OpenPerf：面向开源生态可持续发展的基准测试即服务系统

X-lab 开放实验室，2023 年 7 月

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，是推动一个研究方向甚至学科发展的重要动力。目前，关于开源生态的研究大多基于某一项具体的研究点展开，而缺少对开源生态基准体系的构建。为了推动开源领域的发展，本文提出一种面向开源生态可持续发展的基准测试即服务系统（OpenPerf）。该服务系统旨在为学术界构建并迭代一个研究框架，系统产出基准、任务、数据集等内容，同时在工业界形成业界标准，完成在学术影响力和工业影响力之间的相互转换；OpenPerf 定义了首批开源研究领域中的 9 个基准测试任务，其中涵盖了时间序列数据、文本数据和图数据 3 种数据类型；研究问题包括回归问题、分类问题、推荐问题、排序问题、网络构建问题和异常检测问题共 6 种；根据提出的基准测试任务，实现了 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，为之后从事开源领域研究工作的研究者们提供借鉴与启发；OpenPerf 同时也作为一个开源项目，以基准测试即服务（Benchmarking as a Service，BaaS）的形式给学术界、工业界、基金会等不同组织提供服务，本文最后通过 3 个具体的应用案例说明了 OpenPerf 在推动开源生态健康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；基准测试即服务；基准任务

OpenPerf: Benchmarking as a Service System for Open Source Ecosystems Sustainable Development

Abstract Benchmark test can be referred to as the quantitative and comparable testing of the test object's performance indicator through the design of scientific testing methods, tools, and systems. With the development of artificial intelligence, the AI benchmark test datasets such as ImageNet and DataPerf have gradually become general standards in academia and industry. At present, research on open source ecosystems is mostly based on a single research point, and there is a lack of construction of a benchmark system for open source ecosystems. In order to promote the development of the open source, this paper proposes a benchmarking as a service for the sustainable development of open source ecosystems (OpenPerf) which aims to provide an iterative research framework for the academic community, such as benchmarks, tasks, datasets, etc. The standard of the open source industry can be formed by the test system which can also complete the mutual transformation between academic influence and industrial influence. The paper has defined 9 benchmark testing tasks in the open source domain, with datasets covering three types: time series data, text data, and graph data. The research problems include six types: regression, classification, recommendation, sorting, network construction, and anomaly detection. Based on the proposed benchmark task, three typical task benchmark testing results, two index benchmarks, and one surveyor's pole benchmark were achieved, providing reference and inspiration for researchers engaged in open source research in the future. In addition, the OpenPerf benchmark system is publicly released as an open source project, and services are provided to different organizations such as academia, industry, and foundations in the form of benchmarking as a service (BaaS). Finally, this paper illustrates the crucial role OpenPerf plays in promoting the healthy development of the open source ecosystems through three specific application cases.

Key words benchmark test; open source ecosystems; sustainable development; benchmarking as a service; benchmark tasks

基准测试[1]（Benchmark）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]。

近年来，开源软件的持续发展得到了全球社会的极大关注[5]，在全球数字化创新与转型以及不同规模组织数字主权中的地位等方面得到了共识。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[6]，GitHub 软件生态系统演化过程研究[7-8]，开发者贡献研究[9-10]，开发者对开源项目的评论文本分析[11-14]等。开源领域的开放数据集为研究工作带来了巨大的便利性与创新机会。

虽然这些数据工具和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、一个社区的健康成熟度达到了怎样的水平、企业开源程序办公能力处于行业什么位置、开发者贡献度、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。而这些都需要多方来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。基于此，本文针对开源领域提出一种面向开源生态可持续发展的基准测试体系（OpenPerf）[1]，并以开源项目的形式发布基准测试服务，推动开源研究方向的持续发展。

本文的主要贡献如下：

1）提出一种面向开源生态可持续发展的基准测试体系。该体系通过开源领域知识和应用场景探索基准空间，支撑开源生态可持续发展的目标，使其可以在学术界端迭代研究框架，系统产出基准、任务、数据集等内容；在工业界端形成业界标准，完成在学术影响力和工业影响力之间相互转换。

2）定义了首批开源研究中的 9 个基准测试任务，其中数据类型涵盖了时间序列数据、文本数据和图数据共 3 种；研究问题包括回归问题、分类问题、推荐问题、排序问题、网络构建问题和异常检测问题共 6 种问题，为不同研究方向的学者们提供了关于开源生态的可研究点。

3）根据提出的基准测试任务，实现了 3 项典型的基准测试（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），2 项指数类基准（影响力和活跃度）以及 1 项标杆类基准，并发布了详细评测实验结果，为从事开源领域工作的研究者们提供参考借鉴与启发。

4）以开源项目的形式完成 OpenPerf 基准体系规范的公开发布，并以基准测试即服务（Benchmarking as a Service，BaaS）的形式给学术界、工业界、基金会等不同组织提供服务，并通过 3 个具体的应用案例说明了 OpenPerf 在推动开源生态健康发展中所起到的关键作用。

1 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[15]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。Deng 等人[2]以 WordNet[16]为主体提出 ImageNet 基准数据集，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[17]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[18]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 Pontiki 等人[19]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[20]，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[21]，基于数据集的多样性，研究者可完成的任务也逐

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

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测试集。

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[22]围绕开源健康信息构建软件生态系统。齐等人[7]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[23]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[10]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[11]，代码重构[24]，安全问题[13]，编程语言[25]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。

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，未形成关于开源生态的基准测试体系。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关于基于开源生态的基础测试体系架构，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。

2 OpenPerf 总体架构

2.1 面向开源领域的基准测试

基准测试科学与工程[15]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，它通过可记录的完整校准链与参考相关联，每个校准链都会对导致测量结果产生影响。

