Who Will Stop Contributing to OSS Projects? Predicting Company Turnover Based on Initial Behavior

MIAN QIN, Beijing Institute of Technology, China
YUXIA ZHANG*, Beijing Institute of Technology, China
KLAAS-JAN STOL, University College Cork, Ireland, Lero, Ireland, and SINTEF, Norway
HUI LIU, Beijing Institute of Technology, China

Open Source Software (OSS) projects are no longer only developed by volunteers. Instead, many organizations, from early-stage startups to large global enterprises, actively participate in many well-known projects. The survival and success of OSS projects rely on long-term contributors, who have extensive experience and knowledge. While prior literature has explored volunteer turnover in OSS, there is a paucity of research on company turnover in OSS ecosystems. Given the intensive involvement of companies in OSS and the different nature of corporate contributors vis-a-vis volunteers, it is important to investigate company turnover in OSS projects. This study first explores the prevalence and characteristics of companies that discontinue contributing to OSS projects, and then develops models to predict companies' turnover. Based on a study of the Linux kernel, we analyze the early-stage behavior of 1,322 companies that have contributed to the project. We find that approximately 12% of companies discontinue contributing each year; one-sixth of those used to be core contributing companies (those that ranked in the top 20% by commit volume). Furthermore, withdrawing companies tend to have a lower intensity and scope of contributions, make primarily perfective changes, collaborate less, and operate on a smaller scale. We propose a Temporal Convolutional Network (TCN) deep learning model based on these indicators to predict whether companies will discontinue. The evaluation results show that the model achieves an AUC metric of .76 and an accuracy of .71. We evaluated the model in two other OSS projects, Rust and OpenStack, and the performance remains stable.

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

Additional Key Words and Phrases: Open source software, commercial participation, OSS community, early participation

ACM Reference Format:
Mian Qin, Yuxia Zhang, Klaas-Jan Stol, and Hui Liu. 2025. Who Will Stop Contributing to OSS Projects? Predicting Company Turnover Based on Initial Behavior. *Proc. ACM Softw. Eng.* 2, FSE, Article FSE123 (July 2025), 24 pages. https://doi.org/10.1145/3729393

1 Introduction

Open source software (OSS) has been the cornerstone of our digital world. Sonatype's 2023 report [26] shows that 96% of commercial software now contains OSS, making up 77% of the codebase. Unsurprisingly, companies increasingly contribute to OSS projects and play critical roles [28, 73].

*Corresponding author

Authors' Contact Information: Mian Qin, Beijing Institute of Technology, Beijing, China, qinmian@bit.edu.cn; Yuxia Zhang, Beijing Institute of Technology, Beijing, China, yuxiazh@bit.edu.cn; Klaas-Jan Stol, University College Cork, Cork, Ireland and Lero, Cork, Ireland and SINTEF, Trondheim, Norway, k.stol@ucc.ie; Hui Liu, Beijing Institute of Technology, Beijing, China, liuhui08@bit.edu.cn.

ACM 2994-970X/2025/7-ARTFSE123
https://doi.org/10.1145/3729393

For example, in 2023, companies played a significant role in the Linux kernel development, and were responsible for approximately 70% of total commit activities [74]. Such intense contribution level feeds into the success of OSS projects [86], and as projects become increasingly dependent on corporate participation, it is important that companies become long-term contributors (LTC).

However, companies may face challenges when engaging with OSS projects [35, 65, 86], such as having to adhere to standards and processes while making contributions, getting contributions accepted, and finding fruitful collaborations with others within the OSS project ecosystem that may comprise many other companies. Despite the important role of companies in the development of many OSS projects, these difficulties can ultimately contribute to their decision to stop contributing [82]. Given the critical role of OSS in society, it is important to understand the dynamics of what drives companies to continue and become LTCs, or what stops them from contributing. Because it takes great effort for developers to get involved in OSS and become experienced [70]. Further, companies usually assign a group of developers to OSS projects [86], and losing them will bring more serious consequences, such as abandoned code blocks and knowledge loss. Intervention efforts can be more successful when attention is directed to initial contribution experiences [46, 89]. Thus, understanding the triggers for companies' withdrawal based on their early participation would facilitate the success of OSS projects by assisting timely retain corporate contributors, or look ahead at what might happen when a specific company withdrawal and be ready to handle it.

Prior research on corporate participation in OSS has explored why corporations join OSS communities [8, 14, 30, 63], their business strategies [17, 18, 86, 92], collaboration dynamics [22, 31], and the impact of company engagement on OSS development [75, 85, 87, 92]. We are aware of only one study that has considered company discontinuation [82]; however, what characterizes those companies that discontinue their contributions to OSS projects and their turnover prediction approaches are as of yet lacking. Turnover and retention have been studied at the level of *individual developers* [37, 51, 68, 79]; these studies highlight the challenges in maintaining a stable developer base in OSS. Researchers have explored the relationship between developers' initial behaviors and their long-term participation [90] and the success of OSS projects [78]. To our knowledge, no studies have explored the initial behavior of *companies* that stopped their contributions, despite the major role that companies play in OSS development today. This study bridges this gap by comparing the characteristics of those companies that have become long-term contributors and those that have withdrawn, based on their participation behavior. We propose a prediction model to forecast companies' departure. Of course, not all companies seek to become long-term contributors to the projects they have contributed to. This study mainly focuses on the sustainable development of OSS projects, and attempts to provide insights that can help OSS communities better deal with the potential losses associated with a company's withdrawal.

We begin this study by identifying the prevalence of companies that discontinue contributing to OSS projects. To focus our analysis, we first concentrated on one specific OSS project: the Linux kernel. As one of the most prominent and long-standing OSS projects, with contributions from hundreds of companies, it provides a strong case for exploring company turnover in depth. To understand the scale of corporate turnover, we first ask:

Research Question 1: What is the prevalence of companies that discontinue contributing to the Linux kernel?

We then seek to understand whether we can distinguish those companies that discontinue contributing from those that remain active, with an aim to identify indicators of "early contribution behavior." Hence, we ask:

Research Question 2: How do companies that discontinue contributions differ from those that continue in terms of early-stage contributions?

Finally, we explore whether these indicators can help us predict corporate turnover; the ability to do so would allow for early interventions to ensure the continuity and maintain the health of OSS communities. We ask:

Research Question 3: Can we automatically predict which company will stop contributing based on early behavior?

This paper makes four contributions to the literature on corporate participation in OSS projects. First, this study contributes a comprehensive activity analysis of corporate activity in the Linux kernel. We collected and analyzed data about 1,322 companies that had at least three months of involvement during a period of 18 years, from January 2005 to November 2023. We found that, on average, 12% of companies involved in the Linux kernel discontinued their contributions each year. One in six of these withdrawing companies used to play core roles in terms of their contributions.

Second, this paper presents a novel and multidimensional participation measurement framework. The framework consists of seven constructs (using nine indicators) for measuring the impact of various factors on companies' continued contributions to OSS projects. We used this framework to develop a series of hypotheses to test differences between companies that remained active and those that discontinued contributing. We find that companies that cease contributing to the Linux kernel project initially demonstrate a lower contribution intensity and smaller scope, with a preference for perfective tasks over implementing features or fixing bugs. These companies also engage less in collaborations and are generally smaller in scale.

A third contribution is a Temporal Convolutional Network (TCN)-based deep learning model for effective forecasting of future corporate contributions based on their initial contribution behavior. We formulated a series of models based on nine widely used classification techniques; the evaluation results show that the TCN-based model achieved the most promising performance with an AUC of 0.76. To further validate the robustness and generalizability of this model, we extended the analysis to two other large and popular OSS projects: OpenStack and Rust. The model demonstrated consistently promising performance across these projects, highlighting its potential use in predicting corporate long-term participation in OSS projects. These results allow OSS communities to engage with companies earlier in an attempt to retain these companies or prepare for their leaving.

Finally, this paper provides several strategic insights for open source sustainability: practical implications to help OSS communities grow companies into long-term contributors and future research avenues to gain a more in-depth understanding of companies' long-term involvement.

The rest of this paper is organized as follows: Section 2 describes related work on commercial participation in OSS and developer turnover. Section 3 elaborates on the method we used. Section 4 covers the results of this study. Section 5 discusses the implications for both practice and research. We analyze threats to the validity of our reported findings in Section 6 and conclude this study with a summary in Section 7.

2 Related Work

This section reviews literature along two main themes. First, we review studies of how companies engage in OSS ecosystems, often by assigning employees to contribute code, which overlaps with individual developer participation. Thus, the second theme is research related to developer turnover, to learn possible dimensions to predict.

2.1 Corporate Participation in Open Source Projects

Driven by increasing corporate participation in OSS projects, researchers have focused on four aspects of corporate engagement: motivations, business strategies, contribution patterns, and impacts on OSS projects.

Early research highlighted that companies' motivations stemmed from fostering innovation, reducing costs, and other various drivers [8, 11, 23, 32, 63]. For example, Wang et al. [76] identified social and technological incentives to play a role, while Henkel et al. observed that economic motivations are the primary drivers for such participation [32]. Recently, Guizani et al. [29] have categorized OSS participation motivations into four distinct levels: Founders' Vision, Reputation, Business Advantage, and Reciprocity. Further, research has focused on business strategies for commercial OSS involvement that extend to understanding how companies engage with and benefit from OSS. Zhou et al. [92] examined three projects producing identical software, categorizing company participation into three models: hosting, supporting, and collaborating.

