计算机学报
Chinese Journal of Computers
ISSN 0254-4164,CN 11-1826/TP

《计算机学报》网络首发论文

题目：OpenPerf：面向开源生态可持续发展的数据科学基准测试体系
作者：韩凡宇，毕枫林，张琰彬，李晖，游兰，王伟，钱卫宁，周傲英
网络首发日期：2024-11-01
引用格式：韩凡宇，毕枫林，张琰彬，李晖，游兰，王伟，钱卫宁，周傲英．OpenPerf：面向开源生态可持续发展的数据科学基准测试体系[J/OL]．计算机学报．
https://link.cnki.net/urlid/11.1826.tp.20241101.1453.011

计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS

OpenPerf：面向开源生态可持续发展的数据科学基准测试体系

韩凡宇[1)] 毕枫林[1)] 张琰彬[1)] 李晖[2)] 游兰[3)] 王伟[1)] 钱卫宁[1)] 周傲英[1)]

[1)]（华东师范大学数据科学与工程学院 上海 200062）

[2)]（贵州大学计算机科学与技术学院 贵阳 550025）

[3)]（湖北大学计算机与信息工程学院 武汉 430062）

摘 要 基准测试是指通过设计科学的测试方法、测试工具和测试系统，实现对一类测试对象的某项性能指标进行定量的和可对比的测试。随着人工智能时代的到来，诸如 ImageNet、DataPerf 等这类新型的 AI 基准测试数据集逐步成为学术界和工业界的共识性标准。当前，关于开源生态的研究大多基于某一项具体的研究点展开分析而缺少对开源生态基准体系的构建，一个开源项目处于怎样的发展位置、企业开源程序办公能力处于行业什么位置、开发者活跃度、项目影响力等基础数据与评价，都是数据使用方迫切需要的开源领域知识。为了解决开源领域“有数据无基准”的局面，本文提出一种面向开源生态可持续发展的数据科学基准测试体系（OpenPerf）。该体系自下而上主要包含数据科学任务类基准、指数类基准以及标杆类基准，旨在为学术界，工业界提供不同的基准参考。本文定义了 9 个数据科学任务类基准，给出了 3 项典型的数据科学任务类基准测试结果、2 项指数类基准以及 1 项标杆类基准，其中 2 项指数类基准被中国电子技术标准化研究院作为开源社区治理的评估标准。数据科学任务类基准主要应用于学术界，为不同研究方向的研究者提供自己擅长研究领域的基准。指数类基准主要面向企业界，企业界可以通过影响力和活跃度等基准数据了解当前企业开源程序办公能力所处的行业位置以及旗下开源项目所处的发展位置。标杆类基准是一种可测量的业界最佳水平的成绩，用来比较参考尺度。最后，通过 3 个应用在阿里、蚂蚁以及华东师范大学等国内知名公司和高校的实际案例验证了 OpenPerf 在推动开源生态可持续发展中所起到的关键作用。

关键词 基准测试；开源生态；可持续发展；基准任务；应用案例

中图法分类号 TP391

OpenPerf: A Data Science Benchmark System for Open Source Ecosystem Sustainable Development

HAN Fan-Yu[1)], BI Feng-Lin[1)], ZHANG Yan-Bin[1)], LI Hui[2)], YOU Lan[3)],

WANG Wei[1)], QIAN Wei-Ning[1)], ZHOU Ao-Ying[1)]

[1)]（*School of Data Science and Engineering, East China Normal University, Shanghai,* 200062）

[2)]（*College of Computer Science and Technology, Guizhou University, Guizhou* 550025）

本课题得到国家自然科学基金项目(No.62137001)、上海市教委数字化转型创新研究项目(No.40400-22201)资助. 韩凡宇，博士研究生，主要研究领域为自然语言处理、开源生态. E-mail: hancollins@sina.com. 毕枫林，博士研究生，主要研究领域为机器学习、数据挖掘. 张琰彬，硕士，工程师，硕士生导师，主要研究领域为软件研发效能提升和智能软件测试. 李晖，博士，教授，博士生导师，主要研究领域为大规模数据管理与智能分析处理、人工智能技术及应用. 游兰，博士，教授，硕士生导师，主要研究领域为时空大数据、自然语言处理. 王伟（通信作者），博士，教授，博士生导师，主要研究领域为计算教育学、开源生态系统. 钱卫宁，博士，教授，博士生导师，主要研究领域为 Web 数据管理、社交媒体分析. 周傲英，博士，教授，博士生导师，主要研究领域为 Web 数据管理、数据密集型计算、大数据基准测试和性能优化。

[3)]（*School of Computer Science and Information Engineering, Hubei University, Wuhan,* 430062）

Abstract Benchmarking refers to the quantitative and comparable evaluation of specific performance metrics for a category of test subjects, achieved through scientifically designed test methods, tools, and systems. With the advent of the artificial intelligence era, new AI benchmarking datasets, such as ImageNet and DataPerf, have gradually become consensus standards in both academia and industry. Currently, research on the open-source ecosystem largely focuses on specific research points, lacking a comprehensive framework for open-source ecosystem benchmarks. Data consumers in the open-source domain urgently need foundational metrics and evaluations—such as a project's development stage, an enterprise's open-source program capabilities within the industry, developer activity, and project influence. To address the "data-rich but benchmark-poor" situation in the open-source field, this paper proposes a data science benchmark system for the sustainable development of the open-source ecosystem, termed OpenPerf. This system adopts a bottom-up approach and primarily includes data science task-based, index-based, and benchmark-based categories, aiming to provide diverse benchmark references for academia and industry. This paper define nine task-based data science benchmarks, including open-source behavior data completion and prediction, automated open-source bot identification and classification, sentiment classification of open-source comment texts, risk prediction in open-source software supply chains, open-source project influence ranking, prediction of archived projects, open-source network influence index prediction, anomaly detection in open-source communities, and open-source project recommendation based on link prediction. We present results for three representative task-based benchmarks (open-source behavior data completion and prediction, automated open-source bot identification and classification, and open-source project recommendation based on link prediction), two index-based benchmarks (influence and activity), and one benchmark-based reference. Notably, two of the index-based benchmarks have been adopted by the China Electronics Standardization Institute as evaluation standards for open-source community governance. The task-based benchmarks are primarily aimed at academia, providing researchers in different fields with a relevant reference framework for their areas of expertise. The index-based benchmarks are designed for industry use, enabling enterprises to gain detailed insights through various index-based metrics; unlike task-based benchmarks, index benchmarks serve as standardized units that measure specific attributes of the test subjects. The benchmark-based reference represents an industry best-practice performance level, setting a measurable standard of excellence for specific fields. Through metrics such as influence and activity, enterprises can understand the current industry positioning of their open-source programs and the development stage of their open-source projects. The experimental results demonstrate that, compared to other existing metrics, OpenPerf more effectively evaluates the individual influence of open-source developers, thereby fostering open-source incentives. OpenPerf's benchmark service can analyze the development status of multiple open-source projects within a group from a macro perspective, quantifying the activity and influence of different projects and providing valuable insights for the sustained, healthy growth of these projects. Moreover, in open-source software courses, integrating OpenPerf's benchmark service with traditional grading methods allows for a more comprehensive and fair assessment of students' contributions. The critical role of OpenPerf in promoting the sustainable development of the open-source ecosystem is further illustrated through practical cases from prominent domestic companies and institutions, including Alibaba, Ant Group, and East China Normal University.

Key words benchmark test; open source ecosystem; sustainable development; benchmark tasks; use cases

1 引 言

基准测试[1]在计算机领域非常普遍，从世界 Top 500 的超级计算机，到当下流行的人工智能领域，都能见到基准测试的身影。随着人工智能时代的到来，诸如 ImageNet[2]、DataPerf[3]这类新型的 AI 基准测试数据集、测试任务、性能榜单等开始出现，并逐步成为学术界、研究实验室和工业界的共识性标准，能够为人工智能领域中的各类创新与工程实践，包括系统、算法、模型、任务等带来一个全球化的科学客观评价，这为计算机与人工智能领域的发展提供了巨大的支

撑[4]。

开源生态是指由开源软件、开发者以及相关社区组成的一个生态系统[5-6]。该生态系统的核心是开源软件，它们的源代码是公开的，开发者们可以查看、修改和分发。当前，开源软件已经形成了一种全球化创新发展的生态系统，其中大众参与的软件开发模式正逐步形成一种基于互联网的新型软件生产力，并已在软件开发和应用的各个环节发挥了巨大作用，促进了技术的快速迭代和发展，推动技术的普及和应用[7-9]。一个具有可持续发展的开源生态能够吸引、培养和维护社区参与者，同时保持项目的稳定性、可靠性和活跃性[10]。为了使开源生态可持续健康的发展，较多来自学术界与工业界的研究者们采用数据驱动的研究范式对开源软件生态展开广泛研究，现有研究工作包括企业在开源生态中的合作关系[11]，GitHub 软件生态系统演化过程研究[12-13]，开发者贡献研究[14-15]，开发者对开源项目的评论文本分析[16-19]等等。

当前关于开源生态的研究已经取得了部分成效，但由此所带来的一个重要问题就是缺乏相关的基准、标注与评价规范，造成了一个“有数据无基准”的局面。一个开源项目处于怎样的发展位置、企业开源程序办公能力处于行业什么位置、开发者活跃度、项目影响力等基础数据与评价，都是数据使用方迫切需要的开源领域知识。而这些都需要多方共同开展研究与实践来形成一套与指标、数据相匹配的基准。针对上述问题，课题组成员通过获取 GitHub 全域协作行为数据，并基于该数据构建出全域协作行为网络，联合阿里、蚂蚁等国内知名企业以及木兰开源社区、开放原子基金会共同探索当下开源生态迫切需要解决的问题，课题组通过企业和社区的反馈情况结合现有的数据资源，基于现有的基准测试框架，提出一种面向开源生态可持续发展的数据科学基准测试体系（OpenPerf）[1]，推动开源生态的持续发展。