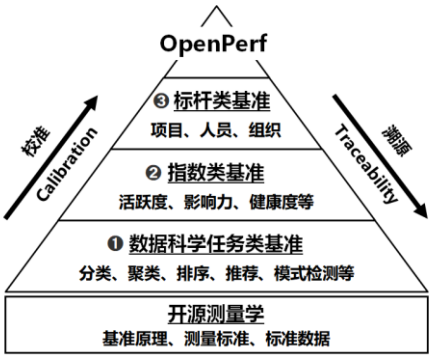

Fig. 1 Benchmarking for the open source

图 1 面向开源领域的基准测试体系

基准测试主要分为以下几类[15]：系统测量标准（如 LINPAC、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）。本文以开源测量学为基础引出关于开源软件开发与生态演化的基准测试层次结构，如图 1 所示。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**；第二类为**指数类基准**，其中包括活跃度、影响力、健康度等，该类基准可以理解为基准单位；第三类则是**标杆类基准**，具体到项目、人员、组织等开源生态中的实体。该基准架构可以以数据科学任务类基准为基础，对指数类基准进行校准，同时指数类基准更进一步完成对标杆类任务的校准。当标杆类基准形成时需要进一步核实该基准的合理性，则可以自顶向下溯源，通过指数类基准以及任务类基准对其完成溯源，这样反复迭代，则最终形成面向开源领域的基准测试。

2.2 OpenPerf 基准服务系统架构

基于上述基准测试体系，本文提出一种面向开源生态可持续发展的基准测试即服务系统。该系统以促进开源软件生态健康可持续发展为目标，包含基准规范、基准实例、公共工具集、以及 BaaS 服务接口四个关键部分，为应用端不同应用场景提供服务。

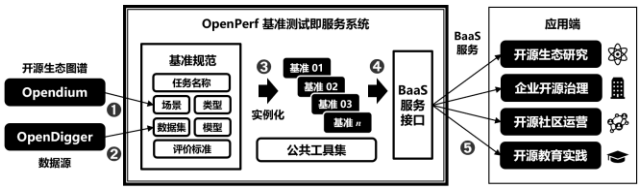

Fig. 2 OpenPerf benchmark as a service system

图 2 OpenPerf 基准测试即服务系统

- **基准规范**：提供一个统一的框架来定义 OpenPerf 中的每个基准实例，一个标准的基准实例由任务名称、应用场景、任务类型、数据集、算法模型、以及评价标准 6 个关键部分组成；
- **基准实例**：是一个具体的基准实例集合，例如开源行为数据补全与预测基准、开源自动化机器人识别与分类基准、基于链路预测的开源项目推荐基准等；
- **公共工具集**：提供了一系列通用的工具，例如数据装载工具、结果评测工具、排行榜等；
- **BaaS 服务接口**：为应用端各应用提供最终的数据服务，包括高校开源研究、企业开源治理、开源社区运营、开源教育实践等场景。

同时，OpenPerf 系统还有两个外部支持项目，协助研究设计人员开发不同的基准实例：

- **OpenDigger**[2]：是一个开源生态数据采集与分析的基础设施开源项目，能够为 OpenPerf 提供各种数据源的制作；
- **Opendium**[3]：则是一个开源生态知识图谱的开源纲目（字典）项目，从不同视角分门别类的梳理了开展基准实例设计所需要的开源知识，包括应用场景分类体系、任务类型分类体系、数据集分类体系、数据科学领域分类体系等。

OpenPerf 服务系统考虑了开源领域的四大应用场景：

- **开源生态研究**：指从经济、社会、教育、学术、政策、法律、文化等层面，对开源生态整体状态的分析与理解，例如开源生态人口信息、供应链识别与态势、技术生态演化等；
- **企业开源治理**：指企业开源管理办公室（OSPO）涉及到的各类业务场景，例如开源组件选型、开源合规、供应链安全、排名与激励等；
- **开源社区运营**：指开源社区测的项目管理与运营的相关业务，例如社区角色管理、项目/开发者推荐、贡献度计算等；
- **开源教育实践**：指通过参与开源项目/社区进行学习与实践，包括 DevOps 下的各类工程性工作，例如 CI/CD 流程、机器人自动化、研发效能提升等。

OpenPerf 体系设计之初就着眼于开源领域的多样性和复杂性，因此它覆盖了各种不同类型的数据和问题，从而实现了广泛的基准测试能力。OpenPerf 的基准测试能力主要体现在以下五个方面：

1). **多元化的数据类型**：OpenPerf 支持对多种数据类型进行基准测试，包括但不限于时序数据、文本数据和图数据。这使得 OpenPerf 可以处理开源项目中的各种数据问题，比如代码变更历史（时序数据）、项目文档和评论（文本数据）、以及项目和开发者之间的关系（图数据）等。

2). **丰富的问题类型**：OpenPerf 不仅能处理基本的分类和回归问题，还可以处理更复杂的排序、网络构建、异常检测和推荐等问题。这使得 OpenPerf 可以广泛应用于开源领域的各种场景，比如项目评估、风险预测、社区分析和推荐等。

3). **深度的任务解析**：OpenPerf 的基准测试不仅仅是运行预定的任务，而是将每个任务深入分解为具体的背景、问题类型、数据类型、评估指标等元素。这有助于用户深入理解每个任务的具体内容和目标，从而更好地选择和调整算法。

4). **全面的评估指标**：OpenPerf 为每个基准测试任务提供了一组全面的评估指标，比如准确性、召回率、F1 值、均方误差等。这使得用户可以从多个角度评估和比较算法的性能，从而做出更全面和准确的评估。

5). **扩展性和灵活性**：用户可以根据需要添加新的基准测试任务，或者修改现有任务的设定。这使得 OpenPerf 可以随着开源领域的发展而不断更新和扩展，保持其在开源数据分析领域的领先地位。

