

Implicit Mentoring: The Unacknowledged Developer Efforts in Open Source

Zixuan Feng, Amreeta Chatterjee, Anita Sarma, and Iftekhar Ahmed

Abstract—Mentoring is traditionally viewed as a dyadic, top-down apprenticeship. This perspective, however, overlooks other forms of informal mentoring taking place in everyday activities in which developers invest time and effort, but remain unacknowledged. Here, we investigate the different flavors of mentoring in Open Source Software (OSS) to define and identify implicit mentoring. We first define implicit mentoring—situations where contributors guide others through instructions and suggestions embedded in everyday (OSS) activities—through formative interviews with OSS contributors, a literature review, and member-checking. Next, through an empirical investigation of Pull Requests (PRs) in 37 Apache Projects, we build a classifier to extract implicit mentoring and characterize it through the dual lenses of experience and gender. Our analysis of 107,895 PRs shows that implicit mentoring occurs (27.41% of all PRs include implicit mentoring) and it does not follow the traditional dyadic, top-down apprenticeship model. When considering the gender of mentor-mentee pairs, we found pervasive homophily—a preference to mentor those who are of the same gender—in 93.81% cases. In the cross-gender mentoring instances, women were more likely to mentor men.

Index Terms—Informal mentoring, Implicit mentoring, Homophily mentoring, Open source software.

1 INTRODUCTION

Open Source Software (OSS) projects are self-organized, volunteer-driven communities where contributors from different parts of the world come together and teach each other to create large, complex software [1], [2], [3], [4]. Mentoring plays a key role in ensuring the sustainability of OSS projects by helping on-board newcomers who need to learn the technical skills and the process and cultural norms of the community.

Mentoring describes an interpersonal relationship where an experienced contributor (the *mentor*) provides functional advice and interpersonal guidance to an inexperienced individual (the *mentee*) [5]. These mentoring relationships between the two parties facilitate the transfer of declarative knowledge—technical facts that mentees need to accomplish their tasks—and procedural knowledge about navigating the contribution process and project culture [6], [7]. Research shows that mentoring is an effective means for newcomer training and improves the on-boarding experience and retention of contributors in OSS [6], [8].

It is no surprise that OSS Foundations have invested in mentorship programs such as, the Google Summer of Code scholarship program that paired 1,289 students with mentors from across OSS organizations in summer 2021 [9]. The Linux foundation currently has 7 mentorship programs and has invested over 2.5M USD in support of first-time and underrepresented OSS contributors [10]. Additionally, OSS projects (e.g., Apache Mentoring Program [11]) con-

nect newcomers to mentors in the project by providing some basic mentoring structure (e.g., mentor-mentee interest matching and guidance on communication, task scope, and progress checkpoints).

Outside of such formal mentoring, contributors actively seek guidance and support from each other via informal channels such as direct contact through emails and video conferencing or meeting at conferences. Moreover, technical guidance is also provided in everyday development activities, such as when contributors review code or design.

Irrespective of the type of mentoring, there is a cost involved in mentoring. Mentors have to spend time and effort in guiding newcomers and can face various challenges. For example, recommending a task that is suited to newcomers' background, interests, and matching the project timeline can be difficult [12]. It can also be challenging for mentors to keep the mentee engaged, especially if the project culture is harsh or the mentees are not proactive/ strongly interested in OSS [7]. Given that mentors are also volunteers, this additional effort in mentoring can lead to a reduction in their technical productivity [13] and sometimes even loss in stature, where women who frequently mentor may be treated as community managers and not engineers [7].

Currently, not much is known about the different types of mentoring in OSS. Without such an understanding, the important contributions made by OSS mentors may go unacknowledged. An experienced OSS mentor in their interview mentioned: “*there is no recognition, no kudos, no kind of positive reinforcement for them to continue being a mentor. So, usually, they are a mentor once and then they leave. [P4]*”

This leads us to the first research question:

RQ1: What are the different approaches to mentoring in OSS?

Through a formative mixed method study of interviews with five senior OSS developers and a literature review, we identified two dimensions in mentoring: *formal—informal*

Zixuan Feng is with Electrical Engineering and Computer Science of Oregon State University (e-mail: fengzi@oregonstate.edu).

Amreeta Chatterjee is with Electrical Engineering and Computer Science of Oregon State University (e-mail: chattera@oregonstate.edu).

Anita Sarma is with Electrical Engineering and Computer Science of Oregon State University (e-mail: anita.sarma@oregonstate.edu).

Iftekhar Ahmed is with Information and Computer Science at the University of California, Irvine. (e-mail: iftekha@uci.edu).

and *explicit-implicit*. While explicit forms of mentoring, both formal and informal, have been investigated, the concept of implicit mentoring—mentoring occurring through everyday OSS activities—although a frequent occurrence, is understudied. As a community, we do not have a clear understanding of the extent at which implicit mentoring occurs and under what conditions? Who provides implicit mentoring? Whether it follows the traditional dyadic models of experienced mentors guiding newcomers? Whether there are any gender imbalances in who is mentoring or who gets mentored? Answers to these questions can help OSS projects better recognize implicit mentors and fix any imbalances that exist. Therefore, we ask:

- RQ2:** What are the characteristics of implicit mentoring?
RQ3: What are the characteristics of implicit mentoring relationships?

We answer these research questions through an empirical study of PRs in 37 Apache projects. We opted to investigate Apache projects because these projects have well-documented discussion procedures. Apache projects follow the principles of open communication logging discussion online [14]. We opted to analyze PRs, as they are used in many scenarios beyond basic patch submission, (e.g., conducting code reviews, discussing new features [15], and providing feedback [16]).

Then, to automatically identify implicit mentoring in PR-comments, we used the Random Forest classifier. Using these identified instances, we investigated the association between implicit mentoring and characteristics of PRs, as well as mentor-mentee relationships through the lens of experience and gender. The significance of our contribution is manifold: (1) defining implicit mentoring in the context of OSS, (2) automatically identifying implicit mentoring from PR-comments, (3) characterizing implicit mentoring and its effects, and (4) characterizing implicit mentoring relationships.

Our results lay the foundation for various future research directions and also identifies several calls for OSS communities such as creating an appreciative community, improving the state of diversity in OSS, and mechanisms to make mentoring sustainable; both of which are important to sustain and create healthy OSS projects. In the words of one of our interviewees : “*in open source, you’re gonna find a lot of professionals that have a lot of experience, but need mentorship to understand and join a specific technology...but I don’t think the mentorship is recognized at all.* [P3]”

2 RELATED WORK

Mentoring has been extensively researched in various domains. In management and organizational literature, works have investigated the extent to which mentoring helps with organizational citizenship behaviors (defined as positive employee attitude to the organization) [17]. They found top-down mentoring to be positively associated with improved employee attitudes. Payne and Huffman [18] investigated the relationship between mentoring and positive organizational attitudes and found it to have a strong association with affective commitment (employee’s emotional attachment or identification with the organization) and continuance commitment with the organization. Similarly, in

education literature, Mullen and Klimaitis [19] conducted a literature review of empirical studies on mentoring to identify other forms in addition to the traditional—formal, dyadic, top-down—mentoring model. These included student mentoring that is done in groups, among peers, in collaborative or cross-cultural forms.

One form of mentoring is informal mentoring. According to Inzer and Crawford [20], informal mentoring occurs in a relationship between two people where one gains insight, knowledge, wisdom, friendship, and support from the other. Multiple works have found informal mentoring to be more beneficial than formal mentoring as it provides higher engagement and skill development opportunities, including coaching, providing challenging assignments, or increasing exposure and visibility of the mentee. Mentees who were informally mentored were much more satisfied with their mentors than mentees with formal mentors [20], [21], [22]. Ragins and Cotton [23] found women had the least to gain from formal mentoring, as the presence of formal mentors reduced coaching, role modelling, and career counselling for women mentees. Informal mentoring on the other hand was beneficial for both men and women. Inzer and Crawford [20] also state that informal mentoring is a valuable tool for grooming an employee as it occurs in a relationship that is voluntary and created by both persons. It is friendship first, learning and career second and third.

Ko et al. [24] found informal mentoring to be better at triggering and maintaining interest in computing among adolescent students. Nandi and Mandernach [25] concluded that students are strongly motivated by informal mentoring relationships. Analysis from surveys, student academic records and source-code commit log data show that students improve significantly when informally mentored.

