From Models to Practice: Enhancing OSS Project Sustainability with Evidence-Based Advice

Nafiz Imtiaz Khan
University of California at Davis
Davis, USA
nikhan@ucdavis.edu

Vladimir Filkov
University of California at Davis
Davis, USA
vfilkov@ucdavis.edu

ABSTRACT

Sustainability in Open Source Software (OSS) projects is crucial for long-term innovation, community support, and the enduring success of open-source solutions. Although multitude of studies have provided effective models for OSS sustainability, their practical implications have been lacking because most identified features are not amenable to direct tuning by developers (e.g., levels of communication, number of commits per project).

In this paper, we report on preliminary work toward making models more actionable based on evidence-based findings from prior research. Given a set of identified features of interest to OSS project sustainability, we performed a comprehensive literature review related to those features to uncover practical, evidence-based advice, which we call Researched Actionables (ReACTs). The ReACTs are practical advice with specific steps, found in prior work to associate with tangible results. Starting from a set of sustainability-related features, this study contributes 105 ReACTs to the SE community by analyzing 186 published articles. Moreover, this study introduces a newly developed tool (ReACTive) designed to enhance the exploration of ReACTs through visualization across various facets of the OSS ecosystem. The ReACTs idea opens new avenues for connecting SE metrics to actionable research in SE in general.

CCS CONCEPTS

• **Software and its engineering** → **Open source model**; *Software design techniques*; **Maintaining software**.

KEYWORDS

Research Actionable, Open Source Software, Sustainability, Literature Review

ACM Reference Format:
Nafiz Imtiaz Khan and Vladimir Filkov. 2024. From Models to Practice: Enhancing OSS Project Sustainability with Evidence-Based Advice. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering (FSE Companion '24), July 15–19, 2024, Porto de Galinhas, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3663529.3663777

FSE Companion '24, July 15–19, 2024, Porto de Galinhas, Brazil
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0658-5/24/07
https://doi.org/10.1145/3663529.3663777

1 INTRODUCTION

Open Source Software (OSS) comprises a thriving billion-dollar industry and serves as a technological cornerstone in the tech era [27] [19]. The collaborative nature of OSS not only fosters innovation and transparency but is also a driving force behind technological advancements and economic efficiency [26]. According to a recent report, 78% of businesses utilize open-source software, and a striking 96% of applications incorporate at least one open-source component [2]. Also, prior research has estimated that the adoption of OSS can save one trillion dollars annually for the Information and Communication Technology (ICT) industry [30].

In the dynamic world of open-source projects, ensuring those projects are sustainable can be a major challenge due to several factors, such as varying levels of contributor commitment, evolving technological trends, and the need for continuous adaptation to maintain relevance in the rapidly changing technology landscape. In the context of OSS projects, sustainability refers to the ability of a project to maintain its longevity, relevance, and effectiveness over time [34]. A sustainable OSS project can offer long-term benefits to OSS communities, users, and contributors.

Research on understanding OSS project sustainability is on the rise [20, 31–33, 35]. While informative, the identified models features by these studies are not directly tunable/ changeable by the developer community. For instance: the article by Yin et al. [35] found that the number of active developers positively associates with the sustainability outcome in the forecasting model. It is not directly possible for the developer community to increase the number of active developers in an OSS project, because participation in such projects relies on individual choices and voluntary contributions, making it challenging to enforce or control the engagement of developers. Therefore, developers need a translation of the theoretical findings into practical advice (e.g., conduct unit tests, make governance explicit, outreach to potential groups of interest, etc.) that can be adopted by the developer community to influence the features of the forecasting model in the right direction, which in turn will increase the sustainability chances of an OSS project. For example, it has been found that ensuring a team of friendly and supportive people is associated with an increase in the number of active project developers [14, 17, 25]. Thus, this is an example of a researched actionable, or ReACT, in the context of our study, which can positively impact the feature *number of active developers* in an OSS project.