3 数据科学任务类基准

本文以开源生态可持续发展为目标，导出开源领域应用场景，并通过开源领域知识和应用场景共同探索基准空间，基准测试任务具体构建流程如下图。

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**：对应开源应用场景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（通用任务）、日志数据分类（研究领域）；
- **数据集**：该任务对应的基准数据集，例如 721 GitHub Apps 数据集；

[2] https://github.com/X-lab2017/open-digger
[3] https://github.com/X-lab2017/open-research/tree/main/Opendium

- **模型设计**：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**：该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[26]，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、异常分析。
- **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。

基于上述内容，本节选取 9 个最具代表性的基准测试任务进行描述（如表 1 所示），并逐一介绍这些任务的背景、所涉及的问题类型以及它们在开源领域的具体应用。

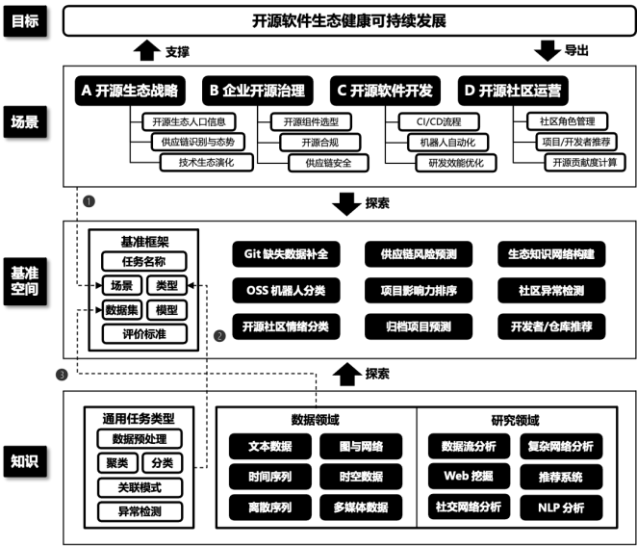

Fig. 3 Benchmarking task construction process

图 3 基准测试任务构建流程

Table 1 Benchmarking Tasks

表 1 基准测试任务

序号	基准测试任务	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[27]	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[28]	时间序列	分类问题	开源软件开发	数据流分析
3	开源社区情绪分类[11, 29]	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[30]	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[31]	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[32]	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[33]	图与网络	回归问题	企业开源治理	数据流分析
8	开源社区异常行为检测[34]	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[35]	图与网络	推荐问题	开源社区运营	推荐系统

3.1 开源行为数据补全与预测

在开源软件领域，高质量的数据治理成为了推动开源发展的重要因素。特别是在数字化进程加速的当下，数据已经转变成一种重要的资源。高质量的数据有助于精准地掌握整个项目的发展现状，而低质量甚至缺失的数据则可能导致研究结论的误差，进而对开源项目的决策造成影响。在从 GitHub 等开源平台采集开发者行为数据时，由于平台的内部机制、API 限制、采集技术和内部相关服务的波动等原因，部分开源项目的行为活动数据无法完全采集，导致部分行为数据缺失，对后续的细粒度研究造成了严重阻碍[36]。

（1）任务描述

如何设计一个既能补全开源行为数据的缺失值，又能对后续趋势进行预测的模型？该模型需要能够充分挖掘开源行为数据的周期性和关联性，保留数据的时间戳信息，且不需要依赖数据的先验信息和概率分布。同时，模型需要能够有效应对数据缺失的情况，例如，GitHub 的采集限制、网络服务的波动等。

（2）任务难点

缺失数据的不确定性：行为数据的缺失可能由许多因素引起，如平台的 API 限制、网络服务的波动等。这种不确定性使得用统一的方式处理所有的缺失数据变得困难，需要能够应对不同缺失情况的方法。

时序数据的处理：开源行为数据是一种时间序列数据，需要在处理缺失数据时保留其时间戳信息，以保持数据的时序性。而时间序列数据的处理，特别是存在缺失值的情况下，比处理静态数据更具挑战性。

缺失数据补全与趋势预测的统一处理：模型需要解决数据的缺失问题并进行趋势预测。该模型需要考虑到数据的周期性、关联性、时序性等多种特性。

3.2 开源自动化机器人识别与分类

合作作为一种社会现象，在软件开发生命周期中的作用越来越重要。当前流行的社交编码平台，如 GitHub，Bit-Bucket 和 GitLab，提供了为开发者共享工作空间的环境，然而，大规模的合作也带来了仓库维护者的巨大工作压力，需要他们完成与贡献者的沟通，审核源代码，处理贡献者许可协议问题，阐述项目准则，执行测试和构建代码，合并拉取请求等多项工作。为了减轻这些重复任务的负担，开源软件项目最近开始利用各种软件机器人来简化他们的运营。机器人的应用也带来了一系列的问题，包括冒充、垃圾邮件、偏见和安全风险等[37]。因此，许多开源研究人员需要识别开源软件机器人账户和行为。

（1）任务描述

如何设计和实现一个模型，以识别并分类开源软件项目中的机器人行为？该模型需要能够准确地识别开源软件项目中的机器人行为，并能够根据他们的行为模式和目标对他们进行有效的分类。除了高精度的预测性能，该模型也需要具有较强的可解释性和合理性。

（2）任务难点

行为模式的多样性：不同的机器人有不同的功能和目标，这就导致他们的行为模式也多样化。机器人可以完成自动回复用户问题、代码审查、代码合并等任务。这种多样性使得机器人行为的识别和分类变得困难。

行为模式的复杂性：例如，一个代码审查机器人可能需要分析代码的复杂度，代码的风格，代码的正确性等多个方面。这种复杂性使得机器人行为的识别和分类需要高度的专业知识和精确的模型。

