problem-196a597aa860/.

Felipe Ebert, Fernando Castor, Nicole Novielli, and Alexander Serebrenik. 2019. Confusion in code reviews: Reasons, impacts, and coping strategies. In *2019 IEEE 26th international conference on software analysis, evolution and reengineering (SANER)*. IEEE, 49–60.

Carolyn D Egelman, Emerson Murphy-Hill, Elizabeth Kammer, Margaret Morrow Hodges, Collin Green, Ciera Jaspan, and James Lin. 2020. Predicting developers' negative feelings about code review. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE, 174–185.

Eran Eidinger, Roee Enbar, and Tal Hassner. 2014. Age and gender estimation of unfiltered faces. *IEEE Transactions on information forensics and security* 9, 12 (2014), 2170–2179.

Isabella Ferreira, Bram Adams, and Jinghui Cheng. 2022. How heated is it? Understanding GitHub locked issues. In *Proceedings of the 19th International Conference on Mining Software Repositories*. 309–320.

Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The" Shut the f** k up" Phenomenon: Characterizing Incivility in Open Source Code Review Discussions. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–35.

Isabella Ferreira, Ahlaam Rafiq, and Jinghui Cheng. 2024. Incivility detection in open source code review and issue discussions. *Journal of Systems and Software* 209 (2024), 111935.

Ben Foley. 2018. What is regression analysis and why should I use it. *Source: https://www. surveygizmo. com/resources/blog/regression-analysis* (2018).

Iginio Gagliardone, Danit Gal, Thiago Alves, and Gabriela Martinez. 2015. *Countering online hate speech*. Unesco Publishing.

GitHub, Inc., Kenyatta Forbes, Kevin Xu, Jeffrey Luszcz, Margaret Tucker, Eva Maxfield Brown, Peter Cihon, Mike Linksvayer, Ashley Wolf, Lukas Speiß, Kevin Crosby, and Jason Meridth. 2024. GitHub Open Source Survey 2024. doi:10.5281/zenodo.13989018

Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An exploratory study of the pull-based software development model. In *Proceedings of the 36th international conference on software engineering*. 345–355.

Georgios Gousios and Diomidis Spinellis. 2012. GHTorrent: GitHub's data from a firehose. In *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*. IEEE, 12–21.

Kruti Goyal, Kartikey Agarwal, and Rishi Kumar. 2017. Face detection and tracking: Using OpenCV. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Vol. 1. IEEE, 474–478.

Nitesh Goyal, Ian D Kivlichan, Rachel Rosen, and Lucy Vasserman. 2022. Is your toxicity my toxicity? exploring the impact of rater identity on toxicity annotation. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–28.

Isuru Gunasekara and Isar Nejadgholi. 2018. A review of standard text classification practices for multi-label toxicity identification of online content. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*. 21–25.

Sanuri Dananja Gunawardena, Peter Devine, Isabelle Beaumont, Lola Garden, Emerson Rex Murphy-Hill, and Kelly Blincoe. 2022. Destructive Criticism in Software Code Review Impacts Inclusion. (2022).

Inge S Helland. 1987. On the interpretation and use of R2 in regression analysis. *Biometrics* (1987), 61–69.

D Hinkle, HW Jurs, and W Wiersma. 1998. Applied Statistics for the behavioral sciences.

Vincent Jacques. 2024. Pygithub. https://pygithub.readthedocs.io/.

Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian. 2016. An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering* 21 (2016), 2035–2071.

Oleksii Kononenko, Olga Baysal, Latifa Guerrouj, and Yaxin Cao. 2015. Investigating code review quality: Do people and participation matter? 111–120. doi:10.1109/ICSM.2015.7332457

Robin M Kowalski, Gary W Giumetti, Amber N Schroeder, and Micah R Lattanner. 2014. *Bullying in the digital age: a critical review and meta-analysis of cyberbullying research among youth*. Vol. 140. American Psychological Association.

Richard A. Kronmal. 1993. Spurious Correlation and the Fallacy of the Ratio Standard Revisited. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 156, 3 (1993), 379–392. http://www.jstor.org/stable/2983064

Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing toxic content classification for a diversity of perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 299–318.

Nolan Lawson. 2017. What it feels like to be an open-source maintainer. https://nolanlawson.com/2017/03/05/.

Megan Lindsay, Jaime M Booth, Jill T Messing, and Jonel Thaller. 2016. Experiences of online harassment among emerging adults: Emotional reactions and the mediating role of fear. *Journal of interpersonal violence* 31, 19 (2016), 3174–3195.

J Scott Long and Jeremy Freese. 2006. *Regression models for categorical dependent variables using Stata*. Vol. 7. Stata press.

Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian Kästner. 2022. "Did You Miss My Comment or What?" Understanding Toxicity in Open Source Discussions. In *In 44th International Conference on Software Engineering (ICSE'22)*.

Emerson Murphy-Hill, Ciera Jaspan, Carolyn Egelman, and Lan Cheng. 2022. The pushback effects of race, ethnicity, gender, and age in code review. *Commun. ACM* 65, 3 (2022), 52–57.

Dawn Nafus. 2012. 'Patches don't have gender': What is not open in open source software. *New Media & Society* 14, 4 (2012), 669–683.

Nachiappan Nagappan and Thomas Ball. 2005. Use of relative code churn measures to predict system defect density. In *Proceedings of the 27th international conference on Software engineering*. 284–292.

Nachiappan Nagappan and Thomas Ball. 2007. Using software dependencies and churn metrics to predict field failures: An empirical case study. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. IEEE, 364–373.