Beyond studying companies at the individual level, prior research also considered the role of collaborations in driving innovation and shaping participation. For instance, Linåker et al. [45] noted that collaboration patterns among companies could drive innovation and reduce time-to-market in the Apache Hadoop ecosystem. Furthermore, Zhang et al. [87] analyzed project characteristics to outline corporate collaborations and participation in extensive OSS ecosystems. Another strand of research focused on the impact of corporate involvement in OSS projects, revealing both positive and negative effects. For example, Valiev et al. [75] studied the PyPI ecosystem, and concluded that corporate contributions significantly bolster project sustainability. In contrast, Zhang et al. [82] observed that divergent roadmaps between companies and the OpenStack OSS project led to companies discontinuing their contributions. More recently, Newton et al. [55] highlighted corporate involvement may constrain a project's contributor base, but subsidizing OSS development can increase project accessibility and openness.

We noticed one study [82] has begun to consider the reasons behind company withdrawal. Zhang et al. [82] conducted an empirical study of OpenStack and observed an increase in the number of companies discontinuing their involvement over time. This study identified eight reasons for company withdrawal by surveying core developers from these companies. To the best of our knowledge, no study has yet attempted to predict company retention or withdrawal in their initial participation. From the perspective of project health and sustainable evolution, the sudden departure of companies can pose negative impacts. Accurately predicting companies' withdrawal based on their early participation behaviors can give project maintainers more time to mitigate the subsequent negative impacts, e.g., code files lose their original maintenance, thereby aiding in the project's sustainability. Therefore, we aim to predict company turnover from an early-stage perspective.

2.2 Turnover of Developers in Open Source Projects

There are extensive previous studies primarily concentrating on the reasons for and impacts of individual developer turnover within these projects. The causes of turnover are multifaceted, and influenced by both external and internal factors. External factors relate to elements outside the developer's control, often involving broader organizational or environmental conditions. For example, regulatory gaps and insufficient incentives are common external factors that lead to volunteer dissatisfaction, prompting developers to discontinue their participation [79]. Similarly, the reputation of the organization plays a role in turnover, as developers may be influenced by how they perceive the organization's external standing [35]. Lastly, social dynamics play a role in retention, Respectfulness in discussions is linked to higher turnover rates, while discussions focused on social power dynamics help improve retention [37].

On the other hand, internal factors are more personal to the developers themselves and their experiences within the OSS ecosystem. For instance, Schilling et al. [68] found developer experience and conversational knowledge critical for retention, applying traditional recruitment concepts (Person-Job and Person-Team fit) to OSS projects. Miller et al. [51] observed long-time contributors leave for various reasons, often due to significant life transitions. Schaarschmidt et al. [67] suggest that the company-employed developers' turnover intention is lowest, when there is a high level of congruence between company and community identification.

Research has also addressed the impact of turnover, revealing key consequences. One prominent issue is knowledge loss [36], when experienced developers leave, taking with them valuable expertise. Foucault et al. [25], analyzing five OSS projects, outlined external and internal turnover concepts, noting external newcomers' activities can deteriorate software quality. Rigby et al. [61] applied financial risk analysis techniques to gauge projects' vulnerability to developer turnover, aiding in risk assessment for potential departures. Technical and social growth in the OSS ecosystem coincides with higher rates of contributor and project abandonment [13].

Finally, considering that our work focuses on making predictions from an early stage, there were two studies concerning developers' early participatory behaviors and initial stages of OSS projects can also lay a foundational basis for our framework: Firstly, Zhou and Mockus [90] emphasized the significance of new participants' initial behaviors and experiences, creating measures for involvement (covering both ability and willingness) and the environment. Regarding the significance of early project activities on its sustainability, Xiao et al. [78] found that during the initial stages of project creation, initial participants with prior OSS experience and focused, steady commitment positively influence future project activities. Although studies focusing on individual developer turnover cannot cover company retention in projects, their proposed dimensions can be reused, because paid developers present company participation.

3 Method

We outline our approach in this section, including project selection (Sec. 3.1), dataset preparation (Sec. 3.2), 'early' participation definition (Sec. 3.3), and hypothesis development (Sec. 3.4).

3.1 Project Selection

Our study aims to explore company turnover in OSS projects. We selected the Linux kernel ecosystem as our primary case study because it is one of the oldest, most well-known OSS projects with significant corporate involvement over a long time. Initiated by Linus Torvalds in 1991, the Linux kernel has become a critical infrastructure and has attracted the involvement of hundreds of companies over the past decades [64]. It accommodates an extensive range of processor architectures, both 32-bit and 64-bit [49], fostering a vast and varied community. By early 2024, it received 1.18 million commits from nearly 39,030 contributors [47]. An annual Linux kernel development report highlighted the involvement of over 1,400 companies [72], underscoring its widespread popularity and use in commercial applications. The intense commercial participation offers an opportunity to explore a rich set of characteristics of companies' early behaviors and long-term contributions.

We also chose other two OSS projects, i.e., OpenStack and Rust, to test the effectiveness of the extracted characteristics from the Linux kernel. We chose these two projects for their importance and impact in distinct domains, namely cloud computing and programming languages. OpenStack, launched in 2010, has been predominantly shaped by corporate contributions [86], with hundreds of companies involved across over 1,000 repositories. Rust [71], which started as a personal project, has grown into a widely loved programming language, now managed by the Rust Foundation, with major tech companies such as Google, Microsoft, and Amazon as key contributors.

3.2 Data Preparation

We accessed all commit data of the Linux kernel up to November 22, 2023, via GitHub's REST API [27], totaling 1,145,070 commits from 27,863 developers. The issue of developers having multiple identities has been extensively recognized in commit history analysis [2, 6, 83, 86]. To address this, we applied a widely used method [93], which augments the developer's name and email address and has been reported to achieve high accuracy with a precision close to 100%, to merge developers' identities. This process merged 2,837 identities, resulting in 25,026 distinct developers.

3.2.1 Affiliation Identification. To analyze companies' participation in the Linux kernel project, it is essential to ascertain the affiliation of each developer, specifically whether a specific company employs them. However, accurately determining a developer's affiliations is challenging because of the absence of resume information in the development history data [83, 86]. We decided to follow a practical and widely used method [83, 92]: determining each developer's affiliation at the moment of their contributions to the Linux kernel by analyzing the domain of their email addresses. Specifically, we implemented the method via four steps, which will be introduced as follows.

(1) Gather all possible organizations. After an initial investigation, we found the Linux Foundation maintains an official organization list [48], which contains 4,436 organizations that have contributed to (not limited to code) the Linux kernel. However, this list is rough and needs additional filtering for three reasons: First, it listed multiple entries for single entities, attributed to capitalization differences or misspellings, such as "Amd", "amd", and "amd.com"; Second, it displayed inconsistent naming, with some reflecting formal business identities like "NVIDIA Corporation", and others as domains like "fr24.com"; Third, it covered a variety of organization types, for example, educational institutions like "bond.edu.au" and volunteer groups represented by public email domains like "gmail.com", whereas our research specifically targets corporate entities.

Before applying the organization list, we address the first two issues by manually cleaning each organization's name, including converting it to lowercase and removing entity types and suffixes. This included common legal entity types, such as "inc.", "corporation", "ltd", and typical internet domain suffixes like ".com", ".org", ".net". For instance, we transformed "Red Hat, Inc." into "redhat". After completing this step, we had a list of 4,312 organization names, which were then paired with developers' email domains from the commit records. If a developer's email ends with an organization domain, such as "@redhat.com", we attribute their affiliation to the corresponding organization name (e.g., "redhat"). We identified 1,924 organizational affiliations in the commit records of the Linux kernel.

(2) Filter free email providers. The 1,924 organizational affiliations may still include some fake 'organization' types brought by the third issue of the official organization list, such as 'gmail.com', which are beyond our corporate focus. Following prior work [82, 87], we used a public domain list of free email providers, which is maintained on GitHub [38], to distinguish corporate participants. Specifically, we labeled developers, whose emails can be found in the public free domain list, as 'volunteer'. After filtering, we still have 1,649 companies pending further verification.

(3) Company verification. To validate that the 1,649 company names are true organizations, we utilized CrunchBase [16], which has been widely used in economics research [19, 81] and hosts detailed profiles of companies, including their founding dates, funding history, key personnel, and industry classifications. We collected company information via CrunchBase's REST API and retrieved each of the 1,649 company names. Company names that could not be accurately matched were deemed less likely to be business entities. Therefore, we omitted these company names from our study. Through this verification, we confirmed the existence of 1,469 distinct companies that assigned 12,305 paid developers to the Linux kernel project.

(4) Attributing contributions. Developers employed by companies may submit commits using their public email addresses. Therefore, we calculated the exact start and end times for each developer's usage of a specific email associated with an organizational domain. The approximate time frame of a developer using the same email is recorded in the 'author_date' field of its contributed commits. Whether commits submitted via the public email are credited to the company depends on the contribution time. If the contribution time falls within the period of using the company email, those commits are attributed to the company; otherwise, they are credited to "volunteer." For example, if a developer contributed 20 commits with a '@gmail.com' account from March to May 2024, while also using an AMD email account (e.g., @amd.com) to submit commits from January to December of the same year, then those 20 commits would be attributed to the AMD Corporation. We find that the cross-use of company and public email is rare in our dataset. Specifically, approximately 4.0% of developers use both their corporate and public emails during the same period, with 5.6% of commits being attributed to the corresponding companies in the Linux kernel. Similarly, when the same developer switches companies, we attribute these contributions to different companies based on the email domains recorded in the commits.

We applied the same approach to collect and clean historical commit data for OpenStack and Rust, which are used to validate the findings and prediction model drawn from the Linux kernel. Specifically, for OpenStack, we selected its two most important repositories, i.e., swift and nova [57]. This dataset comprises a total of 39,175 commits from May 30, 2010, to May 21, 2024. We identified 192 companies employing 1,285 paid developers. For Rust, we collected 162,627 commits from June 23, 2010, to June 29, 2024, contributed by 94 companies. To validate the accuracy of this approach, we randomly selected 50 developers from each of the three projects: the Linux kernel, OpenStack, and Rust. Following [83], we manually verified developers' company affiliations by reviewing publicly available profiles, such as LinkedIn, GitHub, and personal websites. The accuracy of this verification process was found to be 86.8%, 90.6%, and 86.3% for the three projects, respectively. These accuracies are comparable to the validated results reported in [83], which directly asked developers for confirmation. More details about the manual validation process can be found in the online appendix [3].