混合行为的处理：有些机器人可能会和人类开发者一起参与到项目的开发中，这就会产生混合行为。需要模型能够对其进行区分和处理。

3.3 开源社区情绪分类

当前基于开源领域评论文本的情感分析任务相对于餐厅和电商等热门领域较少，开发者的情感会影响任务质量，生产力，创造力，团队和谐以及工作满意度[38, 39]。通过对评论的情感分析可以获取开发者针对项目具体某个方面的行为和见解，有利于项目健康的发展，达到提高开发人员工作效率的目的。

（1）任务描述

Github 大多数的评论均为中性，如何在海量评论文本中挖掘包含开发者观点的评论？不同类型的评论文本表达出的观点也会不同，同时开发者也会针对开源社区的不同方面发表观点，如何获取细粒度的开发者情感也是该任务的难点。

（2）任务难点

情感的多样性：常见的情感分类任务以二分类三分类为主，在 GitHub 中部分开发者的情感是相对复杂的，例如，乐观，厌恶，愤怒，认同，质疑等，通过细粒度的分析可以挖掘更多有价值的观点信息。

评价对象的多样性：开发者往往会针对开源项目某一具体方面进行讨论，例如代码重构，代码风格，安全性，文档可读性等等，如何获取不同方面的评论文本进行情感分析也是难点之一。

3.4 开源软件供应链风险预测

开源软件作为信息技术创新的基石，其依赖关系的复杂性和规模不断增大，带来了指数级增长的安全问题。这些问题可能会在开源软件供应链中产生连锁反应，影响依赖它的所有系统[40]。目前，全球对开源组件的需求增长迅猛，而开源软件供应链的安全情况却在持续恶化。同时，开源软件供应链作为国家博弈的方式和企业运营的基础，其安全性受到高度重视。

（1）任务描述

如何设计和实施一个模型，以预测和管理开源软件供应链中的风险？为了实现这个目标，需要考虑了解和量化软件供应链的各种风险，包括软件的内部属性（如代码行数、软件底层设计、安全漏洞等）和外部属性（如开发团队、活跃度、流行程度等）。需要设计和实施可以对这些风险进行量化评估的预测模型，从而帮助企业和组织更好地管理其供应链。

（2）任务难点

预测模型的设计和优化：由于需要处理大量的特征，并考虑各种可能影响风险的因素，因此设计和优化一个有效的预测模型可能会非常复杂。

评估预测结果的准确性和可用性：需要设计一套评估体系，可以对预测模型的结果进行全面的评估，包括其准确性、稳定性和实用性等。

早期开发过程的预测和管理风险：在早期阶段评

估和管理软件的维护性可以帮助开发团队和使用者尽早发现软件的问题，采取预防和修正的措施，从而大幅度降低后期使用和维护带来的成本。但如何在早期阶段进行有效的风险预测和管理是一个具有挑战性的问题。

3.5 开源项目影响力排序

开源项目不仅提供了成熟、可靠的代码，为开发者节省了大量时间，也为创新和协作提供了广阔的平台。无论是在技术领域，还是在教育、政府、非营利机构等领域，开源项目都发挥着不可忽视的作用。这种趋势使得对开源项目的影响力进行评估变得尤为重要[41]。然而，开源项目影响力的大小取决于多个因素，包括但不限于项目的活跃度、社区规模、贡献者数量、代码质量，以及项目的稳定性等。这些因素的重要性可能会因项目性质、目标用户、发展阶段等差异而有所不同。综合评估这些因素并对其进行合适的权重分配，是评估开源项目影响力的关键环节。对开源项目的影响力进行评估，不仅可以帮助开发者和组织确定哪些项目最有价值，最值得投入时间和资源，还可以为开源项目的优化和改进提供依据。

（1）任务描述

如何构建一个更加精细化的开源项目的评价模型，对项目、开发者等相关元素进行合理的排序？该排序模型需要不仅能评估项目的全局影响力，还能精确地量化每个开发者在项目中的贡献，并赋予相应的激励，以鼓励他们在开源项目中的积极参与。

（2）任务难点

精细化协作单元的确定：在全域协作网络中，项目与开发者被视为节点，以活跃度为边构建了一个庞大的协作网络。然而，在项目内部，需要更加精细化的协作单元，例如 Issue 和 PR，这需要设计一个考虑多种协作单元的模型。

具体贡献者的影响力计算：计算项目内每个开发者的影响力需要确定合适的时间窗口大小，以及各种算法参数。这个挑战涉及到如何根据开发者的行为和项目的特性来评估他们的贡献。

价值导向的设定：如何引导开发者进行高质量的贡献和深度协作？这需要设计一个评价体系，既能公正评价每个人的贡献，又能提供一些引导，鼓励开发者积极参与。

3.6 开源归档项目预测

开源软件生态的可持续性已经成为了一个关键话题。与具有明确交付目标、负责团队和里程碑的传统软件开发生命周期模型不同，开源软件开发在早期阶段高度依赖自组织的贡献者和志愿工作，因此也导致了开源软件项目归档。在开源软件的世界中，大量的项目因各种原因而被归档，即被开发者标记为只读状态，不再接受新的问题、拉取请求或评论。对于在 GitHub 上活跃的开发者以及关注特定开源项目的其他人来说，预测一个项目是否可能会被归档非常重要[32]。

（1）任务描述

如何预测一个开源项目是否可能会被归档？该任务需要对大量的数据进行分析，包括但不限于项目的提交历史、项目的活跃度、项目的维护者信息、项目的成熟度等等。需要通过这些数据来找出可能导致项目被归档的关键因素，然后基于这些因素来制定一个能够预测开源项目是否会被归档的模型。