Software Engineering research has investigated mentoring, especially in the context of onboarding activities. Begel and Simon [26] discuss the importance, advantages and challenges of mentoring for novices in the software industry. Kumar and Wallace [27] investigated communication strategies for mentoring in software development projects, including “Code As Conversation” pattern, where participants on the forum ask or answer questions regarding a code snippet that is shared through communication channels.

In the context of research in OSS, several works have identified the lack of mentoring as a challenge for newcomer onboarding [13], [28], [29]. Others have identified the challenges that mentors face in OSS [7], [12], [30]. Researchers have found that mentoring allowed for a more effective onboarding experience than when newcomers entered a project through a natural, non-deliberate process [8], [13]. Google through its formal Google Summer of Code (GSoc) mentorship program aims to facilitate onboarding to OSS [31]. Several works have investigated the effectiveness of the program [32], the type of contributions made by students [33], and their motivations for joining the program [32]. To the best of our knowledge, no prior work has explored the concept of implicit mentoring in OSS. Our paper fills this gap by investigating how implicit mentoring happens and who are these mentors in the OSS community.

TABLE 1
Demographic information of interview participants

ID	Gender	OSS experiences	Mentor	Mentoring experiences		
				Informal /Formal	Mentee	Informal /Formal
P1	Woman	Over 10 years	Y	Both	Y	Both
P2	Man	Over 10 years	Y	Both	Y	Both
P3	Woman	6-10 years	Y	Both	Y	Both
P4	Woman	6-10 years	Y	Both	Y	Both
P5	Woman	6-10 years	Y	Both	Y	Both

3 METHOD

3.1 Defining Implicit Mentoring

As a formative study to identify the different approaches to mentoring we used semi-structured interviews along with member checking and a literature review of mentoring in the context of OSS, as shown in Figure 1.

3.1.1 Formative Interviews

We first conducted semi-structured interviews with five OSS developers to learn of their experiences regarding mentoring—both from the perspective of being a mentor and a mentee. We recruited two developers, one from the Apache Source Foundation and another from the Linux Foundation, both of whom have spearheaded mentoring programs in OSS. We then used snowballing sampling method to recruit three additional participants. We stopped after five interviews, since we had already reached saturation—participants responses were similar regarding the different types of mentoring they had seen.

The interviews were done remotely, lasting around 30 minutes each. Each interview was recorded with participants' consent (following university-approved IRB protocol) and transcribed. Participants were offered a \$50 gift card as compensation for their time. Table 1 presents the demographic information of our participants.

We briefly explained the goal of the interview to the participants in the beginning of the interview. We asked participants about their roles and experience in OSS, their mentoring experience including formal and informal, what they consider to be informal mentoring and their opinions regarding formal and informal mentoring in OSS. At the end of our interview, we asked participants on how mentoring can be improved in OSS. (See Supplemental [34] for interview questions)

We transcribed the interview recordings and qualitatively analyzed the transcripts, following open coding protocol. We followed the principles of grounded theory to code for types of mentoring, challenges of mentoring and implicit mentoring. Two researchers performed the analysis by independently coding all the transcripts resulting in codes regarding types of mentoring and challenges, how and where mentoring was provided, and recommendations. During this analysis, each emerging code was compared with the existent codes to determine if the emerging code was a discrete category or a subset of an existing code. After this step, the two researchers met to discuss their codes and performed card sorting to arrive at the final code set [35].

3.1.2 Validating findings from member checking and past literature

We validated the insights from our analysis in two ways: Member checking and a literature review.

Member Checking: We contacted each of our participants through email and sent them a survey that comprised 16 questions and was conducted via Qualtrics [34] for survey questions. The survey included four demographics questions, three questions about what constitutes mentoring and where it occurs, and eight samples of PR-comments. All questions had associated text boxes for participants to provide additional feedback. The eight PRs were randomly selected from our dataset based on their verbosity and participants were asked to ascertain if the PR-comment was mentoring (or not) and provide a rationale for their decisions. The open ended responses were used to refine our understanding of the different approaches mentoring and the code set.

Literature review: To corroborate our findings from the interviews, we conducted a small-scale systematic mapping study. According to Kitchenham et al. [36], the goal of such a mapping study is to survey the available knowledge about a topic. To determine the optimal set of search keywords, we first conducted a pilot search on two well-known digital libraries—IEEE and ACM. This process helped us identify relevant words used in mentoring literature in Software Engineering, especially in OSS. Our final list of search keywords included: "mentor", "mentoring", "formal", "Informal".

Next, to discover relevant publications for the survey, we used three of the most popular online paper search engines: ACM Digital Library, IEEE Xplore, and Scopus and searched for papers that were published in top conferences from 2012 to 2021 such as, IEEE-ICSE, IEEE-RSSE, IEEE-ICGSEW, ACM-FSE, ACM-ESEM, ACM-ICSE, IEEE-ICSM, IEEE-MSR, ACM-SIGCSE, ACM-HCI, and ACM-ICPS. Our initial search resulted in 54 publications. Then, the first and second author read the titles and abstracts and only selected those which talked about mentoring in the abstract, which resulted in 17 papers. Our search criteria may have resulted in leaving out some relevant studies. We performed a single iteration of backward snowballing [37] (*i.e.*, looking for additional studies in the reference lists of the selected studies, as suggested by Keele et al. [38]), which provided 6 additional papers. Our final list consisted of 23 papers.

3.2 Characterizing Implicit Mentoring

Figure 1 presents the overview of our methods applied while characterizing implicit mentoring. We use a set of 37 Apache projects as our data corpus to answer RQ#2 and RQ#3. We selected Apache projects since it is one of the most mature OSS ecosystems with well defined policies and a philosophy of transparency where all discussions should be conducted in online forums. This is essential for us as we aim to extract implicit mentoring from archived communication. Further, the Apache Source Foundation is committed to facilitating and improving mentoring in OSS. In addition to providing explicit guidelines for mentors [11], it also oversees one of the most popular formal mentorship programs (Google Summer of Code [32]), which would mean

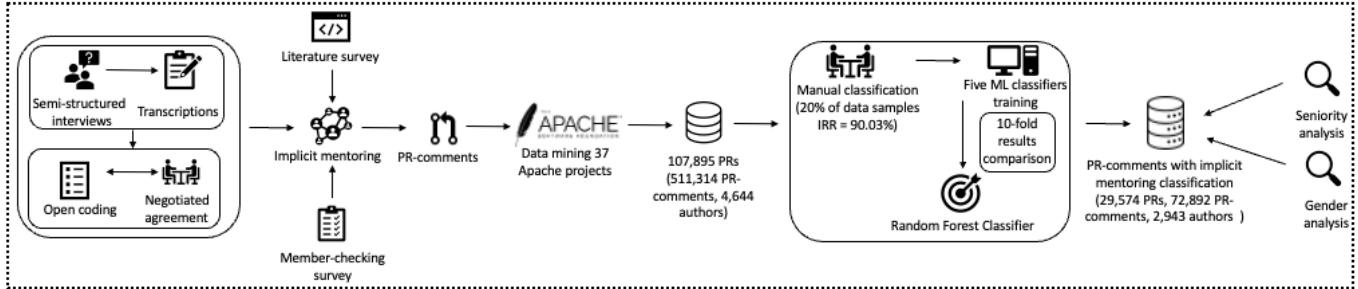


Fig. 1. Overview of research method

TABLE 2
Project statistics of the 37 Apache Projects.

Dimension	Max	Min	Average	Median
Project size (KLOC)	18,475	82	1,770	1,075
Project Age (weeks)	1,063	214	606	560
Developers	1,852	21	226	106
Total Commits	80,227	3,561	22,688	18,084
Total Pull-request (PR)	32,645	6	2,916	651
Total PR-comments(Non-PR author)	316,295	2	14,609	807

that the philosophy of mentoring is instilled in Apache contributors. Finally, Apache projects are often studied in scientific research, which allows our work to be placed in the context of existing research [39], [40], [41], [42], [43]. In fact, we used the list of projects from the dataset curated by Mannan et al [44].