Many prior studies have focused on providing suggestions/advice to influence specific factors contributing to the sustainability of open-source software (OSS), such as increasing engagement of developers, onboarding newcomers, enhancing the contribution of core developers, etc. For example: the study by Balali et al. [15]

found a total of 44 barriers that are commonly faced by the mentors (n=19) and the newcomers (n=34) of the OSS projects. The authors then came up with 10 strategies to deal with the mentioned barriers. In another article [22], the authors analyzed the developer turnover in open source projects and provided three different suggestions (Encourage developers to: (i) start contributing to the project early, (ii) modify existing files instead of creating, (iii) code instead of dealing with documentations) to alleviate the developer turnover rate. In another similar study, Ferreira et al. [18] presented four suggestions (Maintain a small number of core/active developers; Keep the project small and simple; Shorten the project timeline; Shift project ownership to users instead of organizations) to increase the retention rate of core developers.

However, to date, there is no specific literature dedicated to identifying actionable advice for influencing specific features of a sustainability forecasting model. This study aims to address this research gap by answering the following two research questions:

RQ1 - How can Researched Actionables (ReACTs) be identified for each of the features of the forecasting model?
RQ2 - How can these ReACTs be rendered useful and applicable in practice?

This idea paper describes our approach on one specific model, the sustainability model proposed by Yin et al. [35]. However, the methodology followed can be generalized and can be applied to any particular set of features. Future work can focus on automating the costly procedure of manually reviewing hundreds of related papers by e.g. using AI bots and assistants.

2 STUDY METHODOLOGY

2.1 Initial Set of Model Features

In this study, our initial set is the model features obtained in the paper of Yin et al. [35]. The authors used 18 socio-technical features to develop a 3-layer LSTM model there. The features, along with their respective descriptions, are outlined at [12]. [1].

2.2 Article Selection Strategy

To identify ReACTs from prior literature, we formulated a search strategy to systematically explore relevant articles. The primary sources of articles for analysis were IEEE Explorer [5], ACM digital library [1], ScienceDirect [10], Springer Link [11], and Google scholar [4]. To retrieve articles, a set of specific search strings has been employed. These strings are tailored to locate articles based on *direct* and *indirect links*, as discussed in subsection 2.3.

(1) Open Source Project Sustainability
(2) Open Source Projects AND (Emails OR communication OR social network)
(3) Open Source Projects AND Commits
(4) Open Source Projects AND Modularity
(5) Open Source Projects AND core contributors
(6) Open Source Projects AND newcomers
(7) Open Source Projects AND pair programming
(8) Open Source Projects AND Non-hierarchical communications
(9) Open Source Projects AND (developers OR contributors)

To ensure the analysis focused on relevant articles for identifying ReACTs, specific eligibility criteria were applied. The initial search process retrieved a total of 426 articles, with 412 sourced from major scholarly databases and 12 from the grey literature (dissertations, theses, reports). From this pool of 426 articles, a refined subset of 186 articles was selected using the following inclusion-exclusion criteria. For inclusion: (a) it must be a full-text research article, and (b) it was published between 2000 and 2023. We excluded: (a) studies with only abstracts, (b) duplicate articles (if the original was already in the article list), and (c) articles written in languages other than English. If an article unquestionably met one or more of these criteria, it was excluded from further review. The final set of articles was then meticulously analyzed to extract ReACTs.

2.3 ReACTs

We derived Researched Actionables (ReACTs) from prior literature as actions that can positively impact each of the features of the forecasting model. Each of the derived ReACTs is responsible for impacting at least one feature of the forecasting model.

To derive ReACTs, we adopted a two-pronged strategy. First, for a specific feature *X*, we identified a collection of relevant literature on OSS and addressing feature *X*. Then, we extracted the practical advice from these articles, identifying them as ReACTs responsible for positively impacting Feature *X*. We termed this strategy- article identification through *direct links*. For instance, when deriving ReACTs related to *num_act_dev* (Number of active developers), we delved into prior literature focusing on OSS and the onboarding of developers/newcomers. The practical advice extracted from these papers serves as ReACTs for influencing *num_act_dev*. This process returned articles for only three features from the feature list *(num_act_dev, num_emails, num_commits)*.