（2）任务难点

影响因素的多样性：开源项目是否被归档的影响因素包括但不限于项目的生命周期、项目的活跃度、开发者的参与程度、项目的贡献者数量、代码的复杂性、项目的依赖性等等。这些因素可能彼此交互，使得预测结果更加复杂。

时间的动态性：开源项目的状态是随时间不断变化的，因此预测模型需要能够处理时间序列数据，并能够捕捉到项目状态的动态变化。

3.7 开源网络指标预测

在开源软件（OSS）的开发中，开发者的行为数据表现出复杂的关联性和周期性特征，这些特征对于理解开发者的行为模式、项目的进展以及软件质量的影响都具有重要价值。然而，这些复杂的特性使得利用传统的统计方法难以对开发者行为进行准确的预测和分析，因此出现了很多网络指标对开源软件评估。但是开源运营者等更希望从统计指标来找到增强网络指标的方法，因此构建一个能够适应 OSS 数据特性的预测模型，能够拟合网络指标和统计型指标，是一项重要的任务。

（1）任务描述

如何构建一个预测模型，统计型指标拟合网络指标？该模型需要能够适应 OSS 数据的特性，包括各种行为数据之间的复杂关联性和开发者行为数据的周期性。拟合算法需要具有一定的可解释性。

（2）任务难点

周期性：开发者行为数据往往表现出周期性，例如开发者可能在工作日编写更多的代码，而在周末提

交的代码数量减少。这种周期性特征需要预测模型能够捕捉并考虑到。

可解释性：为了理解哪些统计型指标对预测结果更重要，预测模型需要具有一定的可解释性，这对于模型的设计和选择提出了挑战。

动态性：OSS 项目的发展是动态的，开发者的行为模式、项目的规模以及软件的质量都可能随着时间发生变化。因此，预测模型需要能够适应这种动态性，进行实时或近实时的预测。

3.8 开源社区行为异常检测

开源项目中的开发者行为数据是一种重要的信息资源，可以用来监控项目动态，及时发现并处理项目过程中的异常行为，优化项目管理制度[42]。随着项目参与者数量的增加，手动监控所有异常行为变得不现实，需要一种能快速且高效的自动异常监控方案。

（1）任务描述

如何设计和实现一个实时异常检测框架，对实时采集的海量开发者行为数据流进行异常检测，达到以下目标：提前预警类似 NPM 库等异常事件，及时反馈给社区管理人员调整项目管理制度；降低算法的计算规模和耗时，减少对基础设施的需求；过滤异常的行为数据，便于后期对数据进行其他细粒度研究。

（2）任务难点

高通用性：开源项目种类繁多，不同项目中的行为数据具有不同的分布与统计特征，因此设计的检测模型需要具有高通用性，能应用于不同的项目。

计算效率：要实现实时异常检测，需要降低算法的时间复杂度与计算规模，降低空间复杂度，即需要对持续到来的海量行为数据进行有效压缩和筛选，以降低计算规模。

检测精度：在保证实时检测的前提下，需要确保检测方案的高检测精度。

3.9 基于链路预测的开源项目推荐

开源软件开发鼓励全球开发人员以开放协作的方式参与到开发任务中，如错误修复、代码测试和文档改进[43]。每个开发人员都有他们独特的项目需求和技术背景，如果不能将他们与合适的开源项目进行匹配，可能会阻碍开发任务的顺利进行，对开源项目的发展甚至整个开源生态系统的发展产生不利影响。

（1）任务描述

如何基于项目间关系为开发者和社区运营人员推荐感兴趣的开源项目？为开发者推荐感兴趣的开源项目可以帮助他们全面了解目标技术领域的项目，降低开发资源的浪费。

（2）任务难点

复杂的项目关系：开源项目之间的关系复杂多变，开发者和社区运营人员往往需要投入大量时间和精力来了解和掌握这些关系。如何准确地利用这些关系进行高效的开源项目推荐，是一项重大挑战。

项目的多样性：开源项目在技术领域、开发需求、参与者背景等方面存在大量的多样性，这增加了推荐系统的复杂性。

关键词搜索和标签匹配的局限性：虽然 GitHub 等平台提供了基于关键词搜索和标签匹配的功能，但这些方法返回的结果质量参差不齐，且无法完全满足用户的需求。如何提供一种更为有效的方法，以提高开源项目的查找效率和准确度，是需要解决的问题。

4 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），详细分析它们的具体内容与评测结果，为后续研究学者提供参考样例，共同推动开源生态的研究和发展。

4.1 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[4]中采集了 2020 年全年活跃度较高的 10 个开源项目的行为数据集进行测试。其中包括 PyTorch、SkyWalking、Tensorflow、TiDB、VSCode、Flutter、Kibana、Kubernetes、Nixpkgs 和 Rust。该数据集中部分项目的行为数据含有缺失值，为测试基准模型效果提供了较为全面的测试场景[27]。

（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 NMAE、NRMSE、NMSEA 三个评价指标作为评价标准。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 NMSEA 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。

（3）基准模型的实验

使用数据集中有缺失值的 OSS 行为数据作为测试集，然后选择 TRMF、正则化 MF（RMF）、MF、

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

非负 MF（NMF）、概率 MF（PMF）、Basic-SVD 和 BSMF 作为比较算法。为了获得更准确的对比效果，设计了以下实验方案: 首先将七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。最后，得到每组算法在五种数据集上的 NMSE 和 NMAE 值。最终结果如表 2。本节将数据更直观的展示如图 4。

OpenPerf 的基准测试主要基于以上描述的数据集和评价指标展开。为了验证基准模型的通用性和预测精度，各个模型都会在这 10 个项目的数据上分别测试。这种多元化的测试方法可以确保获得的结果不会受到特定项目或特定数据类型的影响。