本文的主要贡献如下：

1）基于现有的基准测试框架[20]，提出一种面向开源生态可持续发展的数据科学基准测试体系。该体系自下而上主要包含数据科学任务类基准、指数类基准以及标杆类基准，旨在为学术界和工业界提供不同的基准参考。

2）在 OpenPerf 基准测试体系的引导下，定义了 9 个数据科学任务类基准，给出了 3 项典型的数据科学任务类基准（开源行为数据补全与预测、开源自动化机器人识别与分类、基于链路预测的开源项目推荐）测试结果，2 项指数类基准（影响力和活跃度）以及 1 项标杆类基准。其中 2 项指数类基准被中国电子技术标准化研究院作为开源社区治理的评估标准。企业界可以通过影响力和活跃度等基准数据了解当前企业开源程序办公能力所处的行业位置以及旗下开源项目所处的发展位置。

3）通过 3 个应用在阿里、蚂蚁以及华东师范大学等国内知名公司和高校的实际案例证明了 OpenPerf 在推动开源生态健康发展中所起到的关键作用。

2 相关工作

基准测试工作在较多研究领域内已经展开，其中包括 AI 基准测试数据集、测试任务、性能榜单等等。Zhan 等人[20]将基准测试上升到一个“基准科学与工程”的高度，并尝试提出系统的研究体系与方法论。近期，该团队提出了一个通用的评价框架，包括统一的评价概念、术语、原理和方法，并对评价、测量和测试的关系进行了辨析，尝试将其应用于计算机、医学、管理学等不同学科[21]。Deng 等人以 WordNet[22]为主体提出 ImageNet 基准数据集[2]，该数据在规模和多样性方面相较于其他图像数据集具备一定的优势。Mazumderd[3]等人提出了一套机器学习数据集和以数据为中心的算法基准套件（DataPerf），DataPerf 可以用于评估训练和测试数据的质量，涵盖了在视觉、语音、采集、调试和扩散等一系列常见的机器学习任务。Hu 等人[23]提出了一套开放图基准数据集（Open Graph Benchmark, OGB），他们通过特定应用程序的数据分割和评估度量来定义统一的评估协议，同时可以支持从社会信息网络到生物网络、分子图和知识图等各种领域的图机器学习任务。Zhou 等人[24]针对图的链接预测问题提出一个新的基准数据集 TeleGraph，该数据集是一个高度稀疏和分层的信息网络，具有丰富的节点属性，可以用于评估和促进链接预测技术。上述方法主要围绕图计算方向展开基准数据集的研究。此外，不少研究者针对自然语言处理提出标准数据集。以情感分析为例，Maria 等人[25]针对方面级情感分析发布了餐厅和电商领域两类数据集，该数据集旨在挖掘用户针对实体不同方面的情感极性，吸引了广大研究者们的深入研究。随后，他们在此基础上新增了方面词类别属性，同时添加了酒店领域数据集[26]，为研究者设计了更多情感分析子任务。一年后，该团队针对该领域提供了涉及 8 种语言和 7 个不同领域的数据集[27]，基于数据集的多样性，研究者可完成的任务也逐渐多样化。然而，现有的工作都没有形成关于开源生态可持续发展的基准数据集。

[1] https://github.com/X-lab2017/open-perf

近年来，随着开源作为全球性数字化发展战略，开源领域的研究工作受到国内外研究者们的青睐。不少研究者对 GitHub 中的软件生态演化过程进行了研究。Hewapathirana 等人[28]围绕开源健康信息构建软件生态系统。齐等人[12]基于动态社区发现方法检测 GitHub 中不断演化的生态系统，识别并比较 GitHub 中的不同演化事件，分析了生态系统存货或消亡的原因。针对开发者的分析工作一直是软件生态的研究热点。Mockus 等人[29]的研究结果表明，核心开发者作为项目的核心和领袖，在代码的工作量上一般远大于边缘开发者。吴等人[15]以 9 个 Apache 项目为基础，分析了开发者对项目的贡献度，并以此有效地区分核心开发者和边缘开发者。关于开源评论文本的研究也一直是学者们的兴趣点，其中包括对项目文档[16]，代码重构[30]，安全问题[18]，编程语言[31]等开源项目不同方面的评价，从开发者情感的角度分析开发者对项目的看法。近期，Jimenez 等人[32]提出了 SWE-bench 评估基准，用来评测大模型解决实际软件工程存在的问题，结果表明，即使是当下表现最好的大语言模型，也只能解决 1.96%的问题。Tao 等人[33]引入了 MAGIS 框架，由四种类型的智能体组成：管理者、仓库管理员、开发人员和质量保证工程师。该框架通过规划编码过程中不同智能体的协作来挖掘大模型解决实际问题的潜力，在 SWE-bench 评测的结果表明，MAGIS 可以解决 13.94%的问题。

综上，当前较多基准测试工作主要围绕某一具体学术研究方向展开，而缺少针对开源生态基准测试的研究；同时，现有开源领域的研究工作较多针对某一具体的研究点进行展开，且大多研究以数据分析为主，未形成具体的基准测试标准。基于此，本文结合数据科学领域知识、现有数据集以及开源领域应用场景提出基于开源生态的数据科学基础测试体系，该体系以领域知识为基础，基准空间为任务，应用场景为导向，为开源生态可持续发展提供一套较为完整的基准测试体系架构。

3 OpenPerf 总体架构

基准测试科学与工程[20]的主要目标之一是建立跨学科的标准基准层次结构。通过推出多学科基准、标准和评价指标以交流基准科学和工程的最新技术和实践状态。

基准测试最重要的是保持基准的一致性，一般通过以下方式来实现：（1）统一定义测量标准和计量单位；（2）实现具有不同精度的测量标准和计量单位；（3）基准测试体系的溯源性和校准。溯源性是测量结果的一种特性，它包含可记录的完整校准链与参考相关链，每个校准链都会对测量结果产生影响。校准是将结果为未知或未经鉴定的某个测量实体与具有已知特性的测量实体进行比较的过程。

基准测试主要分为以下几类[20]：系统测量标准（如 LINPACK、TPC）、代表性工作负载（如 MLPerf）、数据科学标准任务与标准数据集（如 ImageNet、OGB）、代表性数据集（如金融领域各种指数）、最佳实践基准（如各个行业/商业领域的最佳实践）。

本文参考上述基准分类结果以及开源领域不同的应用场景引出关于开源软件开发与生态演化的数据科学基准测试层次结构体系，如图 1 所示。本文将开源领域下的基准原理、测量标准和标准数据统称为开源测量学，基准的提出需要标准数据的支撑和测量标准的确定。标准数据是指基于具体研究方向包含明确定义的标准数据集，数据包含确定的标签，例如 ImageNet 就是一个标准数据集。测量标准即为被测量对象的可测量特性，例如长度，高度等。在机器学习中，评价指标与测量标准类似。

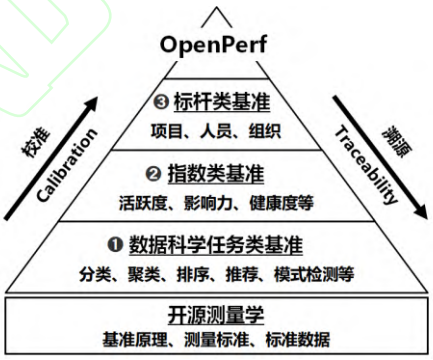

图 1 面向开源生态可持续发展的数据科学基准测试体系

OpenPerf 体系第一类是以分类、聚类、排序等标准任务为主的**数据科学任务类基准**，该类基准主要应用于学术界，为不同研究方向的研究者提供自己擅长研究领域的基准。该类基准测试任务具体构建流程如图 2。

其中，基准空间中的每一个基准包括六个要素：

- **任务名称**：具体的任务名称，如“开源自动化机器人识别与分类”；
- **开源场景**：对应开源应用中的一个或多个场景，如 DevOps 机器人自动化；
- **任务类型**：对应知识域中的一个或多个抽象任务，如分类任务；
- **数据集**：该任务对应的基准数据集，例如 GitHub 协作行为数据集；
- **模型**：给出现有优秀模型的实验结果，方便

研究者与其进行对比；

- **评价标准**：任务对应的评价标准，例如 Accuracy、Precision/Recall、F1-score、AUC。

同时，参考数据分析与挖掘领域的知识体系[34]，可以将开源领域的任务根据不同的视角进行划分：

- **任务视角**：数据预处理、聚类、分类、关联模式、异常检测。
- **数据视角**：文本、时序、离散序列、图与网络、多媒体、时空。
- **研究视角**：数据流分析、复杂网络分析、Web 挖掘、社交网络分析、NLP 分析、推荐系统。

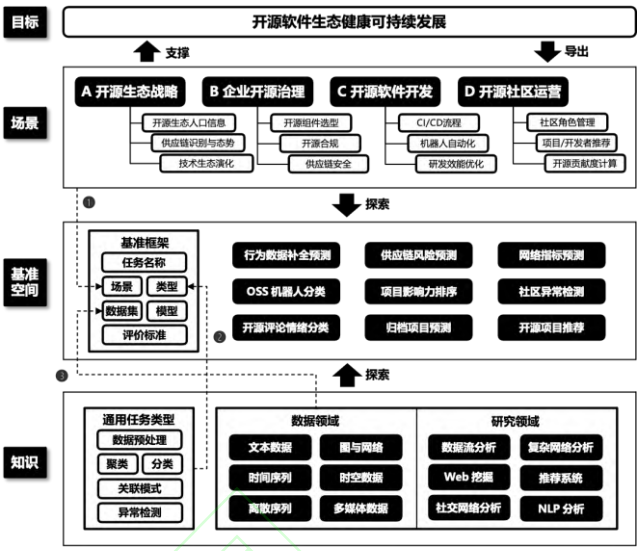

图 2 数据科学任务类基准构建流程

表 1 数据科学任务类基准测试任务

序号	任务名称	数据类型	问题类型	主要应用场景	研究领域
1	开源行为数据补全与预测	时间序列	回归问题	企业开源治理	数据流分析
2	开源自动化机器人识别与分类	时间序列	分类问题	开源软件开发	数据流分析
3	开源评论文本情绪分类	文本数据	分类问题	开源社区运营	NLP 分析
4	开源软件供应链风险预测	时间序列	回归问题	开源生态战略	复杂网络分析
5	开源项目影响力排序	图与网络	排序问题	开源社区运营	复杂网络分析
6	开源归档项目预测	时间序列	回归问题	企业开源治理	Web 挖掘
7	开源网络指标预测	图与网络	回归问题	企业开源治理	复杂网络分析
8	开源社区异常行为检测	时间序列	异常检测问题	企业开源治理	复杂网络分析
9	基于链路预测的开源项目推荐	图与网络	推荐问题	开源社区运营	推荐系统