When direct links were not available (e.g., for the 15 remaining features above), we added a level of indirection, i.e., intermediate level of features. We call this strategy- article identification through *indirect links*. For example, no prior literature explicitly addressed OSS and *top_c_fract* (The percentages of commits performed by the top 10% contributors). Therefore, for this particular feature, we established Core Developers' contribution as the intermediate layer feature. Subsequently, we compiled a list of articles that focused on OSS and highlighted the contributions of core developers. ReACTs were then derived from these articles. In this context, the assumption is that ensuring significant contributions by core developers will positively impact the feature *top_c_fract*.

The features and their corresponding intermediate layers are presented in Table 1. In the table, each intermediate layer, involving contribution, is followed by one of two terms: (i) Technical or (ii) Social. In this context, "Technical" entails contributing to the project through technical aspects like coding, debugging, and system design. On the other hand, "Social" involves making contributions in areas such as community engagement, collaboration, and communication. Hence, if an intermediate layer involving technical

[1]The article employed a LIME [29] model to obtain the magnitude and sign (positive or negative) of each of the features, signifying their impact in the LSTM model. For this study, the impact of these features was not considered.

Table 1: Features and the intermediate layers

Feature Name	Intermediate Layer
num_act_dev	No intermediate layer has been used
num_emails	
num_commits	
num_files	Code refactoring, Modularity, Modular Design
c_interruption	Discouraging contribution (Technical)
e_interruption	Discouraging contribution (Social)
top_e_fract	Core developers contribution (Social)
top_c_fract	Core developers contribution (Technical)
c_nodes	Newcomers contribution (Technical)
c_edges	Newcomers contributions (Social), Pair Programmers contribution (Social), Contribution (Social) from Collaboration
c_c_coef	Pair Programmers contribution (Technical), Contribution (Technical) from Collaboration
c_mean_degree	Pair Programmers contribution (Technical), Contribution (Technical) from Collaboration
c_long_tail	Pair Programmers contribution (Technical), Contribution (Technical) from Collaboration
e_nodes	Newcomers contributions (Social), Core developers contribution (Social), Contribution (Social) from Collaboration
e_edges	Newcomers contributions (Social), Pair Programmers contribution (Social), Contribution (Social) from Collaboration
e_c_coef	Pair Programmers contribution (Social), Contribution (Social) from Collaboration
e_mean_degree	Newcomers contributions (Social), Pair Programmers contribution (Social), Contribution (Social) from Collaboration
e_long_tail	Newcomers contributions (Social), Pair Programmers contribution (Social), Contribution (Social) from Collaboration

contributions is specified for Feature X and an intermediate layer involving social contributions is defined for Feature Y, then the ReACTs derived from the technical intermediate layer will exert an influence on Feature X, and those from the social intermediate layer will impact Feature Y. In cases where the derived ReACT involves both technical and social impacts, it will influence both Features X and Y. For example, ReACTs discouraging technical contributions of developers in an OSS project will positively impact *c_interruption*. Conversely, ReACTs discouraging social contributions will positively impact *e_interruption*. However, if the ReACT discourages both technical and social contributions, it will affect both features.

2.3.1 *ReACTs Synthesis and Interpretation.*

Upon thorough analysis of the final set of selected articles, 28 articles were identified from which ReACTs could be derived.

The derivation process relies on two criteria:

Criterion 1: The article should address specific research problem(s) and offer actionable recommendation(s) to address the identified problem(s). For instance, the research conducted by Balali et al. [15] focused on identifying barriers encountered by newcomers and their mentors in an OSS project. Through a semi-structured interview involving 10 experienced developers, the authors identified 44 barriers faced by newcomers and their mentors. Additionally, the study presented 10 actionable recommendations to mitigate these barriers. These actionable recommendations are then reformulated as ReACTs, with the potential to impact the feature *num_act_dev*.