4.2 开源自动化机器人识别与分类

（1）数据集

该任务选择了 GHTorrent 数据集中 2021 年 3 月至 2022 年 3 月最活跃的仓库，以确保泛化能力和准确性。同时将活动数据超过 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。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，对 BIMAN 和 BoDegHa 的数据进行了处理，获取了他们的账户的 GitHub ID，并选择了过去一年内活动的账户（活动数据大于 10 条日志）。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的全面性。最后经过科学的标签标注流程和一个可视化标注系统，尽可能的提高标签标注的准确性，构成了 OSS 机器人分类数据集[40]。

（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，包括精确度（Accuracy）、查准率（Precision）、查全率（Recall）、F1 分数（F1-score）以及 AUC 值（Area Under the ROC Curve）。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）基准模型的实验

OSS 机器人识别任务中，OpenPerf 采用了多种机器学习模型进行实验，包括逻辑回归、决策树、支持向量机、高斯朴素贝叶斯、K 近邻、随机森林，以及专门针对 OSS 机器人识别任务设计的模型 BotHunter、BoDeGHa 和 BotHawk。最终结果如表 3。图 5 给出了不同算法在不同评价指标下的对比情况。

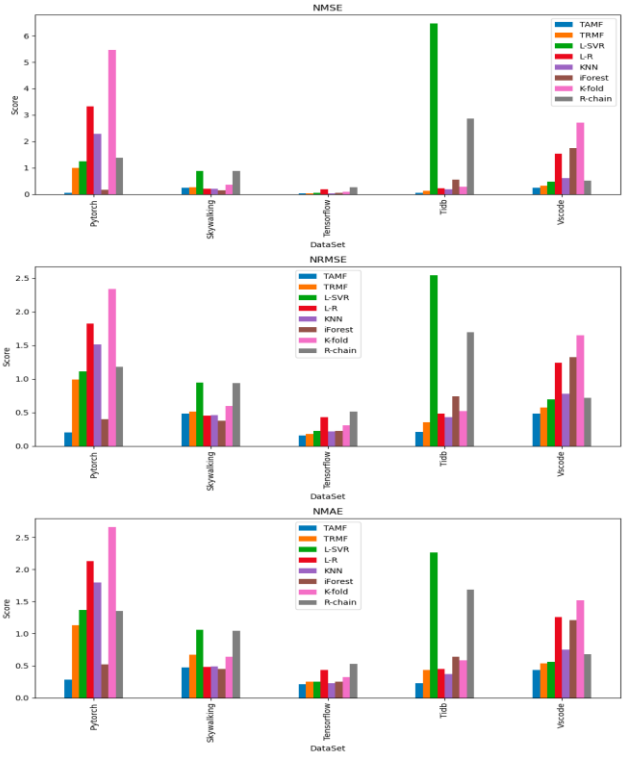

Fig. 4 Results of eight algorithms on five OSS behavior datasets

图 4 八种算法在五组 OSS 行为数据集上的结果

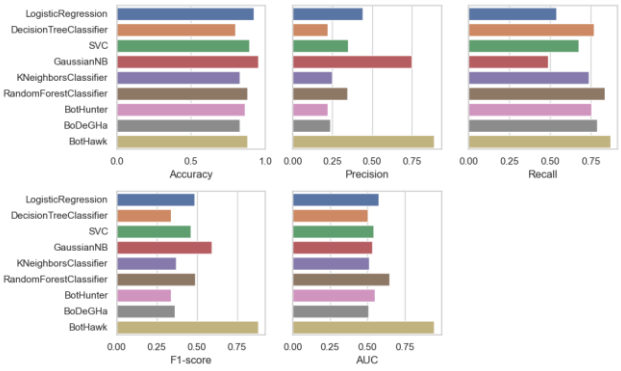

Fig. 5 Classifying results of nine algorithms for OSS bots

图 5 9 种算法对开源自动化机器人分类结果

OpenPerf 为不同模型提供了统一的评估标准，可以使研究人员在相同的条件下将自己设计的模型与基准结果进行比较。通过 OpenPerf 可以帮助识别模型存在的弱点，比如在哪些类型的机器人分类任务上性能较差，从而为改进优化模型提供了一定的方向。它还可以帮助研究者理解当前模型的性能边界，即在当前技术下部分模型的最好表现，为研究人员提供目标，推动他们进行新模型的开发和旧模型的优化。

Table 2 The prediction results of eight algorithms on five groups of OSS behavior datasets

表 2 八种算法在五组 OSS 行为数据集上的预测结果

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

Table 3 Classifying results of nine algorithms for OSS bots

表 3 九种算法对 OSS 机器人分类结果

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

4.3 基于链路预测的开源项目推荐

（1）数据集

为了开发和评估开源项目推荐系统，本节构建了一个包含项目活跃度和项目间的协作关联度开源项目基准数据集。该数据集选择了 2020 年内表现活跃的项目。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经过一系列的处理步骤后，得到了三个带有权重的开源项目基准数据集，分别是 Repo_topic、Repo_relation、Repo_relation_topic。其中，在 Repo_topic 数据集中，节点表示一个开源项目，节点间连边表示两个项目的标签中存在相同标签，权重表示相同标签的个数（数据集中相同标签个数均大于）；在 Repo_relation 数据集中，节点表示一个开源项目，连边表示一个开发者在两个项目中都有过贡献，权重表示项目间协作关联度（数据集中项目间协作关联度的值均大于 5）；Repo_relation_topic 数据集为 Repo_topic 和 Repo_relation 的聚合数据集[35]。

（2）评价指标

链接预测算法主要使用 AUC 和运行时间指标来衡量预测结果的优劣。AUC 是基于算法预测的整体结果来评价算法的预测精度，而运行时间可以反映算法的计算性能。

（3）基准模型的实验

OpenPerf 选择了四种常用的基于网络嵌入的算

法 NodezVee、Attri2vec、GraphSAGE 和 GCN 和基于节点局部信息的 RA、IRA、WRA、WICRA 作为对比算法。基准测试结果如表 4，可视化结果如图 6.