OpenPerf 构建的数据科学任务类基准任务如表 1 所示，包含了任务名称、数据类型，问题类型、应用场景和研究领域等要素，具体任务细节见第 4 章表 2。

如图 2，OpenPerf 通过最底层的数据科学知识体系，结合具体的数据领域、研究领域以及多种通用任务类型，同时考虑了上层具体的业务场景，共同探索开源领域的基准空间。开源领域业务场景包含开源生态战略、企业开源治理、开源软件开发以及开源社区运营，不同场景下包含详细的业务，以上场景均从开源领域的各个方面支撑开源软件生态健康的可持续发展。为了保持开源软件生态健康的可持续发展，又可以导出开源软件生态的具体业务场景，再根据详细的业务场景探索具体任务的基准空间。在开源领域知识体系和具体业务场景驱动下，确定基准框架中的具体任务、任务类型、数据集、评价标准以及模型设计，则完成一个数据科学任务类基准的构建。

第二类为**指数类基准**，主要面向工业界，企业可以通过不同的指数类基准展开详细洞察，与数据科学任务类基准不同，该类基准可以理解为基准单位，该基准单位可以对测试对象某一具体属性进行测量，例如活跃度、影响力、健康度等。

第三类则是**标杆类基准**，是一种可测量的业界最佳水平的成绩，用来比较参考尺度；得到认可的绩效水平，作为特定领域的卓越标准。本质上讲，这类基准测试的目的就是向标杆学习。对应到开源生态中，可以有标杆开源项目、标杆开源开发者、以及标杆开源企业等。这些实体拥有可量化的、业界最佳水平的成绩或得到认可的绩效水平，能够作为相关领域的卓越标准。

数据科学任务类基准由于其涵盖较广的任务类型、研究方向等多项因素，相对其他两类基准较易实现。指数类基准需要提出独有的测量单位，该单位可以对实体的某一项属性进行描述且需要被业界认可，难度更进一步。标杆类基准依托于指数类基准，当某个实体在指数类基准的测量下达到一定值时，可以将其视为标杆，所以指数类基准的定义会影响到标杆类

基准最后的实现效果。

以阿里巴巴开源开发者贡献激励榜（详见 6.3 节）为例，该应用旨在通过精准地评价个人在项目中的影响力从而进一步形成对开发者的激励，鼓励更多开发者参与到项目的开发中。若将影响力排名前 3 的开发者视为开发者标杆，对其溯源发现评价的主要依据为影响力指数基准，影响力指数基准通过计算开发者在协作网络中的贡献度而来，而最终的排序结果则通过开源影响力排序任务产出最终数据，自上而下对标杆类基准完成溯源。同时，可以通过开源影响力排序任务来判定影响力指数的合理性，对每次给出的排序结果进行一次次校准，使其逐步合理；而影响力指数计算的最终结果会直接涉及到标杆类基准最终的判定，通过不断优化调整影响力指数的计算方式则可以校准标杆类基准最后的结果，使其更加符合实际情况。

OpenPerf 基准测试体系的设计初衷是希望涵盖开源生态下的大多数基准测试任务，目标相对宏大。当前课题组已对三类基准分别给出一定的测试结果，在未来会针对该体系进一步优化，对不同子类下的基准测试展开详细定义，完善整个体系下的基准测试结果。

4 数据科学任务类基准

本章结合课题组现有研究成果，选取 9 个最具代表性的数据科学任务类基准测试任务（简称任务类基准）进行描述（如表 2 所示），列出了项目数量、输入、输出、标签来源和评价指标。其中较多任务均采用了 GitHub 平台 star 数排名较高的项目下的协作行为数据，本文认为头部项目下的数据从时序的角度来看数据相对完整，同时也有大量的日志数据去做各方面的分析，数据具备一定的典型性，而对于一些不活跃的仓库，产生的数据较少，不作为主要的分析对象。此外，本文基于特定任务也将现有优秀的一些方法应用于构建的开源数据集中，均具有良好的表现。

“开源行为数据补全与预测”、“开源自动化机器人识别与分类”和“基于链路预测的开源项目推荐”详细内容见第 5 章。开发者的个人情绪在软件开发的过程中会影响生产效率、任务质量、工作满意度[35-38]等，故本文将开源评论文本情绪分类任务作为基准测试任务之一，选取了 GitHub star 数排名前 50 的仓库，提取了 2022 年中这些项目的 issue 评论，对数据进行了预处理，移除了非英文评论，为了保证数据集的质量，由 3 名开源生态学研究方向的硕士研究生对长度不超过 128 的短评论文本进行打标签，得票高的情绪为该条数据的最终结果，在选择文本时尽量选择了带有明显情感词的评论，暂未考虑中性评论，最后产出 5000 条正向评论以及 5000 条负向评论数据。

开源软件供应链风险预测任务旨在通过量化软件供应链的各种风险（测试完整度、外部依赖度、团队健康度等 9 种风险指标），最终得出可维护性评分。通过实证分析的方式验证可维护性评分的有效性。

开源项目影响力排序任务通过对 GitHub 中 2022 年全域 17 万左右活跃项目进行获取，项目与项目之间通过开发者产生关联从而形成一个开源项目网络图，该图中包含了 280 多万条连边。该任务旨在通过高效的图排序算法对图中含有不同影响力值的开源项目进行排序。

表 2 数据科学任务类基准测试任务(任务细节)

序号	基准测试任务	项目数量	输入	输出	标签来源	评价指标
1	开源行为数据补全与预测[39,40]	5	含有缺失数据的协作行为指标时序数据。	缺失的行为指标数据	随机地对项目中的行为指标数据进行删除。	NRMSE、NMSE、NMAE
2	开源自动化机器人识别与分类[41,42]	19779	含有 17 个相关特征的 GitHub 账户数据	判断该账户是否为机器人账户	人工标注+BIMAN[43]和 BoDegHa[44]	F1、Precision、AUC
3	开源评论文本情绪分类[16, 45]	50	issue 评论文本	积极/消极	人工标注	F1、Precision
4	开源软件供应链风险预测[46,47]	约 166 万	含有团队健康、软件活跃度、影响力、测试完整度、外部依赖度等 9 种属性的开源项目	可维护性评分	/	/
5	开源项目影响力排序[48]	约 17 万	含有影响力值的开源项目协作网络	排序结果	/	算法运行时间

6	开源归档项目预测[49]	361	含有贡献者巴士系数，issue 响应时间，issue 解决时间，PR 响应时间，PR 接收比例 5 个主要特征的归档项目	判断项目在未来是否会被归档	GitHub API	F1、Precision
7	开源网络影响力指标预测[50]	300	含有参与者、PR 周期、issue 和 PR 活跃度、issue 评论数、新建 Issue 数等 20 种指标的开源项目	拟合出的该项目影响力值	各个项目的影响力值[48]	MSE、RMSE
8	开源社区异常行为检测[51,52]	1	含有异常行为数据的协作行为时序数据。	判断行为数据是否异常	通过聚类模型和专家意见将与正常数据偏差较大的数据定义为异常数据。	F1、Precision
9	基于链路预测的开源项目推荐[53,54]	10000	由开发者与开源项目共同构成的开源协作网络图，详见 5.3 节	开发者在未来可能会参与的项目	随机的删除开发者参与过的部分开源项目	AUC、算法运行时间

5 代表性任务类基准测试与结果

本节选取 3 个代表性任务类基准测试（开源行为数据补全与预测、开源自动化机器人识别与分类、基于链路预测的开源项目推荐），给出数据集、评价指标与评测结果，为后续研究学者提供基准参考样例。

5.1 开源行为数据补全与预测

（1）数据集

为了验证基准模型的通用性和预测精度，该任务从 OpenLeaderboard[2]中采集了 2020 年全年活跃度较高的 5 个开源项目的行为数据集进行测试。其中包括 Pytorch、Skywalking、Tensorflow、Tidb、Vscode。然后随机地对项目中的行为数据进行删除，尽可能的涵盖多种类型的日志数据，为测试基准模型效果提供了较为全面的测试场景[39]。

（2）评价指标

为了验证各个模型对缺失值和未来值的预测效果，选取了 NMSE、NRMSE、NMAE 三个评价指标作为评价标准。如下所示：

$$NMSE = (\frac{1}{|\Omega|}\sum_{i,t\in\Omega}(\widehat{Y_{it}} - Y_{it})^2)/(\frac{1}{|\Omega|}\sum_{i,t\in\Omega}Y_{it}^2) \quad (1)$$

$$NRMSE = \sqrt{(\frac{1}{|\Omega|}\sum_{i,t\in\Omega}(\widehat{Y_{it}} - Y_{it})^2)}/\sqrt{(\frac{1}{|\Omega|}\sum_{i,t\in\Omega}Y_{it}^2)} \quad (2)$$

$$NMAE = (\frac{1}{|\Omega|}\sum_{i,t\in\Omega}|\widehat{Y_{it}} - Y_{it}|)/(\frac{1}{|\Omega|}\sum_{i,t\in\Omega}|Y_{it}|) \quad (3)$$

