Social Network Analysis and Mining (2024) 14:146
https://doi.org/10.1007/s13278-024-01312-y

ORIGINAL ARTICLE

Popularity and discourse in tech-related online communities

Abduljaleel Al Rubaye[1] · **Gita Sukthankar**[1]

Received: 29 February 2024 / Revised: 19 July 2024 / Accepted: 19 July 2024

Abstract
Tech-related online communities on GitHub, Reddit, and Stack Overflow are an invaluable resource for software engineers, allowing them to find solutions to problems and connect with other professionals. Much of the discourse on these platforms is conducted using commenting mechanisms in which one user responds to content posted by another user. Even though these communities lack formal organizational structures, these technologists are often followed by other software developers who monitor their posts; users who regularly post useful solutions are recognized using platform-specific mechanisms such as stars or karma points. This article investigates the relationship between popularity and discourse in tech-related online communities. To do this, we create comment timelines from sequences of user interactions and extract commenting networks from comment response patterns. We study how the popularity of the post authors and other commenters shapes community interaction. Although there are some commonalities, there are distinct differences between the commenting behavior of GitHub users vs. Reddit and Stack Overflow. Popularity influences the length of commenting timelines on GitHub, whereas this effect isn't observed on Reddit or Stack Overflow. However on all three platforms, user seniority appears to have a stronger impact on the structure of commenting networks than popularity. By understanding how popularity affects user interactions, we can design online communities that are more effective at supporting knowledge sharing and problem solving. Our cross-platform comment dataset is available for download at: https://bit.ly/abdul-dissertation-dataset.

Keywords Social coding platforms · Tech-related forums · Popularity · Knowledge sharing

1 Introduction

Modern software developers rely on knowledge sharing communities, including social coding platforms, question and answer sites, and news forums, to keep up with ever-changing technologies. Although many users passively search for solutions without contributing, others actively participate in these knowledge sharing forums by posting code and articles, commenting, and upvoting good solutions. Most platforms have mechanisms in place to allow users to publicly endorse the contributions of others. Expert software developers who regularly provide valuable content may be viewed as "gurus" or "wizards" by the community. Identifying the experts in a online forum can make it easier to find high quality posts; however high quality code and accurate technical responses are often generated by users who possess few followers. Although many high status users earn their popularity through the regular production of high quality content, some users more actively "game" platform popularity measures (Richterich 2014).

Popular users in online communities can influence others' actions and attitudes. Their views and actions can shape the conversation, setting the agenda, tone, and behavior of others. When a popular user expresses a strong viewpoint or shares specific content, it can trigger a chain reaction of similar opinions and content among other users creating temporal bursts in posting activity (Gorovits et al. 2021). This article tackles the research question: how does popularity affect the commenting behavior of other users within the same community? We hypothesize that popular users affect both comment timelines and commenting networks. This article builds on previous work that appeared in Al-Rubaye and Sukthankar (2023) by presenting a new analysis of the centrality of popular users within network

✉ Gita Sukthankar
gita.sukthankar@ucf.edu

Abduljaleel Al Rubaye
aalrubaye@ucf.edu

[1] Department of Computer Science, University of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816, USA

Published online: 31 July 2024

communities. Our aim is to understand the impact of social role (newbie, rising star, longstanding member, and tech guru) on user engagement. We introduce the Commenting Community Engagement Score (CCES) for quantifying contributions to community discourse and have made our commenting behavior dataset publicly available to stimulate further research in this area. Reassuringly, our study reveals that in tech communities, popularity plays less of a role in influencing debate compared to purely social discussion forums.

2 Related work

In his survey on social media popularity, Woods (2023) states: "Although the act of clicking a like button may seem simple, perhaps trivial, its causes and contingencies may be varied and complex." He endorses the usage of the Barlund (2008) transactional model to simulate the back and forth communication style of social media where users are constantly engaged in context-dependent impression management.

Tech-related commenting represents a very specialized category of user posting behavior in which users are participating an existing technical discussion, simultaneously engaging in problem-solving behavior while questing for social capital. Previous studies have been conducted on commenting behavior on GitHub (Destefanis et al. 2018), Reddit (Choi et al. 2015; Buntain and Golbeck, 2014), and Stack Overflow (Zhang et al. 2019; Sengupta and Haythornthwaite, 2020). Many GitHub users exclusively post comments and never open issues or commit code changes. Destefanis et al. (2018) found that the comments from these users are less positive, less polite, and also less emotive than comments from contributors. Stack Overflow encourages constructive comments by awarding badges like "pundit" and "commentator"; Anderson et al. (2013) introduced a utility-based model that predicts how these badges affect user engagement. Commenting behavior on Stack Overflow is also correlated with user characteristics, such as experience level and social activity (Zhang et al. 2019).

The comments of a relatively small group of users can affect the discourse of the whole community. Choi et al. (2015) found that a small number of users drive the most critical conversations on Reddit. Their study also showed that users involved in multiple areas tend to participate more actively in conversations. Sengupta and Haythornthwaite (2020) studied the impact of commenting on community interactions through content-based comment analysis, categorizing comments based on how well they support knowledge sharing and learning. On Reddit, there are users who assume role of "answer-person" within the community; these users can be identified using network structure alone, without any content analysis (Buntain and Golbeck 2014). Our research extends on previous work by presenting an in depth analysis of comment timelines, which have not been used in previous studies.

3 Method

3.1 Dataset

To perform a cross-platform analysis of user commenting behavior, we gathered datasets from platforms that host tech-related discussions: GitHub, Reddit, and Stack Overflow. GitHub is a social coding platform for collaboration between software developers that offers code version control. Collaborative communication on GitHub is facilitated through discussion forums and issue reporting. Stack Overflow is a community-based Q & A website dedicated to answering technical problems. Reddit allows for asynchronous discussions between users; it comprises communities called subreddits dedicated to specific topics, including software development.

We developed a Python crawler utilizing platform-specific REST APIs. This crawler was then used to retrieve *posts*: GitHub issues, Reddit submissions, and Stack Overflow questions. The current popularity of AI has led to a boom in posting activity in this area so we selected keywords related to artificial intelligence, machine learning, and robotics (Fig. 1). The crawler used the same set of keywords to retrieve posts across all three platforms. Social media websites have mechanisms in place to support load balancing and throttle demands. Since these constraints are designed to obstruct high-speed data gathering, we limited the number of data objects fetched to 5000 per platform. Our cross-platform comment dataset is available for download at: https://bit.ly/abdul-dissertation-dataset.