3.2.2 Identifying Companies that Discontinued Participation. Prior studies identified individual developer abandonment by considering a defined period without contributions, for example, developers who did not contribute for 180 days were supposed to have left [25, 44]. Commercial participation in OSS can change based on their business goals, domain, and development rhythms [82, 86], and companies usually assign more than one developer to make contributions. So applying a uniform threshold to identify departure would risk overlooking the unique operational dynamics and contribution patterns inherent to different companies.

In this study, we followed the measurement of Zhang et al. [82] that empirically studies company turnover in OpenStack, and looked at the contribution interval pattern for each company, namely, how long a *typical* break lasts between contributions for a given company, to determine if they ceased contributing. If a company has not contributed for a period longer than the longest contribution break (up to the last date we have data for), we consider that it discontinued contributing. This method provides a more tailored assessment of company engagement, rather than taking a one-size-fits-all approach [25, 44] that may be imprecise, because companies may work at different paces. As a result, the longest contribution breaks of the 1,469 companies in the Linux kernel range from 0 to 5,732 days, with a median interval of 360 days. It demonstrates the variety of companies' development rhythms. Among the 1,469 companies in the Linux kernel, we used their longest historical contribution intervals as a threshold and identified 629 discontinued companies. For companies that only contributed once, since no historical intervals were available, we followed Zhang et al.'s [82] method, using the median of other companies' longest historical contribution intervals (e.g., 360 days in the Linux kernel) as the threshold. Specifically, if a one-time company's last commit occurred more than 360 days before the data collection cutoff date, we classified it as "withdrawn".

3.3 Early Participation

A central hypothesis in this study is that we can predict companies' discontinuation based on their 'early-stage' participation. Determining an appropriate observation window for companies' initial participation is necessary. Zhou and Mockus [91] model individual developers' initial behavior in their first month of joining an OSS project to predict who will become long-term contributors. Xiao et al. [78] measure early participation using development activities from the first one, three, and five months, respectively. They found that a developer's engagement during the initial three months is more crucial for predicting the sustainability of OSS projects. Given the possible differences between companies and individual developers, we analyzed the monthly contribution of each company in the Linux kernel within their initial six months. The results show that companies averagely have commit activities over the span of three months, which is in line with Xiao et al. [78]. Thus, we designated the three months following a company's first contribution as the time frame that constitutes 'early-stage' participation. Considering the three-month window's limitation for companies joining close to our dataset's cutoff date (22 November 2023), we excluded companies whose first contribution was on, or after, 22 August 2023. Thus, our final dataset contains contributions from 1,322 companies, 620 of which did not continue to contribute.

Table 1. Dimensions to characterize contribution behavior of companies

Dimension	Construct	Variable	Data type	Explanation
Willingness and Capacity	Contribution Intensity	#Commits	discrete	The number of commits made to the project.
		#LoCs	discrete	The amount of code lines modified.
		#Developers	discrete	The number of developers from the company who are contributing.
	Contribution Scope	#Module	discrete	The number of different modules the company has contributed to.
	Contribution Type	Type	nominal	The type of the company's contributions, whether they are for new features (adaptive), improvements (perfective), or bug fixes (corrective).
Collaborative Environment	Scope of Collaboration	Degree centrality	continuous	A measure of a node's centrality in a social network.
	Dominance	Is_influenced	boolean	Whether the company has participated in a module where dominance exists.
Company Attributes	Business Domain	Domain	nominal	The primary domain or field of the company.
	Company Size	Employee size range	ordinal	The size of the company, typically measured by the number of employees.

3.4 Hypothesis Development

To conduct a thorough comparison between companies that discontinue contributing and those that do, we draw on prior literature [7, 33, 90] to determine which dimensions to consider and develop a comparison framework (see Table 1). Specifically, we defined a first dimension called 'Willingness and Capacity' to capture the motivational and capability aspects [7, 90] of corporate contributions. The second dimension is 'Collaborative Environment' which seeks to capture interaction within an open source project's ecosystem [90]. Finally, recognizing the potential impact of a company's specific traits [82] on early participation in an open source ecosystem, we defined a third dimension 'Company Attributes' that captures the characteristics of the company. To answer RQ2, we developed seven hypotheses based on the framework above.

3.4.1 Willingness and Capacity. Willingness refers to the forces that motivate companies to engage and continuously contribute, such as strategic business goals [53]. Capacity refers to the internal resources and technical expertise that make effective participation feasible. In OSS projects, it is the developers' willingness that ultimately converts their capabilities into meaningful contributions [90]. We combined willingness and capacity into one dimension, aiming to comprehend companies' early participation through their strategic interests, and outward contributions. To measure this dimension, drawing on insights from previous research [82], we focused on three specific constructs: Contribution Intensity, Contribution Scope, and Contribution Type. These were selected to collectively measure a company's level of involvement in OSS projects comprehensively.

Contribution Intensity: Given that integrating code changes into an OSS repository can be an exhaustive and time-intensive task for contributors [60], we used the number of commits to reflect the frequency of contributions and the modified lines of code [83] as indicators of the workload of each contribution. We used the number of participating developers as an indicator of the level of resources allocated. Together, these three indicators reflect a company's contribution intensity. Using these measurements, we propose the following hypothesis:

> Hypothesis 1: Compared to companies that continue to contribute, companies that discontinued their contributions exhibit lower contribution intensity.

We measured the early engagement during the first three months using the median of each month's three metrics. The median [80] is less influenced by extreme variations than the mean, and reflects a central tendency, providing a consistent representation of a company's average activity level during that period, ensuring that the overall participation assessment is not skewed by anomalous monthly activities.

Contribution Scope: A company's focus on OSS projects potentially enhances its technical learning and integration within the open source ecosystem. Conversely, companies contributing to fewer modules might have a narrower impact or a more targeted engagement approach, possibly leading to reduced participation. Thus, we propose:

> Hypothesis 2: Companies that discontinued contributions tend to have a more restricted scope of contribution than companies that continued to contribute.

To assess the scope of a company's contribution, we used the median number of modules contributed to the Linux kernel over the first three months of a company's engagement. Given that the Linux kernel does not have officially defined modules, and commit logs do not explicitly state the module to which a commit belongs, we devised an approach based on the file path recorded in the commit log. Drawing from methods observed in the analysis of the Linux kernel architecture [9], where directory paths are crucial for organizing and identifying subsystems, we defined a module as the first directory in the file path. This approach is also similar to the analysis of the open source project's components by examining directory structures [12]. For example, if a commit involved modifications to "arch/arm/mach-s3c6410/mach-smdk6410.c", the corresponding module would be identified as "arch".

Contribution Type: This variable reveals the nature of companies' contributions, showcasing their technical capabilities and support strategies for open source project development. Contributions can be aimed at developing new features, improving existing functionalities, or bug fixing [69], and the type of contributions may indicate a company's strategic engagement. Companies participating in open source projects typically have specific objectives to serve their business, such as integrating OSS components into their products. This frequently involves allocating paid developers to essential tasks that align with the company's strategic goals and product roadmap, indicating a targeted approach to their open source contributions [86]. Companies that eventually cease contributions may exhibit a pattern of short-term or non-strategic engagements, such as bug fixing, rather than ongoing development of novel features. Thus, we propose:

> Hypothesis 3: Companies that discontinued contributions are more likely to engage in corrective or perfective maintenance rather than adaptive maintenance for new feature development.

We determined a company's contribution type during the initial three-month period by identifying the most frequent type of commits they made. To categorize commits, given the absence of direct classification within commit messages, we used a classifier developed by Sarwar et al. [66] to classify commits into distinct types. This classifier relies on commit messages as its input, employing a bi-directional neural network that has been carefully adjusted to identify the subtle linguistic nuances characteristic of software development communication. It then classifies each commit into one of three traditional types [69]: Corrective (fixing bugs and faults), Perfective (improving existing functionalities), or Adaptive (implementing new features), reported with a high level of precision with an F1-score of 87%. We validated the performance of the classification model by randomly selecting 100 commits from each of the three projects (Linux kernel, OpenStack, and Rust) and manually labeling based on commit messages and diffs. The results show that 72%, 74%, and 75% of the commits in the respective projects are given the same labels from both the classification model and the manual validation. The high consistency can demonstrate the availability of the classification model we selected. More details about the manual validation process can be found in the online appendix [3].

3.4.2 Collaborative Environment. This dimension focuses on the extent and nature of a company's interactions within open source projects. It involves two constructs: Degree centrality evaluates the company's central role in the project's social network (ecosystem), indicating its influence and connectivity. Dominance is determined by whether the company has contributed to modules where dominance by one company can be observed. These concepts together provide insight into how actively and influential a company participates in the collaborative ecosystem of OSS projects and how open the collaboration environment is.

Scope of Collaboration: The environment significantly impacts the likelihood of individual participants becoming long-term contributors in OSS projects [90]. For companies involved in open source projects, engagement in intentional and passive collaborations is crucial. A company's position within the collaboration network positively correlates with its productivity in OSS projects [87]. The perceived value of the community, alignment of individual and community values, and the influence of coworker interactions play pivotal roles in driving participation motivation [41, 88]. Therefore, we propose:

> Hypothesis 4: Companies that discontinued contributing exhibit a lower level of collaboration than companies that continued contributing.