其中$\widehat{Y_{it}}$是预测值，Y_{it}是对应的真实值。在实际应用中的效果 NMAE 和 NRMSE 主要关注预测值与实际值之间的差异，而 NMSE 则更侧重于预测值和实际值之间的相对差异，这三种指标综合起来能对模型的预测性能进行全面的评估。

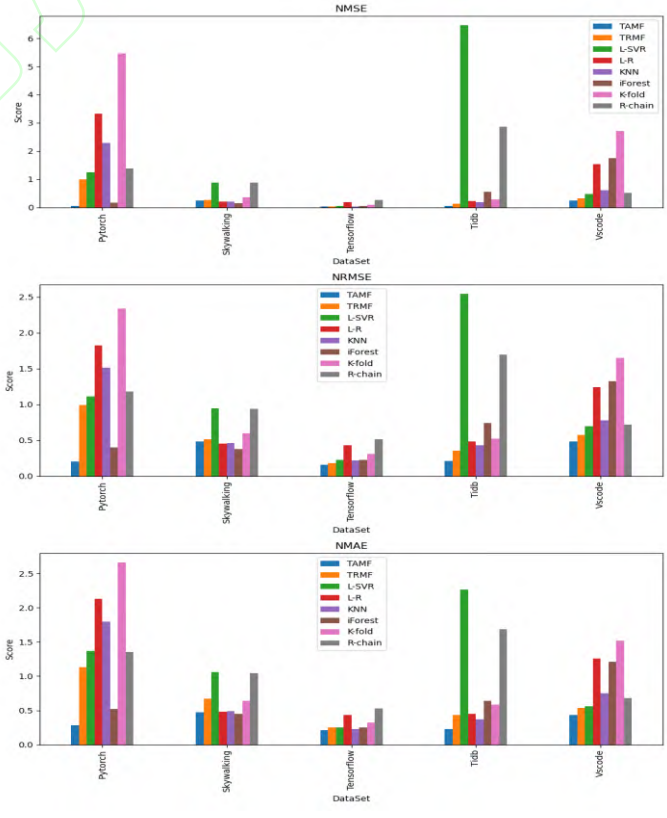

图 3 OSS 行为数据预测实验结果

（3）基准模型实验

[2] https://github.com/X-lab2017/open-leaderboard

该任务选择了 TAMF[39]、TRMF[55]、L-SVR(Linear-SVR)、L-R(Linear-Regression)、KNN、iForest、K-fold 和 R-chain(Regressor-chain)作为基准测试算法。TAMF 是一种结合矩阵分解与自回归的预测模型，挖掘了时间序列数据的趋势和周期性特征。TRMF 是基于图方案提出的模型，对数据补全方面有较好的性能。L-SVR 是一种经典的多维预测模型，L-R 算法是常用的回归模型；KNN 为结构简单，较易实现的模型；iForest 为常用于工业界的算法；K-fold 和 R-chain 为高效的机器学习算法。本实验首先将 8 种算法的迭代次数设置为固定值 1000(可调)，然后计算除缺失值外所有其它正常值的预测误差。在实际的实验中发现，算法的迭代次数往往还未达到 1000 便已经收敛，如果迭代次数过小则算法可能无法收敛，次数过大则与真实迭代次数相差较大，故经过多次实验后，设置为 1000 最接近真实迭代次数。最后，得到每组算法在 5 种数据集上的 NMSE、NRMSE 和 NMAE 值，图 3 可以看出不同评价指标的对比情况。

从表 3 可以得出，除了在 Skywalking 数据集上的效果略低于 KNN、L-R 和 iForest 外，TAMF 是其余算法中预测精度最高的方法。通过分析 Skywalking 数据集发现其周期性较弱，则表明 TAMF 对具有较强周期性的数据集预测结果更好。TAMF 的预测效果明显优于其他算法，该方法通过挖掘时间序列数据的趋势和周期性特征比其余利用单一特征挖掘的方法更合适对开源项目行为数据进行预测。

5.2 开源自动化机器人识别与分类

（1）数据集

该任务选择了 GitHub 中 2021 年 3 月至 2022 年 3 月活跃的仓库以确保泛化能力和准确性。同时将协作数据超过 100 条日志的账户识别为“活跃账户”数据集，从全局账户中随机选择了一部分账户作为“随机账户”数据集。BIMAN[43]和 BoDeGha[44]算法是两个优秀的 GitHub 机器人识别算法，BIMAN 账户表示 BIMAN 算法下识别出来的机器人账号，BoDeGha 账户同理，为了扩大数据集并确保与 BIMAN 和 BoDeGha 算法的比较实验的可信度，故将这两个算法识别出的机器人账户也考虑进来。随后，对 BIMAN 和 BoDegHa 的数据进行预处理，获取了他们的 GitHub ID，选择了 2023 年内活跃的账户（活跃数据大于 100 条日志）。同时，将"活跃账户"、"随机账户"、"BIMAN 账户"和"BoDeGha 账户"的数据合并成一个数据集，称为"混合账户"。然后，对"混合账户"的数据进行清理，选择了 17 个相关特征以确保数据特征的全面性，构成了 OSS 机器人分类数据集[41]。

（2）评价指标

为了评估机器人识别模型的性能，本节采用了一系列标准的机器学习评估指标，包括准确度（Accuracy）、精确度（Precision）、召回率（Recall）、F1 分数（F1-score）以及 AUC 值（Area Under the ROC Curve）。其主要公式如下：

$$Accuracy = \frac{TN+TP}{TP+FN+FP+TN} \tag{4}$$

其中 FP 表示假正例，被模型预测为正的负样本；TP 表示真正例，为被模型预测为正的正样本；FN 表示假负例，被模型预测为负的正样本；TN 表示真负例，被模型预测为负的负样本。

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

$$F1-score = \frac{2*Precision*Recall}{Precision+Recall} \tag{7}$$

AUC 值即 ROC（Receiver operating characteristic curve）曲线下的面积，ROC 曲线是以假正例率为横轴，真正例率为纵轴绘制的曲线，AUC 值能够度量模型对正例和负例的区分能力，值越大，说明模型的区分能力越好。

准确度、精确度和召回率主要评估了模型的分类能力，即模型正确识别正负类的能力；F1 分数是精确度和召回率的综合指标，它能在一定程度上平衡精确度和召回率的权重；AUC 值则反映了模型在面对不同分类阈值时的性能表现。

（3）基准模型实验

OSS 机器人识别任务采用了多种机器学习模型进行实验，包括逻辑回归（LogisticRegression）、决策树（DecisionTreeClassifier）、支持向量机（SVM）、高斯朴素贝叶斯（GaussianNB）、K 近邻（KNeighborsClassifier）、随机森林（RandomForestClassifier），以及专门针对 OSS 机器人识别任务设计的模型 BotHunter[56]、BoDeGHa 和 BotHawk。BotHunter 是一种通过用户推文的文本内容来判断给定的 Twitter 用户是否为机器人的方法。BoDeGHa 通过分析与评论相关的多维特征来识别 GitHub 上的软件机器人账户。BotHawk 可以检测更多类型的机器人账户，其中包含 CI/CD 和扫描机器人。图 4 可以看出不同算法在不同评价指标下的对比情况。

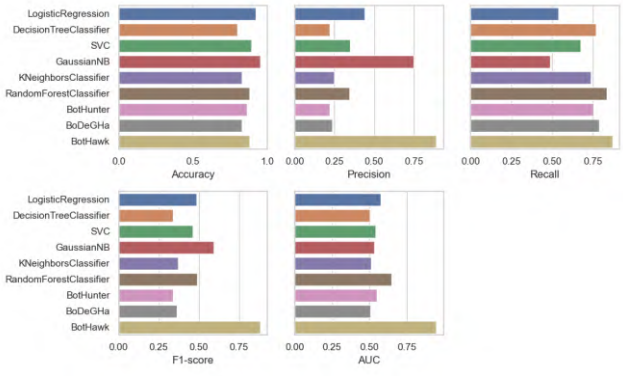

图 4. 开源自动化机器人分类实验结果

从表 4 可以看出 ,BotHawk 模型相对其他优秀的模型在 Precision(0.8930)、Recall(0.8715)、F1-score (0.8821) 以及 AUC (0.9472) 4 个指标上的表现均显示出了最佳的性能。该方法考虑了 GitHub 中多种类型的机器人账户的特征，相对其他方法更适合对 GitHub 中的机器人账户进行挖掘。

表 3 OSS 行为数据预测实验结果

数据集	指标	TAMF	TRMF	L-SVR	L-R	KNN	iForest	K-fold	R-chain
Pytorch	NMSE	0.041	0.987	1.236	3.326	2.28	0.161	5.457	1.388
Pytorch	NRMSE	0.203	0.993	1.112	1.823	1.51	0.401	2.336	1.178
Pytorch	NMAE	0.282	1.132	1.371	2.133	1.80	0.518	2.660	1.356
Skywalking	NMSE	0.235	0.264	0.886	0.202	0.21	0.143	0.353	0.875
Skywalking	NRMSE	0.484	0.514	0.941	0.449	0.46	0.379	0.594	0.935
Skywalking	NMAE	0.471	0.671	1.059	0.482	0.49	0.448	0.636	1.042
Tensorflow	NMSE	0.024	0.031	0.051	0.182	0.04	0.049	0.094	0.262
Tensorflow	NRMSE	0.157	0.176	0.225	0.426	0.22	0.223	0.307	0.512
Tensorflow	NMAE	0.210	0.250	0.253	0.429	0.23	0.250	0.318	0.526
Tidb	NMSE	0.043	0.124	6.462	0.23	0.18	0.546	0.272	2.858
Tidb	NRMSE	0.208	0.352	2.542	0.479	0.43	0.739	0.521	1.691
Tidb	NMAE	0.227	0.435	2.261	0.449	0.37	0.640	0.586	1.682
Vscode	NMSE	0.234	0.326	0.482	1.528	0.61	1.753	2.709	0.516
Vscode	NRMSE	0.484	0.571	0.694	1.236	0.78	1.324	1.646	0.718
Vscode	NMAE	0.435	0.535	0.559	1.259	0.75	1.207	1.516	0.677