3.2 General statistics

The commenting mechanisms on online social networks let users leave comments in response to an initial post. Users can also comment on other users' comments. We divide users into two categories:

- *Authors* users who initiate conversations by creating Issues on GitHub, adding a submission on Reddit, or posting a question on Stack Overflow
- *Commenters* users who have submitted comments.

Table 1 provides some comment-specific statistics about our dataset. It can be observed that our GitHub

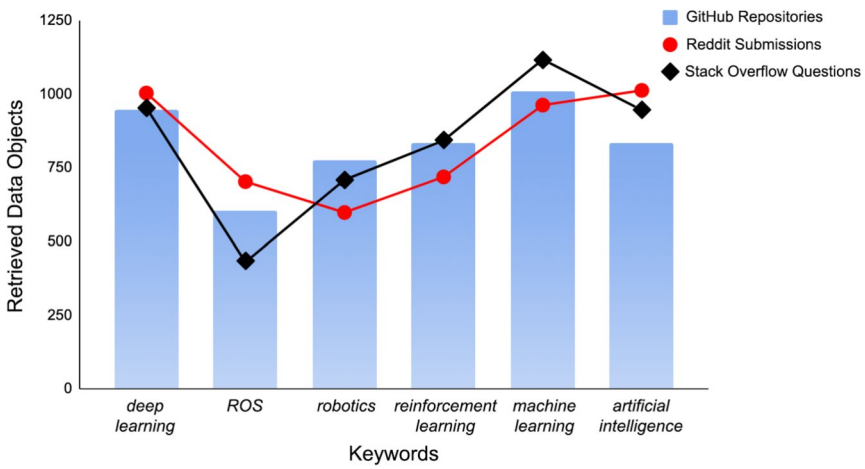

Fig. 1 Retrieved posts grouped by search term

Table 1 Comment-related statistics

	GitHub	Reddit	Stack OF
Avg. comment count per post	17.24	6.38	7.29
Avg. authors' comments	5.53	1.46	1.50
Avg. commenters count	3.87	5.34	4.82
% One-time commenters	38%	89%	74%

dataset has a higher number of comments per post, but that those comments are generated from a smaller number of commenters. On average, 20–23% of the comments in Reddit and Stack Overflow are made by the posts' authors, while authors in GitHub showed a slightly higher rate of commenting, submitting 32% of all the collected comments. This is unsurprising given that GitHub comment chains are often focused on resolving code issues and involve in-depth discussion between a small number of developers.

3.3 Comment timeline

To explore time-related commenting behaviors, we constructed comment timelines. Figure 2 depicts the comment timeline and the distribution of daily comments for one GitHub Issue. In our dataset, GitHub commenters spent around four months on average between creating and closing issues. This time interval is substantially larger than the time required to close questions on Stack Overflow. On Reddit, the majority of comments are submitted over a few days. Comment density indicates how closely or widely dispersed the comments in a specific post's comment timeline are. The density of comments in open source software platforms may be affected by project characteristics such as code size and project age, as well as other factors such as code functionality and code quality (Arafat and Riehle 2009). We leverage comment density to infer implicit network connections between commenters engaged in short bursts of high frequency communication.

3.4 User popularity

The mechanisms for quantifying popularity on social media are platform-specific. For instance, GitHub allows users to follow each other, much like most online networks. We utilize the follower count to measure users' popularity on GitHub as was done in other studies (Al-Rubaye and Sukthankar 2020; Blincoe et al. 2016). On Reddit, users' popularity can be measured by *karma*. Users can vote on a submission or comment, and Reddit uses an algorithm to calculate *karma* from user votes. *Karma* encourages

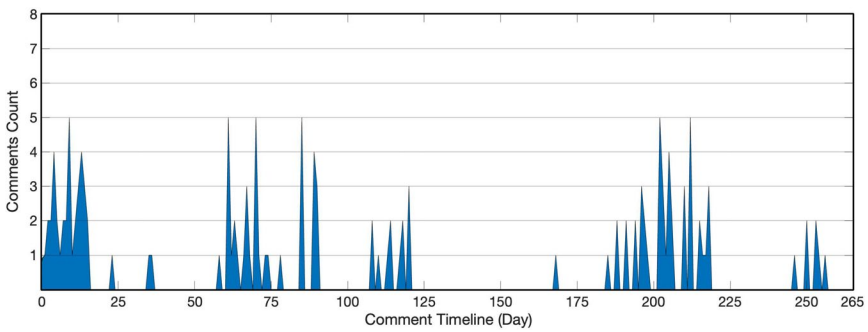

Fig. 2 An example comment timeline for one GitHub Issue

and enables community engagement, and users seek to boost their scores by sharing their opinions, knowledge, and expertise (Richterich 2014). On Stack Overflow, users earn rewards and gain reputation through their activities (Movshovitz-Attias et al. 2013). Users gain (or lose) reputation based on how many people vote on their postings. Posting good questions and meaningful replies is the most effective strategy to acquire reputation. Various studies on the Stack Overflow platform (e.g., (Merchant et al. 2019)) suggest that *reputation* score is highly correlated to users' popularity. We gather popularity statistics for all authors and commenters in our dataset.

3.5 Role categorization

To analyze the effects of user features on community discourse, users were clustered into four social roles using popularity and seniority. These can be colloquially described as follows:

- *Newbies* (SR1) new users who also have low popularity scores.
- *Rising Stars* (SR2) users who rapidly achieve a high level of popularity. This is unusual since there is often a correlation between popularity and account age.
- *Longstanding Members* (SR3) longstanding members of tech communities who do not enjoy a high level of popularity. Many of them post infrequently.
- *Tech Gurus* (SR4) experienced users who have gained the confidence of others through their interactions.

Table 2 shows the distribution of commenter social roles on the three platforms. On GitHub and Stack Overflow, most commenters fall in the category of members (SR3); whereas on Reddit, commenters are split between newbies (SR1) and members (SR3). Across all three platforms, rising stars (SR2) are the rarest category. Gurus (SR4) are most commonly found commenting on Stack Overflow. Figure 3 depicts the distribution of commenters' count ratios based on their social role classification across all three platforms.