通过以上的数据集和评价指标，OpenPerf 能够全面地评估模型在链路预测任务上的性能，包括预测精度和计算效率等各方面，不仅可以帮助找出模型的优缺点，还可以为相关研究者提供优化和改进模型的方向，从而进一步提升开源项目推荐的质量和效率

Table 4 Results of Eight Weighted Link Prediction Algorithms

表 4 八种加权链接预测算法结果

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

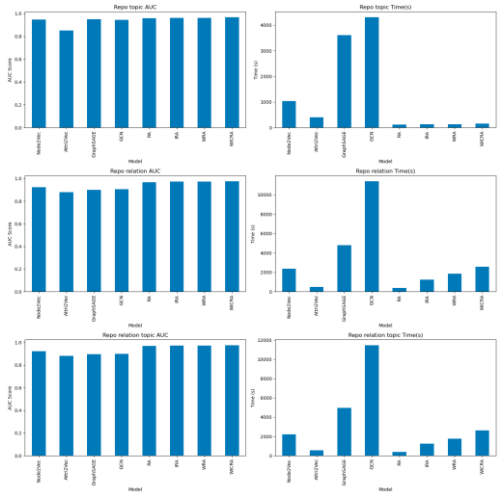

Fig. 6 Results of Eight Weighted Link Prediction Algorithms

图 6 八种加权链接预测算法结果

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 5 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 与其他两种指标不同，它是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。

Table 5 OpenActivity Ranking Comparison Results

表 5 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

5 指数类与标杆类基准及其应用

5.1 指数类基准

5.1.1 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[5]。OpenActivity 可以作为量化一个项目活跃度的基准单位。

[5] https://blog.frankzhao.cn/how_to_measure_open_source_1/

5.1.2 影响力指数

开源项目影响力的大小取决于多个因素，对开源项目的影响力进行评估，不仅可以帮助开发者和组织确定哪些项目值得投入精力，还可以为开源项目的优化和改进提供针对性意见。OpenPerf 选择了一种加权 PageRank 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。

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，表 6 列出 OpenRank 排名前 10 的对比结果。

由表 6 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。

Table 6 OpenActivity Ranking Comparison Results

表 6 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

5.2 标杆类基准：OpenLeaderboard 排行榜

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，对应到开源数字生态中，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，即那些拥有可测量的、开源业界最佳水平的成绩或认可的绩效水平，可以作为相关领域的卓越标准的对象实体。

OpenLeaderBoard[6] 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。该项目旨在洞察开源世界的脉络，帮助用户可以轻松获取一系列有关开源项目的信息，不仅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，如图 7 所示。

该排行榜使用了 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。

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、国际开源局势分析、企与开源社区观测等领域，均得到了较好的应用。该项目主要使用了 OpenPerf 中的活跃度和影响力基准对开源项目进行排序，不仅能够帮助用户了解开源项目的变化趋势，还可以进一步树立业内项目标杆基准，进一步推动开源项目的发展。

Fig. 7 the presentation of OpenLeaderBoard

图 7 OpenLeaderBoard 项目展示

5.3 行业应用 1：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，还包括学习开源社区的协作模式来提升内部的效率和质量等。目前，国内外 IT 企业的开源办公室（Open Source Program Office，OSPO）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越来越多，OpenPerf 也在此过程中成为越来越多企业的首选。

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

Fig. 8 Ant Group Open Source Dashboard

图 8 蚂蚁集团开源治理大屏

蚂蚁集团 OSPO 开源治理大屏[44]（图 8）使用了 OpenPerf 中的影响力、活跃度、风险度等指数类基准，而这些指数类基准又通过“开源软件供应链风险预测”、“开源社区异常行为检测”等任务进行定义并产出最终数据。

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。

5.4 行业应用 2：阿里巴巴开源开发者贡献激励榜

开发者的个人影响力旨在通过开发者在不同项目间的协作情况来对全域开源生态中的所有开发者进行影响力的计算。精准地评价个人贡献度和影响力有利于对开发者付出和回报进行匹配，可以推动开源协作的持久性和开源社区的规模化发展。当前，国内外不少企业使用 OpenPerf 中提出的开发者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 9 所示。

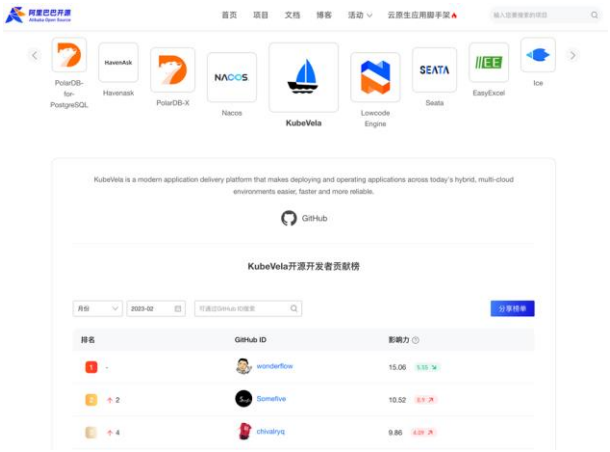

Fig. 9 Alibaba Open Source Contribution Leaderboard

图 9 阿里巴巴开源开发者贡献榜

以阿里巴巴开源开发者贡献榜[7]为例，该贡献激励榜使用了 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。