表 4 开源自动化机器人分类实验结果

Model	Accuracy	Precision	Recall	F1-score	AUC
LogisticRegression	0.9234	0.4427	0.5376	0.4856	0.5760
DecisionTree	0.7995	0.2188	0.7707	0.3408	0.5024
SVM	0.8936	0.3495	0.6767	0.4609	0.5414
GaussianNB	0.9548	0.7514	0.4887	0.5923	0.5319
KNeighborsClassifier	0.8309	0.2472	0.7406	0.3706	0.5119
RandomForest	0.8817	0.3441	0.8383	0.4880	0.6486
BotHunter	0.8649	0.2200	0.7528	0.3405	0.5512
BoDeGHa	0.8286	0.2354	0.7910	0.3628	0.5049
BotHawk	0.8799	0.8930	0.8715	0.8821	0.9472

5.3 基于链路预测的开源项目推荐

（1）数据集

该任务构建了一个包含项目活跃度和项目协作关联度的开源项目基准数据集。如果两个项目中存在相同的开发者，则说明项目之间通过开发者产生关联。该任务通过计算相同开发者在不同开源项目中的活跃度调和平均值，来量化开发者对项目间协作关联度的贡献。公式如下：

$$R_{ab} = \sum_i \frac{A_{ia}A_{ib}}{A_{ia}+A_{ib}} \tag{8}$$

其中，A_{ia} 为开发者 i 在项目 A 的活跃度，A_{ib} 同理。

该数据集选择了 2020 年内较为活跃的项目。通过获取每个项目的标签数据以及计算项目间的协作关联度等一系列的处理步骤后，获得了 3 个数据集：Repo_relation 、Repo_topic 和 Repo_relation_topic。其中，Repo_relation 数据集节点代表开源项目，若不同的项目被同一开发者参与，则连一条边；Repo_topic 中，节点也表示一个开源项目，两个项目之间存在相同的标签则连一条边，标签的重合数量则为权重；Repo_relation_topic 数据集为 Repo_topic 和 Repo_relation 的聚合数据集[53]。

表 5 链接预测算法实验结果

Model	Repo relation Time(s)	Repo relation AUC	Repo topic Time(s)	Repo topic AUC	Repo relation topic Time(s)	Repo relation topic AUC
RA	394.6628	0.9676	119.3958	0.9593	392.3663	0.9692
IRA	1240.5222	0.9715	138.4095	0.9618	1236.6029	0.9718
WRA	1855.0142	0.9727	126.3657	0.9609	1755.1089	0.9731
WICRA	2561.7305	0.9759	161.4153	0.9676	2626.3151	0.9755
GCN	11396.1526	0.9064	4309.6158	0.9449	11460.5472	0.9006
GraphSAGE	4779.3441	0.8986	3614.3415	0.9514	4945.2739	0.8956
Attri2Vec	497.9778	0.8806	403.2008	0.8513	547.8489	0.8806
Node2Vec	2373.1777	0.9220	1033.8656	0.9475	2193.7499	0.9239

（2）评价指标

AUC 可以被理解为一种概率。在指定某条不存在的边，其分数值为 P_1 的情况下，从测试集中选取一条存在的边，其分数值为 P_2，则 P_2 的值存在大于 P_1 的可能性。如果 $P_2 > P_1$，就为此次比较加 1 分；如果 $P_2 = P_1$，则为此次比较加 0.5 分。通过进行 n 次这样的独立比较，可以得到 n'次加 1 分和 n"次加 0.5 分。因此，AUC 的计算公式可以表述如下：

$$AUC = \frac{n' + 0.5n''}{n} \tag{9}$$

算法运行时间是评估算法效率的一个重要指标，该任务使用该指标来衡量预测算法执行完成所需要的时间。

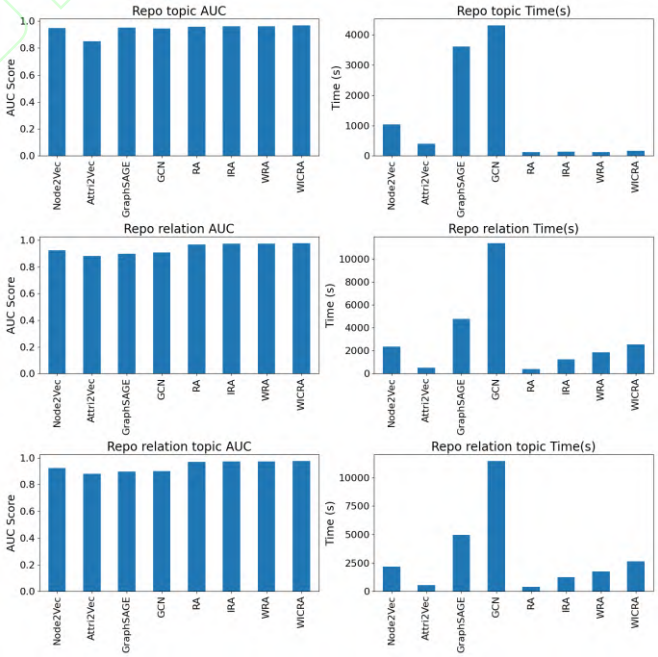

图 5 链接预测算法实验结果

（3）基准模型的实验

该任务对比了 4 种基线方法 Node2Vec[57]、Attri2Vec[58]、GraphSAGE[59]、GCN[60]和基于节点局

部信息的 RA、IRA、WRA、WICRA 作为对比算法，Node2Vec 是一种基于 DeepWalk 改进的算法，可以学习网络中节点的低维表示。Attri2Vec 是一种节点嵌入方法，通过结合节点属性和网络结构信息来学习高质量的节点表示。GraphSAGE 是对图网络进行归纳表示学习的框架。GCN 是一种可扩展的、用于图数据的半监督学习方法。RA 的主要理念为度小的共同邻居节点对新链接形成的影响要大于度大的共同邻居节点。WRA 指标为 RA 指标的加权网络版本。IRA 为结合节点之间亲密度的 RA 指标。WICRA 为结合节点亲密度和接近中心性的 RA 指标。实验对比结果如图 5.

从表 5 可以看出，RA 指标虽然未考虑节点亲密度和网络边权信息，但其 AUC 值仍然高于其他 4 种基线方法，且运行时间更短。基于节点局部信息的算法在 AUC 指标上表现更佳，整体而言，基于节点局部信息的链接预测算法效果更为出色。同时，RA 模型在算法运行时间上也优于 IRA、WRA 和 WICRA 模型。

6 指数类与标杆类基准及其应用

6.1 指数类基准

6.1.1 活跃度指数

判断一个项目是否长期处于活跃状态对于开发者的技术选型，组件选取，是否参与该项目成为贡献者有着重要的意义。OpenPerf 选择了一种通过统计开发者协作行为数据并加权求和的方式(OpenActivity)来计算项目活跃度的计算方法。

某开发者在特定仓库上当月的活跃度为该月中该开发者所有协作行为的加权和的平方根，即：

$$A_d = \sqrt{\sum w_i c_i} \qquad (10)$$

仓库的活跃度即为当月该仓库中所有开发者的活跃度总和，公式如下：

$$A_r = \sum A_d \qquad (11)$$

其中，c_i 为某协作行为的总次数，w_i为对应协作行为的权重，在确定权重方面，本文使用了层次分析法 AHP[61]，在构建判断矩阵时，开源专家首先根据经验对各因素进行两两比较，并赋予相应的权重值。这些值反映了在特定准则下各因素的相对重要性。然后通过计算判断矩阵的最大特征值和对应的特征向量，可以得到各因素的权重。根据该方法可得各类协作行为的权重[48]如下：发起 issue 权重为 2.2235、issue 评论为 0.5252、发起 PR 为 4.0679、PR review 评论为 0.7427、PR 合入为 2.0339，上述权重值同样应用于影响力指数的计算中。

本节对 2023 年 6 月 GitHub 中全域活跃项目进行统计与排名，将参与人数与日志增量作为项目活跃度衡量的指标。参与人数表示当月参与项目贡献的开发者数量；日志增量为当月该项目的日志总量。

从表 6 中可以看出，pytorch/pytorch 日志增量最高，但 OpenActivity 仅排第 6。NixOS/nixpkgs 当月参与的开发者数量较少，但 OpenActivity 排第 1。不同指标下可以看出整体排序结果差异较大。这是由于 OpenActivity 根据开发者的协作行为数据加权求和而得出的。OpenPerf 提供了 2023 年 6 月 GitHub 全域 OpenActivity 排名前 10 的项目统计结果，以便其他开发者在提出新的活跃度指数类指标时，与当前产出的基准结果进行对比，从而验证指标的合理性。

表 6 项目活跃度排名对比结果

仓库	参与人数	日志增量	OpenActivity
NixOS/nixpkgs	1908	33956	5163.91
home-assistant/core	3384	18402	4380.23
microsoft/vscode	3557	16907	3643.1
flutter/flutter	2649	15300	2938.33
MicrosoftDocs/azure-docs	1774	9944	2884.33
pytorch/pytorch	2102	52636	2839.17
odoo/odoo	1285	34123	2470.15
dotnet/runtime	862	16641	2298.13
godotengine/godot	2247	11204	2114.51
microsoft/winget-pkgs	539	26600	1703.35

6.1.2 影响力指数

OpenPerf 选择了一种加权 PageRank[62]算法 OpenRank 来计算项目的影响力，并作为量化一个项目影响力的基准单位[48]。

OpenRank 通过开发者和仓库之间的各种协作行为构成协作网络图，在该图中每个节点v_i在每次迭代中的 OpenRank 值由以下公式表示：

$$v_i = (1 - a_i) \sum_{j=1}^{|V|} \frac{w_{ji}}{d_{oj}} v_j + a_i v_0 \qquad (12)$$