Table 2 Role Distribution

	GitHub (%)	Reddit (%)	Stack OF (%)
SR1 Newbies	19.4	38.2	9.8
SR2 Stars	5.0	7.8	2.2
SR3 Members	61.8	36.5	65.4
SR4 Gurus	13.8	17.5	22.6

4 Results

Our study examined three aspects of social commenting behavior. First, we looked at the general patterns of commenting behavior among users. Second, we compared the impact of popularity on the timelines of posts. Finally, we investigated the social commenting communities that form around posts.

4.1 Commenting behaviours

Figure 4 shows the distribution of the posts in our dataset based on the comment counts. According to the illustrated data, on GitHub, over 82% of the issues had 20 or fewer comments, whereas, on Reddit, nearly 95% of the submissions had eight or fewer comments. Similarly, 96% of the questions on Stack Overflow received only five or fewer comments. Moreover, we observed that there are a considerable number of posts that received zero or one comments. Like many social media datasets (Barabási 2002; Takac and Zabovsky 2012), our data follows a power law distribution in which a small number of posts have very long comment chains but most do not.

Commenters can be divided into two groups based on the extent of their commenting participation:

- *One-time commenters* Users that contribute by leaving only one comment on a post.
- *Multi-time commenters* Active users who comment more than once during a post's timeline.

The investigation reveals that individuals comment on GitHub issues an average of 4.45 times (the ratio of the average number of commenters over the comment count), indicating a significant number of repeat commenters.

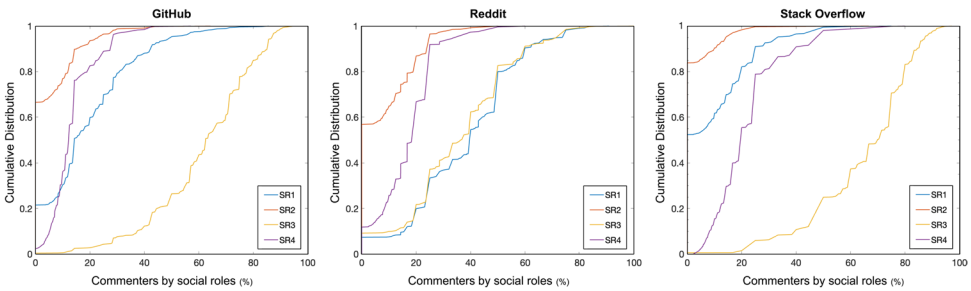

Fig. 3 Cumulative commenter distribution broken down by role

Approximately 38% of GitHub users are one-time commenters. In contrast, this ratio is substantially higher on the other two platforms, where 89% and 74% of Reddit and Stack Overflow users, respectively, leave only one comment.

Figure 5 depicts the distribution of captured posts based on the commenting ratio of their contributors. Reddit and Stack Overflow have right-skewed distributions that contain a larger number of one-time commenters, whereas commenters on GitHub are more likely to make repeated comments.

This reveals clear platform specific differences in user commenting behavior. On social coding platforms that involve groups of developers and reviewers working as a team (e.g., GitHub), the results suggest that users exhibit conversational behavior when posting. Posts are utilized as part of an ongoing conversation to review work and exchange information. On Reddit and Stack Overflow, commenters are more likely to express their viewpoint once and refrain from further comments.

4.2 User popularity effect

We investigated the popularity of two user categories, authors and most popular commenters (MPCs), to understand the impact of popularity on other communication-related features.

4.2.1 Authors' popularity effect

Table 3 details the correlation of the author's popularity to the following measurements: (1) number of comments left on the same post, (2) the total number of commenters that engaged with the post, (3) the popularity of the commenters, and (4) the author's comments count. Across all three platforms, we observe a statistically significant positive correlation between the authors' popularity and the number of comments to posts they initiate, as well as to the total number of commenters engaging with their posts. The correlation between authors popularity and their own comment count is not statistically significant; it isn't the case that simply commenting more guarantees an increase in popularity. On GitHub and Stack Overflow, we found that

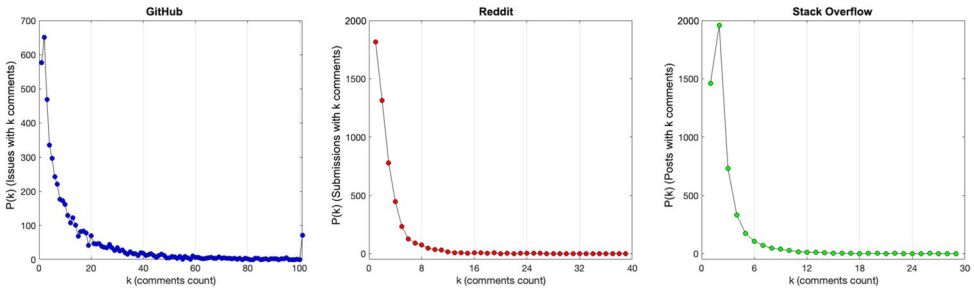

Fig. 4 Distribution of posts based on extracted comments

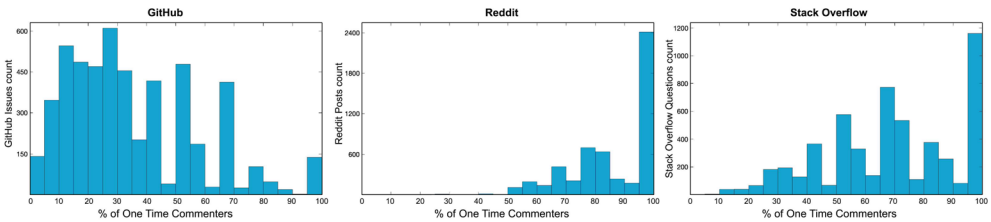

Fig. 5 Distribution of posts based on the ratio of single commenters

Table 3 Author's popularity correlations to other comment-related measurements

	GitHub		Reddit		Stack OF	
	r-val	*p* val	r-val	*p* val	r-val	*p* val
Comments count	0.30	0.01	0.12	0.00	0.39	0.02
Commenters count	0.21	0.00	0.10	0.00	0.31	0.03
Author's comment count	− 0.05	0.29	− 0.03	0.49	− 0.11	0.09
Commenters' popularity	0.38	0.01	0.08	0.57	0.40	0.01

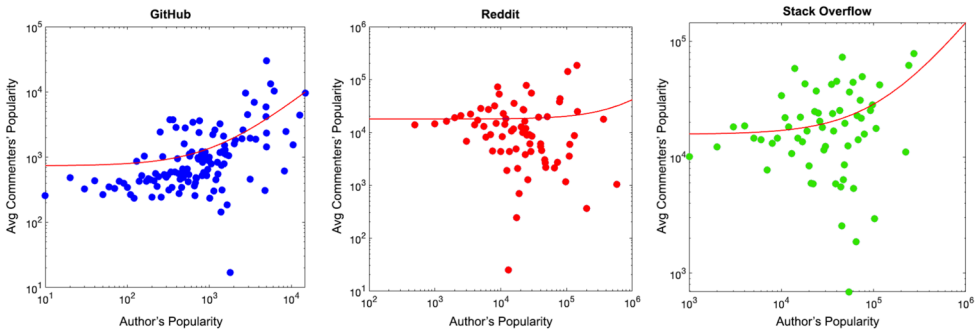

Fig. 6 Log-log plot of authors' popularity correlation with the average commenters' popularity

popular authors are more likely to receive comments from popular users. However, this trend is not observed on Reddit (Fig. 6).