Table 2 presents the statistics of the 37 projects in our dataset. The criteria for selecting the dataset was that projects are mature and include sizeable codebase (Table 2 provides the project size in KLOC—thousands of lines of code) and contributors. This dataset comprised 107,990 PRs with 836,729 PR-comments, logged by 12,668 contributors. As our analysis required GitHub profile data, we removed contributions of 42 (0.04%) user profiles whose GitHub accounts were deleted at the time of data collection. Additionally, we filtered out the PR-comments made by the PR author since we only wanted to analyze mentoring comments by other contributors. These steps resulted in 107,895 PRs with 511,314 PR-comments, and 12,626 contributors (out of which 4,644 contributors were PR-comment authors). See Supplemental [34] for further details about each project in the dataset.

Manual Classification of Sample Data: To answer our research questions, we needed to differentiate between PR-comments with/without implicit mentoring. As manual classification is not a practical option to classify 511,314 PR-comments, we used machine learning techniques. To train the machine learning classifier, we manually classified a training set. We determined the size of this training dataset by using a 95% confidence interval and a margin of error of 5% [45] on the dataset, giving us a sample size of 384 PR-comments. We then randomly selected 384 PR-comments from the dataset.

Next, the first two authors manually labeled a subset of the PR-comments in the training set with binary labels based on whether the PR-comment included implicit mentoring to calculate Inter Rater Reliability (IRR) [46]. A PR-comment was considered to contain implicit mentoring, if it included

TABLE 3
Classification rule book

Mentoring Action	PR_comment sample
Instruction	M-284: "...run [tool] on the project before creating a PR. You would have noticed [problem]..."
Suggestion	B-553: "I would still duplicate [action] like I did in [certain PR] because it's widely used in [tests]. Maybe this could be removed after [situation]."
Mechanisms to fix errors	I-1376: "Would you mind just doing [action] again to kick off [framework]? I think [framework] is just not happy when it has a lot of loads."

TABLE 4
precision, recall, F1, and AUC for classifiers

	Precision	Recall	F1	AUC
RandomForest	0.86	0.91	0.88	0.93
Support Vector Machine	0.84	0.86	0.84	0.91
NaiveBayes	0.89	0.68	0.76	0.90
DecisionTree	0.77	0.69	0.72	0.81
K-neighbors	0.87	0.30	0.44	0.71

an “explanation” in addition to giving suggestions, instructions or helping fix errors. Table 3 shows the rule book we used with examples of PR-comments.

The two authors independently labeled 20% of the PR-comments and reached high IRR (90.03% Cohen Kappa [47]). The remaining 80% of the training dataset was split evenly between the two authors who manually classified the PR-comments.

Machine Learning Classifier: Using the manually classified corpus, we trained five different machine learning classifiers, a Random Forest [48], a Bernoulli [49], a Support Vector [50], a KNeighbors [51], and a Decision trees [52].

To ensure the best performance, we applied hyper-parameter adjustments [53] from Python Scikit learn library [54] to all five classifiers. Randomized look sets up a lattice of hyper-parameter values and chooses arbitrary combinations to train each classifier. By applying *ScikitLearnRandomizedSearchCV*, we found the optimal parameter for each classifier. The final tuned Random forester parameters for our classifier were *n_estimators=2800, max_features='auto, max_depth=73, min_samples_split=20, min_samples_leaf=2,* and *bootstrap=True*. The model was trained and evaluated using a 10-fold cross validation methodology. That is, the data was randomly divided into 10 equal sets, and nine sets were used for training and one for testing. We trained our model using this method 10 times and report the mean

performance.

Table 4 shows the precision, recall, F1, and AUC scores of the five classifiers. Random Forest Classifier (RFC) had the best overall performance when considering both the F-measure (0.88) as well as the AUC scores (0.93). Therefore, we used RFC for further analysis.

Identifying PR complexity: We measured the complexity of PRs using two different metrics. Complexity based on the length of the description [55], [56] and based on whether a PR was reopened [57]. Next we check the relationship between implicit mentoring and each of these metrics using Welch Two Sample t-tests. Since we perform multiple-hypotheses testing, we applied Bonferroni correction to adjust P values [58], which gives an adjusted $\alpha = 0.017$.

Identifying Expertise of Mentors and Mentees: To investigate the extent to which top-down interactions comprise implicit mentoring (RQ3a), we need to first identify when someone is a mentee or a mentor and their experiences. We considered the PR author as a mentee if any of the PR-comments associated with that PR is classified as “implicit mentoring”. For those PR-comments identified as implicit mentoring, their contributors are identified as a mentor.

To investigate the dynamics of implicit mentoring, especially if it comprises top-down interactions (experienced to inexperienced) we calculate experience at two levels. First, experience within the specific project, which we calculated based on the difference between the date of their first contribution (PR or PR-comment) within the project. However, in OSS, contributors can participate in multiple projects or can leave a project to join another. In such cases, they may bring experiences from working on another project when mentoring a developer in a project. Therefore, we also calculated the overall experience of a contributor in GitHub based on the date on which they created their GitHub account.

Identifying Gender: We used the “Namsor” API to identify the gender of contributors in our dataset. Namsor is a name recognition API that estimates the gender of a (full) name on a -1 to +1 probability scale based on geographic information [59]. Multiple studies have addressed the reliability of the “Namsor” gender classifications, the error is less than 10% [60]. Our gender-analysis dataset consists of 6,359 contributors (out of which 1,976 contributors were PR-comment authors), for whom Namsor gender identification probability was over 90%.

4 RESULTS

4.1 Different mentoring approaches (RQ#1)

Mentoring is an important part of onboarding and career progress in OSS [7], [8], [61]. However, despite formal programs, it is hard for mentees to find mentors: “*I always have trouble getting mentors...*” [P1], [12], [29], [62], [63]. On the flip side, it is hard for mentors to make a commitment without knowing whether their effort would be worthwhile: “*It’s hard to assess...whether the return on investment is going to be effective.*” [P2], [33]. There are two dimensions to mentoring. The first is whether the mentorship is Formal or Informal. The second is whether mentoring is Explicit or Implicit.

Formal mentoring includes instances where mentors and mentees are formally connected either through scholarships

TABLE 5
Mentoring in OSS and its characteristics

		Interview	Member Checked	Literature Survey
Mentoring Tasks	Suggestions	P1, P3, P5	✓	[7], [13], [24], [30], [68]
	Instructions	P2, P3, P4, P5	✓	[7], [13], [24], [30], [68]
	Mechanisms to fix errors	P1, P3, P5	✓	[7], [13], [24], [68], [69]
Mentoring Channels	Email Code review tool (e.g. PR-comments)	P1, P3, P4, P5	✓	[7], [68]
	In person/Remote	P4, P5	✓	[24]

or mentoring programs [7], [12]. However, despite the formal programs having guidance and financial support, it is difficult for mentees to figure out how to seek guidance: “*I really got overwhelmed with all the information...I didn’t have much idea what to ask for my mentor to guide me with*” [P4], [12], [62], or follow up with their mentors: “*There was no continuous progress of mentoring, we didn’t know how to follow up on things*” [P4], [12], [64]. Sustaining the mentor-mentee relationship was further difficult because of (a) diverging interests: “*...interests tend to diverge, you tend to look for new mentors, new areas, and new people to serve.*” [P5], [7], [12] and (b) limited resources: “*if the mentee leaves...a more likely outcome. Am I going to be left with code to maintain that I might have been better for me to write in the first place*” [P5], [32].

Informal mentoring, perhaps because of the above challenges, can be more effective than formal mentoring [20]. Informal mentoring is “interest-driven” when a mentor or mentee reaches out to the other to seek/give guidance in a particular area. Informal mentoring occurs frequently: “*I have always had mentors, all informal mentors because I chose to learn from them*” [P1], [20], [24], [65]. Such mentoring can be explicit, where the mentee or mentor reaches out to the other or can be implicit (discussed next). In either case, such mentoring is “invisble” and the effort made by mentors remain unacknowledged: “*I believe the employee doesn’t get any recognition for [informal] mentorship.*” [P1], [66].

A majority of formal and informal approaches that have been investigated in literature are explicit [7], [13], [32], [67], where mentees seek out or are formally paired with mentors. However, mentoring need not always be explicit and can be done during contributors’ everyday activities, such as code review: “*When somebody reviews a patch, that gives a feedback to you, that’s a form of mentoring*” [P1].