其中，v_0代表节点的初始值，a_i表示节点对其初始值的依赖程度，d_{oj}是节点j的加权出度，w_{ji}是从节点j到节点i的边的权重。如果将归一化权重w_{ji}/d_{oj}组织成矩阵S，并将a_i作为对角值组织成矩阵A，根据收敛性证明，所有节点的 OpenRank 值将收敛到一个向量，如下所示：

$$v = \lim_{k \to \infty} v^{(k)} = \lim_{k \to \infty} \left[AS v^{(k-1)} + (E - A) v^{(0)} \right]$$

$$= \lim_{k\to\infty}[(AS)^k v^{(0)} + \sum_{t=0}^{k-1}(AS)^t (E-A)v^{(0)}]$$

$$= (E-AS)^{-1}(E-A)v^{(0)} \quad (13)$$

OpenRank 在计算中心性时考虑了每种类型节点的初始值。如果节点在上个月参与了计算并已有中心性值，其初始值将沿用上个月的值。未获得中心性值的节点，其初始值将设为 1。对于 issue 或 PR 节点，初始值会根据开发者点赞数增加。若 PR 节点在当月合并，其初始值将额外提升 50%。在确定不同类型节点的中心性时，会参考节点的初始值及其与其他节点互动所产生的值，从而判断节点对初始值的依赖程度。OpenRank 参考了 PageRank 的默认阻尼因子，所有节点的中心性中有 15%基于初始值，85%则依赖于网络内的价值传递，各类协作行为的权重值则可以参考 6.1.1 节。

表 7 项目影响力排名对比结果

仓库	度中心性	PageRank	OpenRank
home-assistant/core	0.015660	0.0035	2393.86
NixOS/nixpkgs	0.008743	0.0008	2207.5
microsoft/vscode	0.015247	0.003	1960.39
flutter/flutter	0.012138	0.002	1460.34
pytorch/pytorch	0.009624	0.0012	1421.18
MicrosoftDocs/azure-docs	0.239616	0.08	1216.01
dotnet/runtime	0.004141	0.0006	1181.12
microsoft/winget-pkgs	0.061954	0.0075	1106.3
godotengine/godot	0.203330	0.045	1105.51
odoo/odoo	0.175534	0.043	907.97

本节对 2023 年 6 月 GitHub 中全域活跃项目进行统计与排名，使用了经典的度中心性算法和 PageRank 算法与 OpenRank 进行对比。PageRank 根据网页之间的链接关系来衡量每个网页的重要性，同样可以适用于 GitHub 协作网络中。度中心性算法主要基于节点的连接数量来评估其在网络中的中心性和影响力。

由表 7 可得，MicrosoftDocs/azure-docs 项目的度中心性和 PageRank 值最高，但 OpenRank 相对其他项目较低，home-assistant/core 项目的 OpenRank 值排第 1。由于 OpenRank 通过加权 PageRank 算法来计算项目的中心度，其计算出的值相对其他指标都偏高。该算法考虑了不同协作行为对项目产生的影响，故与其他指标排序结果不同。OpenPerf 提供了 2023 年 6 月 GitHub 全域 OpenRank 排名前 10 的项目统计结果，以便其他开发者在提出新的项目影响力指数类指标时，与当前 3 类影响力指标进行对比。

6.2 标杆类基准：OpenLeaderboard 排行榜

OpenLeaderboard[3] 是华东师范大学数据科学与工程学院推出的一款开源工具，并于 2022 年 6 月北美开源峰会上正式发布。该项目可以看到项目的排名和热度，还可以了解到相关企业在开源领域的贡献和地位，使用了 OpenPerf 中的影响力、活跃度等指数类基准，而这些指数类基准又通过“开源项目影响力排序”等任务进行定义并产出最终数据，同时将排名靠前（例如 Top100）的对象作为标杆集。如图 6 所示。

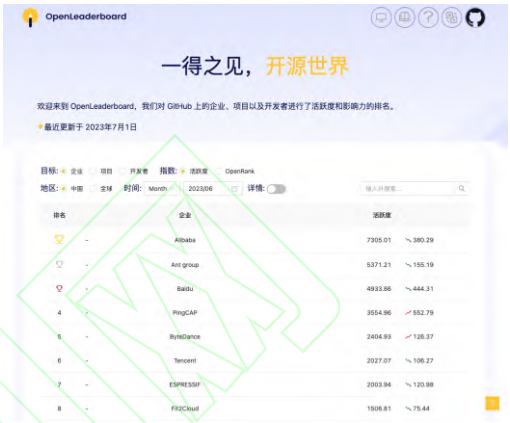

图 6 OpenLeaderboard 项目展示

目前，OpenLeaderboard 已经成为软件行业开源领域的风向标。包括开源组件选型、企业与开源社区观测等领域，均得到了较好的应用。

6.3 行业应用 1：阿里巴巴开源开发者贡献激励榜

当前，不少企业使用 OpenPerf 中提出的影响力基准对企业开源项目的开发者进行影响力评估，从而进一步形成对企业开源项目开发者的激励，鼓励更多新人能参与到开源项目的开发中，如图 7 所示。

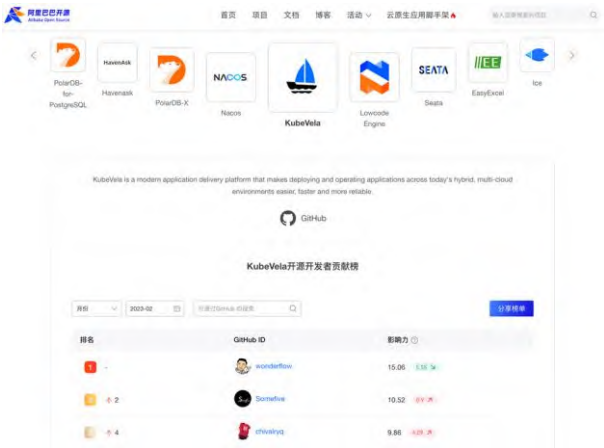

图 7 阿里巴巴开源开发者贡献榜

阿里巴巴积极推动旗下多个开源项目的发起和

[3] https://open-leaderboard.x-lab.info/

捐赠，投入较多的员工进行运维并吸引外部贡献者，致力于建设繁荣的开源社区。公司采用传统的开源社区运营策略，包括发布技术文章、维护沟通渠道、组织活动等吸引更多的开发者参与开源。然而，这些运营实践面临高成本、低转化率等现实问题，无法直观的判断开发者的社区影响力。因此行业需要一种可量化的方法来评估开发者的贡献和价值，从而能够根据开发者的实际贡献持续地激励他们。

OpenPerf 在阿里巴巴公司落地了影响力指数排行榜，即阿里巴巴开源开发者贡献激励榜，该排行榜每月更新，根据每位开发者的影响力数值给予一定的奖励，例如，当影响力数值达到 50 时，可以兑换一定的周边礼品（如文化衫、键盘、鼠标等）。为了验证影响力指数的激励作用，本文选取了 2022 年 4 月-2023 年 4 月一年的协作行为数据展开分析，统计出了每月项目的 issue 评论数，发起 PR 数，表情数，采用了断点回归设计法来分析上述指标。如图 8 所示，横轴表示月份，纵轴表示统计值，2022 年 10 月开始采用 OpenPerf 中的影响力指数激励机制。

可以看出，在采用该机制前，每月 issue 评论数最高在 1100 左右，且整体趋势较为稳定。在采用影响力指数激励机制后，每月的 issue 评论数呈现明显的上升趋势，最高达到 2200 左右。发起 PR 数同样体现出明显的增长，根据影响力基准各协作行为权重的设置，该指标权重相对较高，故较多开发者采用提交 PR 修复 bug 的方式提升影响力。在表情数指标趋势图中可以发现，该指标在使用激励机制后呈现上升趋势，中途略显回落后增长到最高点。这一现象可能源于开发者在最开始并未习惯采用表情符号的方式进行交互，而该指标是作为影响力基准中调整 issue 和 PR 协作行为的权重要素，该指标最后呈现的上升趋势反映了开发者对于该部分算法逻辑的认可，也更愿意为自己认为有价值的 issue 或 PR 进行点赞。通过以上 3 个指标的分析研究，可以看出阿里巴巴在采用影响力指数激励后，对旗下的开源项目产生了积极的影响，在参与问题讨论，提交新的 PR 以及认可其他开发者工作方面均取得了显著的促进作用。由此可知，合理的评估开源开发者的个人影响力可以形成一定的激励，从而进一步促进开源生态的健康发展。

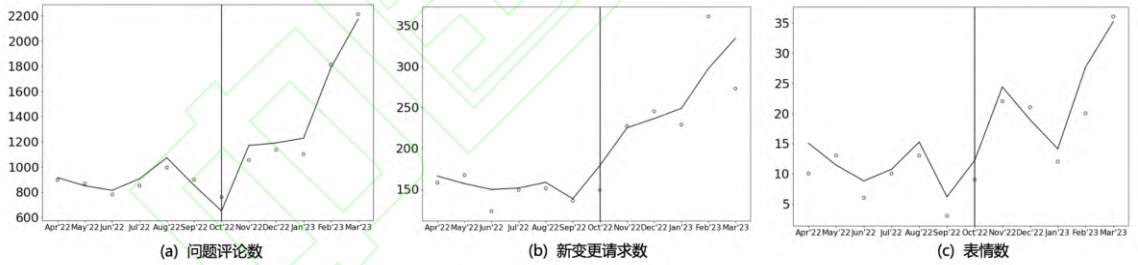

图 8 OpenRank 激励下各项指标变化趋势图

6.4 行业应用 2：蚂蚁集团 OSPO 开源治理大屏

企业开源治理主要与企业所使用的开源软件和企业内外部协作相关，包括如何选择和使用、如何管理和维护、如何与外部企业和社区合作、是否创建自己的开源项目来获取利益等等。目前，国内外 IT 企业的开源办公室将企业开源治理过程与数据可视化看板结合起来的最佳实践逐渐增多，OpenPerf 也在此过程中成为越来越多企业的首选。