4.2.2 MPCs' popularity effect

MPCs are the commenters with the highest popularity all the users who contribute to a post. Like other commenters, MPCs may contribute to a post by commenting more than once. Our data shows that 85% of GitHub posts include MPCs with multiple comments per post, resulting in 27% of the total number of comments. On the other two platforms, MPCs are less likely to make multiple comments. Most of our collected posts in Reddit and Stack Overflow (79% and 62%, respectively) have only one MPC contribution. This indicates that MPCs drive more of the conversation in GitHub.

MPC's Comment Tail From the comment timelines, we extract a comment tail in order to quantify changes in the number and the popularity of the commenters before and after the MPC's comment.

Assuming the comment timeline of the post (P) includes the comments $\{c_0, c_1, ..., c_k, c_{k+1}, ..., c_{n-1}, c_n\}$, (n) is the total number of the comments, (k) is the order of the MPC's comment, and $k \leq n$. Therefore we define the comment tail to be equal to:

$$CT_P = \{c_{k+1}, c_{k+2}, ..., c_n\}$$

which is equal to a set of the post (P)'s comments starting from the comment at order (k+1) to the last comment at order (n).

As depicted in Fig. 7 we define three types of comment tails based on their length:

- *Short comment tails* MPCs comment closer to the end of the comment timeline. In these conversations, a smaller

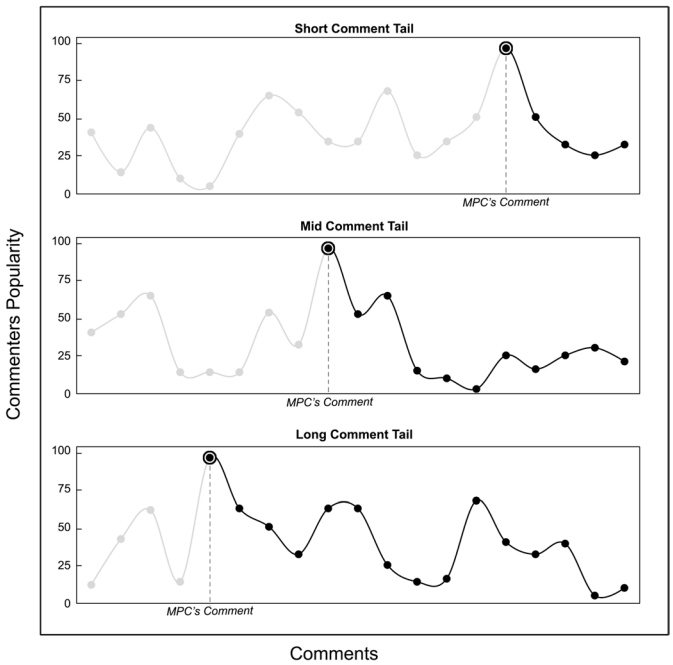

Fig. 7 Three sample comment timelines, each showing one of the comment tail length categories: long, medium, and short. The comment tail includes all comments that are posted after the MPC comments

number of individuals leave comments after the MPCs compared to the ones who comment before the MPC:

$$|CT_P| < |\{c_0, ..., c_{k-1}\}|$$

- *Medium comment tails* MPCs contribute closer to the middle of the conversation, and a similar number of comments are observed before and after the MPC's comment:

$$|CT_P| \simeq |\{c_0, ..., c_{k-1}\}|$$

- *Long comment tails* MPCs tend to comment earlier in the conversation, leading to a longer comment tail. Hence most commenters engage with the post after the MPC comments:

$$|CT_P| > |\{c_0, ..., c_{k-1}\}|$$

From the comment timelines, we extracted each post's MPC-related comment tail and categorize them based on length. Table 4 details the ratio of each category and the commenting rate per user, including the authors, before and after the MPCs comment.

Table 4 MPC comment tail statistics

	GitHub	Reddit	Stack OF
Short comment tail ratio	69.14%	40.39%	43.19%
Medium comment tail ratio	11.79%	20.56%	18.37%
Long comment tail ratio	19.07%	39.05%	38.44%
Comments per user before MPC	3.093	1.026	2.022
Comments per user after MPC	2.181	1.040	1.416
Authors' comments count before MPC	5.938	0.905	1.040
Authors' comments count after MPC	1.795	1.001	0.997

We observed a similar pattern of comment tail lengths on Reddit and Stack Overflow, with similar proportions of short, medium and long tails. There is no noticeable MPC-related change in users' comments count in Reddit and Stack Overflow. In GitHub, however, more than two third of the posts fall in the category of short comment tails. Short comment tails indicate that the MPCs have commented mainly toward the end of the conversation. In other words, GitHub users' commenting behavior changes after an MPC comments; MPCs seem to have the "final word" in the conversation, bringing commenting to a close. We see a similar pattern in the authors' comment rates. In GitHub, there is a noticeable reduction in authors comments after the MPC comment.

Figure 8 depicts the popularity distribution of different types of users. The data show that the posts on all platforms exhibit a similar pattern, where the popularity of the authors is slightly lower than most commenters. The users who comment before or after MPCs have similar levels of popularity on Reddit and Stack Overflow. However, on GitHub, the average popularity of the commenters decreases after the contribution of MPCs.