We constructed an early-stage collaboration network between companies using the NetworkX library for Python [20]. Each node in this network represents a distinct company, identified by the company's name. Given that the commit log records the file path of every commit modification, an edge is created from one company node to another if the first company has made contributions involving a file path that has also been contributed to by the second company within a specified timeframe (three months following the initial contribution). Through this approach, we employed degree centrality [21, 58, 77] to measure the extent of a company's collaboration, thereby revealing the extent of their engagement and their roles within the collaborative network of the open source ecosystem.

Dominance: In the realm of open source projects, a phenomenon of dominance by certain companies within repositories is observable; for example, within the OpenStack ecosystem, a single company's dominance can extend up to 70% within some repositories [84]. The presence of a dominating contributing company can deter other companies from participating [84, 85]. Thus, we posit:

> Hypothesis 5: Companies that discontinued their contributions are more likely to have been influenced by dominance than companies that continued their contributions.

As companies may contribute to multiple modules in the initial three months, we focused on identifying the module to which each company contributed the most during this period as their representative module.

We determined whether dominance existed for each company's representative module during the initial three-month phase of their involvement in OSS projects. Specifically, we adopted a method proposed by Zhang et al. [84] to determine the existence of dominance. We calculated the Contribution Amount (CA) for the top two companies with the highest commit counts to this module. Dominance is defined based on the disparity in their contribution rates, as follows [84]:

$$\text{Dominance} = 0.5 \times (1 - (CA_1^2 - CA_2^2)),$$

where CA_1 and CA_2 represent the contribution rates of the top two companies, respectively. A module is considered to exhibit dominance if:

$$CA_1^2 > \text{Dominance}.$$

The results of these calculations were used to update the dominance status in the dataset. For each company, the is_influenced attribute was set to true if dominance was observed, and false otherwise. This adjustment allows the dataset to accurately reflect the influence dynamics of company contributions in the early stages of open source projects.

3.4.3 Company Attributes. In exploring the Inherent Attributes dimension, we aimed to uncover the influence of companies' internal characteristics on their initial involvement in open source projects. Within this framework, we focused on two variables: business domain and size. These elements shed light on the core attributes that can guide a company's approach to engaging with OSS communities.

Business Domain: The business domain reflects the primary industry or sector a company operates within. Jullien et al. proposed (but did not test) that the extent to which a company invests in an OSS project depends on the role of that asset in the company's business, and that if it is no longer perceived to be of value, a company will stop investing [39]. Companies tend to contribute more effectively to projects aligned with their core expertise, utilizing their specialized knowledge to foster innovations that complement their business objectives [86]. We argue that companies with business domains closely aligned to the technical nature and goals of the Linux Kernel would exhibit more sustained and significant contributions, reflecting a strong willingness and capacity for engagement. Conversely, companies from less relevant or less technically intensive domains might show a lower propensity for long-term contribution, potentially leading to withdrawal from the project. Thus, we propose:

Hypothesis 6: Companies that discontinued contributing are likely to have a less technical domain than companies that continued contributing.

In our endeavor to accurately determine each company's domain, we initially analyzed the "industries" information from CrunchBase records, revealing 507 unique industry sectors from diverse fields such as "Automotive" to "Food and Beverage." Acknowledging the complexity of companies operating in multiple sectors, we employed ChatGPT-4 [56], known for its broader general knowledge and problem-solving abilities, to perform an initial classification into sixteen broad categories. To further enhance clarity and simplify our analysis, we consolidated these into five primary categories: Software and Digital Innovation, Hardware and Manufacturing, Services, Energy and Environment, and Others. This concise categorization framework allowed us to assign a representative domain to each company, based on the most prevalent category within its industries.

Company Size: An organization's size is a key factor in its ability to sustain contributions to complex OSS projects [82]. Specifically, smaller companies may initiate participation in OSS initiatives but could find their scale limiting in continuing long-term engagements, especially in demanding projects like the Linux kernel. Based on this premise, we proposed:

Hypothesis 7: Companies that discontinued contributing are likely smaller in size than companies that continued contributing.

In defining the Company Size construct, we conducted an analysis based on employee data sourced from CrunchBase, which provides only the range of companies' employee counts. Following previous studies [5, 82], we manually defined five size categories: 1-10, 11-500, 501-5,000, 5,001-10,000, and 10,001+ employees.

To provide a clear overview of our analytical framework and the predictive metrics under investigation, we summarized all dimensions, constructs, and the corresponding nine variables along with their detailed interpretation in Table 1. Data and analysis scripts for this study are available in a replication package [3].

4 Results

4.1 RQ1: Prevalence of Companies Discontinuing Contributions

The first research question explores the prevalence of companies discontinuing their contributions to the Linux kernel project. We answer this question by analyzing the number of companies that discontinue their contribution each year and the importance of these companies. Figure 1 shows the participation trends of two types of companies in the Linux kernel, where: (1) blue bars represent the number of companies that contributed at least one commit in a given year; (2) orange bars represent the number of companies that have discontinued their contributions to the Linux kernel for a given year; (3) the red line shows the yearly ratio of companies that stopped contributing vs. those that contributed. We observe that, on average, 302 companies contributed to the Linux Kernel annually, exhibiting an upward trend from 2005 to 2016, with a peak of 361 companies in 2016. On average, 35 companies discontinued contributing each year, with a peak of 64 companies in 2014. While there is some fluctuation year by year, the numbers indicate a slight upward trend up to 2014, at which point the trend turned slightly downwards. The median discontinued-to-contributed ratio is 12.4%, with a peak in 2014 reaching 18.6%. The ratio increased rapidly from 2005 to 2007, then fluctuated and stabilized post-2014.

Figure 2 presents distributions of companies that discontinued contributions annually by volume of contributions; Following existing classifications [15, 52, 78], we explore two groups: those that rank in the top 20% of all companies by the total number of commits historically, up to the end of the given year, and those in the lower 80%. Companies in the top 20% account for approximately

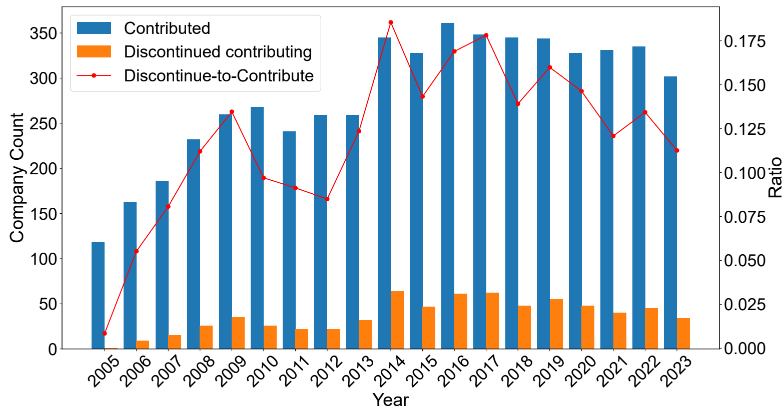

Fig. 1. Turnover and retention of companies in the Linux kernel ecosystem (left y-axis) and annual ratio of companies discontinued contributing: contributed (right y-axis)

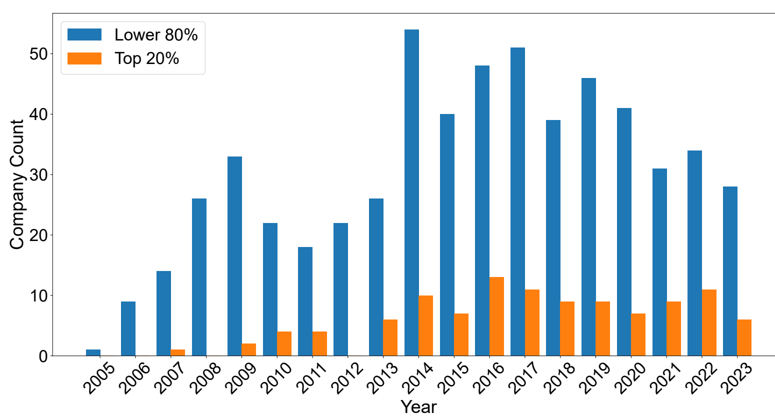

Fig. 2. Companies that discontinued contributing: Top 20% vs. Lower 80%

16% of the withdrawn companies each year. The median number of such companies annually is six, peaking before 2016 and then stabilizing at an average of around ten. Companies outside the top 20% are indicated by those peripheral in discontinued companies. A median of 31 such companies stopped contributing each year, with the most considerable fluctuation observed in, again, 2014, when as many as 54 companies withdrew. Subsequently, these numbers stabilized to a lower count.

Summary for RQ1: Annually, on average 12% of companies involved in the Linux kernel discontinue their contributions; 16% of these withdrawn companies were used to be the top 20% contributors ranked by their total commits. The results highlight a significant portion of major contributing companies choosing to discontinue their involvement each year.

4.2 RQ2: Characteristics of Companies that Discontinue Contributing

To address RQ2, we categorized companies into two distinct groups, based on whether they had discontinued contributing. Considering the variety of variables involved, which include both

Table 2. Summary of statistical analysis results.

Dimension	Hypothesis / construct	Variable	Statistical Test	Adjusted p-value
Willingness and Capacity	H1 Contribution intensity	#Commits	Mann-Whitney U	$7.09e^{-28}$
		#LoCs	Mann-Whitney U	$1.26e^{-27}$
		#Developers	Mann-Whitney U	$3.92e^{-31}$
	H2 Contribution scope	#Modules	Mann-Whitney U	$4.59e^{-13}$
	H3 Contribution type	Type (adaptive, corrective, perfective)	Chi-square	$6.56e^{-03}$
Collaborative Environment	H4 Scope of collaboration	Degree centrality	Mann-Whitney U	$3.64e^{-5}$
	H5 Dominance	Is_influenced	Mann-Whitney U	0.3481
Company Attributes	H6 Business Domain	Domain	Chi-square	0.9146
	H7 Company size	Employee size range	Chi-square	$4.44e^{-2}$

numerical (discrete and continuous) and categorical data (see Table 1), we used different statistical non-parametric tests suitable for non-normal distributions [24]. Specifically, we used the Mann-Whitney U test [54] for numerical variables and the Chi-square test [1] for categorical variables. To reduce false discoveries because of multiple hypothesis testing, we adjusted all p-values using the Benjamin-Hochberg correction method [4]. Table 2 presents the results.