Criterion 2: The article should address a specific research problem(s) and elucidate the underlying reasons, causally upstream of the identified problem(s). These underlying reasons are then reformulated as ReACTs. For example, Lin et al. [22] analyzed the developer turnover in OSS projects by doing a survival analysis. The authors found that developers have higher chances of survival in software projects when they: 1) initiate contributions to the project early; 2) predominantly modify files rather than create them; and 3) predominantly engage in coding rather than dealing with documentation. These discoveries from the article are then reformulated to create three distinct ReACTs (ReACT-62, ReACT-63, and ReACT-64), which are as follows:

1. ReACT-62: Encourage developers to start contributing to the project early
2. ReACT-63: Encourage developers to modify existing files instead of creating
3. ReACT-64: Encourage developers to code instead of dealing with documentation

The derivation process provided 105 ReACTs, which were further analyzed.

2.3.2 *ReACTs & Entities.*

The entities refer to individuals responsible for performing specific ReACTs. Based on prior studies [16, 24], 20 distinct roles have been defined, each with a specific set of responsibilities. Next, two independent researchers separately assigned entities for each ReACT. The inter-rater agreement score (Cohen's Kappa [23]) between the raters was 0.89, and the disagreements were resolved through discussion. The description of the entities is available at [13].

3 REACTIVE

ReACTive is an interactive tool that visualizes the connection between ReACTs and the features, and the associativity of ReACTs with entities. The tool is primarily designed for developers, with secondary utility extending to all individuals engaged in the OSS Community.

ReACTive displays an interactive network graph to visualize ReACTs and their impact on each of the features. A snapshot of the graph is shown in Figure 1. This dynamic network graph adopts a directed acyclic graph (DAG) structure, featuring circular nodes (color: light blue) representing ReACTs, rectangular nodes (color: light red) delineating features, and edges symbolizing articles.

Edges connect circular nodes (ReACTs) to rectangular nodes (features). The edges are weighted, representing the count of articles observing the effect.

The tool's front-end is crafted with a blend of HTML, CSS, and JavaScript. Python's PyVis library [28] has been used to generate the interactive network graphs, while python's Plotly [21] has been harnessed to generate the other interactive plots (bar-chart, pie-chart, scattered plot). The plots offer interactivity by displaying tooltips when users hover over or click on any component of the

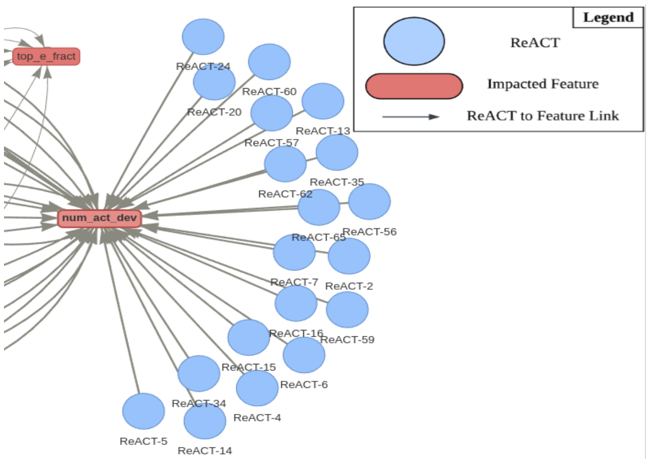

Figure 1: A snapshot of the network graph showing the relationship between ReACTs and features

graph. A tooltip is a small pop-up box that appears when a user hovers over an element on a web page. It provides additional information or a brief description related to the element. To preserve the interactivity of the plots, each plot was individually transformed into an HTML file. Subsequently, these files were embedded into the front-end using HTML's inline element, the *iframe*. This element facilitates the creation of a nested browsing context, allowing the display of a distinct HTML document within the main document. ReACTive can be publicly accessed at [7].

3.1 Use Cases

3.1.1 Use Case 1: Finding ReACTS to Tune Features. ReACTive can be used to find the ReACTs, responsible for tuning a particular feature. Hovering over a node provides a detailed description of the node, as tooltips [6]. Hovering over an edge reveals relevant information, including article count, references, and links. This feature enables users to directly access articles from the tool that observed the relationship between a ReACT and a feature.