Table 5 details the correlation between MPCs' popularity to other communication features. There is a positive relationship between the popularity of the post's MPC and the number of comments a post receives on all three platforms. It is likely that a post will receive a higher number of comments if its MPC's popularity is relatively high. Also the other commenters (both before and after the MPC's comment) are likely to have a high popularity themselves. The more popular MPCs are, the higher the likelihood of popular users' participation in the same post. However, MPC popularity is not strongly correlated with the post author

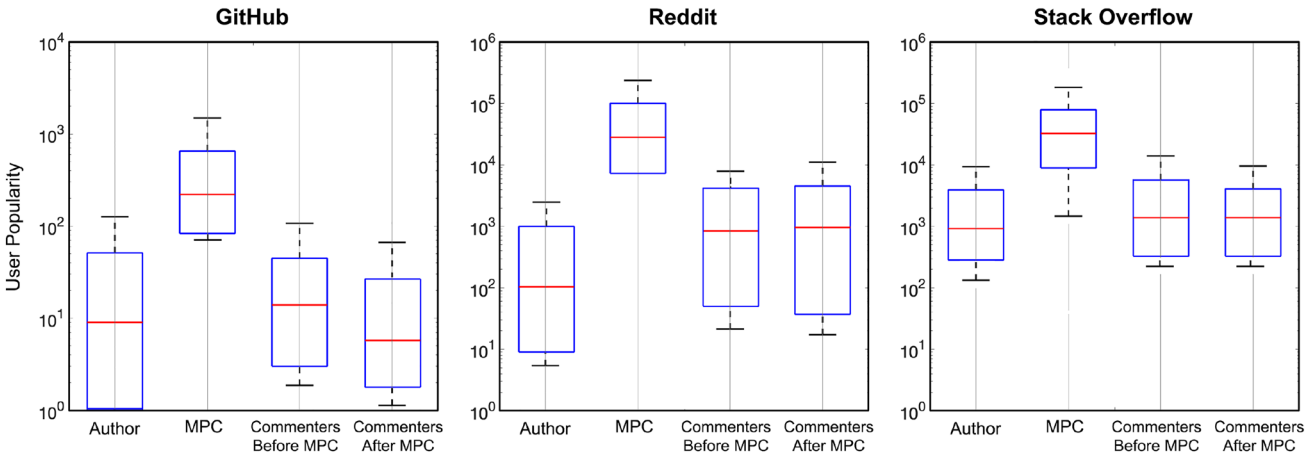

Fig. 8 Popularity of authors, MPCs (the Most Popular Commenters), and commenters (partitioned by those who comment before and after the MPC)

Table 5 MPC popularity correlations to other users' communications features

	GitHub		Reddit		Stack OF	
	r-val	*p* val	r-val	*p* val	r-val	*p* val
Comment count	0.22	0.00	0.16	0.01	0.22	0.01
Authors' popularity	0.07	0.02	0.00	0.83	0.10	0.03
Commenters' popularity pre MPC	0.41	0.00	0.22	0.01	0.48	0.01
Commenters' popularity post MPC	0.35	0.00	0.11	0.00	0.33	0.04

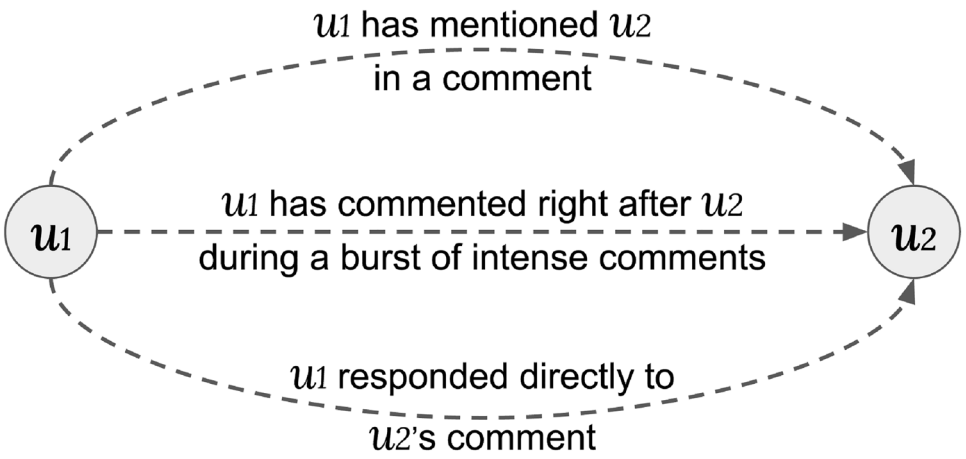

Fig. 9 Construction of the the commenting network. The nodes can have a direct weighted link through social tagging, conversation involvement, and directly responding to comments

popularity, particularly on Reddit. Popular commentators often comment on posts that interest them, regardless of the popularity of the original poster.

4.3 Commenting communities

To study the influence of status on network structure, we constructed a weighted directed network where the nodes represent the users involved in commenting activities. Comments were used to construct the network using the following heuristics (shown in Fig. 9):

- Links by direct *social tagging*: Social tagging refers to cases in which one user mentions another user and links them to a comment or social post (Zappavigna and Martin 2018). We utilize social tagging to identify users of a specific social platform inside the comments by searching for @mention tags or finding mentioned valid usernames. Therefore, in this network, the node (u_1) will have a direct outgoing link to the node (u_2) if the latter is tagged in (u_1)'s comment.
- Links by being involved in *conversation-like commenting*: Social interactions among peers through comments can be seen as a form of conversation (Pace and Buzzanca 2016). Within the commenting timeline, we often observed bursts of intense comments, usually between several users who comment multiple times within a relatively short period. These bursts of comments were used to link users. Therefore, during burst intervals, the node (u_1) will be linked directly to the node (u_2) if the former has commented right after (u_2) within a short period.
- Links by directly *responding to comments*: Most social networks permit users to leave a direct comment, reply, or answer other users' comments. Hence, the node (u_1) is linked to the node (u_2) by an outgoing edge if the former has a direct comment on (u_2)'s comment.