Nicole Novielli, Fabio Calefato, Filippo Lanubile, and Alexander Serebrenik. 2021. Assessment of off-the-shelf SE-specific sentiment analysis tools: An extended replication study. *Empirical Software Engineering* 26, 4 (2021), 77.

Nicole Novielli, Daniela Girardi, and Filippo Lanubile. 2018. A benchmark study on sentiment analysis for software engineering research. In *Proceedings of the 15th International Conference on Mining Software Repositories*. 364–375.

Christopher A Paul. 2018. *The toxic meritocracy of video games: Why gaming culture is the worst*. U of Minnesota Press.

Huilian Sophie Qiu, Bogdan Vasilescu, Christian Kästner, Carolyn Egelman, Ciera Jaspan, and Emerson Murphy-Hill. 2022. Detecting interpersonal conflict in issues and code review: cross pollinating open-and closed-source approaches. In *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society*. 41–55.

Md Shamimur Rahman, Zadia Codabux, and Chanchal K Roy. 2024. Do Words Have Power? Understanding and Fostering Civility in Code Review Discussion. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 1632–1655.

Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*. 57–60.

Jim Salter. 2021. The Perl Foundation is fragmenting over Code of Conduct enforcement. https://arstechnica.com/gadgets/2021/08/the-perl-foundation-is-fragmenting-over-code-of-conduct-enforcement/.

Lucía Santamaría and Helena Mihaljević. 2018. Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science* 4 (2018), e156.

Jaydeb Sarker, Sayma Sultana, Steve Wilson, and Amiangshu Bosu. 2023a. ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. In *Proceedings of the 17th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*.

Jaydeb Sarker, Asif Kamal Turzo, and Amiangshu Bosu. 2020. A benchmark study of the contemporary toxicity detectors on software engineering interactions. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 218–227.

Jaydeb Sarker, Asif Kamal Turzo, and Amiangshu Bosu. 2025. Replication Package. https://doi.org/10.5281/zenodo.14802294

Jaydeb Sarker, Asif Kamal Turzo, Ming Dong, and Amiangshu Bosu. 2023b. Automated Identification of Toxic Code Reviews Using ToxiCR. *ACM Transactions on Software Engineering and Methodology* 32, 5, Article 118 (jul 2023), 32 pages. doi:10.1145/3583562

WS Sarle. 1990. SAS/STAT User's Guide: The Varclus Procedure. SAS Institute. *Inc., Cary, NC, USA* (1990).

Thomas J Smith and Cornelius M McKenna. 2013. A comparison of logistic regression pseudo R2 indices. *Multiple Linear Regression Viewpoints* 39, 2 (2013), 17–26.

Laerd Statistics. 2013. Spearman's rank-order correlation. *Laerd Statistics* (2013).

Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*. 1379–1392.

Igor Steinmacher, Igor Wiese, Ana Paula Chaves, and Marco Aurélio Gerosa. 2013. Why do newcomers abandon open source software projects?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 25–32.

Vikram N Subramanian, Ifraz Rehman, Meiyappan Nagappan, and Raula Gaikovina Kula. 2020. Analyzing first contributions on github: What do newcomers do? *IEEE Software* 39, 1 (2020), 93–101.

Sayma Sultana. 2022a. Identification and mitigation of gender biases to promote diversity and inclusion among open source communities. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. 1–5.

Sayma Sultana. 2022b. Identifying Sexism and Misogyny in Pull Request Comments. In *37th IEEE/ACM International Conference on Automated Software Engineering*. 1–3.

Sayma Sultana, Jaydeb Sarker, and Amiangshu Bosu. 2021. A Rubric to Identify Misogynistic and Sexist Texts from Software Developer Communications. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–6.

Sayma Sultana, Asif Kamal Turzo, and Amiangshu Bosu. 2023. Code reviews in open source projects: how do gender biases affect participation and outcomes? *Empirical Software Engineering* 28, 4 (2023), 92.

Patanamon Thongtanunam, Shane McIntosh, Ahmed E Hassan, and Hajimu Iida. 2017. Review participation in modern code review: An empirical study of the android, Qt, and OpenStack projects. *Empirical Software Engineering* 22 (2017), 768–817.

Asif Kamal Turzo and Amiangshu Bosu. 2024. What makes a code review useful to opendev developers? an empirical investigation. *Empirical Software Engineering* 29, 1 (2024), 6.

Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511.

Steven Vaughan-Nichols. 2018. Linus Torvalds takes a break from Linux. https://www.zdnet.com/article/linus-torvalds-takes-a-break-from-linux/.

Michael R Veall and Klaus F Zimmermann. 1994. Evaluating Pseudo-R2's for binary probit models. *Quality and Quantity* 28, 2 (1994), 151–164.

Sawyer X. 2021. I Am Stepping Down from PSC and Core, Effective Immediately. https://perl.topicbox.com/groups/perl-core/T7a4f1bf9e069641f-Mebbcc218eb006f0da34c7a41

Li Xu, Chris Gotwalt, Yili Hong, Caleb B King, and William Q Meeker. 2020. Applications of the fractional-random-weight bootstrap. *The American Statistician* 74, 4 (2020), 345–358.

Sara Zaheri, Jeff Leath, and David Stroud. 2020. Toxic comment classification. *SMU Data Science Review* 3, 1 (2020), 13.

Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. 2021. A comparative study of using pre-trained language models for toxic comment classification. In *Companion Proceedings of the Web Conference 2021*. 500–507.

Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. 2020. How has forking changed in the last 20 years? a study of hard forks on github. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 445–456.

Received 2024-09-13; accepted 2025-01-14