图 9 蚂蚁集团开源治理大屏

蚂蚁集团在成立开源项目办公室（OSPO）时意识到，作为一家公司推动开源面临较高的成本和风险，需要较长的时间来建立势头。公司设有业务目标，评估投资回报的周期可能较长，有些努力难以量化。OSPO 的开源项目经理希望通过一些具体的指标了解项目的总体健康状况、影响力等，从而洞察企业整体的开源能力。

OSPO 关注生态系统级别的数据，即开发者和仓库之间的协作关系，这与 OpenPerf 下的指数类基准具有相关性。蚂蚁集团 OSPO 开源治理大屏[63]（图 9）利用 OpenPerf 中的 OpenRank 和 OpenActivity 指数评估各开源项目的当前影响力和活跃度。OSPO 的开源项目经理每月通过观察各项目指标的具体情况来判断其是否处于健康发展状态。某次项目经理发现旗下

某一开源项目指数类基准下降严重，从而判断协作行为数据出现缺失，随即使用“开源行为数据补全与预测”任务对在数据采集阶段丢失的数据进行补全，使其治理大屏稳定运维。相对于单一的统计型指标，活跃度和影响力指数更能体现开源项目的协作影响力。该大屏不仅协助负责人从开源战略角度做出决策，还能激励企业内部员工。在指数类基准的激励下，开发者们更倾向于在开源平台中进行协作讨论，从而促进开源生态的发展。

该基准服务从宏观的角度分析集团下多个开源项目的发展现状，有效地量化不同项目的活跃度与影响力，为项目未来持续健康的发展提供一定的参考。

6.5 行业应用 3：开源软件课程过程性评价

华东师范大学自 2019 年开始逐步开设了面向研究生、计算机类本科生以及全校通识类的开源课程[64]。《开源软件通识》[4]是一门面向高校全体学生的开源通识性课程。课程的学习过程、作业、实训等均在代码仓库中开展。如今，国内外学者在评估学生对开源项目的贡献方面进行了大量探索。然而，现有研究大多关注于项目中代码工作的评估，忽略了学生在开源社区中的协作能力。因此，《开源软件通识》课程使用了 OpenPerf 中的影响力指数类基准对学生的协作能力进行评价，如图 10 所示。

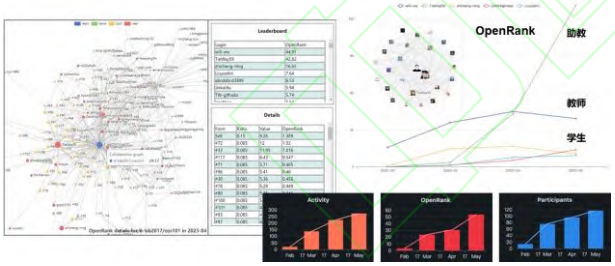

图 10 开源通识课程学生影响力网络

为了验证影响力基准在评价学生贡献度方面的精准度和合理性，华东师大开源课程获取了学生 2023 年 3 月-5 月的影响力值，采用了 Pearson 和 Spearman 两种不同的相关性分析方法，将影响力的最终得分与期中作业评分、期末作业评分、平时分以及总成绩进行了对比研究，结果如表 8。

通过两项相关系数可以看出，OpenRank 评分与教师评分存在一定程度的关联，与期中作业和期末作业的关联程度较弱，这是因为传统评分方法更加关注课堂作业的打分，而 OpenRank 重点关注学生在开源项目中的协作行为来给出评分。OpenRank 与其他评分方法的 Spearman 相关性和 Pearson 相关性的 P 值都小于 0.01，即表示相关性系数的结果不是由随机机会造成的。

表 8 OpenRank 与传统评分方法相关性分析

学生成绩	Pearson	Pearson 的 p 值	Spearman	Spearman 的 p 值
平时分	0.54	1.11E-07	0.60	1.07E-09
期中作业	0.25	1.95E-02	0.28	9.79E-03
期末作业	0.31	4.19E-03	0.33	1.93E-03
总成绩	0.53	1.57E-07	0.58	8.19E-09

为更深入探讨 OpenRank 指标与传统教师评分之间的差异，该课程对某些学生进行了案例分析。用户名为 Surefour 的学生课程成绩为 73 分，为班级最低，而 OpenRank 分数处于中等，为 5.12 分。通过查看该生在开源课程中的各项表现发现，其期中和期末作业得分较低，作业完成质量欠佳，且出勤率较低，影响了平时成绩。然而，该生在课程的 GitHub 仓库中较为活跃，积极参与讨论，修复了仓库的一些 bug，并贡献了大量代码。该生的 OpenRank 分数反映出他在课程实践中对开源仓库做出了较多贡献，而传统评分方式难以全面评估这部分贡献。通过结合 OpenRank 与传统评分对学生进行评价，可以更加全面、公正地衡量学生的综合能力和实际贡献，促进学生的团队合作与开源协作能力的培养，不仅能够让学员积极在仓库上开展开源活动，推动开源项目的发展；同时还能通过该过程让学员深入了解开源协作的本质。

7 总结

近年来，开源软件的持续发展得到了全球社会的极大关注，针对开源领域的研究工作逐渐引起了广大学者们的兴趣。本文结合基准测试科学与工程的核心内容，提出一种面向开源生态可持续发展的数据科学基准测试体系，定义了开源方向的 9 个数据科学任务类基准测试，并给出了 3 项典型的任务类基准测试结果、2 项指数类基准以及 1 项标杆类基准，帮助研究者更好地理解和使用 OpenPerf，最后通过 3 个行业应用证明了 OpenPerf 在推动开源生态发展中起到的关键作用。

OpenPerf 主要聚焦于开源领域的数据科学任务

[4] https://github.com/X-lab2017/oss101/

类基准，在未来会逐步迭代，针对不同的研究子领域提出详细的基准测试结果，例如项目健康度、成熟度、社区化程度等。

OpenPerf 基准体系的设计紧密结合学术界与工业界，已经产出数篇高质量学术论文，持续给学术界、工业界、基金会等不同组织提供服务，其中，2 项指数类基准已被中国电子技术标准化研究院作为开源社区治理的评估标准[2]。

参 考 文 献

[1] Zhan J. A BenchCouncil view on benchmarking emerging and future computing. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 2022, 2(2): 100064.

[2] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database//Proceedings of the 2009 IEEE conference on computer vision and pattern recognition. Miami, USA, 2009: 248-255.

[3] Mazumder M, Banbury C, Yao X, et al. Dataperf: Benchmarks for data-centric ai development. arXiv preprint arXiv:2207.10062, 2022.

[4] Ma Y, Dey T, Bogart C, et al. World of code: enabling a research workflow for mining and analyzing the universe of open source VCS data. Empirical Software Engineering, 2021, 26: 1-42.

[5] Yin Gang, Wang Tao, Liu Bing-Xun, et al. Survey of Software Data Mining for Open Source Ecosystem. Journal of Software, 2018, 29(8): 2258-2271(in Chinese).
（尹刚,王涛,刘冰珣,等.面向开源生态的软件数据挖掘技术研究综述.软件学报, 2018, 29(08): 2258-2271.）

[6] Kilamo T, Hammouda I, Mikkonen T, et al. From proprietary to open source—Growing an open source ecosystem. Journal of Systems and Software, 2012, 85(7): 1467-1478.

[7] Chen C, Golshan B, Halevy A Y, et al. BigGorilla: An open-source ecosystem for data preparation and integration. IEEE Data Engineering Bulletin, 2018, 41(2): 10-22.

[8] Xu R, Xiang H, Han X, et al. The OpenCDA open-source ecosystem for cooperative driving automation research. IEEE Transactions on Intelligent Vehicles, 2023, 8(4): 2698-2711.

[9] Yang L, Geng H, Zhang Y, et al. Open Source Ecosystem in New Era: Pattern and Trend. China’s e-Science Blue Book 2023. Singapore: Springer Nature Singapore, 2024.

[10] Ye Y, Kishida K. Toward an understanding of the motivation of open source software developers//Proceedings of the 25th International Conference on Software Engineering. Washington, USA, 2003: 419-429.

[11] Zhang Y, Zhou M, Stol K J, et al. How do companies collaborate in open source ecosystems? an empirical study of openstack//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. Seoul, Republic of Korea, 2020: 1196-1208.

[12] Qi Qing, Cao Jian, Liu Yan-Cen. The Evolution of Software Ecosystem in GitHub. Journal of Computer Research and Development, 2020, 57(3): 513-524. (in Chinese)
（齐晴, 曹健, 刘妍岑. GitHub 中软件生态系统的演化. 计算机研究与发展, 2020, 57(3): 513-524.）

[13] Dong Rui-Zhi, Li Bi-Xin, Wang Lu-Lu, et al, Review of Research on Software Ecosystems. Chinese Journal of Computers. 2020,43(02):250-271.(in Chinese)
（董瑞志,李必信,王璐璐等.软件生态系统研究综述.计算机学报,2020,43(02):250-271.）

[14] Tsay J, Dabbish L, Herbsleb J. Let's talk about it: evaluating contributions through discussion in GitHub//Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering. Hong Kong, China, 2014: 144-154.

[15] Wu Zhe-Fu, Zhu Tian-Tong, Xuan Qi, et al. Evaluation of Core Developers in Open Source Software by Contribution Allocation. Journal of Software, 2018, 29(8): 2272-2282(in Chinese).
（吴哲夫,朱天潼,宣琦等.基于贡献分配的开源软件核心开发者评估.软件学报,2018,29(8):2272-2282.）

[16] Venigalla A S M, Chimalakonda S. Understanding emotions of developer community towards software documentation//Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). Madrid, Spain, 2021: 87-91.

[17] Huq S F, Sadiq A Z, Sakib K. Is developer sentiment related to software bugs: An exploratory study on github commits//Proceedings of the 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER). London, Canada, 2020: 527-531.

[18] Pletea D, Vasilescu B, Serebrenik A. Security and emotion: sentiment analysis of security discussions on github//Proceedings of the 11th working conference on mining software repositories. Hyderabad, India, 2014: 348-351.