Table 6 Community size by detection method

	GitHub	Reddit	Stack OF
Louvain	58	55	45
Statistical inference	30	37	17
Label propagation	32	10	12

A commenting network was constructed for each platform using these defined linking conditions. Then community detection methods were used to isolate subgroups of users who have direct interactions. Table 6 shows the average size of the detected communities categorized by detection method: (1) Louvain (Newman 2006), (2) statistical inference (Zhang and Peixoto 2020), (3) label propagation (Raghavan 2007). The Louvain method tended

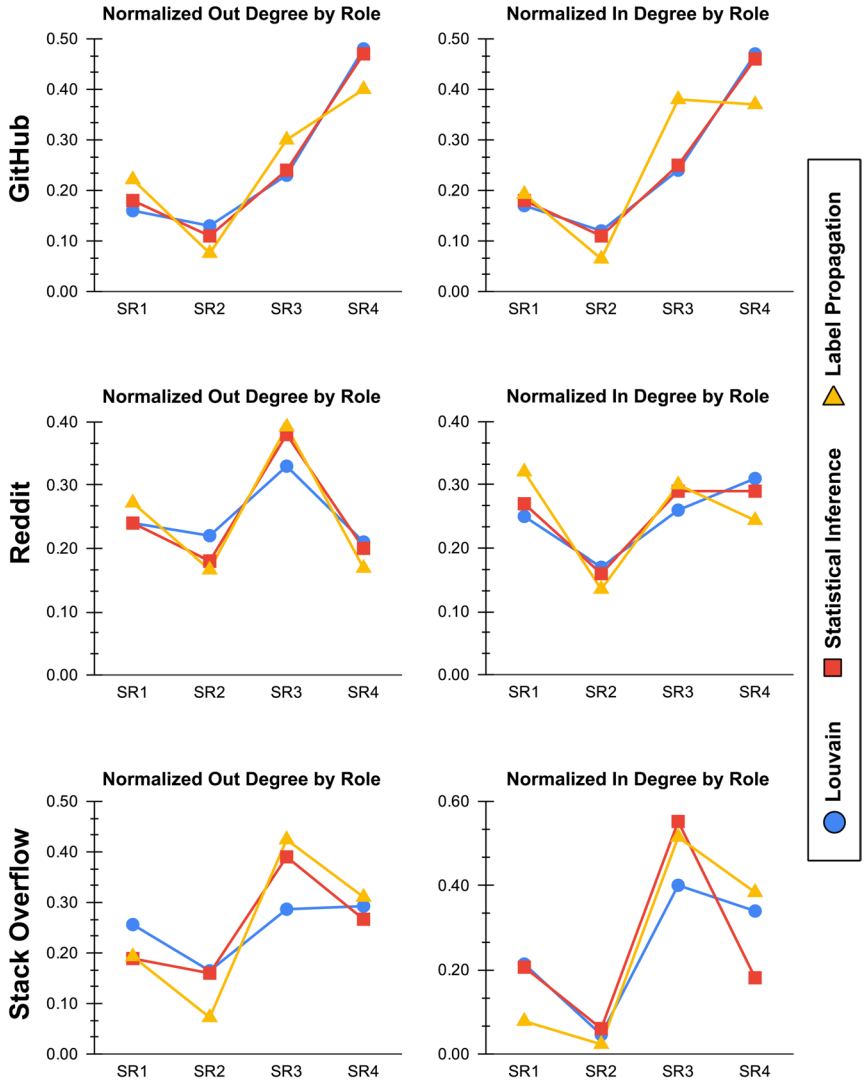

Fig. 10 Constructed networks' node-specific measurements: weighted out-degree and weighted in-degree by social roles based on the user community detection methods: Louvain, statistical inference, and label propagation

to separate the network into larger communities, whereas the label propagation created smaller communities.

Social Roles After extracting communities, we examine the effects of user role on community discourse by annotating the nodes with social role: newbie (SR1), rising star (SR2), longstanding member (SR3), and tech guru (SR4). Then we calculate user *weighted out-degrees* and *weighted in-degrees*. The weighted edges indicate the number of outgoing (out-degree) and incoming (in degree) links; link weights are proportional to the number of comments exchanged (Opsahl and Agneessens 2010). Figure 10 shows how the users' social roles affect the normalized weighted in-degree and out-degree. The figure shows that the pattern is stable across different methods of community detection (Louvain statistical inference, and label propagation). In GitHub, tech gurus (SR4) have the highest weighted in degree and out degrees, indicating that they are highly active at all forms of discourse (socially tagging, conversing and responding to comments). However in Reddit and Stack Overflow, the majority of discourse is driven by longstanding members (SR3). Interestingly rising stars (SR2) do not contribute much to the discourse vs. newbies (SR1).

Commenting Community Effective Engagement (CCEE) In addition to degree centrality, we also examined the impact of social roles on other types of network measures: 1) betweenness centrality and 2) clustering coefficient.

- *Betweenness Centrality* The betweenness centrality measures how centrally a node is placed in a network based on connections to other nodes. Because they are located on the most significant number of message pathways, they are the ones whose removal from the network will have the most impact on communication between other vertices. The links' weights were considered while determining the betweenness centrality (Opsahl and Agneessens 2010).

- *Clustering Coefficient* The clustering coefficient measures the node's tendency to cluster with other nodes and quantifies how close the network is to having a clique structure (Albert and Barabási 2002).

Betweenness centrality and clustering coefficient capture different aspects of users' importance to social discourse. Betweenness expresses the user's overall importance to connecting disparate members of the community; these users comment on many issues and participate broadly in discussions. Users with high clustering coefficients are highly active within a tight-knit group. In commenting networks, both measurements are useful for quantifying contributions to community discourse. Therefore, we combine these two measures to create the Commenting Community Effective Engagement Score (CCEE) for nodes in commenting communities:

$$CCEE(u) = \beta(u)_c \delta(u)_c$$

where $\beta(u)_c$ is the betweenness centrality and $\delta(u)$ is the clustering coefficient of the node (u) in community c.

- The betweenness measurement $\beta(u)$ is computed using the following formula:

$$\beta(u)_c = \sum_{i,u,j \in c} \frac{\sigma_{ij}(u)}{\sigma_{ij}}$$

where σ_{ij} represents the total number of the shortest paths connecting i to j, and $\sigma_{ij}(u)$ a subset of σ_{ij} and only includes the shortest paths that only pass through node u where i, u and j all are nodes in commenting community c (Brandes, 2008).

- The clustering coefficient $\delta(u)$ is calculated using the following formula:

$$\delta(u) = \frac{2t(u)}{k(u)[k(u)-1] - 2k^{\leftrightarrow}(u)}$$

where $k(u)$ is the sum of in-degree and out-degree of node (u), $t(u)$ is the number of triangles through (u) that equals the total number of edges between the $k(u)$ neighbors of node (u), and $k^{\leftrightarrow}(u)$ is the bilateral degree of node (u) and equals the sum of all the two sided links between node (u) and its neighbor (Fagiolo 2007).