4.2.1 Willingness and Capacity. We analyzed the two groups of companies using the Mann-Whitney U test [54] for Hypothesis 1 (Contribution Intensity, measured using three variables) and Hypothesis 2 (Contribution Scope). This analysis revealed statistically significant differences across all four indicators (all p-values well below .05). We compared the median ranks [40] of the two groups in the Mann-Whitney U test: The discontinued companies have median ranks of 373 for all the three measures, i.e., 'Number of developers', 'Lines of code', and 'Number of commits', whereas continuing companies show higher median ranks of 913.5, 856.5, and 1,013 for the same variables. This underscores that the median values of contribution intensity metrics for continuing contributing companies surpass those of companies that discontinued contributions. The 'Number of modules' variable shows the same median ranks at 478.5 for both groups. We explored the mean ranks to gain additional insight, finding that continuing discontinued companies' mean rank is 605.00 and continuing contributing companies' is 725.48. These results lend support to Hypotheses 1 and 2; in comparison to companies that continued to contribute, companies that discontinued demonstrate a lower contribution intensity and a more limited scope of contributions in the early stages.

To test Hypothesis 3 (Contribution Type), we used the Chi-square test [1], and the results also showed significant differences between the two groups of companies (p-value < 0.05). We examined the differences in contribution types between the two groups: among the companies that continued to contribute, 48.7% of them contributed most to perfective tasks, 26.5% of them preferred adaptive tasks, and 24.8% of them made more corrective contributions. Companies that discontinued contributing showed a more distinct skew towards perfective contributions at 58.2%, with these companies making adaptive and corrective contributions at 20.6% and 21.1%, respectively. This analysis reveals a preference for perfective tasks among *all* companies, and it also lends support to Hypothesis 3, namely that companies that discontinued contributions appear to focus even more on perfective tasks (58.2% *vs.* 48.7%).

4.2.2 Collaborative Environment. The second dimension captures two constructs: Scope of Collaboration and Dominance. We used the Mann-Whitney U test for testing degree centrality and is_influenced, respectively. The results highlighted significant differences (p-value=$3.64e^{-5}$) in network centrality in the early stage between the two kinds of companies, with discontinued companies having a median rank of 544.5, compared to a higher 761.5 for those that continued contributions. This result lends support to Hypothesis 4, indicating that companies that discontinued contributing exhibited a smaller scope of collaboration.

In contrast, the results don't support Hypothesis 5. Contributing to a module that is dominated (in terms of contributions) by a company shows no significant variance, suggesting similar initial behavior in engaging with modules that are dominated regardless of whether companies continue or not.

4.2.3 Company Attributes. We explored the characteristics of companies, focusing on their business domain and size range using the Chi-square test. For the business domain, we observed no significant difference (p-value ≈ 0.91), thus rejecting Hypothesis 6. It indicates that the primary business domain of a company does not appear to influence whether a company continues or ceases its contributions significantly. The Chi-square test result for size highlighted a significant difference (p-value ≈ 0.04), underscoring that the size of a company, as determined by its (approximate) employee count, plays a role in determining its continued participation or withdrawal from the project. To substantiate Hypothesis 7, we analyzed the distribution of companies across different size categories. We found that companies that remained active are more frequently of larger size (501-5,000 and 10,001+ employees) compared to companies that discontinued, which are more prevalent in the smallest size category (1-10 employees). Hence, the analysis lends support to Hypothesis 7, indicating that companies that have discontinued their contributions tend to be smaller in size.

Summary for RQ2: Compared to companies that remained active, companies that discontinued contributing have a lower intensity and smaller scope of contributions with a preference for perfective contributions in the early stage of contributing. They tend to engage less in collaborative efforts within other companies and tend to be smaller-scale companies. There is no significant difference between the two groups of companies in terms of Dominance and the business domains in which they operate.

4.3 RQ3: Prediction Models of Companies Stopping Contributing

We build prediction models based on nine widely-used classification techniques, including traditional machine learning models such as Logistic Regression [50] and Random Forest [10] and advanced deep learning models like Multilayer Perceptrons (MLPs) [59] and Long Short-Term Memory Networks (LSTMs) [34]. To balance between interpretability and performance, we chose not to use larger-scaled deep-learning models, which we leave as a future research avenue. We conducted a comprehensive evaluation of these models using a ten-fold cross-validation approach [42],[1] which is widely recognized for its effectiveness in providing robust estimates of model performance. We randomly divided the dataset, integrating the significant variables identified in RQ2, into ten similarly sized subsets. Each iteration of the validation process trained the model with nine subsets and tested it with one, ensuring each subset was tested once. This process was repeated ten times, and we reported the average performance across all rounds, offering a thorough evaluation of the predictive accuracy of each model.

We leverage two widely-used metrics, i.e., Area Under the ROC Curve (AUC) and Accuracy, as the main metrics for their ability to evaluate model performance across all classification thresholds,

[1] Note that data leakage does not exist because each record contains one company's features and records classified into different folds do not have temporal relationships.

enhancing the reliability of the predictive assessment. AUC effectively reflects the likelihood that a model will rank a randomly chosen positive instance (meaning a company discontinuing contributions to an OSS project in this study) higher than a negative one (i.e., a company continuing contributions), even in datasets with imbalanced distributions. Accuracy measures the proportion of correctly classified instances among all predictions.

To explore the generalizability of the significant variables we identified in the Linux kernel, we validated our models in two other OSS projects, i.e., OpenStack and Rust. The details of collecting and processing the datasets are in Sec. 3.2. Similar to our approach with the Linux kernel dataset, we employed a ten-fold cross-validation technique, and the results are averaged to provide a comprehensive assessment of model performance. Table 3 presents the performance of these models.

Table 3. Model Performance on the Linux Kernel, OpenStack, and Rust Projects

	The Linux kernel		OpenStack		Rust	
Model	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy
TCN	**0.76**	**0.71**	**0.79**	**0.76**	**0.84**	**0.79**
MLP	0.74	0.69	0.77	0.73	0.76	0.70
LSTM	0.72	0.68	0.75	0.70	0.71	0.65
Bi-LSTM	0.72	0.68	0.77	0.73	0.76	0.66
GRU	0.72	0.68	0.77	0.71	0.76	0.66
Logistic Regression	0.68	0.65	0.70	0.63	0.72	0.66
Random Forest	0.65	0.63	0.69	0.69	0.57	0.57
Gradient Boosting	0.69	0.66	0.68	0.69	0.60	0.63
CatBoost	0.66	0.64	0.64	0.66	0.56	0.55

The Temporal Convolutional Network (TCN) achieved the best performance when compared with other models in all three OSS projects with an average AUC of 0.80 and an average accuracy of 0.75. TCN is a deep learning architecture designed for sequential data analysis, featuring flexible receptive fields and causal convolutional design, and is particularly well-suited for capturing temporal dependencies accurately [43], making it highly effective for both classification and prediction tasks. These characteristics likely contributed to the TCN model's good performance in this study, indicating its potential for broader application in predictive modeling scenarios.

The promising performance of these prediction models may be because companies that contribute just once are easily identified correctly. To verify this conjecture, we excluded companies that contributed only once within their first three months and re-ran these models with the remaining data. The results show that the TCN model continues to perform the best and remains stable, achieving an AUC of 0.78 and an ACC of 0.76. This suggests that the inclusion or exclusion of one-time contributors has minimal impact on the overall performance of our model. More details can be found in the online appendix [3].

Table 4 presents the precision, recall, and F1 scores of the TCN model in the Linux kernel project. The *Positive Class* refers to companies that discontinue contributions, while the *Negative Class* represents those that still make contributions to the Linux kernel. The precision of 0.74 and recall of 0.82, point to promising aspects of the model's performance. More work is to be done, such as considering additional factors for improvement.

Table 4. Performance Metrics of the TCN Model on the Linux kernel Dataset

Metric	Positive Class	Negative Class
Precision	0.74	0.69
Recall	0.59	0.82
F1 Score	0.65	0.75

Summary for RQ3: Based on the identified features that significantly differentiate companies that continue to contribute from those that discontinue, our TCN-based model achieves an AUC of up to 0.76 in the Linux kernel ecosystem and maintains stable (or even improved) performance in the other two projects.

5 Discussion

5.1 Implication for Practice

5.1.1 Strategically Improving Contributor Retention. The findings from RQ1 reveal that annually, about 12% of companies disengaged from the Linux kernel, of which approximately 16% represent companies that rank in the top 20% by total number of commits. These results provide a first assessment of company turnover in the Linux kernel project. Given an increasing corporate involvement in OSS projects, empirical evidence on such trends is important but has been hitherto lacking. We acknowledge that not all companies pursue sustained participation in projects they engage with. The initiative of whether to retain companies identified as likely to withdraw lies in the hands of the open source community. If determined to do so, our results can help the OSS communities identify such companies. Moreover, data on early corporate behaviors that may lead to turnover can offer valuable insights to develop retention strategies. Specifically, OSS communities could establish early warning systems tracking key metrics like commit frequency decline (e.g., >20% over 3 months), or reduced module ownership. These metrics can trigger targeted interventions, such as technical support through maintainer pairing or strategic roadmap alignment workshops, fostering a stable, dynamic, and committed corporate contributor base for OSS ecosystem sustainability. This can contribute to maintaining a stable, dynamic, and committed corporate contributor base and the sustainability of OSS ecosystems.