Implicit mentoring can be defined as “mentoring that occurs in everyday development activities such as code reviews, where a mentor provides an underlying explanation when providing suggestions, instructions, or mechanisms to address errors”. As P2 described: “*what you do in your day-to-day activities where you mentor...teachable moments, where you explain why you are doing certain things. So you’re essentially communicating knowledge about the system that goes beyond the knowledge necessary to act on the particular item that you have.*” Such mentoring can be through multiple channels: emails, PR-comments, in person meetings, or through online communication tools. As P3 commented, “*There’s going to be mentors who are more like code reviewers or the design reviewers...mentorship that happens every single day*”. Table 5 presents the different aspects of mentoring our interviews

and literature survey identified, along with the channels where mentors can “teach” in an OSS project.

4.2 Characterizing Implicit Mentoring (RQ#2)

Our goal here is to investigate how to automate the identification of implicit mentoring. Identifying such mentors has two direct benefits. First, an ability to (formally) acknowledge the effort of mentors, as P3 stated: “*people within [project], who have dedicated their entire careers to mentor interns and they don't get recognized for it.*”

Second, projects currently struggle to identify mentors, as P5 said: “*our leads are burned out by too much of [mentoring]...I'm trying to figure out how can I identify the people who've been mentored or who have been at the intermediate level, and get them engaged in the mentoring.*”

RQ #2a: How often does implicit mentoring occur? As a first step, we investigate how often implicit mentoring occurs via PR-comments. A single PR can have multiple rounds of discussion between the PR-author and other contributors, recorded via PR-comments. In our dataset on average, a PR had 4.74 PR-comments ($sd = 8.81$) which were not made by the PR-author. It is possible that one or more of these PR-comments could embed implicit mentoring. In our dataset, 27.41% of PRs included implicit mentoring comments. These comments were made by 2,943 out of 4,644 PR-comment authors (63.37%).

Observation 1: Implicit mentoring occurs (27.41% of PRs), with a majority of PR-comment contributors serving as implicit mentors (63.37%).

RQ #2b: Do complex PRs need more implicit mentoring?

Dabbish et al. [56] showed that visible cues, such as PR-comments made by contributors serve as an important signal of community support (or lack thereof). They found that PR-comments to a commit (or PR) signal that activity is interesting, controversial or worth looking at. It is possible that “complex” PRs that address such interesting or controversial changes elicit more PR-comments, including implicit mentoring PR-comments. Recall, we measure the complexity of PRs using two different metrics. Complexity based on the length of the description [55], [56] and based on whether a PR was reopened [57].

PRs with more *wordy description* on an average had more implicit mentoring than those with less wordy descriptions. The difference between the two groups was statistically significant (Welch Two Sample t-test, $t = 25.48$, $df = 49,161$, P value < 0.001). The effect size measured using Cohens d [70] is 0.18 which is nearly small effect (where small size is 0.2). This corroborates the finding that complex pull requests warrant additional attention or guidance.

Next we considered complexity based on whether the PR was *reopened*. 2,606 out of 107,895 PRs had been reopened in our dataset, out of which 1,294 PRs included implicit mentoring. Complex PRs (based on whether it was reopened) on average, had more mentoring than the non-reopened group. The difference in mean was statistically significant (Welch Two Sample t-test, $t = 9.31$, $df = 1,355$, P value < 0.001). The effect size measured using Cohens d [70] is

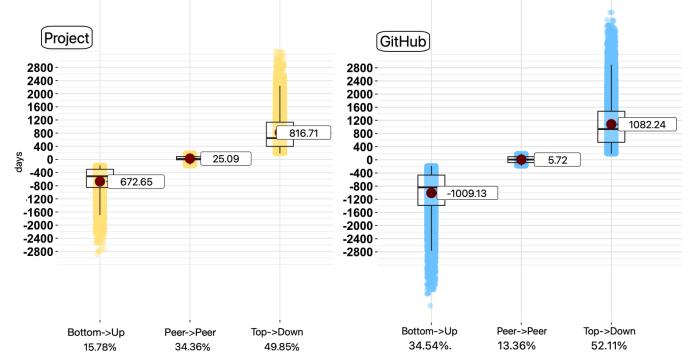


Fig. 2. Number of instances of implicit mentoring PR-comments vs. difference in experiences of mentor/mentee

0.36 which is a small to medium effect. Research has found that reopened PRs have lower acceptance rates, more PR-comments, and longer evaluation time, which might mean either these changes might be complex or controversial and thus might need more guidance [71].

Observation 2: Complex PRs (reopened and wordy descriptions PRs) include instances of implicit mentoring.

4.3 Implicit Mentoring Relationships (RQ#3)

4.3.1 Interaction Types in Implicit Mentoring

Traditionally, mentoring has been viewed as a dyadic relationship wherein an experienced individual (the mentor) provides practical advice and guidance to an inexperienced individual (the mentee) [72], [73], helping them gain technical and job-related skills [74], [75]. Therefore, we wanted to analyze if the same sort of mentoring dynamics occur in implicit mentoring, leading to:

RQ #3a: To what extent is implicit mentoring characterized by dyadic, top-down interactions?

We answer the first part of the question by considering the interactions for each PR. The PR-author is considered the mentee. Contributors of PR-comments that included implicit mentoring are considered mentors. If the discussions occurred only between the mentee and one mentor, this implicit mentoring is considered *dyadic*. Similarly, if the discussions occur between an author and two mentors, its considered a *triad*. Finally, we group all mentee-mentor groups equal to or greater than four as *≥quadrads*. Note, that for a PR, if there were other contributors who commented, but were not mentors (PR-comment was not considered implicit mentoring), they are ignored in this analysis.

Of the 29,574 PRs with implicit mentoring, the majority was dyadic (66.79%), lining up with the traditional view of mentor-mentee relationship. However, because of the open, voluntary-based nature of OSS, we found 21.85% of implicit mentoring occurring in triads and another 11.36% in *≥quadrads*. This is akin to “it takes a village...” adage, where the community of contributors work together to mentor new contributors and providing perhaps different perspectives and guidance.

Observation 3: A majority of implicit mentoring is dyadic (66.79%), but in a substantial number of cases (9,823 instances) multiple mentors provide support.

Next, we investigate what role experience—within the project and overall—plays in implicit mentoring. There can be three types of mentoring directionality:

- *top-down*, the mentor is more experienced than the mentee,
- *peer-to-peer*, the mentor and mentee have same level of experience,
- *bottom-up*, mentee is more experienced than the mentor.

To obtain the direction of implicit mentoring, we subtracted the date of first contribution (a PR or a PR-comment) made by the mentee from that of the mentor.

We use 6-months (183 days) as a threshold to classify the mentoring approaches. If the difference is greater than 6-months, then the PR-comment interaction is flagged as *top-down* and if less than 6-months, it is *bottom-up*. If mentor and mentee had their PR/PR-comments date within 6-months, we classify it as *peer-to-peer*. Figure 2 shows the distribution of implicit mentoring when considering within the project as well as overall experience, where we use the GitHub account creation date as a proxy (blue).

About 49.85% of implicit mentoring falls in the traditional top-down mentoring interaction style. The average difference between mentee and mentor experience was 2.24 years (817 days, $sd = 536$). 34.36% implicit mentoring was in peer-to-peer category, the mean difference between mentor and mentee was 25 days ($sd = 83$). 15.78% of implicit mentoring was in the bottom-up category with the mean difference between mentor-mentee being 1.84 years (673 days, $sd = 488$).

In OSS, contributors frequently contribute to multiple projects or migrate across projects [76]. In such cases, they may accrue skills and experiences relevant to the project elsewhere. Therefore, we look at the mentoring directionality when considering the overall experience in GitHub. The results show that the traditional top-down mentoring instances are 52.11% out of 72,892 PR-comments, the mean differences in experience being 2.96 years (1,082 days, $sd = 682$). This was followed by *peer-to-peer* mentoring (13.36%) with the mean difference being 6 days ($sd = 104$); and *bottom-up* mentoring (34.54% of cases) with mean difference in experience being 2.76 years (1,009 days, $sd = 675$).

Analyzing from the perspective of the contributors who do implicit mentoring, the story remains the same, with about 50% of contributors having done peer-to-peer or bottom-up mentoring when considering within the project as well as overall GitHub experience. This shows that “*seniority is not necessary for implicit mentoring*” [P3].

Observation 4: Non-traditional mentoring (Peer-Peer and Bottom-Up) models are common, accounting for nearly half of implicit mentoring cases.