Since the detected communities are of different sizes, $\beta(u)$ and $\delta(u)$ scale the community's size. Therefore to compute these two measurements, we have normalized them to be in the range [0,1]. Figure 11 illustrates how nodes are sized based on nodes' centrality, clustering tendency, and CCEE score in one GitHub commenting community. Nodes with high CCEE scores are more evenly distributed across the whole network, compared to those with a high betweenness centrality or clustering coefficient.

Our aim is to understand the impact of social role (newbie, rising star, longstanding member, and tech guru) on user engagement. Figure 12 shows the average CCEE broken down by social roles across all three platforms. Tech gurus (SR4) have a higher CCEE score on both GitHub and Stack Overflow, indicating a high level of engagement with other users. On Reddit, longstanding members (SR3) dominate commenting discourse. Interestingly, rising stars (SR2) are not as engaged with fellow users as their high level of popularity would suggest. Account age (SR3 and SR4) is more predictive of community engagement than popularity. Unfortunately, the community detection method used affects the CCEE score more

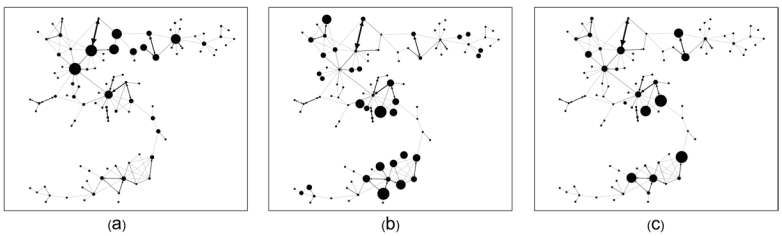

Fig. 11 A GitHub commenting community (detected using Louvain method) where nodes are sized based on three measurements: **a** the size of the nodes are proportional to the betweenness value, **b** the nodes' sizes are indicating the value of clustering coefficient, and **c** shows the nodes where they are sized based on CCEE score

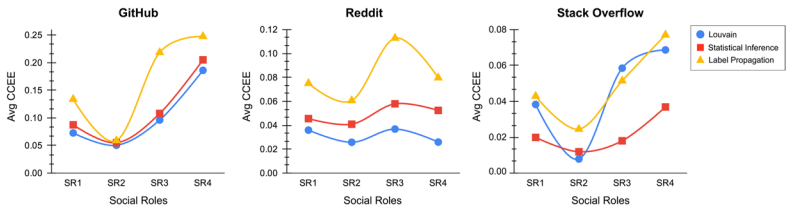

Fig. 12 Average commenting community effective engagement score (CCEE) categorized by social roles across all platforms' detected communities

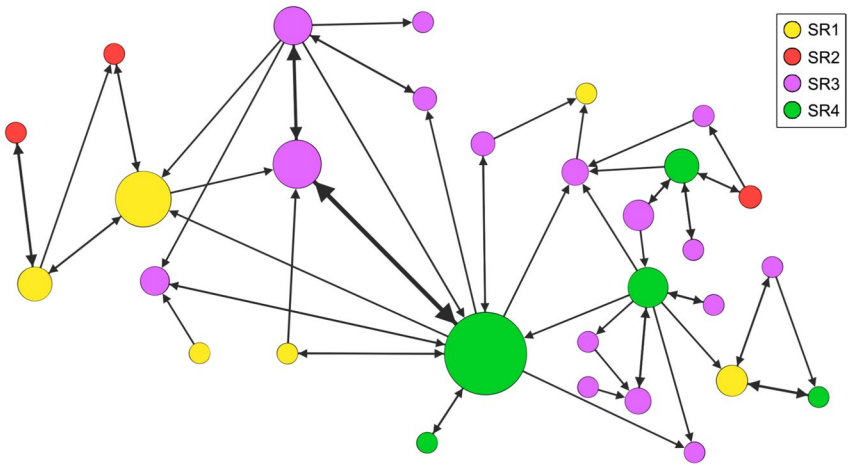

Fig. 13 One of the detected communities in GitHub, which was uncovered using the label propagation method. The node size is relative to the CCEE score of the users in this sub-community

than the weighted degree which remains more stable across different detection algorithms. CCEE engagement positively correlates to user popularity and seniority on GitHub and Stack Overflow, whereas on Reddit normal members drive more of the commenting discourse. Figure 13 illustrates the distribution of CCEE scores, broken down by social role, in an example GitHub community.

5 Conclusion and future work

The objective of this research study was to assess the engagement of individuals with technology-related posts on online social platforms. Our findings revealed clear differences in commenting behavior on different social platforms, particularly for GitHub users. GitHub users seem to use the platform in a more conversational manner when communicating through posts. For example, GitHub users post more comments and are more likely to comment multiple times. Additionally, GitHub discussion timelines are longer, and feature conversations spanning months rather than days.

We aim to understand how the popularity of users can affect the trajectory of discourse. To achieve this goal, we evaluated the impact of popularity on two types of users: the authors of posts and the most popular commenters (MPCs). Our data suggests that posts initiated by popular users tend to elicit more comments and involve a larger group of people in the discussion. The participation of high-popularity MPCs is positively correlated with the number of comments and the participation of other high-popularity commenters. By studying comment timelines, we observed that users' commenting rates and involvement change before and after MPCs contribute to discussions on GitHub. However, this pattern was not evident on Reddit or Stack Overflow.

In online communities, users' social roles are shaped by their tenure and posting activity. Unlike a corporate organizational hierarchy, social roles in online communities emerge organically and evolve over time. Since users lack formal authority, they cultivate a perception of expertise to influence others. We cluster users into four social roles using popularity and seniority: (1) newbies, (2) rising stars, (3) longstanding members, and (4) tech gurus. Commenting networks were constructed to represent the interactions between users on each platform. To evaluate user engagement within these communities, we defined a metric, the Commenting Community Effective Engagement (CCEE) score. This score aimed to identify users who are highly engaged in communication with their peers and important within their communities, better than just relying on degree centrality. The results of this study revealed that high popularity newcomers have relatively low CCEEs, whereas tech gurus (who have a high seniority as well as popularity) had the highest CCEEs. This indicates that popularity alone does not predict community engagement.

By understanding how popularity affects user interactions, we can design communities that are more effective at supporting collaboration. One of our primary concerns was that popular users had the implicit authority to prematurely terminate productive technical discussions. On GitHub, we found that commenting behavior changes after an MPC comments, typically ending the discussion. This pattern is not observed on Reddit or Stack Overflow. This may occur when GitHub MPCs also possess the formal technical responsibility for code review, which gives their comments more weight. In future work, we plan to use NLP to analyze comment contents and augment our timelines with additional content information.