该榜单可以获取不同项目下的开发者影响力排名，阿里巴巴针对影响力的数值给予开发者一定奖励。在这样的激励下，更多地开发者愿意在 Github 平台中进行交流，同时项目的 issue、PR、点赞数量有明显提升，且 issue 的评论量提升巨大，相关性极高。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。

5.5 行业应用 3：开源软件课程过程性评价

开源生态的可持续发展离不开开源人才的支持，开源教育是关键，国内外诸多高校均陆续开设了开源相关的课程。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。并率先在课程是通过开源的模式进行开源的评价。

该过程性评价使用了 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，如图 10 所示。

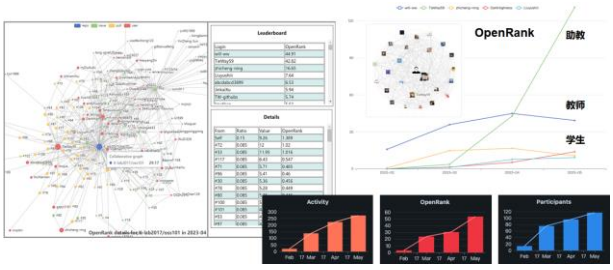

Fig. 10 Analysis of Student Influence in Open Source General Education Courses

图 10 开源通识课程学生影响力网络

《开源软件通识》[8]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。课题组基于 OpenPerf 中的活跃度、影响力以及参与度等基准对学员进行过程性评价，取得了较好的效果。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精髓与本质，为卓越开源人才的培养奠定了基础。

[7] https://opensource.alibaba.com/contribution_leaderboard/details?projectValue=sealer&timeType=month&time=1685548800000

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

6 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基准测试服务系统。该系统包含开源基准规范、公共工具集、服务接口等核心内容，基于上述系统，定义了首批开源方向的 9 个数据科学任务类基准测试，并实现了 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，推动开源领域的研究和发展。最后，列举了 3 项基准测试即服务的应用案例，持给学术界、工业界、基金会等不同组织提供服务。

当前，OpenPerf 主要聚焦于开源领域的数据科学标准任务与标准数据集，在未来会逐步迭代，形成更多其他类别的基准并提出更多的基准测试任务。

参 考 文 献

[1] Zhan J. A BenchCouncil view on benchmarking emerging and future computing[J]. 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[C]//2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009: 248-255.

[3] Mazumder M, Banbury C, Yao X, et al. Dataperf: Benchmarks for data-centric ai development[J]. 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[J]. Empirical Software Engineering, 2021, 26: 1-42.

[5] Li Y, Zhan J. SAIBench: Benchmarking AI for science[J]. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 2022, 2(2): 100063.

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

[7] Qi Qing, Cao Jian, Liu Yancen. The Evolution of Software Ecosystem in GitHub[J]. 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[J]. 计算机研究与发展, 2020, 57(3): 513-524.

[8] Dong Ruizhi, Li Bixin, Wang Lulu, et al, Review of Research on Software Ecosystems[J]. 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[J].计算机学报,2020,43(02):250-271.

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

[10] Wu ZF, Zhu TT, Xuan Q, Yu Y. Evaluation of Core Developers in Open Source Software by Contribution Allocation[J]. Journal of Software, 2018, 29(8): 2272-2282(in Chinese).
吴哲夫,朱天潼,宣琦,余跃.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.

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

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

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

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

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

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

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

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

[19] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis[C]//In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 27 – 35, Dublin, Ireland. Association for Computational Linguistics.

[20] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 486 – 495, Denver, Colorado. Association for Computational Linguistics.

[21] Pontiki M, Galanis D, Papageorgiou H, et al. Semeval-2016 task 5: Aspect based sentiment analysis[C]//ProWorkshop on Semantic Evaluation

(SemEval-2016). Association for Computational Linguistics, 2016: 19-30.

[22] Hewapathirana R, Amarakoon P, Braa J. Open Source Software Ecosystems in Health Sector: A Case Study from Sri Lanka[C]//Information and Communication Technologies for Development: 14th IFIP WG 9.4 International Conference on Social Implications of Computers in Developing Countries, ICT4D 2017, Yogyakarta, Indonesia, May 22-24, 2017, Proceedings 14. Springer International Publishing, 2017: 71-80.

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

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

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

[26] 蒋盛益, 李霞, 郑琪. 数据挖掘原理与实践[M]. Dian zi gong ye chu ban she, 2011.

[27] Chen L, Yang Y, Wang W. Temporal Autoregressive Matrix Factorization for High-Dimensional Time Series Prediction of OSS[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023.

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

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

[30] 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[J].华东师范大学学报(自然科学版),2022(05):90-99.

[31] 赵生宇. 基于 OpenRank 的开源项目内开发者贡献评价 [EB/OL].2022. https://blog.frankzhao.cn/openrank_in_project/

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

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

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

[35] 王皓月. 基于链接预测的同质开源项目推荐[D].华东师范大学,2022.DOI:10.27149/d.cnki.ghdsu.2022.003970.

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

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

[38] Rishi D. Affective sentiment and emotional analysis of pull request comments on github[D]. University of Waterloo, 2017.

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

[40] Ohm M, Plate H, Sykosch A, et al. Backstabber’s knife collection: A review of open source software supply chain attacks[C]//Detection of Intrusions and Malware, and Vulnerability Assessment: 17th International Conference, DIMVA 2020, Lisbon, Portugal, June 24 – 26, 2020, Proceedings 17. Springer International Publishing, 2020: 23-43.

[41] Singh P V. The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2010, 20(2): 1-27.

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

[43] Xia X, Lo D, Wang X, et al. Accurate developer recommendation for bug resolution[C]//2013 20th Working Conference on Reverse Engineering (WCRE). IEEE, 2013: 72-81.

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