5.1.2 Creating Spaces for Diverse Companies. RQ2 findings indicate that companies that stopped contributing display a lower intensity and a narrower scope of initial contributions, compared to those that continued contributing. This highlights a need for OSS communities to cultivate more inclusive and collaborative environments that accommodate companies of varying sizes and types of contributions, particularly those smaller enterprises with limited resources but a willingness and interest to contribute. Specifically, OSS communities can support them by lowering participation barriers, such as identifying areas where they can contribute and recommending specific tasks. For example, an AI-assistant matching system based on historical data could recommend tasks to companies by considering their technical expertise (e.g., hardware) and business interests (e.g., market expansion). Besides, the OSS communities can highlight the voices of these small companies to help them get better involvement and construct more collaborations. By designing more flexible contribution processes, policies, and participation models, communities can better cater to these companies, encouraging a diverse range of contributions including code submissions, documentation, design, and marketing efforts. This approach not only aims to attract and retain a broader and more diverse base of corporate contributors but also ensures that valuable insights and innovations are fully embraced, thereby fostering long-term prosperity and growth of the community.

5.1.3 Proactive Corporate Engagement Management. The adoption of predictive models provides a transformative method for the OSS community to manage corporate contributions more effectively. Drawing on the findings of RQ3 that demonstrated the performance of a TCN-based prediction model that successfully forecasts whether a company will cease contributing, we recognize that community leaders can use these models to precisely identify companies at higher risk of disengagement. This insight empowers them to initiate forward-looking strategies to encourage sustained participation.

5.1.4 Early Engagement Predicts Longevity. Our research establishes that characteristics of early engagement can be good indicators of a company's long-term commitment to OSS projects. For example, companies that stop contributing typically exhibit distinctive characteristics such as a lower intensity of initial participation. Companies are encouraged to leverage the early stage of involvement as an opportunity to establish a foundation for long-term engagement, actively contributing to the project.

5.2 Implications for Research

5.2.1 Explore More Factors Influencing Company Turnover in OSS. Our finding in RQ2, i.e., companies that discontinued contributing have a lower contribution intensity, aligns with Zhang et al.'s conclusion [82] that contribution intensity is significantly associated with increased survival rates. Additionally, our conclusion that smaller companies are more likely to stop contributing is contrary to Zhang et al.'s finding [82] that company scale is negatively associated with company withdrawal. Furthermore, building on previous work, we have identified three new significant dimensions: companies that discontinued contributing have a smaller scope of contributions, show a preference for perfective contributions in the early stage of contributing, and tend to engage less in collaborative efforts with other companies. These indicators of early contributing behavior identified in this study can predict companies' departure from an OSS ecosystem, and the performance of the TCN model remains promising. There are likely other factors at play. Further study should explore more early indicators of corporate engagement. For instance, future work could expand the analysis by including a broader range of contributions beyond commits, such as code review, funding, or community involvement. Investigating how these different types of contributions correlate with company retention or withdrawal could offer deeper insights into the dynamics of company turnover within OSS ecosystems. Additionally, future research could explore which factor contributes most to predicting long-term involvement, potentially leading to refined predictive models. As supplementary validation, we provide the logistic regression coefficients of all the factors we explored in RQ2 in the appendix [3], where degree centrality has the greatest impact while the company size has the least contribution.

5.2.2 Investigte the Impact of Company Withdrawal. Our study mainly focuses on predicting company withdrawal based on prior empirical evidence that contributor attrition undermines OSS sustainability [25, 36]. Future research could explore its impact on the overall open source ecosystem in depth. Specifically, it could examine how the absence of key company contributors affects project direction, the ability to attract new contributors, and the community's resilience in maintaining project momentum. Further investigation into how company withdrawal influences project development could deepen our understanding of the adaptive strategies OSS projects employ to thrive amid changing participation dynamics.

5.2.3 Variability in OSS Ecosystems. The findings of this study indicate that the behavior of participating in dominant modules has no significant correlation with a company's long-term contributions to the Linux kernel. Further, we discovered that most Linux kernel modules do not exhibit dominance-related behaviors, which contrasts sharply with findings of studies of OpenStack [84], where dominance phenomena are common and negatively associated with project survival. These differences underscore the unique characteristics of OSS ecosystems, but at the same time suggest that OSS ecosystems vary in terms of how companies collaborate and contribute to the central assets.

6 Threats to Validity

The design of this study involved a number of decisions, or trade-offs [62], which we now discuss as follows.

External Validity. In this study, we focus on the Linux kernel project, which means that the findings cannot be generalized to other projects. Together with the over 1,300 companies that contribute to the project, this forms an ecosystem in which different actors (i.e. contributors of any kind) operate. We selected a single project as this provides a shared context for all companies that are involved, which means that we can exclude any context-specific factors as confounding factors in addressing the three research questions. For example, the OpenStack ecosystem is different than the Linux kernel. We selected the Linux kernel because it is among the best-known and most important OSS projects that enjoy extensive corporate involvement. While our findings do not generalize to other OSS ecosystems without further inquiry, the positioning and design of this study offer a useful starting point to develop a better understanding of corporate turnover in OSS projects. Our focus is on the project's internal dynamics through analyzing commit data. We do not extensively cover how external factors such as market trends and regulations influence corporate strategies in OSS. However, considering commit data as a direct reflection of the company's valuable record in participating in OSS, we can develop an understanding of the company's contributions to a major OSS project.

Internal Validity. We identified developers' affiliations by matching their email domains to company names. Although this method was widely used in prior studies [83, 86, 92], a known potential shortcoming is that developers can use personal emails or change jobs, which would make it hard to determine in what capacity and role they make contributions. Zhang et al. [83, 86] have conducted validations of the identified affiliations by contacting developers. Their evaluation results suggested a high accuracy (up to 93%). We acknowledge that there are many methods (such as advanced deep learning models) and data sources (e.g., LinkedIn) that can improve the accuracy of developers' affiliation identification. This study strives to strike a balance between the ease of use of prediction models and the accuracy of company identification, and we deem the current email domain-based method sufficiently accurate for our study.

The second internal threat lies in how we locate a company's collaboration scope. To identify collaborations between companies, we relied on the files that different companies modified. While such trace data affords analysis of overlapping efforts and shared interests, it does not capture any informal collaborations in which no trace data exists. This remains a limitation.

The third internal threat relates to our method of determining company withdrawal based on different contribution breaks. Prior studies [86, 92] have shown that companies' contribution performance can vary because of different business strategies. Thus, we adopted a flexible approach, i.e., identifying withdrawal companies based on their historical contribution intervals (i.e., the time between two adjacent code contributions). Compared to using fixed contribution breaks, our approach allows for a more nuanced assessment of company turnover. However, we acknowledge that this approach may misclassify companies that previously made frequent contributions but have recently slowed down the pace. These companies could be unfairly classified as withdrawn. To improve the validity, future work could investigate company withdrawal based on multiple activities.

Another threat lies in commit frequency measurement. OSS projects may require developers to submit clean, well-structured patches with clear commit messages. This often involves squashing multiple commits into a single, more concise commit before submitting it for review. Thus, squashing commits can result in a lower contribution frequency. However, since all companies contributing to the same OSS project are subject to the same community guidelines, we believe this has a limited impact on our findings. Future work that is sensitive to contribution frequency should consider squashing commits.

The final limitation of our study is the limited interpretability of the TCN model. The main focus of RQ3 is designing comprehensive prediction models with promising performance. Although the ten-fold cross-validation strengthens the validity of our model, OSS communities or companies may still not know what to do to avoid this departure risk. The significant characteristics of companies' initial participation identified in RQ2 can be used as a coarse reference. Future research could explore ways to enhance the interpretability of such models (e.g., SHAP values or attention mechanisms), including identifying key factors that contribute to withdrawal. We have also included the coefficient results of the logistic regression model in the appendix [3], where degree centrality has the greatest impact, while company size has the least contribution.

7 Conclusion

Large OSS projects like the Linux kernel have attracted corporate contributions, forming complex ecosystems with hundreds or even thousands of actors. As the importance of OSS to business and society grows, questions arise around the sustainability of these projects. Companies continuously join and leave these OSS ecosystems, but very little is known about companies that discontinue contributing. This study aims to close this gap by (1) establishing how prevalent this is; (2) how companies that discontinue contributing differ from companies that remain; and (3) whether we can predict which companies will discontinue contributing.

By examining early participation behaviors, we aim to identify predictors of potential company withdrawal. We observed that companies that discontinued contributions show distinct patterns in their early stage, such as lower initial contribution levels and less involvement in collaborative activities, emphasizing a preference for perfective over adaptive or corrective tasks, compared with companies that continued. We demonstrate that a TCN-based model to forecast a company's future contributions has promising performance, which underscores its potential as a strategic tool for project maintainers, enabling proactive measures to foster a stable and sustainable OSS ecosystem. This study not only sheds light on the factors influencing company turnover in OSS projects but also offers practical solutions to mitigate such risks. It underscores the importance of early detection and intervention in maintaining the vitality of OSS communities, contributing to the broader goal of ensuring their continued growth and success.

Data Availability

To facilitate replications or future work, we publicly open our data, analysis scripts, and other resources used in this study in [3].

Acknowledgments

This work is supported by the National Natural Science Foundation of China Grants (62202048, 62232003, and 62141209), Taighde Éireann – Research Ireland under Grant number 13/RC/2094_P2, and CCF-Huawei Populus Grove Fund.