Additionally, clicking on a specific feature node in the graph highlights all associated ReACT nodes, offering insights into the ReACTs to consider when tuning a particular feature. This interactive functionality enhances user experience and facilitates a more in-depth exploration of the relationships within the data.

3.1.2 Use Case 2: Finding Roles and Responsibilities. ReACTive can be used to understand the roles of the personnel who are responsible for performing each of the ReACTs. An interactive scattered plot has been utilized to show the relationship between ReACTs and the entities [9]. The graph displays entities along the X-axis, with ReACTs represented on the Y-axis. Interconnections between a ReACT and an entity are visually depicted as square boxes. Clicking on these boxes reveals tooltips, providing insights into the specific connection between a ReACT and an entity.

4 DISCUSSION AND CONCLUSION

Principal Outcomes. This article presents two significant contributions. Firstly, we introduce 105 Researched Actionables (ReACTs) derived from an extensive review of papers spanning the past two decades. These ReACTs offer practical, actionable suggestions that, when implemented, yield tangible outcomes. By adopting these ReACTs, OSS projects can tailor the features of their sustainability model, enhancing their potential for long-term viability.

Secondly, we introduce a tool called ReACTive, designed to visualize the impact of ReACTs on individual features, as well as the associations between ReACTs and entities. This tool can serve as a valuable resource for the OSS community, providing a deeper understanding of the interplay between ReACTs and project features.

Limitations and Future work. The study exhibits certain limitations. Firstly, we relied on features from a specific study [35] as baseline, potentially overlooking essential features utilized by other papers contributing to sustainability forecasting models [20, 31–33]. Future research could explore a combined set of features from multiple studies to enhance comprehensiveness. Secondly, causal effects of ReACTs on other ReACTs were not examined. The adoption of a particular ReACT may not only impact its associated feature(s) but also influence other ReACTs, a facet not considered in this study. Subsequent research should delve into these internal effects. Thirdly, project-specific factors like complexity, size, and technology stack were not factored into ReACT extraction. A ReACT effective for a smaller project might not be suitable for a larger one. Thus, future studies should incorporate project-specific considerations when extracting ReACTs from existing literature. Fourthly, specific search strings are used to retrieve papers from scholarly databases which may not retrieve all the articles. Future studies may focus on more generalized search strings to retrieve all the relevant papers in the context of software engineering and sustainability.

Conclusion. As per the GitHub Octoverse Report, OSS repository creation witnessed substantial growth in recent years [3]. With a rapid exponential increase in the number of projects, ensuring the sustainability of these endeavors has emerged as a major concern. While prior research has introduced OSS sustainability models, their practical utility for developers has been limited due to the lack of direct tunability or changeability of model features. This study pioneers a solution by addressing feature tunability, presenting 105 ReACTs directly adoptable by developers to fine-tune sustainability model features. Additionally, the contribution of ReACTive, a visualization tool, enhances comprehension of ReACT impacts across various dimensions. The collective outcome of this study can empower OSS projects to maintain sustainability.

5 DATA-AVAILABILITY STATEMENT

The study's data and code can be accessed publicly at [8].

ACKNOWLEDGMENTS

The research received funding from the National Science Foundation under Grant No. 2020751. Additionally, Nafiz was supported by the UC Davis GGCS Fellowship throughout the project. The authors would like to thank Dr. Toufique Ahmed for participating in the ReACT to Entity annotation process.