TABLE 6
Summary of implicit mentoring by gender.

	women	men	Total
PR-comment developers	81 (4.10%)	1,895 (95.90%)	1,976
Implicit mentors	54 (4.18%)	1,237 (95.82%)	1,291
PR-comments	1,036 (1.07%)	95,906 (98.93%)	96,942
PR-comments w/ Implicit mentoring	390 (1.41%)	27,331 (98.59%)	27,721

4.3.2 Gender Analysis of Implicit Mentoring

Past work has found that women, especially in OSS, are more tuned to community building roles and end up being mentors more often [7]. This often translates to women being seen as community managers and losing their “engineer” voice in decision making [69]. Further, given the gender imbalance in OSS (past work having found women contributors to range around 10% [69]) and past research having shown that same-gender mentor-mentee dyadic relationships are more common [77], it can mean that fewer women have mentors. Both of these phenomena disadvantage women in OSS. Therefore, we wanted to investigate if implicit mentoring is also disproportionately done by women, leading to the research question:

RQ #3b: What role do women play in implicit mentoring? To answer this question, we analyzed a reduced dataset comprising 6,359 PR/PR-comment contributors (out of which 1,976 contributors were PR-comment authors). Recall, we only kept data of individuals where the Name-sor [59] API predicted the gender with high confidence ($> 90\%$). Table 6 shows the distribution of the gender of contributors in this dataset and their mentoring activity. The gender distribution shows that 4.10% (81) were women and the 95.90% (1,895) men. A majority of these contributors—54 women (66.67% of women) and 1,237 men (65.28% of men)—served as implicit mentors.

Past research has shown that the conventional view of mentoring is that women tend to do more mentoring [7], [78], [79]. In our dataset, we see that trend with slightly more women (66.67%) serving as mentors as compared to men (65.28%). To investigate if the differences between the proportions of implicit mentoring comments provided by men compared to women are statistically significant, we performed a two sample Z-test of proportions [80] (see Table 7 (Overall column)).

The results for *overall implicit mentoring* show that differences are significant ($estimate=-0.09$, P value <0.001), with women providing implicit mentoring 9% times more than men. However, as the estimated ratio of differences is small, we calculated the effect size using Cohen’s d [70]. An effect size of $d = 0.19$ (Table 7, Overall Column) indicates the differences are small [70].

Observation 5: Overall, more women than men provide implicit mentoring, but the difference is small.

Next, we investigate the different implicit mentoring approaches, such as top-down, peer-to-peer, and bottom-up mentoring (we only use project-specific data for this analysis). Since we are analyzing the mentoring approaches

TABLE 7

Proportion test comparing implicit mentoring provided by men vs. women through PR-comments.

	Overall	Top→Down	Peer→Peer	Bottom→Up
Z-score	6.48	1.08	5.17	4.52
P-value	< 0.001	0.09	< 0.001	< 0.001
Estimated differences	-0.09	-0.03	-0.11	-0.12
Cohen d	0.19	0.06	0.23	0.27

$H_0 : P_1 - P_2 = 0$, $H_a : P_1 - P_2 \neq 0$, adjusted $\alpha = 0.017$

P_1 :P(PR-comments by men: Mentoring/All)

P_2 :P(PR-comments by women: Mentoring/All)

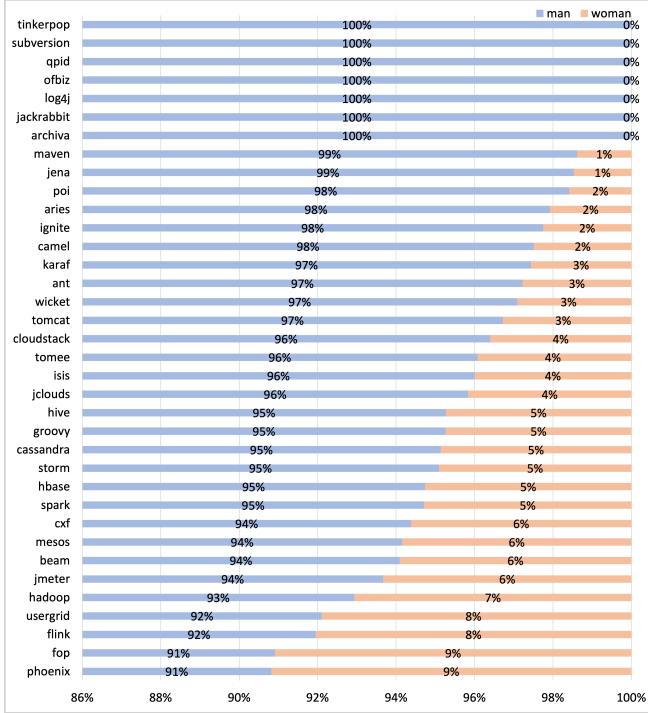


Fig. 3. Gender distributions of contributors in the projects in our dataset.

in pairs. Table 7 presents the results of two sample Z-test of proportions and the effect sizes. Since we perform multiple-hypotheses testing, we applied Bonferroni correction to adjust P values [58], which gives an adjusted $\alpha = 0.017$.

There are no significant differences between implicit mentoring done by men and women for Top→Down ($estimate = -0.03$, P value=0.09). When considering peer implicit mentoring, more women than men did implicit mentoring ($estimate=-0.11$, P value<0.001). While the differences are significant, the effect size is small (Cohen $d = 0.23$). In addition, more women than men were involved in Bottom→Up implicit mentoring ($estimate = -0.12$, P value<0.001), with small effect size (Cohen $d = 0.27$).

Observation 6: There are significant differences between implicit mentoring done by men and women, with women providing more peer-peer and bottom-up implicit mentoring.

Past work has found *homophily*—the tendency for people to seek out or be attracted to those who are similar to

TABLE 8

Implicit mentoring gender information.

	Overall	Top→Down	Peer→Peer	Bottom→Up
W→W	95 (0.36%)	5 (0.04%)	86 (0.87%)	4 (0.13%)
W→M	295 (1.11%)	83 (0.61%)	111 (1.12%)	101 (3.29%)
M→W	1,353 (5.08%)	729 (5.32%)	577 (5.85%)	47 (1.53%)
M→M	24,887 (93.45%)	12,876 (94.03%)	9,095 (92.16%)	2,916 (95.05%)
Total	26,630	13,693	9,869	3,068

TABLE 9

Proportion test results of cross-gender mentoring

	Overall	Top→Down	Peer→Peer	Bottom→Up
Z-score	57.35	35.22	27.49	44.46
P-value	< 0.001	< 0.001	< 0.001	< 0.001
Estimated differences	-0.70	-0.89	-0.50	-0.95
Cohen d	1.65	2.19	1.20	2.50

$H_0 : P_1 - P_2 = 0$, $H_a : P_1 - P_2 \neq 0$, adjusted $\alpha = 0.017$

P_1 : (m→w) / (m→m+m→w)

P_2 : (w→m) / (w→w+w→m)

themselves—by gender among mentor-mentee pairs [23].

Before analyzing for homophily in implicit mentoring, we first analyzed the dataset to ensure whether there are women contributors in the projects for men to mentor. Figure 3 shows the gender distribution of contributors in the projects in our dataset. There were seven projects with no women contributors, and in one project Namesor did not identify the gender of contributors with >90% confidence, so we excluded these eight projects from our analysis.

Table 8 (Overall column) presents the number of implicit mentoring PR-comments grouped by the genders of the mentor→mentee pair. That is, w→w means a woman mentored another woman and a w→m means a woman mentored a man. Our dataset shows that homophily is present in a majority of cases (93.81%). A two-sample proportionality test between homophilic implicit mentoring and cross-gender implicit mentoring show that these differences are significant ($estimate=0.88$, P value<0.001) and the effect size large (Cohen $d = 2.13$). Similarly, homophilic mentoring was significantly more (P value<0.001) than cross-gender mentoring for top-down, peer-to-peer, and bottom-up mentoring with large effect sizes (Cohen $d > 2.0$).

In the small percentage of cases (6.19%) where cross-gender implicit mentoring occurred, there is evidence that more women than men “crossed” gender boundaries when giving implicit mentoring feedback (see Table 9, Overall column: $estimate=-0.70$, P value<0.001) with large effect size (Cohen $d = 1.65$). We see similar results for the different implicit mentoring approaches, with more women than men participating in cross-gender implicit mentoring; where the differences are significant (P value<0.001) with large effect size (Cohen $d > 1.00$).