References

[1] Alan Agresti. 2012. *Categorical data analysis*. Vol. 792. John Wiley & Sons.
[2] Sadika Amreen, Audris Mockus, Russell Zaretzki, Christopher Bogart, and Yuxia Zhang. 2020. ALFAA: Active Learning Fingerprint based Anti-Aliasing for correcting developer identity errors in version control systems. *Empirical Software Engineering* (03 Jan 2020). https://doi.org/10.1007/s10664-019-09786-7
[3] Anonymous. 2024. FSE 2025 Scripts Data. https://figshare.com/s/b1985c3da484910df9c5
[4] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.
[5] Gentrit Berisha and Justina Shiroka Pula. 2015. Defining Small and Medium Enterprises: a critical review. *Academic Journal of Business, Administration, Law and Social Sciences* 1, 1 (2015), 17–28.
[6] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. 2006. Mining email social networks. In *Proceedings of the 2006 international workshop on Mining software repositories*. ACM, 137–143. https://doi.org/10.1145/1137983.1138016
[7] Melvin Blumberg and Charles D. Pringle. 1982. The Missing Opportunity in Organizational Research: Some Implications for a Theory of Work Performance. *The Academy of Management Review* 7, 4 (1982), pp. 560–569.
[8] Andrea Bonaccorsi and Cristina Rossi. 2006. Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology & Policy* 18, 4 (2006), 40–64. https://doi.org/10.1007/s12130-006-1003-9
[9] Ivan T Bowman, Richard C Holt, and Neil V Brewster. 1999. Linux as a case study: Its extracted software architecture. In *Proceedings of the 21st International Conference on Software Engineering*. 555–563.
[10] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.
[11] Peter G Capek, Steven P Frank, Steve Gerdt, and David Shields. 2005. A history of IBM's open-source involvement and strategy. *IBM Systems Journal* 44, 2 (2005), 249–257.
[12] Andrea Capiluppi, Klaas-Jan Stol, and Cornelia Boldyreff. 2011. Software Reuse in Open Source: A Case Study. *International Journal of Open Source Software and Processes* 3, 3 (2011), 10–35. https://doi.org/10.4018/978-1-4666-4301-7.ch090
[13] Eleni Constantinou and Tom Mens. 2017. Socio-technical evolution of the Ruby ecosystem in GitHub. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 34–44. https://doi.org/10.1109/SANER.2017.7884607
[14] Kevin Crowston, Kangning Wei, James Howison, and Andrea Wiggins. 2012. Free/Libre Open-Source Software Development: What We Know and What We Do Not Know. *Comput. Surveys* 44, 2 (2012). https://doi.org/10.1145/2089125.2089127
[15] Kevin Crowston, Kangning Wei, Qing Li, and James Howison. 2006. Core and periphery in free/libre and open source software team communications. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, Vol. 6. IEEE, 118a–118a. https://doi.org/10.1109/HICSS.2006.101
[16] Crunchbase documentation. 2024. Using the Crunchbase API. https://data.crunchbase.com/docs/using-the-api.
[17] Carlo Daffara. 2007. Business models in FLOSS-based companies. In *Workshop presentatioon at the 3rd Conference on Open Source Systems*.
[18] Linus Dahlander and Mats Magnusson. 2008. How do firms make use of open source communities? *Long Range Planning* 41, 6 (2008), 629–649. https://doi.org/10.1016/j.lrp.2008.09.003
[19] Jean-Michel Dalle, Matthijs Den Besten, and Carlo Menon. 2017. Using Crunchbase for economic and managerial research. (2017).
[20] NetworkX Developers. 2023. NetworkX. https://networkx.org/, last accessed 20 Aug 2024.
[21] Reinhard Diestel. 2005. Graph theory. 2005. *Grad. Texts in Math* 101 (2005).
[22] Anh Nguyen Duc, Daniela S Cruzes, Claudia Ayala, and Reidar Conradi. 2011. Impact of stakeholder type and collaboration on issue resolution time in OSS projects. In *IFIP International Conference on Open Source Systems*. Springer, 1–16. https://doi.org/10.1007/978-3-642-24418-6_1
[23] Joseph Feller, Patrick Finnegan, Brian Fitzgerald, and Jeremy Hayes. 2008. From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information Systems Research* 19, 4 (2008), 475–493.
[24] Andy Field. 2013. *Discovering statistics using IBM SPSS statistics*. sage.
[25] Matthieu Foucault, Marc Palyart, Xavier Blanc, Gail C Murphy, and Jean-Rémy Falleri. 2015. Impact of developer turnover on quality in open-source software. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. 829–841. https://doi.org/10.1145/2786805.2786870
[26] Fred Bals. 2024. 2024 Open Source Security and Risk Analysis Report. https://www.synopsys.com/blogs/software-security/open-source-trends-ossra-report.html/.
[27] GitHub. 2021. GitHub REST API. https://docs.github.com/en/rest.

[28] Jesus M Gonzalez-Barahona and Gregorio Robles. 2013. Trends in free, libre, open source software communities: From volunteers to companies. *it–Information Technology it–Information Technology* 55, 5 (2013), 173–180. https://doi.org/10.1515/itit.2013.1012

[29] Mariam Guizani, Aileen Abril Castro-Guzman, Anita Sarma, and Igor Steinmacher. 2023. Rules of Engagement: Why and How Companies Participate in OSS. (2023), 2617–2629. https://doi.org/10.1109/ICSE48619.2023.00218

[30] Dietmar Harhoff, Joachim Henkel, and Eric Von Hippel. 2003. Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research Policy* 32, 10 (2003), 1753–1769. https://doi.org/10.1016/S0048-7333(03)00061-1

[31] Joachim Henkel. 2003. Software development in embedded Linux—Informal collaboration of competing firms. In *Wirtschaftsinformatik 2003/Band II*. Springer, 81–99. https://doi.org/10.1007/978-3-642-57445-0_5

[32] Joachim Henkel. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy* 35, 7 (2006), 953–969. https://doi.org/10.1016/j.respol.2006.04.010

[33] Guido Hertel, Sven Niedner, and Stefanie Herrmann. 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy* 32, 7 (2003), 1159–1177.

[34] S Hochreiter. 1997. Long Short-term Memory. *Neural Computation MIT-Press* (1997).

[35] Dirk Homscheid and Mario Schaarschmidt. 2016. Between organization and community: investigating turnover intention factors of firm-sponsored open source software developers. In *Proceedings of the 8th ACM Conference on Web Science*. 336–337. https://doi.org/10.1145/2908131.2908200

[36] Daniel Izquierdo-Cortazar, Gregorio Robles, Felipe Ortega, and Jesus M Gonzalez-Barahona. 2009. Using software archaeology to measure knowledge loss in software projects due to developer turnover. In *2009 42nd Hawaii International Conference on System Sciences*. IEEE, 1–10. https://doi.org/10.1109/HICSS.2009.498

[37] Jack Jamieson, Naomi Yamashita, and Eureka Foong. 2024. Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*. 1–13. https://doi.org/10.1145/3597503.3623340

[38] Brian T Jones. 2023. free email provider domains.txt. https://gist.github.com/tbrianjones/5992856/, last accessed 20 Aug 2023.

[39] Nicolas Jullien, Klaas-Jan Stol, and James Herbsleb. 2019. A Preliminary Theory for Open Source Ecosystem Micro-economics. In *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability*. Springer, 49–68. https://doi.org/10.1007/978-981-13-7099-1_3

[40] Andreas Keller, Hanyi Zhuang, Qiuyi Chi, Leslie B Vosshall, and Hiroaki Matsunami. 2007. Genetic variation in a human odorant receptor alters odour perception. *Nature* 449, 7161 (2007), 468–472.

[41] Joon Koh, Young-Gul Kim, Brian Butler, and Gee-Woo Bock. 2007. Encouraging participation in virtual communities. *Communication of ACM* 50, 2 (Feb. 2007), 68–73.

[42] R Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Morgan Kaufman Publishing* (1995).

[43] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 156–165.

[44] Bin Lin, Gregorio Robles, and Alexander Serebrenik. 2017. Developer turnover in global, industrial open source projects: Insights from applying survival analysis. In *IEEE 12th International Conference on Global Software Engineering*. 66–75. https://doi.org/10.1109/ICGSE.2017.11

[45] Johan Linåker, Patrick Rempel, Björn Regnell, and Patrick Mäder. 2016. How firms adapt and interact in open source ecosystems: analyzing stakeholder influence and collaboration patterns. In *International Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 63–81. https://doi.org/10.1007/978-3-319-30282-9_5

[46] Dana H Lindsley, Daniel J Brass, and James B Thomas. 1995. Efficacy-performing spirals: A multilevel perspective. *Academy of management review* 20, 3 (1995), 645–678. https://doi.org/10.5465/amr.1995.9508080333

[47] Linux Foundation. 2023. The Linux Kernel Organization Trends. https://insights-v2.lfx.linuxfoundation.org/korg/trends.

[48] Linux Foundation Insights. 2024. Organizations Report. https://insights.lfx.linuxfoundation.org/foundation/korg/reports/organizations?project=korg.

[49] Linux Kernel Organization. 2022. About Linux Kernel. https://www.kernel.org/linux.html.

[50] Scott W Menard. 2010. *Logistic regression: From introductory to advanced concepts and applications*. Sage.