REFERENCES

[1] 2024. *ACM Digital Library*. https://dl.acm.org/.
[2] 2024. *Open Source Software Statistics*. Retrieved January 10, 2024 from https://gitnux.org/open-source-software-statistics/.
[3] 2024. *The state of open source software*. Retrieved January 10, 2024 from https://octoverse.github.com/.
[4] 2024. *Google Scholar*. https://scholar.google.com/.
[5] 2024. *IEEE Xplore*. https://ieeexplore.ieee.org/.
[6] 2024. *Network Graph: ReACTs & Features*. https://nafiz43.github.io/ReACTive/feature.html.
[7] 2024. *ReACTive App*. https://nafiz43.github.io/ReACTive.
[8] 2024. *ReACTive: Visualizing ReACTs and Their Impact on Features*. https://doi.org/10.5281/zenodo.11180765.
[9] 2024. *Scattered Plot: ReACTs & Entities*. https://nafiz43.github.io/ReACTive/entity.html.
[10] 2024. *ScienceDirect*. https://www.sciencedirect.com/.
[11] 2024. *Springer Link*. https://link.springer.com/.
[12] 2024. *Supplementary File 1: Description of Features*. https://nafiz43.github.io/ReACTive/index.html.
[13] 2024. *Supplementary File 2: Description of Entities*. https://nafiz43.github.io/ReACTive/entities_desc.html.
[14] Guilherme Avelino, Eleni Constantinou, Marco Tulio Valente, and Alexander Serebrenik. 2019. On the abandonment and survival of open source projects: An empirical investigation. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–12.
[15] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurelio Gerosa. 2018. Newcomers' barriers... is that all? an analysis of mentors' and newcomers' barriers in OSS projects. *Computer Supported Cooperative Work (CSCW)* 27 (2018), 679–714.
[16] Flore Barcellini, Françoise Détienne, and Jean-Marie Burkhardt. 2014. A situated approach of roles and participation in Open Source Software Communities. *Human–Computer Interaction* 29, 3 (2014), 205–255.
[17] Jailton Coelho, Marco Tulio Valente, Luciana L Silva, and André Hora. 2018. Why we engage in FLOSS: Answers from core developers. In *proceedings of the 11th international workshop on cooperative and human aspects of software engineering*. 114–121.
[18] Fabio Ferreira, Luciana Lourdes Silva, and Marco Tulio Valente. 2020. Turnover in Open-Source Projects: The Case of Core Developers. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 447–456.
[19] Alfonso Fuggetta. 2003. Open source software—-an evaluation. *Journal of Systems and software* 66, 1 (2003), 77–90.
[20] Amir Hossein Ghapanchi. 2015. Predicting software future sustainability: A longitudinal perspective. *Information Systems* 49 (2015), 40–51.
[21] Plotly Technologies Inc. 2015. *Collaborative data science*. Montreal, QC. https://plot.ly
[22] Bin Lin, Gregorio Robles, and Alexander Serebrenik. 2017. Developer turnover in global, industrial open source projects: Insights from applying survival analysis. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE, 66–75.
[23] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282.
[24] Reed Milewicz, Gustavo Pinto, and Paige Rodeghero. 2019. Characterizing the roles of contributors in open-source scientific software projects. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 421–432.
[25] 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.
[26] Del Nagy, Areej M Yassin, and Anol Bhattacherjee. 2010. Organizational adoption of open source software: barriers and remedies. *Commun. ACM* 53, 3 (2010), 148–151.
[27] Bruce Perens. 2005. The emerging economic paradigm of open source. *First Monday* (2005).
[28] Giancarlo Perrone, Jose Unpingco, and Haw-minn Lu. 2020. Network visualizations with Pyvis and VisJS. *arXiv preprint arXiv:2006.04951* (2020).
[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.
[30] Michael Tiemann and President Open Source Initiative. 2009. How open source software can save the ICT industry one trillion dollars per year. *White paper.[Online] http://www. opensource. org/files/OSS-2010. pdf* (2009).
[31] 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.
[32] Jing Wang. 2012. Survival factors for Free Open Source Software projects: A multi-stage perspective. *European Management Journal* 30, 4 (2012), 352–371.
[33] 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.
[34] Likang Yin, Mahasweta Chakraborti, Yibo Yan, Charles Schweik, Seth Frey, and Vladimir Filkov. 2022. Open source software sustainability: Combining institutional analysis and socio-technical networks. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–23.
[35] Likang Yin, Zhuangzhi Chen, Qi Xuan, and Vladimir Filkov. 2021. Sustainability forecasting for apache incubator projects. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1056–1067.

Received 29-JAN-2024; accepted 2024-04-09