Observation 7: There is a strong homophily effect in implicit mentoring. In the few cases of cross-gender mentoring, women tend to cross gender boundaries more often than men (70%).

5 DISCUSSION

Mentoring has been shown to be effective in helping with onboarding newcomers to OSS [13] as well as for existing contributors, as P3 said: *"I've gotten mentors, technically, my entire career in tech. And I mostly have looked for these mentors".* To the best of our knowledge, we are the first to identify and investigate implicit mentoring in OSS. This section discusses the implications of our research and practice for OSS communities.

5.1 Creating an Appreciative Community

Creating an appreciative community where mentoring activities are visible and mentors acknowledged is an important consideration for OSS projects, especially since research has shown that mentoring takes effort, which reduces the technical productivity of mentors [7], [13]. Currently, mentors are unacknowledged that causes them to disengage, as P4 said: *"there is no recognition, no kudos, no kind of positive reinforcement for them to continue being a mentor. So, usually, they are a mentor once and then they leave."*

OSS communities can explore different mechanisms to "bake in" appreciation for mentors in their project.

- 1) The code of conduct of OSS projects can explicitly mention thanking mentors for different activities, including code reviews.
- 2) A lightweight mechanism to acknowledge mentors could be creating an attribution tag, such as @mentor to allow contributors to formally acknowledge mentoring they received when creating their contribution.
- 3) OSS communities can use our approach to identify implicit mentoring, which they can then use to create "karma" points to recognize mentoring (or other non-code related activities).
- 4) Project hosting sites such as GitHub can use our approach to identify implicit mentors and include implicit mentoring on contributor profile pages.
- 5) OSS projects can use our approach to identify and highlight the amount of (implicit) mentoring in the project. Past work has found signals that attract newcomers to the project, where a welcoming community is one of the top signals [81].

OSS researchers can investigate implicit mentors in more depth. For example, what motivates mentors? Past work has found that motivations to join OSS versus remain in OSS changes. As contributors become experienced members, their motivations change from extrinsic to intrinsic [82]. Another research question is to what extent do the professional and topical interests of the mentors and mentees need to align? For example, P2 stated *"...why would I help them unless they are going to help me achieve my own personal my own business goals?"* Finally, researchers can explore what barriers exist for implicit mentors in giving feedback within the constraints of code review tools.

5.2 Homophily in Implicit Mentoring

In our study, we find the occurrence of homophily in implicit mentor-mentee pairs especially for men. There might be multiple reasons behind this, such as unconscious bias against other genders, personal preferences or mentor and

mentee's common expectations from the relationship. As McPherson et al. [83] found, connections and friendships are based on social processes and personal preferences and are not randomly made. Therefore, in non-random mentorships, it is more likely to find homophily than it is when assigned formally. Another reason could be the mentee's view of same-gender mentors as being able to empathize with issues specific to their gender [84]. The extensive amount of homophily in (implicit) mentoring (> 90%) amplifies an important call to action of improving diversity in OSS. The already low number of women in OSS and the deleterious effects of cross-gender mentoring on women creates a negative feedback loop that further disadvantages women in OSS.

5.3 Effects of Implicit Mentoring...

...on Mentees: Past work has shown that informal mentoring helps build effective commitment—deeper engagement and identification with the project [18]. It is not clear whether implicit mentoring, which by its very nature is brief and topical, can achieve such results. Further work is needed to investigate questions such as, Does implicit mentoring promote such effective commitment? Do mentees who received implicit mentoring improve their productivity? Do mentees become more engaged with the community and give back by becoming mentors themselves? Does implicit mentoring improve the retention of contributors?

...on Projects: In our work, we investigated the relation between PR complexity and implicit mentoring. We used two simple metrics as a proxy for PR complexity. Other metrics could be investigated, such as, the complexity of the code snippet that was committed for the PR, the complexity of the proposed change based on its centrality or its impact. Further investigation can shed light on why complex PR attracted more implicit mentoring. Was it a cascading effect, where a complex or controversial PR attracted attention, which in turn invited more attention [56], or are there other factors at play? Finally, if the association of implicit mentoring and PR reopen is strong even after considering other factors that were investigated in [57], it could be used as a possible predictor to identify the likelihood of PR being opened in the future.

...on Mentors: A recurring problem raised by our interviewees and literature is mentor "burnout" and work overload [7], [13], [30]. Since by its very nature, implicit mentoring constitutes a brief interaction, it is possible that if this kind of mentoring is recognized by OSS projects, mentoring can become more sustainable. It is also possible that the brief interactions characterized in implicit mentoring can be transformed into longer-lasting interactions. Research has shown that those informal mentoring relationships are more satisfying and often rooted in friendship [20], [21], [22]. One of the problems with mentor-mentee matching is diverting interests. Given implicit mentoring is topical, it is perhaps feasible to match mentor-mentees together who share a common passion for some technical aspects of the project. In fact, informal mentor networks can be created based on individuals who share topical interests. Mullen and Klimaitis [19] identified effective alternate mentoring models, such as group mentoring or collaborative mentoring. Such a collaborative mentoring model can reduce the

workload among mentors, while at the same time creating a cohort of like-minded individuals who support each other.

6 THREATS TO VALIDITY

Like any other empirical research, our study also has threats. We have taken all possible measures to offset the impact of these potential threats as we detail below.

External validity. The first threat to our study is our analysis of implicit mentoring in only Apache projects. The findings from our dataset might not generalize to other OSS projects.

Construct validity. The second threat to validity can occur due to our use of the ‘Namsor’ API [59] to identify the gender of mentors and mentees in our dataset. “Namsor” requires a person’s full name and geographic location to predict a gender. However, not all GitHub contributors provide this information due to privacy concerns, which might cause noise in our data. To mitigate this threat, we only included contributors who had both their geographical location available on GitHub and only use the prediction result if the probability of prediction is greater than 90%, as done in an earlier research [60].

Additionally, some of the projects in our dataset might use external communication channels such as Jira to provide feedback on changes that we cannot identify from GitHub.

Internal validity. The final threat to validity is the subjectivity of the data. To create the training dataset for our classifier, two researchers manually labelled implicit mentoring in PR-comments. To mitigate this threat, we calculated the inter-rater reliability [46] among the two researchers. We describe our method in detail in the methodology section to allow replication of the study.

7 CONCLUSIONS AND FUTURE WORK

In this work, we define the concept of implicit mentoring—everyday development activities such as code reviews, where a mentor provides an underlying explanation when providing suggestions, instructions, or mechanisms to address errors. Through an empirical investigation of PR-comments in 37 Apache projects, our results show that—27.41% of PR includes implicit mentoring created by 63.37% out of 4,644 PR-comment contributors.

Implicit mentoring bucks the traditional dyadic, top-down mentoring model, instead comprising a large portion of non-traditional mentoring (bottom-up and peer to peer). Given the large amount of implicit mentoring taking place that is currently unacknowledged in a project, mechanisms to acknowledge implicit mentors through badging or other mechanisms of appreciation can help make mentoring sustainable. As in other mentorship models, homophily was dominant in implicit mentoring (> 90% of all implicit mentoring), especially for men. While this is expected, it raises serious concerns.

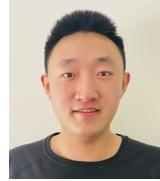
In our future work we plan to address how implicit mentoring impacts mentees, mentors and the organization. We will also investigate the effects of implicit mentoring on diversity in OSS and whether fostering lightweight mentor networks can help to reduce the long-standing problem of lack of women participation in OSS.