Fig. 14 Dataset directory on mega platform

6 Threats to validity

This article uses data gathered in the wild to study commenting timelines and networks. However, without a control dataset, we cannot be sure that popularity modifies the commenting behavior of other users. Additionally, our dataset focused on a narrow range of topics within AI, robotics, and machine learning and may not generalize to other tech forums. Finally, our study focuses solely on quantitative data and does not explore the qualitative experiences of developers.

Appendix A: Commenting behavior dataset

A.1 Collection

Our commenting behavior dataset was created by downloading public GitHub data provided by the GHTorrent project (Gousios 2013) using the GitHub API within a Python crawler. Data was cleaned and categorized into three levels: (1) repository, (2) user, and (3) event. MongoDB was then used to store and organize the data objects. For visualization and analysis purposes, the data was extracted into a tabular format.

For this study, we randomly selected 5000 GitHub repositories through the GitHub API using search terms related to artificial intelligence and robotics (see Fig. 1). We gathered comparable data from groups of users on two other social platforms with the same tech-related focus, to establish a benchmark for a cross-platform comparison of user behavior. Using the same search terms, we gathered 5000 Reddit submissions and 5000 Stack Overflow questions.

A.2 Usage

Our data is available for download on the Mega platform at: https://bit.ly/abdul-dissertation-dataset under folder Chapter06Commenting_Behavior Fig. 14.

Under the folder, `000_Processed_data` we have provided Excel spreadsheets summarizing the dataset entitled `Network Stats` and `The Collected Data's General Info`. The files required to reconstitute the MongoDB database are separated by platform under the directories (1) GitHub, (2) Reddit, and (3) Stack Overflow.

Author contributions AAR carried out the technical work, dataset preparation, statistical analysis, prepared the illustrations, and wrote the initial draft. GRS supervised the technical work and revised the article.

Funding No funding was received to assist with the preparation of this manuscript.

Data availability We have released the data for this article at: https://bit.ly/abdul-dissertation-dataset. See the appendix for details about the dataset collection process and usage.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

Albert R, Barabási A-L (2002) Statistical mechanics of complex networks. Rev Mod Phys 74(1):47

Anderson A, Huttenlocher D, Kleinberg J, Leskovec J (2013) Steering user behavior with badges. In: Proceedings of the international conference on world wide web. Association for computing machinery, New York, NY, USA. pp 95–106. https://doi.org/10.1145/2488388.2488398

Arafat O, Riehle D (2009) The comment density of open source software code. In: International conference on software engineering-companion volume. pp 195–198. IEEE

Al-Rubaye A, Sukthankar G (2020) Scoring popularity in GitHub. In: International conference on computational science and computational intelligence (CSCI), pp 217–223. IEEE

Al-Rubaye A, Sukthankar G (2023) How popularity shapes user interaction in tech-related online communities. In: Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining

Barabási A-L (2002) The new science of networks. Perseus, Cambridge

Barlund D (2008) Communication theory. Routledge, London, pp 47–57

Buntain C, Golbeck J (2014) Identifying social roles in Reddit using network structure. In: Proceedings of the international conference on world wide web, pp 615–620

Brandes U (2008) On variants of shortest-path betweenness centrality and their generic computation. Soc Netw 30(2):136–145

Blincoe K, Sheoran J, Goggins S, Petakovic E (2016) Understanding the popular users: following, affiliation influence and leadership on GitHub. Inf Softw Technol 70:30–39

Choi D, Han J, Chung T, Ahn Y-Y, Chun B-G, Kwon TT (2015) Characterizing conversation patterns in Reddit: from the

perspectives of content properties and user participation behaviors. In: Proceedings of the ACM conference on online social Networks. pp 233–243
Destefanis G, Ortu M, Bowes D, Marchesi M, Tonelli R (2018) On measuring affects of GitHub issues' commenters. In: Proceedings of the international workshop on emotion awareness in software engineering. pp 14–19
Fagiolo G (2007) Clustering in complex directed networks. Phys Rev E 76(2):026107
Gousios G (2013) The GHTorrent dataset and tool suite. In: 2013 10th working conference on mining software repositories (MSR). pp 233–236. IEEE
Gorovits A, Zhang L, Gujral E, Papalexakis E, Bogdanov P (2021) Mining bursty groups from interaction data. In: Proceedings of the ACM international conference on information & knowledge management. Association for Computing Machinery, New York, NY, USA. pp 596–605. https://doi.org/10.1145/3459637.3482370
Movshovitz-Attias D, Movshovitz-Attias Y, Steenkiste P, Faloutsos C (2013) Analysis of the reputation system and user contributions on a question answering website: stack overflow. In: IEEE/ACM international conference on advances in social networks analysis and mining. pp 886–893. IEEE
Merchant A, Shah D, Bhatia GS, Ghosh A, Kumaraguru P (2019) Signals matter: understanding popularity and impact of users on stack overflow. In: The world wide web conference, pp 3086–3092
Newman ME (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582
Opsahl T, Agneessens F, Skvoretz J (2010) Node centrality in weighted networks: generalizing degree and shortest paths. Soc Netw 32(3):245–251
Pace S, Buzzanca S, Fratocchi L (2016) The structure of conversations on social networks: between dialogic and dialectic threads. Int J Inf Manag 36(6):1144–1151
Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
Richterich J (2014) Karma Precious karma!: Karmawhoring on Reddit and the front page's econometrisation. J Peer Prod 4(1):1–12
Sengupta S, Haythornthwaite C (2020) Learning with comments: an analysis of comments and community on stack overflow. In: Proceedings of the Hawaii international conference on system sciences
Takac L, Zabovsky M (2012) Data analysis in public social networks. In: International scientific conference and workshop present day trends of innovations, vol. 1
Woods J (2023) A systematic literature review of predictors of social media popularity. J Digit Soc Res 5(4):62–92. https://doi.org/10.33621/jdsr.v5i4.181
Zappavigna M, Martin JR (2018) # Communing affiliation: social tagging as a resource for aligning around values in social media. Discourse, Context Media 22:4–12
Zhang L, Peixoto TP (2020) Statistical inference of assortative community structures. Phys Rev Res 2(4):043271
Zhang H, Wang S, Chen T-H, Hassan AE (2019) Reading answers on stack overflow: not enough! IEEE Trans Softw Eng 47(11):2520–2533