[19] Sinha V, Lazar A, Sharif B. Analyzing developer sentiment in commit logs//Proceedings of the 13th international conference on mining software repositories. Austin, USA, 2016: 520-523.

[20] Zhan J. Call for establishing benchmark science and engineering. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 2021, 1(1): 100012.

[21] Zhan J, Wang L, Gao W, et al. Evaluatology: The science and engineering of evaluation. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 2024, 4(1): 100162.

[22] WordNet: An electronic lexical database[M]. MIT press, 1998.

[23] Hu W, Fey M, Zitnik M, et al. Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems, 2020, 33: 22118-22133.

[24] Zhou M, Li B, Yang M, et al. TeleGraph: A Benchmark Dataset for Hierarchical Link Prediction. arXiv preprint arXiv:2204.07703, 2022.

[25] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, et al. SemEval-2014 Task 4: Aspect Based Sentiment Analysis//Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). Dublin, Ireland, 2014: 27-35

[26] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, et al. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. //Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Denver, USA, 2015: 486-495.

[27] Pontiki M, Galanis D, Papageorgiou H, et al. Semeval-2016 task 5: Aspect based sentiment analysis//Proceedings of the ProWorkshop on Semantic Evaluation (SemEval-2016). Berlin, Germany, 2016: 19-30.

[28] Hewapathirana R, Amarakoon P, Braa J. Open Source Software Ecosystems in Health Sector: A Case Study from Sri Lanka.//Proceedings of the Information and Communication Technologies for Development: 14th IFIP WG 9.4 International Conference on Social Implications of Computers in Developing Countries. Yogyakarta, Indonesia, 2017: 71-80.

[29] Mockus A, Fielding R T, Herbsleb J D. Two case studies of open source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology (TOSEM), 2002, 11(3): 309-346.

[30] Jurado F, Rodriguez P. Sentiment Analysis in monitoring software development processes: An exploratory case study on GitHub's project issues. Journal of Systems and Software, 2015, 104: 82-89.

[31] Guzman E, Azócar D, Li Y. Sentiment analysis of commit comments in GitHub: an empirical study//Proceedings of the 11th working conference on mining software repositories. Hyderabad, India, 2014: 352-355.

[32] Jimenez C E, Yang J, Wettig A, et al. Swe-bench: Can language models resolve real-world github issues?. arXiv preprint arXiv:2310.06770, 2023.

[33] Tao W, Zhou Y, Zhang W, et al. MAGIS: LLM-Based Multi-Agent Framework for GitHub Issue Resolution. arXiv preprint arXiv:2403.17927, 2024.

[34] J. J. Cuadrado-Gallego, Y. Demchenko. The Data Science Framework: A View From the EDISON Project. Cham, Switzerland: Springer Cham, 2020.

[35] Obaidi M, Nagel L, Specht A, et al. Sentiment analysis tools in software engineering: A systematic mapping study. Information and Software Technology, 2022, 151: 107018.

[36] Fisher C D, Ashkanasy N M. The emerging role of emotions in work life: An introduction. Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior, 2000, 21(2): 123-129.

[37] Kaur R, Chahal K K, Saini M. Analysis of Factors Influencing Developers' Sentiments in Commit Logs: Insights from Applying Sentiment Analysis. e-Informatica Software Engineering Journal, 2022, 16(1): 220102.

[38] Miller C, Cohen S, Klug D, et al. " Did you miss my comment or what?" understanding toxicity in open source discussions//Proceedings of the 44th International Conference on Software Engineering. Pittsburgh, Pennsylvania, 2022: 710-722.

[39] Chen L, Yang Y, Wang W. Temporal Autoregressive Matrix Factorization for High-Dimensional Time Series Prediction of OSS. IEEE Transactions on Neural Networks and Learning Systems, 2024,35(10):13741-13752.

[40] Vasilescu B, Serebrenik A, Filkov V. A data set for social diversity studies of GitHub teams//Proceedings of the 2015 IEEE/ACM 12th working conference on mining software repositories. Florence, Italy, 2015: 514-517.

[41] Bi F, Zhu Z, Wang W, Xia X, Khan H A, Pu P. BotHawk: An Approach for Bots Detection in Open Source Software Projects. 2023. arXiv preprint arXiv:2307.13386.

[42] Shvets A A, Rakhlin A, Kalinin A A, et al. Automatic instrument segmentation in robot-assisted surgery using deep learning//Proceedings of the 2018 17th IEEE international conference on machine learning and applications (ICMLA). Orlando, USA, 2018: 624-628.

[43] Dey T, Mousavi S, Ponce E, et al. Detecting and characterizing bots that commit code//Proceedings of the 17th international conference on mining software repositories. Seoul, Republic of Korea, 2020: 209-219.

[44] Golzadeh M, Decan A, Legay D, et al. A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments. Journal of Systems and Software, 2021, 175: 110911.

[45] You L, Han F, Peng J, et al. ASK-RoBERTa: A pretraining model for aspect-based sentiment classification via sentiment knowledge mining. Knowledge-Based Systems, 2022, 253: 109511.

[46] Sun Qing, Liang Guan-Yu, Wu Yan, et al. Data-driven open source software supply chain maintenance risk analysis method. Journal of East China Normal University(Natural Science), 2022, 2022(5): 90-99. (in Chinese)
（孙晴,梁冠宇,武延等.数据驱动的开源软件供应链可维护性风险分析方法.华东师范大学学报(自然科学版),2022(05):90-99.）

[47] Ohm M, Plate H, Sykosch A, et al. Backstabber’s knife collection: A review of open source software supply chain attacks//Proceedings of the Detection of Intrusions and Malware, and Vulnerability Assessment: 17th International Conference. Lisbon, Portugal, 2020: 23-43.

[48] Zhao S, Xia X, Fitzgerald B, et al. OpenRank Leaderboard: Motivati ng Open Source Collaborations Through Social Network Evaluation i n Alibaba//Proceedings of the 46th International Conference on Softw are Engineering: Software Engineering in Practice. Lisbon, Portugal, 2 024: 346-357.

[49] Xia X, Zhao S, Zhang X, et al. Understanding the Archived Projects on GitHub//Proceedings of the 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). Macao, China, 2023: 13-24.

[50] Xia X, Weng Z, Wang W, et al. Exploring activity and contributors on GitHub: Who, what, when, and where//Proceedings of the 2022 29th Asia-Pacific Software Engineering Conference (APSEC).Tokyo, Japan, 2022: 11-20.

[51] Chen L, Wang W, Yang Y. CELOF: Effective and fast memory efficient local outlier detection in high-dimensional data streams. Applied Soft Computing, 2021, 102: 107079.

[52] Al-Fawa'reh M, Al-Fayoumi M, Nashwan S, et al. Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior. Egyptian Informatics Journal, 2022, 23(2): 173-185.

[53] Wang Hao-Yue. Homogeneous open source project recommendation based on link prediction. East China Normal University. Shanghai, 2022.(in Chinese)
(王皓月. 基于链接预测的同质开源项目推荐.华东师范大学, 上海, 2022.)

[54] Xia X, Lo D, Wang X, et al. Accurate developer recommendation for bug resolution//Proceedings of the 2013 20th Working Conference on Reverse Engineering (WCRE). Koblenz, Germany, 2013: 72-81.

[55] Yu H F, Rao N, Dhillon I S. Temporal regularized matrix factorization for high-dimensional time series prediction//Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 847-855.

[56] Abdellatif A, Wessel M, Steinmacher I, et al. BotHunter: an approach to detect software bots in GitHub//Proceedings of the 19th International Conference on Mining Software Repositories. Pittsburgh, Pennsylvania, 2022: 6-17.

[57] Grover A, Leskovec J. node2vec: Scalable feature learning for networks//Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. San Francisco California, USA, 2016: 855-864.

[58] Zhang D, Yin J, Zhu X, et al. Attributed network embedding via subspace discovery. Data Mining and Knowledge Discovery, 2019, 33: 1953-1980.

[59] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. Advances in neural information processing systems, 2017, 30.

[60] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[61] Ossadnik W, Lange O. AHP-based evaluation of AHP-Software. European journal of operational research, 1999, 118(3): 578-588.

[62] Li Q, Zhong J, Zheng L, et al. Streaming graph partitioning for large graphs with limited memory//Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications. Guangzhou, China, 2017: 1269-1271.

[63] Xia X, Wang W, Zhao S, et al. Lessons Learned From the Ant Group Open Source Program Office. Computer, 2023, 56(4): 92-97.

[64] Wang Jie, Huang Wen-Rui, Zhao Sheng-Yu, et al. OpenRank contribution evaluation method and empirical study in open-source course.Journal of East China Normal University(Natural Science), 2024, 2024(5): 11-19.(in Chinese)
(王婕, 黄温瑞, 赵生宇等. OpenRank 贡献度评估方法及其在开源课程中的实证研究. 华东师范大学学报（自然科学版）, 2024, 2024(5): 11-19.)

HAN Fan-Yu, Ph.D. candidate. His main research interests include natural language processing and open source ecosystems.

BI Feng-Lin, Ph.D. candidate. His main research interests include machine learning and data mining.

ZHANG Yan-Bin, M.S., engineer, M.S. supervisor. Her main research interests include software development efficiency improvement and intelligent software testing.

LI Hui, Ph.D., professor, Ph.D. supervisor. His main research interests include big data management and intelligent analysis, artificial intelligence technology and applications.

YOU Lan, Ph.D., professor, M.S. supervisor. Her main research interests include spatio-temporal big data and natural language processing.

WANG Wei, Ph.D., professor, Ph.D. supervisor. His main research interests include computational pedagogy and open-source ecosystem.

QIAN Wei-Ning, Ph.D., professor, Ph.D. supervisor. His main research interests include web data management and social analysis and mining.

ZHOU Ao-Ying, Ph.D., professor, Ph.D. supervisor. His main research interests focus on data management and applications, inclusive of web data management, data intensive computing, in-memory cluster computing, big data benchmarking and performance optimization.