[51] Courtney Miller, David Gray Widder, Christian Kästner, and Bogdan Vasilescu. 2019. Why do people give up flossing? a study of contributor disengagement in open source. In *Open Source Systems: 15th IFIP WG 2.13 International Conference, OSS 2019, Montreal, QC, Canada, May 26–27, 2019, Proceedings 15*. Springer, 116–129. https://doi.org/10.1007/978-3-030-20883-7_11

[52] Audris Mockus, Roy T Fielding, and James D Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *Acm Transactions on Software Engineering Methodology* 11, 3 (2002), 309–346. https://doi.org/10.1145/567793.567795
[53] Eunyoung Moon and James Howison. 2014. Modularity and organizational dynamics in open source software (OSS) production. (2014).
[54] Nadim Nachar et al. 2008. The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution. *Tutorials in quantitative Methods for Psychology* 4, 1 (2008), 13–20.
[55] Olivia B Newton and Stephen M Fiore. 2023. Leveraging Corporate Engagement for Diversity in Free/Libre and Open Source Software Projects. In *2023 IEEE/ACM 4th Workshop on Gender Equity, Diversity, and Inclusion in Software Engineering (GEICSE)*. IEEE, 41–48. https://doi.org/10.1109/GEICSE59319.2023.00010
[56] OpenAI. 2023. GPT-4: OpenAI's Generative Pre-trained Transformer 4. https://openai.com/gpt-4. Accessed: 2024-02-27.
[57] OpenStack. 2024. OpenStack GitHub Repository. https://github.com/openstack. Accessed: 2024-07-28.
[58] Tore Opsahl, Filip Agneessens, and John Skvoretz. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks* 32, 3 (2010), 245–251. https://doi.org/10.1016/j.socnet.2010.03.006
[59] Sankar K Pal and Sushmita Mitra. 1992. Multilayer perceptron, fuzzy sets, classifiaction. (1992).
[60] Peter C. Rigby, Daniel M. German, and Margaret-Anne Storey. 2008. Open Source Software Peer Review Practices: A Case Study of the Apache Server. In *30th International Conference on Software Engineering* (Leipzig, Germany) *(ICSE '08)*. ACM, New York, NY, USA, 541–550. https://doi.org/10.1145/1368088.1368162
[61] Peter C Rigby, Yue Cai Zhu, Samuel M Donadelli, and Audris Mockus. 2016. Quantifying and mitigating turnover-induced knowledge loss: case studies of Chrome and a project at Avaya. In *Proceedings of the 38th International Conference on Software Engineering*. 1006–1016. https://doi.org/10.1145/2884781.2884851
[62] Martin P Robillard, Deeksha M Arya, Neil A Ernst, Jin LC Guo, Maxime Lamothe, Mathieu Nassif, Nicole Novielli, Alexander Serebrenik, Igor Steinmacher, and Klaas-Jan Stol. 2024. Communicating Study Design Trade-offs in Software Engineering. *ACM Transactions on Software Engineering and Methodology* (2024). https://doi.org/10.1145/3649598
[63] Cristina Rossi and Andrea Bonaccorsi. 2006. *4 – Intrinsic Motivations and Profit-Oriented Firms in Open Source Software : Do Firms Practise What They Preach?* 83–109 pages.
[64] David A. Rusling. 1999. The Linux Kernel. https://tldp.org/LDP/tlk/tlk.html.
[65] Ioannis Samoladas, Ioannis Stamelos, Lefteris Angelis, and Apostolos Oikonomou. 2004. Open source software development should strive for even greater code maintainability. *Commun. ACM* 47, 10 (2004), 83–87. https://doi.org/10.1145/1022594.1022598
[66] Muhammad Usman Sarwar, Sarim Zafar, Mohamed Wiem Mkaouer, Gursimran Singh Walia, and Muhammad Zubair Malik. 2020. Multi-label classification of commit messages using transfer learning. In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 37–42. https://doi.org/10.1109/ISSREW51248.2020.00034
[67] Mario Schaarschmidt, Klaas-Jan Stol, and Brian Fitzgerald. 2025. The Insider's Dilemma: Employed Open Source Developers' Identification Imbalance and Intentions to Leave. *European Journal of Information Systems* (2025), 1–20. https://doi.org/10.1080/0960085X.2025.2463984
[68] Andreas Schilling, Sven Laumer, and Tim Weitzel. 2012. Who will remain? an evaluation of actual person-job and person-team fit to predict developer retention in FLOSS projects. In *2012 45th Hawaii International Conference on System Sciences*. IEEE, 3446–3455. https://doi.org/10.1109/HICSS.2012.644
[69] E. B. Swanson. 1976. The dimensions of maintenance. In *2nd International Conference on Software Engineering*. San Francisco, California, USA, 492–497.
[70] Xin Tan, Minghui Zhou, and Zeyu Sun. 2020. A first look at good first issues on GitHub. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 398–409. https://doi.org/10.1145/3368089.3409746
[71] Rust Team. 2022. Rust, A language empowering everyone to build reliable and efficient software. https://www.rust-lang.org/.
[72] The Linux Foundation. 2017. The Linux Foundation Releases Annual Kernel Development Report. https://www.linuxfoundation.org/press/press-release/linux-foundation-releases-annual-kernel-development-report.
[73] The Linux Foundation. 2022. Participating in Open Source Communities. https://www.linuxfoundation.org/tools/participating-in-open-source-communities/.
[74] The Linux Foundation. 2024. LFX Insights - Linux Kernel Overview. https://insights.lfx.linuxfoundation.org/foundation/korg/overview/github?project=korg&projectId=41401e90-6ded-46e9-a5b5-081c24df80d0&routedFrom=Github&bestPractice=false. Accessed: 2024-08-27.
[75] Marat Valiev, Bogdan Vasilescu, and James Herbsleb. 2018. Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 644–655. https://doi.org/10.1145/3236024.3236062

[76] Fei-Rong Wang, Dan He, and Jin Chen. 2005. Motivations of individuals and firms participating in open source community. In *2005 International Conference on Machine Learning and Cybernetics*, Vol. 1. IEEE, 309–314.
[77] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440–442.
[78] Wenxin Xiao, Hao He, Weiwei Xu, Yuxia Zhang, and Minghui Zhou. 2023. How early participation determines long-term sustained activity in github projects?. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 29–41. https://doi.org/10.1145/3611643.3616349
[79] Yiqing Yu, Alexander Benlian, and Thomas Hess. 2012. An Empirical Study of Volunteer Members' Perceived Turnover in Open Source Software Projects. In *Hawaii International Conference on System Sciences*. https://doi.org/10.1109/HICSS.2012.97
[80] Zach. 2023. Advantages & Disadvantages of Using Median in Statistics. https://www.statology.org/advantages-disadvantages-of-median-in-statistics/. Accessed: 2024-03-08.
[81] Kamil Żbikowski and Piotr Antosiuk. 2021. A machine learning, bias-free approach for predicting business success using Crunchbase data. *Information Processing & Management* 58, 4 (2021), 102555. https://doi.org/10.1016/j.ipm.2021.102555
[82] Yuxia Zhang, Hui Liu, Xin Tan, Minghui Zhou, Zhi Jin, and Jiaxin Zhu. 2022. Turnover of Companies in OpenStack: Prevalence and Rationale. *ACM Transactions on Software Engineering and Methodology* 31, 4 (2022).
[83] Yuxia Zhang, Mian Qin, Klaas-Jan Stol, Minghui Zhou, and Hui Liu. 2024. How Are Paid and Volunteer Open Source Developers Different? A Study of the Rust Project. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–13. https://doi.org/10.1145/3597503.3639197
[84] Yuxia Zhang, Klaas-Jan Stol, Hui Liu, and Minghui Zhou. 2022. Corporate dominance in open source ecosystems: a case study of OpenStack. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1048–1060. https://doi.org/10.1145/3540250.3549117
[85] Yuxia Zhang, Xin Tan, Minghui Zhou, and Zhi Jin. 2018. Companies' domination in FLOSS development: an empirical study of OpenStack. In *40th International Conference on Software Engineering*. ACM, 440–441. https://doi.org/10.1145/3183440.3195047
[86] Yuxia Zhang, Minghui Zhou, Audris Mockus, and Zhi Jin. 2021. Companies' Participation in OSS Development–An Empirical Study of OpenStack. *IEEE Transactions on Software Engineering* 47, 10 (2021), 2242–2259. https://doi.org/10.1109/TSE.2019.2946156
[87] Yuxia Zhang, Minghui Zhou, Klaas-Jan Stol, Jianyu Wu, and Zhi Jin. 2020. How Do Companies Collaborate in Open Source Ecosystems? An Empirical Study of OpenStack. In *International Conference on Software Engineering* (Seoul, South Korea). Association for Computing Machinery, New York, NY, USA, 1196–1208.
[88] Minghui Zhou and Audris Mockus. 2011. Does the initial environment impact the future of developers?. In *ICSE 2011*. Honolulu, Hawaii, 271–280.
[89] Minghui Zhou and Audris Mockus. 2012. What make long term contributors: Willingness and opportunity in OSS community. In *Software Engineering (ICSE), 2012 34th International Conference on*. IEEE, 518–528.
[90] Minghui Zhou and Audris Mockus. 2015. Who Will Stay in the FLOSS Community? Modeling Participant's Initial Behavior. *IEEE Transactions on Software Engineering* 41, 1 (Jan 2015), 82–99.
[91] Minghui Zhou and A Mockus. 2015. Who Will Stay in the FLOSS Community? Modeling Participant's Initial Behavior. *Software Engineering IEEE Transactions on* 41, 1 (2015), 82–99. https://doi.org/10.1109/TSE.2014.2349496
[92] Minghui Zhou, Audris Mockus, Xiujuan Ma, Lu Zhang, and Hong Mei. 2016. Inflow and retention in OSS communities with commercial involvement: A case study of three hybrid projects. *ACM Trans Softw Eng Methodol* 25, 2 (2016), 13. https://doi.org/10.1145/2876443
[93] Jiaxin Zhu and Jun Wei. 2019. An empirical study of multiple names and email addresses in oss version control repositories. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 409–420. https://doi.org/10.1109/MSR.2019.00068

Received 2025-02-26; accepted 2025-04-01