REFERENCES

- [1] C. de Souza Costa, J. J. Figueiredo, J. F. Pimentel, A. Sarma, and L. G. P. Murta, “Recommending participants for collaborative merge sessions,” *IEEE Transactions on Software Engineering*, 2019.
- [2] N. Ducheneaut, “Socialization in an open source software community: A socio-technical analysis,” *Computer Supported Cooperative Work (CSCW)*, vol. 14, no. 4, pp. 323–368, 2005.
- [3] P. N. Courant and R. J. Griffiths, “Software and collaboration in higher education: A study of open source software,” *New York: Ithaca*. Retrieved January, vol. 30, p. 2009, 2006.
- [4] L. Zhao and F. P. Deek, “User collaboration in open source software development,” *Electronic Markets*, vol. 14, no. 2, pp. 89–103, 2004.
- [5] K. E. Kram, *Mentoring at work: Developmental relationships in organizational life*. University Press of America, 1988.
- [6] A. Schilling, S. Laumer, and T. Weitzel, “Train and retain: the impact of mentoring on the retention of floss developers,” in *Proceedings of the 50th annual conference on Computers and People Research*, 2012, pp. 79–84.
- [7] S. Balali, I. Steinmacher, U. Annamalai, A. Sarma, and M. A. Gerosa, “Newcomers’ barriers... is that all? an analysis of mentors’ and newcomers’ barriers in oss projects,” *Computer Supported Cooperative Work (CSCW)*, vol. 27, no. 3, pp. 679–714, 2018.
- [8] F. Fagerholm, A. S. Guinea, J. Münch, and J. Borenstein, “The role of mentoring and project characteristics for onboarding in open source software projects,” in *Proceedings of the 8th ACM/IEEE international symposium on empirical software engineering and measurement*, 2014, pp. 1–10.
- [9] “Google summer of code,” Google summer of code, accessed: 2021-07-30. [Online]. Available: <https://summerofcode.withgoogle.com/about/>
- [10] “Mentorship programs give back to the open source community,” The linus Foundation, accessed: 2021-07-30. [Online]. Available: <https://linuxfoundation.org/about/diversity-inclusivity/mentorship/>
- [11] T. A. S. Foundation, “The apache software foundation: Mentoringprogramme,” accessed: 2021-05-30. [Online]. Available: <https://community.apache.org/mentoringprogramme.html>
- [12] S. Balali, U. Annamalai, H. S. Padala, B. Trinkenreich, M. A. Gerosa, I. Steinmacher, and A. Sarma, “Recommending tasks to newcomers in oss projects: How do mentors handle it?” in *Proceedings of the 16th International Symposium on Open Collaboration*, 2020, pp. 1–14.
- [13] F. Fagerholm, A. S. Guinea, J. Borenstein, and J. Münch, “Onboarding in open source projects,” *IEEE Software*, vol. 31, no. 6, pp. 54–61, 2014.
- [14] Apache, “The apache software foundation-the apache way,” accessed: 2021-09-30. [Online]. Available: <https://www.apache.org/theapacheaway/>
- [15] G. Gousios, A. Zaidman, M.-A. Storey, and A. Van Deursen, “Work practices and challenges in pull-based development: The integrator’s perspective,” in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 1. IEEE, 2015, pp. 358–368.
- [16] O. Kononenko, T. Rose, O. Baysal, M. Godfrey, D. Theisen, and B. De Water, “Studying pull request merges: a case study of shopify’s active merchant,” in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, 2018, pp. 124–133.
- [17] L. T. Eby, M. M. Butts, B. J. Hoffman, and J. B. Sauer, “Cross-lagged relations between mentoring received from supervisors and employee ocb: Disentangling causal direction and identifying boundary conditions.” *Journal of Applied Psychology*, vol. 100, no. 4, p. 1275, 2015.
- [18] S. C. Payne and A. H. Huffman, “A longitudinal examination of the influence of mentoring on organizational commitment and turnover,” *Academy of Management Journal*, vol. 48, no. 1, pp. 158–168, 2005.
- [19] C. A. Mullen and C. C. Klimaitis, “Defining mentoring: a literature review of issues, types, and applications,” *Annals of the New York Academy of Sciences*, vol. 1483, no. 1, pp. 19–35, 2021.
- [20] L. D. Inzer and C. B. Crawford, “A review of formal and informal mentoring: Processes, problems, and design,” *Journal of leadership Education*, vol. 4, no. 1, pp. 31–50, 2005.
- [21] S. G. Baugh and E. A. Fagenson-Eland, “Formal mentoring programs,” *The handbook of mentoring at work: Theory, research, and practice*, pp. 249–271, 2007.

- [22] Y. P. Bynum, "The power of informal mentoring," *Education*, vol. 136, no. 1, pp. 69–73, 2015.
- [23] B. R. Ragins and J. L. Cotton, "Mentor functions and outcomes: a comparison of men and women in formal and informal mentoring relationships," *Journal of applied psychology*, vol. 84, no. 4, p. 529, 1999.
- [24] A. J. Ko, L. Hwa, K. Davis, and J. C. Yip, "Informal mentoring of adolescents about computing: Relationships, roles, qualities, and impact," in *Proceedings of the 49th ACM technical symposium on computer science education*, 2018, pp. 598–603.
- [25] A. Nandi and M. Mandernach, "Hackathons as an informal learning platform," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 2016, pp. 346–351.
- [26] A. Begel and B. Simon, "Novice software developers, all over again," in *Proceedings of the fourth international workshop on computing education research*, 2008, pp. 3–14.
- [27] S. Kumar and C. Wallace, "Communication strategies for mentoring in software development projects," in *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2014, pp. 111–114.
- [28] I. Steinmacher, I. S. Wiese, and M. A. Gerosa, "Recommending mentors to software project newcomers," in *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)*. IEEE, 2012, pp. 63–67.
- [29] I. Steinmacher, M. A. G. Silva, M. A. Gerosa, and D. F. Redmiles, "A systematic literature review on the barriers faced by newcomers to open source software projects," *Information and Software Technology*, vol. 59, pp. 67–85, 2015.
- [30] B. Trinkenreich, M. Guizani, I. Wiese, A. Sarma, and I. Steinmacher, "Hidden figures: Roles and pathways of successful oss contributors," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–22, 2020.
- [31] M. Johnson and M. Senges, "Learning to be a programmer in a complex organization: A case study on practice-based learning during the onboarding process at google," *Journal of Workplace Learning*, 2010.
- [32] J. O. Silva, I. Wiese, D. M. German, C. Treude, M. A. Gerosa, and I. Steinmacher, "Google summer of code: Student motivations and contributions," *Journal of Systems and Software*, vol. 162, p. 110487, 2020.
- [33] J. D. O. Silva, I. S. Wiese, D. M. German, I. F. Steinmacher, and M. A. Gerosa, "How long and how much: What to expect from summer of code participants?" in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2017, pp. 69–79.
- [34] GitHub, "Paper supplement documents," accessed: 2021-09-30. [Online]. Available: [https://github.com/ZixuanSteveFeng/ImplicitMentoring\\$](https://github.com/ZixuanSteveFeng/ImplicitMentoring$)
- [35] M. Guizani, A. Chatterjee, B. Trinkenreich, M. E. May, G. J. Noa-Guevara, L. J. Russell, G. G. Cuevas Zambrano, D. Izquierdo-Cortazar, I. Steinmacher, M. A. Gerosa *et al.*, "The long road ahead: Ongoing challenges in contributing to large oss organizations and what to do," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW2, pp. 1–30, 2021.
- [36] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*. CRC press, 2015, vol. 4.
- [37] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.
- [38] S. Keele *et al.*, "Guidelines for performing systematic literature reviews in software engineering," Technical report, Ver. 2.3 EBSE Technical Report. EBSE, Tech. Rep., 2007.
- [39] M. Gharehyazie, D. Posnett, B. Vasilescu, and V. Filkov, "Developer initiation and social interactions in oss: A case study of the apache software foundation," *Empirical Software Engineering*, vol. 20, no. 5, pp. 1318–1353, 2015.
- [40] S. Kabinna, C.-P. Bezemer, W. Shang, and A. E. Hassan, "Logging library migrations: A case study for the apache software foundation projects," in *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. IEEE, 2016, pp. 154–164.
- [41] B. Chen and Z. M. J. Jiang, "Characterizing logging practices in java-based open source software projects—a replication study in apache software foundation," *Empirical Software Engineering*, vol. 22, no. 1, pp. 330–374, 2017.
- [42] T. Chełkowski, P. Gloor, and D. Jemielniak, "Inequalities in open source software development: Analysis of contributor's commits in apache software foundation projects," *PLoS One*, vol. 11, no. 4, p. e0152976, 2016.
- [43] G. A. Oliva, J. T. da Silva, M. A. Gerosa, F. W. S. Santana, C. M. L. Werner, C. R. B. de Souza, and K. C. M. de Oliveira, "Evolving the system's core: a case study on the identification and characterization of key developers in apache ant," *Computing and Informatics*, vol. 34, no. 3, pp. 678–724, 2015.
- [44] U. A. Mannan, I. Ahmed, C. Jensen, and A. Sarma, "On the relationship between design discussions and design quality: a case study of apache projects," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 543–555.
- [45] J. Kotrlik and C. Higgins, "Organizational research: Determining appropriate sample size in survey research appropriate sample size in survey research," *Information technology, learning, and performance journal*, vol. 19, no. 1, p. 43, 2001.
- [46] K. A. Hallgren, "Computing inter-rater reliability for observational data: an overview and tutorial," *Tutorials in quantitative methods for psychology*, vol. 8, no. 1, p. 23, 2012.
- [47] A. J. Viera, J. M. Garrett *et al.*, "Understanding interobserver agreement: the kappa statistic," *Fam med*, vol. 37, no. 5, pp. 360–363, 2005.
- [48] M. Pal, "Random forest classifier for remote sensing classification," *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.
- [49] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and bernoulli naïve bayes for text classification," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*. IEEE, 2019, pp. 593–596.
- [50] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [51] J. Laaksonen and E. Oja, "Classification with learning k-nearest neighbors," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, vol. 3. IEEE, 1996, pp. 1480–1483.
- [52] R. Kothari and M. Dong, "Decision trees for classification: A review and some new results," *Pattern recognition: from classical to modern approaches*, pp. 169–184, 2001.
- [53] T. Horváth, R. G. Mantovani, and A. C. de Carvalho, "Effects of random sampling on svm hyper-parameter tuning," in *International Conference on Intelligent Systems Design and Applications*. Springer, 2016, pp. 268–278.
- [54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [55] Y. Yu, H. Wang, V. Filkov, P. Devanbu, and B. Vasilescu, "Wait for it: Determinants of pull request evaluation latency on github," in *2015 IEEE/ACM 12th working conference on mining software repositories*. IEEE, 2015, pp. 367–371.
- [56] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in github: transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 conference on computer supported cooperative work*, 2012, pp. 1277–1286.
- [57] A. Mohamed, L. Zhang, J. Jiang, and A. Ktob, "Predicting which pull requests will get reopened in github," in *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, 2018, pp. 375–385.
- [58] M. A. Napierala, "What is the bonferroni correction," *AAOS Now*, vol. 6, no. 4, p. 40, 2012.
- [59] "Free name gender api, by namsor," Dec 2019, accessed: 2021-07-30. [Online]. Available: <https://namesorts.com/api/>
- [60] L. Santamaría and H. Mihaljević, "Comparison and benchmark of name-to-gender inference services," *PeerJ Computer Science*, vol. 4, p. e156, 2018.
- [61] I. Steinmacher, C. Treude, and M. A. Gerosa, "Let me in: Guidelines for the successful onboarding of newcomers to open source projects," *IEEE Software*, vol. 36, no. 4, pp. 41–49, 2018.
- [62] I. Steinmacher, I. S. Wiese, T. Conte, M. A. Gerosa, and D. Redmiles, "The hard life of open source software project newcomers," in *Proceedings of the 7th international workshop on cooperative and human aspects of software engineering*, 2014, pp. 72–78.
- [63] B. Trinkenreich, I. Wiese, A. Sarma, M. Gerosa, and I. Steinmacher, "Women's participation in open source software: A survey of the literature," *arXiv preprint arXiv:2105.08777*, 2021.
- [64] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles, "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, 2015, pp. 1379–1392.
- [65] D. E. Okurame and S. Balogun, "Role of informal mentoring in the career success of first-line bank managers: A nigerian case study," *Career Development International*, 2005.
- [66] I. Steinmacher, S. Balali, B. Trinkenreich, M. Guizani, D. Izquierdo-Cortazar, G. G. Cuevas Zambrano, M. A. Gerosa, and A. Sarma, "Being a mentor in open source projects," *Journal of Internet Services and Applications*, vol. 12, no. 1, pp. 1–33, 2021.
- [67] J. S. Mota, M. V. Okimoto, E. D. Canedo, and J. S. Mota, "Google summer of code gender diversity: An analysis of the last 4 editions," *Anais do Computer on the Beach*, vol. 12, pp. 117–124, 2021.
- [68] M. Zhou and A. Mockus, "What make long term contributors: Willingness and opportunity in oss community," in *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 2012, pp. 518–528.
- [69] A. Bosu and K. Z. Sultana, "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, 2019, pp. 1–11.
- [70] J. Cohen, *Statistical power analysis for the behavioral sciences*. Academic press, 2013.
- [71] J. Jiang, A. Mohamed, and L. Zhang, "What are the characteristics of reopened pull requests? a case study on open source projects in github," *IEEE Access*, vol. 7, pp. 102751–102761, 2019.
- [72] K. E. Kram, *Mentoring at work: Developmental relationships in organizational life*. University Press of America, 1988. [Online]. Available: <http://psycnet.apa.org/record/1988-97625-000>
- [73] L. Gandhi, "Reverse mentoring-a bottom up approach in mentoring system," *EDITORIAL Food Processing Industry*, p. 41, 2012.
- [74] T. G. Brashear, D. N. Bellenger, J. S. Boles, and H. C. Barksdale Jr, "An exploratory study of the relative effectiveness of different types of sales force mentors," *Journal of Personal Selling & Sales Management*, vol. 26, no. 1, pp. 7–18, 2006.
- [75] R. Hale, "To match or mis-match? the dynamics of mentoring as a route to personal and organisational learning," *Career Development International*, 2000.
- [76] C. Jergensen, A. Sarma, and P. Wagstrom, "The onion patch: migration in open source ecosystems," in *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, 2011, pp. 70–80.
- [77] M. Kogovšek and I. Ograjenšek, "Effects of the same-gender vs. cross-gender mentoring on a protégé outcome in academia: An exploratory study," *Metodoloski Zvezki*, vol. 16, no. 1, pp. 61–78, 2019.
- [78] J. J. Sosik and V. M. Godshalk, "The role of gender in mentoring: Implications for diversified and homogenous mentoring relationships," *Journal of Vocational Behavior*, vol. 57, no. 1, pp. 102–122, 2000.
- [79] B. Trinkenreich, "Please don't go-a comprehensive approach to increase women's participation in open source software," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 2021, pp. 293–298.
- [80] F. LD and G. Van Belle, "Biostatistics: a methodology for health sciences," Wiley, New York, pp. 496–595, 1993.
- [81] H. S. Qiu, Y. L. Li, H. S. Padala, A. Sarma, and B. Vasilescu, "The signals that potential contributors look for when choosing open-source projects," *Proc. ACM Hum. Comput. Interact.*, vol. 3, no. CSCW, pp. 122:1–122:29, 2019. [Online]. Available: <https://doi.org/10.1145/3359224>
- [82] M. A. Gerosa, I. Wiese, B. Trinkenreich, G. Link, G. Robles, C. Treude, I. Steinmacher, and A. Sarma, "The shifting sands of motivation: Revisiting what drives contributors in open source," in *43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021, Madrid, Spain, 22-30 May 2021*. IEEE, 2021, pp. 1046–1058. [Online]. Available: <https://doi.org/10.1109/ICSE43902.2021.00098>
- [83] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.
- [84] B. R. Ragins and D. B. McFarlin, "Perceptions of mentor roles in cross-gender mentoring relationships," *Journal of Vocational Behavior*, vol. 37, no. 3, pp. 321–339, 1990. [Online]. Available: <https://www.sciencedirect.com/>



Zixuan Feng is a Ph.D. student at Oregon State University in the School of Electrical Engineering and Computer science. His research interest is in supporting collaboration in distributed teams on software engineering and data mining.



Amreeta Chatterjee is Ph.D student in the School of Electrical Engineering and Computer science, at Oregon State University. Her research lies in the intersection of Software Engineering (SE) and Human Computer Interaction (HCI) and her interests include making software more inclusive to people.



Anita Sarma is an Associate Professor at Oregon State University in the School of Electrical Engineering and Computer science. Her research investigate human factors in software engineering. She has over 100 papers in journals and conferences. Her work has been recognized by an NSF CAREER award as well as several best paper awards.



Iftekhar Ahmed is an Assistant Professor of Informatics in the Donald Bren School of Information and Computer Science at the University of California, Irvine. His research focuses on software engineering in general and combining software testing, analysis and data mining to come up with better tools and techniques in particular. He finished his B.Sc. in Computer Science and Engineering from Shahjalal University of Science and Technology, Bangladesh and after working in the industry for 4 years, did his